Reti di Calcolatori

APPUNTI DELLE LEZIONI

UNIVERSITA' TELEMATICA UNINETTUNO

A.A. 2015 – 2016

Luca Agostini

Lezione n. 1

Introduzione alle reti di calcolatori

- **Concetti di base**

Le reti di calcolatori son utilizzate per comunicare, per controllare dispositivi remoti, per condividere risorse. L'uso di reti di calcolatori permette la riduzione dei costi.

- **Protocolli e standardizzazione**

Essi sono fondamentali per il funzionamento delle reti di calcolatori. Per **protocollo** si intende una serie di regole che governano lo scambio dei dati ed in particolare il loro formato, incluse le tempistiche e le procedure da eseguire per scambiare i dati.

I protocolli devono essere standardizzati affinché ci sia interoperabilità tra marche diverse. Nel campo delle reti di calcolatori operano diversi enti di standardizzazione: ITU-T, ISO, IEEE.

Le reti di calcolatori possono essere distinte dalla loro distanza, le tre tipologie più importanti sono le LAN, le MAN e le WAN.

LAN è una Local Area Network, con elevata velocità di trasmissione, di 100Mb/s ed oltre. Ha una copertura limitata, dell'ordine dei km; è conforme allo standard ISO/IEEE.

MAN è una Metropolitan Area Network, con velocità di trasmissione medio-alta, >2Mb/s ed ha una estensione dell'ordine della città. E' conforme allo standard ITU-T e anche allo standard ISO/IEEE.

WAN è la Wide Area Network, con bassa velocità di trasmissione e conforme allo standard ITU-T.

- **Tipi di canale**

Un **canale di comunicazione** è un collegamento fisico oppure logico per trasportare informazioni tra due entità, ad esempio due calcolatori.

I canali possono essere punto-punto, con connessione tra due nodi, con la trasmissione che può essere sia unidirezionale che bidirezionale; multipunto, in cui la connessione è tra più nodi. Uno dei nodi ha il controllo del canale, questo tipo di connessione, detta anche master-slave, non è più molto diffusa; broadcast, è una connessione che collega più nodi e ogni trasmissione raggiunge più nodi, sono necessari indirizzi per poter raggiungere il destinatario della comunicazione.

- **Topologie di interconnessione**

Esse possono essere a stella, con un hub che centralizza il passaggio delle informazioni; a bus, in cui le stazioni sono collegate allo stesso mezzo e la trasmissione di una stazione raggiunge tutte le altre stazioni; ad anello, in cui l'informazione viaggia in un anello di stazioni; a maglia parziale, in cui ci sono maglie di canali tra coppie di calcolatori per cui c'è ridondanza di possibilità di comunicazione tra i calcolatori; miste, ad esempio reti satellitari, a stella in una direzione, a bus in un'altra.

- **Multiplazione**

Essa consiste nella condivisione di un canale fisico, il termine inglese è multiplexing. Può essere basata sulla frequenza, usando tecniche di modulazione, per cui si modula il segnale, ovvero si usano diverse frequenza di modulazione, con la conseguenza che i segnali trasmessi sul canale non interferiscono. La multiplazione può anche essere basata sui codici.

La soluzione trasmissiva affinché i dati trasmessi siano riconoscibili è quella di adottare pacchetti con una intestazione. L'informazione aggiuntiva necessaria a questo tipo di soluzione è detta overhead.

Per rendere riconoscibili i pacchetti si può usare la tecnica TDM (Time-Division-Multiplexing), in cui l'appartenenza è codificata nella posizione temporale. Multiplexer e demultiplexer devono sincronizzarsi, inoltre, anche in assenza di dati da trasmettere, deve essere trasmesso qualcosa

- **Commutazione (switching)**

Potendo multiplare comunicazioni diverse sullo stesso canale, possiamo creare dei dispositivi che sono collegati a diversi canali: abbiamo dunque dei dispositivi detti switch (commutatori), router (instradatori), che sono nodi della rete, detti anche intermediate systems, cioè sistemi che stanno nel mezzo di una comunicazione.

Lezione n. 2

Architettura protocollare ISO/OSI

- **Modelli a livelli**

Il modello a livelli deriva dalla logica divide et impera, per cui un problema complesso come quello della comunicazione viene diviso in sotto problemi più semplici, dove al livello più basso si tratta di inviare o ricevere bit. Sotto questo aspetto il generico livello n-esimo fornisce servizi al livello superiore ed interagisce con un livello n-esimo remoto. Inoltre il livello n-esimo usa i servizi forniti dal livello sottostante, livello n – 1.

- **L'architettura OSI**

E' una architettura di riferimento che non è utilizzata perché troppo complessa la sua realizzazione. Essa è formata da 7 livelli:

Livello **7. Applicazione**, o Application. I dati e le funzionalità sono tipiche dell'applicazione, una pagina HTML, un messaggio di posta elettronica.

Livello **6. Presentazione**, o Presentation. Riguarda il contenuto informativo dei dati. Si occupa di fare una traduzione di sintassi dal formato usato nell'End System mittente ad una sintassi di comunicazione nel trasferimento. La cifratura è una funzionalità di questo livello.

Livello **5. Sessione**, o Session. Ha a che fare con transazioni, che sono comunicazioni più complicate, con procedure che coinvolgono eventuali scambi di dati.

Livello **4. Trasporto**, o Transport. I dati possono essere bit o byte. Esso opera sempre End-To-End. In questo livello possono essere fatti i controlli di errore; può essere fatto il controllo di flusso, cioè evitare che il trasmettitore sovraccarichi la rete. Il livello trasporto si occupa di adattare i pacchetti, che devono essere organizzati in gruppi per metterli in pacchetti di livello 3.

Livello **3. Rete**, o Network. Lavora sui pacchetti, cioè un insieme di bit, detti anche PDU (Protocol Data Unit), come da terminologia OSI. Il livello 3 si occupa della consegna attraverso nodi intermedi (Intermediate System) ed ha funzionalità di instradamento dei pacchetti (routing) e di inoltro (forwarding), con i pacchetti che sono ricevuti da collegamenti in ingresso e mandati su collegamenti in uscita. Il livello rete si occupa di definire il formato e la modalità di utilizzare degli indirizzi, che identificano gli End System all'interno dell'intera rete, per poter consegnare i dati all'End System

giusto passando attraverso un certo numero di Intermediate System.

Livello **2. Collegamento**, o Data Link. Lavora sui frame, che sono gruppi di bit. So occupa di capire dove inizia e finisce un frame. Esso aggiunge i suoi dati sia in testa che in coda ai dati che riceve dal livello superiore. Il livello collegamento ha funzionalità di accesso al mezzo (Medium Access Control, MAC), per evitare comunicazioni contemporanee. Inoltre ha funzionalità di rilevamento e correzione degli errori, e di controllo del flusso.

Livello **1. Fisico**, o Physical. Lavora sui bit. Ne definisce la codifica, ovvero come i bit vengono rappresentati da un segnale che viene trasmesso sul mezzo. Definisce le caratteristiche fisiche dei mezzi usati per la trasmissione ed i connettori che devono essere usati.

- **Interazione tra i livelli**

L'interazione tra livelli avviene tramite la cosiddetta Protocol Entity (entità protocollare) che realizza le funzionalità di un livello N, per cui si parla di N-entity, in riferimento alla realizzazione delle funzionalità del livello N. Una N-entity comunica con una N-entity remota dello stesso livello ed usa i servizi del livello inferiore. La comunicazione tra una N-entity ed una N-1-entity avviene attraverso un Service Access Point (SAP).

Le informazioni che vengono mandate da una entità di un certo livello ad una entità remota dello stesso livello sono dette Protocol Data Unit (PDU), in particolare N-PDU, riferite al livello N. In sostanza una N-PDU diventa una (N-1)-SDU, Service Data Unit, quando è passata dal livello N al livello N – 1, che aggiunge le cosiddette Protocol Control Information (PCI), ottenendo quini un N – 1 – PCI. Le PCI aggiunte all'SDU nel livello N – 1 vanno a costituire, finalmente, l'N-1-PDU che saranno mandate al livello N-1 remoto. Questa operazione di inserimento dati si chiama Encapsulation, incapsulamento, o anche imbustamento.

Due livelli adiacenti interagiscono attraverso interfacce, che definiscono servizi e primitive offerti al livello superiore. Il livello inferiore fornisce servizi al livello superiore ed il livello superiore usa i servizi forniti dal livello inferiore.

Ci sono due grosse famiglie nel contesto di interazione e servizi offerti dai livelli: servizi non connessi e servizi connessi, quindi comunicazioni non orientate oppure orientate alla connessione.

Nelle comunicazioni non orientate alla comunicazione, connectionless, non è

necessario un contatto o un'azione preliminare. Le unità dati, cioè le PDU (Protocol Data Unit) vengono mandate ognuna a se stante (si parla di servizio datagram). Questo tipo di servizio non richiede normalmente informazione di stato, non richiede di mantenere traccia degli scambi precedenti, né negli End System (ES) né negli Intermediate System (IS). I servizi non connessi sono di tipo best-effort (non affidabile), cioè si fa il possibile ma non si garantisce nulla, non ci sono conferme e quindi i messaggi possono andare persi. Inoltre non c'è controllo di flusso, quindi i dati possono essere troppi per il destinatario. Non c'è controllo della congestione e quindi i dati possono essere troppi per gli intermediate system (IS). Il servizio connectionless funziona con qualsiasi tipo di canale (sia punto-punto, sia multipunto, multicast o broadcast). E' un servizio più semplice e flessibile, per cui le funzionalità sofisticate sono demandate ad altri livelli, sia sotto che sopra.

Nelle comunicazioni orientate alla connessione, Connection Oriented, più sofisticate, richiedono un coordinamento precedente alla comunicazione, quindi occorre un meccanismo (protocollo) di segnalazione e occorrono informazioni di stato negli Intermediate System e negli End System. In questo caso è possibile garantire che i dati siano consegnati correttamente, una sola volta ed in ordine, ma ad un costo e ad una complessità aggiuntiva.

Comunicazioni non orientate alla connessione sono UDP (User Datagram Protocol) e IP (Internet Protocol); connessioni orientate alla connessione sono TCP (Transmission Control Protocol), GPRS (General Packet Radio Service, una delle tecnologie di telefonia mobile cellulare).

Lezione n. 3

Livello fisico

- **Trasmissione numerica (Digital Transmission)**

Il trasferimento di informazioni è delegato alla propagazione di un segnale con caratteristiche variabili. Il tipo di segnale può essere elettrico (voltaggio tra due fili), ottico (intensità della luce all'interno di una guida d'onda che è la fibra ottica), onda elettromagnetica (che si propaga o nell'etere o in una guida d'onda). La trasmissione avviene in un mezzo che rende possibile la propagazione dei segnali che può essere un conduttore, oppure l'aria stessa.

La trasmissione può essere parallela o seriale. Nella trasmissione parallela si trasmettono più bit per volta ed occorrono più mezzi trasmettitori e più mezzi ricevitori, con un numero di mezzi pari al numero di bit. Si usa per comunicazioni di coro raggio, ad esempio nei chip o nelle schede. Per estensioni più elevate si usa la trasmissione seriale in cui i bit vengono trasmessi uno dopo l'altro. La trasmissione può essere analogica oppure numerica. Nella trasmissione numerica il segnale varia tra un insieme limitato di valori discreti. La codifica di linea definisce una corrispondenza tra valori del segnale, o transizioni del segnale, ed i bit.

Ci sono diversi tipi di codifica: NRZ (un bit a 1 è rappresentato da un valore alto del segnale; E' molto usata la codifica Manchester nelle reti Ethernet, una transizione dall'alto al basso rappresenta un bit che ha valore 1.

Le forme d'onda generate dalle diverse codifiche sono molto diverse. Si deve notare che la durata dei bit è costante, per cui viene definito il tempo di bit, bit time, corrispondente a tempo di trasmissione di un bit, ad esempio 1 nanosecondo. Si ha dunque una velocità di trasmissione costante, transmission rate o bit rate nel caso di bit.

Il ricevitore ed il trasmettitore devono operare in modo sincrono.

- **Interazione con il canale**

Un canale, attraverso il quale viene fatto propagare un segnale, ha una certa banda (bandwidth) che è l'intervallo di frequenze, di sinusoidi a frequenze determinate, che possono propagarsi attraverso il canale. La banda del canale si riduce con la lunghezza, cioè l'intervallo di frequenze a cui il canale consente di propagarsi diventa sempre più piccolo al diventare più lungo il canale. Se la banda del canale è tale per

cui passano tutte le frequenze del segnale originale allora al ricevitore arriverà esattamente lo stesso segnale; viceversa se la banda passante è più stretta le frequenze più alte saranno tagliate per cui il segnale risulterà distorto e la sua campionatura in ricezione potrà contenere degli errori.
Un'altra caratteristica del canale è l'attenuazione, che cresce con la distanza, e consiste nella perdita di ampiezza del segnale in funzione, appunto, della distanza. Questo rende difficile al ricevitore distinguere un valore alto da un valore basso, oppure di vedere le transizioni. Per aumentare l'ampiezza occorre aumentare la potenza, con maggior consumo energetico e quindi più calore e con minore dinamica, ovvero il segnale riesce a cambiare meno velocemente.
Il livello fisico comporta problematiche legate all'attenuazione, alla dissipazione di potenza e alla banda del segnale, che devono essere risolte sia a livello di linea, sia a livello dei mezzi trasmissivi utilizzati per fare i canali.

- **Mezzi in rame**

Si tratta di coppie di conduttori usati per trasmettere segnali elettrici. Con il rame si hanno problemi di interferenza, in quanto le coppie di conduttori funzionano come spire per cui il campo magnetico circostante crea una corrente che si somma al segnale. Questo non succede nelle fibre ottiche, costruite con un materiale dielettrico.
Per risolvere il problema si usano cavi coassiali, con un conduttore al centro, in rame, e con l'altro conduttore che è uno schermo a foglio messo nel cavo che viene coperto con uno schermo intrecciato, separato dal nucleo da un materiale dielettrico. Non avendo buone proprietà meccaniche, il cavo coassiale è sostituito dal doppino attorcigliato, detto twisted pair, in cui la corrente indotta in una spira si elide con quella indotta nella spira adiacente e quindi l'interferenza di una spira cancella l'interferenza dell'altra. Il cavo più usato è l'UTP, in cui ci sono 4 coppie di cavi di rame attorcigliati. Un ulteriore tipo di cavo è il cavo STP, che ha una doppia schermatura, una a foglio e una a maglia, per mantenere all'esterno il campo magnetico che provoca interferenza. Essendo più costoso è usato per collegamenti ad alta velocità, con spettri del segnale molto larghi ed è quindi necessario un cavo con una banda molto elevata.
I connettori per tali cavi sono connettori di tipo RJ-45, di cui l'interfaccia ha la presa (receptacle) ed il cavo la spina (plug).

- **Amplificatori e ripetitori**

Gli amplificatori sono dei dispositivi intermedi del livello fisico che aumentano l'ampiezza del segnale prima che questa diventi troppo bassa; l'amplificatore opera su un segnale analogico, aumentandone l'ampiezza, ovvero l'intensità; esso, però, amplifica anche le interferenze.

Il ripetitore lavora sui bit e non sul segnale analogico, esso riceve e ritrasmette ogni singolo bit. E' anch'esso un dispositivo intermedio a livello fisico che è a conoscenza della codifica di linea. Quindi esso rigenera il segnale, “elimina” la distorsione e le interferenze accumulate. Trasmettitore e ricevitore, con l'introduzione dei ripetitori, possono essere arbitrariamente lontani, a differenza di quanto accade con gli amplificatori.

L'uso dei ripetitori non può essere indiscriminato perché, essendo una trasmettitore ed un ricevitore, hanno bisogno di alimentazione. Non è consuetudine fare tratte con un numero elevato di ripetitori che, oltretutto, non si accorgono di errori in trasmissione, cosa che viene fatta al livello 2 della pila protocollare ISO/OSI.

Lezione n. 4

Controllo dell'errore

- **Rilevamento e correzione dell'errore**

Ci sono due tipi di errori, uno legato alla trasmissione ed uno legato all'elettronica (negli End system. Oppure PDU, Protocol Data Unit, scartate dagli Intermediate System a seguito di congestione). In trasmissione gli errori sono legati all'interferenza e all'attenuazione, che sono condizioni del canale variabili.

Il controllo dell'errore può essere fatto a livello trasporto oppure a livello data link, in cui è tradizionalmente fatto poiché il livello fisico aveva molti errori in tempi passati, mentre gli attuali mezzi trasmissivi hanno tassi di errore molto inferiori. Nel rame e nella fibra sono dell'ordine di 10^{-10}, ovvero un bit ogni 10 miliardi subisce un errore di trasmissione, per cui il controllo dell'errore non viene più fatto a livello data link, ma a livello trasporto. Allo stato attuale sono i canali wireless ad avere alti tassi di errore.

Per il rilevamento dell'errore ci si avvale di informazioni aggiuntive, che quindi costituiscono un overhead. Esse vengono aggiunte nella Protocol Control Information della pila OSI, nell'intestazione (header) dei pacchetti, o delle trame; possono essere aggiunte anche in coda (trailer).

Esempi semplici di rilevamento errori sono i bit di parità: ad esempio devo trasmettere 8 bit, quindi ne aggiungo uno in modo che il numero totale dei bit a 1 sia pari (oppure dispari; nel primo caso si parla di parità pari, nell'altra di parità dispari). Questa soluzione ha il pregio di avere basso overhead, ma ha il difetto di rilevare un numero di errori molto basso, ovvero si rileva un numero pari di errori. Il codice di ripetizione è un altro esempio di rilevamento errori. Esso consiste nell'aggiungere gli stessi bit da trasmettere. C'è un grosso overhead, ma ha il pregio di rilevare molti errori, ma non su bit corrispondenti.

Quello che si vuole ottenere idealmente nella rilevazione degli errori è massima protezione, con minimo overhead e una bassa complessità di calcolo. In pratica se una stazione deve trasmettere dei dati, essa combina i dati con un codice di rilevamento errori generato da un algoritmo trasmettendo il tutto verso il destinatario, che riceve i dati e, tramite l'algoritmo noto grazie al tipo di protocollo usato determina il codice di rilevamento errori. Poi confronta i due codici di rilevamento errore, quello ricevuto e quello calcolato. Se sono uguali allora non c'è stato errore oppure ci sono stati errori

che il codice di rilevamento non ha potuto rilevare. Se i due codici sono diversi allora c'è stato un errore per cui occorre scartare il pacchetto e poi cercare di recuperare l'errore.

Su come calcolare il codice di rilevamento errore ci sono molti modi: parità su righe e colonne, Cyclic redundancy check, Checksum, funzioni di hash criptografico. Ci sono inoltre algoritmi di correzione degli errori, il cui nome generico è FEC, forward error correction. Con essi il codice dice anche quali sono i bit trasmessi che sono sbagliati. La capacità di correzione è però limitata, a causa della complessità di calcolo, che rende la soluzione poco usata.

La cosa più semplice da fare in caso di errore è quella di chiedere di ritrasmettere, che è quello che fa la tecnica ARQ.

- **ARQ: Automatic Retransmission Request (Richiesta automatica di ritrasmissione**

Essa è una tecnica che si basa sulla richiesta al trasmettitore di ritrasmettere in caso di errore. Si può anche usare per le PDU non arrivate, ma in questo caso occorre un timer. Nel fare ARQ occorre avere delle informazioni di controllo dei dati, sia di avere dei codici per il rilevamento dell'errore (Error Protection Code), sia di avere delle informazioni di controllo per numerare le PDU, informazioni incluse nel PCI, Protocol Control Information. I messaggi di controllo sono in sostanza delle PDU che contengono solo PCI e non trasportano SDU (Service Data Unit). Esempi di questo sono gli ACKnowledgment (conferme) , che il ricevitore invia al trasmettitore informandolo di aver ricevuto una particolare PDU. Altri esempi sono le NACK (not ACK), in cui al trasmettitore viene detto di non aver ricevuto una particolare PDU.

Il modo più semplice di fare ARQ è detto “Stop and Wait”: quando il trasmettitore manda una PDU, cioè una SDU+PCI, al ricevitore, il trasmettitore si ferma, ed aspetta che il ricevitore risponda con un ACK. A questo punto il ricevitore manderà la successiva PDU e riceverà il successivo ACK. Questo ha un grosso limite in termini di prestazioni.

Quindi occorre gestire l'ARQ impostando un timer che è fatto partire all'atto dell'invio di una PDU; quando scade il timer la PDU è di nuovo inviata, ma se il timer è troppo lungo allora la sorgente aspetta troppo, se è troppo corto c'è il rischio di mandare inutilmente una PDU, perché potrebbe essere arrivata dopo. Inoltre potrebbe essere

perso l'ACK, per cui vene messo un numeratore nelle PDU affinché il ricevitore possa distinguere copie della stessa PDU ed il numero della PDU viene messo nell'ACK e nel NACK. Nel caso dello Stop and Wait è sufficiente un bit, numerando da 0 a 1, e quindi distinguendo una PDU dalla precedente. Così facendo si elimina la possibilità di fare confusione, infatti, se il ricevitore riceve una PDU con lo stesso riferimento allora la ignora. Inoltre, se viene ricevuta una SDU diversa dalla precedente, il ricevitore vede che è diversa da quella di prima e allora la riceve e la conferma con ACK.

Quello che fa la sorgente è prendere una SDU (Service Data Unit) dal livello superiore e formare la sua PDU in cui aggiunge Protocol Control Information che è il numero N della PDU; memorizza la PDU localmente associata a un nuovo numero N, la invia e fa partire un timer. Quando il timer scade allora vuol dire che la PDU o l'ACK sono andati persi (non si sa) e allora deve inviare di nuovo la PDU, che è stata memorizzata. Se invece è stato ricevuto l'ACK della PDU N allora si può cominciare da capo, cioè accettare una nuova SDU, formare una PDU eccetera.

La destinazione, quando ha ricevuto una PDU e ha rilevato il codice di rilevamento dell'errore e questo risulta essere corretto allora manda un ACK per la PDU numero N, che era scritto nella PCI della PDU appena ricevuta. Se la destinazione stava aspettando la PDU numero N allora estrae la SDU dalla PDU e la passa al livello superiore, quindi riceve effettivamente la PDU. Se sta aspettando la N+1 allora la PDU è arrivata due volte e quindi la PDU N non deve essere ricevuta e deve essere scartata.

Questo è il modo in cui funzionano mittente e destinatario. Essi devono accordarsi sul numero iniziale da usare, quindi il meccanismo di ARQ funziona su protocolli connessi, cioè si pare una connessione per comunicare, ma prima ci si mette d'accordo sul numero da cui partire. L'ARQ non funziona su protocolli connectionless.

- **Prestazioni con ARQ**

Le prestazioni con ARQ sono condizionate da alcuni parametri. Notare che la sorgente si ferma ed aspetta per un RTT (Round Trip Time), cioè per un tempo di andata e ritorno, pari al tempo della PDU per arrivare a destinazione più quello dell'ACK a tornare alla sorgente. Quindi se il canale è lungo e veloce l'efficienza è bassa in quanto si trasmette velocemente ma si aspetta molto quindi il canale rimane inutilizzato. Se consideriamo il throughput, cioè la quantità di informazione trasferita con successo nell'unità di tempo, il massimo valore valore nello Stop and Wait, $T_{S\&W}$ è $T_{S\&W} = L_{PDU}/RTT$, in cui L_{PDU} è la lunghezza della PDU in numero di bit e RTT è il Round Trip Time misurato in secondi. Per aumentare il throughput dello Stop and Wait occorre aumentare la quantità di informazione inviata, prima di aspettare. Quindi il protocollo deve mandare più di una PDU prima di fermarsi ed aspettare, cosa che fa il Window-based ARQ.

- **Window-based ARQ (ARQ a finestra)**

L'idea è quella di avere una finestra di trasmissione, in cui il trasmettitore può inviare fino a Wt PDU prima di fermarsi ed aspettare gli ACK; Wt è la dimensione della finestra, Idealmente, se la trasmissione è sufficientemente alta, la sorgente non si ferma mai. Se la finestra è abbastanza grande, le PDU si propagano sul canale, con una PDU inviata che arriva a destinazione mentre le altre sono ancora in transito con il trasmettitore che continua a mandare, ed altre ricevute, per le quali il ricevitore manda gli ACK. Quindi, prima che il trasmettitore si fermi ad aspettare ACK comincia ad arrivare un nuovo ACK per cui il trasmettitore può mandare nuove PDU; nel frattempo il trasmettitore non si era mai fermato. Se la finestra è grande il trasmettitore non si ferma mai. Questo si traduce in $(Wt \bullet L_{PDU})/ Tc \geq RTT$. Si noti che i PDU devono essere numerati con più di un bit. Una finestra abbastanza grande può essere un problema in quanto i pacchetti in sospeso vanno memorizzati, quindi occorre un certo quantitativo di memoria, che in certi casi può essere rilevante.

Il throughput con finestra di trasmissione è proporzionale alla finestra (e non alla lunghezza del canale. Su reti moderne ARQ non è usato a livello 2 perché vorrebbe dire molta memoria sul trasmettitore e numeri di sequenze molto grandi per cui si avrebbero molti bit da trasmettere, memorizzare, elaborare, ARQ è usato a livello 4, dove le connessioni sono tra Intermediate System, con più sessioni su ogni collegamento. Ogni sessione avrà un throughput limitato, che non arriva al massimo, ma tutte insieme riusciranno a usare la capacità del canale.

La gestione della finestra è tale per cui prende il nome di “sliding window”, finestra che scorre. Il trasmettitore invia solo i dati per possono stare in una finestra di trasmissione prima di ricevere la conferma. All'ACK il trasmettitore sposta la finestra in avanti per cui nella finestra si apre uno spazio per trasmettere nuovi dati e aspettare nuovi ACK. Quando il trasmettitore si ferma vuol dire che la finestra non è sufficientemente grande rispetto al Round Trip Time. E l'ARQ sta limitando le prestazioni. Il trasmettitore potrebbe trasmettere di più se avesse una finestra più grande.

Quando i dati sono persi occorre ritrasmetterli e questo viene fatto con due soluzioni: una detta Go-back-N e una detta Selective repeat. Nella prima si ripetono i dati dall'inizio della finestra e questo succede o allo scadere di un timer oppure se si continuano a ricevere ACK del primo byte della finestra. Nella seconda soluzione il NACK specifica quali sono i PDU mancanti. In questo caso l'inizio della finestra è spostato alla prima PDU non confermata e all'interno della finestra vengono ritrasmesse solo le PDU evidenziate come perse. Se scade il timer, però, viene ritrasmessa l'intera finestra, anche in questo caso.

Lezione n. 5

Ethernet e le reti IEEE 802.3. Medium access control

- **Requisiti e caratteristiche**

I principali standard, di cui sistemi di cablaggio e protocolli sono gli elementi principali, definiscono come devono essere fatti i mezzi trasmissivi (cavi, fibre ottiche ecc.), meccanicamente, fisicamente e quali sono le proprietà trasmissive che devono avere. Inoltre definiscono come devono essere posati, ad esempio indicandone la lunghezza massima.

- **Il modello protocollare IEEE 802**

Il modello protocollare IEEE802 è organizzato su due livelli, uno di nome LLC ed uno di nome MAC, rispettivamente Logical Link Control e Medium Access Control. Questi livelli non corrispondono esattamente a dei livelli OSI però possono essere messi in corrispondenza con dei livelli che stanno al di sotto del livello rete; inoltre le funzionalità dell'LLC sono tipiche del Data Link Layer; il livello MAC ha funzionalità tipiche del livello Data Link Layer ed altre tipiche del Physical Layer.

A livello LLC c'è un unico protocollo, di nome LLC e con la sigla IEEE 802.2.

Per quanto riguarda il Medium Access Control ce ne sono tanti diversi. Esso si occupa di definire chi può trasmettere, chi deve ricevere è stabilito da indirizzi, denominati indirizzi MAC, ovvero indirizzi del sottolivello MAC. Chi deve trasmettere è stabilito da un accesso al mezzo, di cui ne vediamo tre: IEEE 802.3, 802.5 e FDMI. Il primo è uno standard che usa un meccanismo di accesso al mezzo di nome CSMA/CD.

I protocolli Ethernet v2.0 e IEEE 802.3 sono molto simili e possono coesister sulla stessa rete. Ethernet v2.0 ha una unica specifica che ha funzionalità di livello Data Link e di livello fisico. Il sottolivello MAC, sia di Ethernet v2.0, sia di 802.3, ha la caratteristica di essere progettato per topologie a bus, in cui l'accesso al canale è condiviso, e l'accesso deve essere permesso ad una sola stazione per volta, una stazione trasmette, molte ricevono. Questo al fine di evitare collisioni, quindi di evitare interferenze che rendono intellegibile il segnale. Il meccanismo di accesso al canale è non deterministico, cioè una stazione non può sapere a priori quanto tempo passa prima che possa usare il canale.

- **Indirizzi MAC**

Ci sono molte stazioni collegate al canale e quella che deve ricevere i dati è specificato

usano i cosiddetti indirizzi MAC, che sono lunghi 6 byte, quindi 48 bit, in base esadecimale, con i primi due bit trasmessi sul canale che informano sul tipo di indirizzo. Gli indirizzi possono essere di tipo individuale o di gruppo. Se si vuole trasmettere a più stazioni il primo bit trasmesso ha valore 1, viceversa si tratta di un indirizzo individuale. Il primo bit trasmesso sul cavo è il bit meno significativo del primo byte.

Il secondo bit permette di capire se l'indirizzo è universale (valore 0) oppure locale (valore 1). Gli indirizzi globali che individuano una singola stazione si chiamano indirizzi unicast ed essi identificano univocamente una interfaccia.

Ci sono poi indirizzi particolari, detti broadcast, che sono costituiti da tutti 1, per cui un pacchetto o una trama mandati ad un tale indirizzo dovranno essere ricevuti da tutte le stazioni della LAN.

Tra gli indirizzi di gruppo ci sono quelli denominati multicast, che identificano un gruppo di interfacce. Questi indirizzi hanno la caratteristica di avere la seconda cifra esadecimale dispari.

- **CSMA/CD**

CSMA/CD è l'algoritmo che decide chi può trasmettere, Carrier Sense Multiple Access with Collision Detection. Gli indirizzi delle stazioni servono per capire chi deve ricevere. L'algoritmo si basa sul concetto di "carrier sense": la stazione che vuole trasmettere, prima di trasmettere "ascolta" la rete e guarda se c'è una portante, cioè se c'è qualche altra stazione che sta trasmettendo e quindi sta modulando il suo segnale su una portante, si tratta cioè di capire se c'è un segnale sul canale, visto che il protocollo 802.3 trasmette in banda base. Se c'è un segnale sul canale allora la stazione aspetta e quando ci si accorge che nessuno sta trasmettendo allora essa trasmette. C'è una collision detection, cioè un controllo di collisione, in cui ci si può comunque imbattere perché i segnali si propagano con una certa velocità finita. In caso di collisione il segnale è intellegibile. Quando la stazione si accorge che c'è stata collisione essa smette di trasmettere, ma non subito, aspetta un certo tempo e poi riprova. In realtà la stazione trasmette una "jamming sequence", cioè una sequenza di segnali, non identificabili come bit, ma tali da far capire a tutte le stazioni che c'è stata una collisione e devono pertanto ritrasmettere. La stazione riprova a trasmettere, dopo un tempo casuale, perché le stazioni non devono tutte aspettare lo stesso tempo.

L'accesso al mezzo è dunque non deterministico, in quanto la stazione non sa quando potrà trasmettere. Il tempo per accorgersi che è avvenuta una collisione dipende da quanto sono lontane le stazioni, cioè dal diametro della rete, ovvero da quanto è grande la rete.

Per fare in modo che una stazione sia sempre in grado di rilevare collisioni deve essere che il round trip time sia minore o uguale al tempo che la stazione trasmittente impiega a trasmettere il pacchetto minimo, cioè il pacchetto che riesce a trasmettere più velocemente possibile; RTD ≤ min T_{TX}

Il round trip time dipende in sostanza dal diametro della rete diviso la velocità di propagazione del segnale, che dipende dal mezzo trasmissivo, vicino a c, la velocità della luce. Il tempo minimo di trasmissione dipende dalla dimensione minima di un pacchetto e dalla velocità di trasmissione e cioè dal bit rate. Il protocollo 802.3 e Ethernet funzionano a 10Mb/s, che è il bit rate R. Lo standard definisce in 64 byte (512 bit) la dimensione minima dei pacchetti, il tempo minimo di trasmissione è dell'ordine dei 576 tempi di bit, tempi richiesti per trasmettere un bit. Da questi dati possiamo capire quale è la distanza massima tra due stazioni, ovvero il diametro della rete, che risulta essere 5760 m, a 10 Mb/s, quindi quasi 6 km.

Con Ethernet a 100 Mb/s la dimensione diventa 10 volte più piccola e con Ethernet a 1 Gb/s diventa 100 volte più piccolo, ovvero 50 m. In realtà saranno presi degli accorgimenti, ma questo è il calcolo della dimensione minima con l'algoritmo CSMA/CD.

CSMA/CD è un protocollo MAC (Media Access Control), quindi posto al secondo livello del modello ISO/OSI.

- **Formato delle trame**

La trama inizia con un preambolo di 7 byte che serve ai ricevitori per sincronizzarsi. Siccome il canale è diviso, avremo tanti trasmettitori diversi ed ogni trasmettitore ha il suo clock che determina quanto sono lunghi i bit. Un bit a 10 Mb/s sarà lungo 0,1 microsecondi. ma questo valore viene misurato con un clock che ha un valore diverso sulle stazioni. Quindi il ricevitore, usando il suo clock, potrebbe campionare il segnale nel momento sbagliato, leggere due volte lo stesso bit, o saltarne uno.

Per evitare questo la trasmissione deve essere sincrona, il ricevitore deve sincronizzare il proprio clock su quello del trasmettitore, perché lui può individuare

guardando transizioni nel segnale, che vengono fatte in istanti predefiniti. Il segnale del preambolo ha una serie di transizioni che il ricevitore può usare per sincronizzarsi. Il ricevitore sa che il preambolo è finito perché trova una configurazione di bit (1 byte) particolari che si chiama Starting Frame delimiter (SFD). Poi c'è l'indirizzo destinazione (6 byte), un indirizzo MAC, un indirizzo del mittente (6 byte) e la lunghezza del payload; se il payload, cioè i dati, è inferiore a 46 byte, i rimanenti byte sono di riempimento (PADDING). Con la lunghezza del payload il ricevitore deve sapere quanti sono i bit significativi, ovvero quanti portano informazione e quanti no. Il ricevitore passerà i bit significativi al livello superiore. alla fine della trama abbiamo 4 byte che conteggiano una Frame Check Sequence (FCS) per verificare s ci sono stati errori di trasmissione. Se c'è un errore di trasmissione la trama viene scartata, come dal protocollo Ethernet e 802.3, che sono "best-effort", cioè non fanno nulla se non scartare la trama con errore di trasmissione.

La stazione, oltre a capire dove inizia la trama deve anche capire dove finisce, che è una delle funzionalità importanti di livello Data Link. La fine della trama è identificata da silenzio, detto Inter Packet Gap in terminologia Ethernet v2.0, oppure Inter Frame Spacing interminologia 802.3. Quindi lo standard dice che quando la trama finisce le stazioni non devono trasmettere per un certo tempo che è 9,6 microsecondi (9 μs). Questo permette al ricevitore di capire che la trama è finita e permette anche ad un trasmettitore diverso di occupare il mezzo.

Le trame hanno un payload massimo di 1500 byte e quindi una dimensione massima di 1518 byte e questo per non far tenere il mezzo troppo occupato, ovvero affinché le stazioni rilascino il mezzo abbastanza spesso.

- **Ricezione delle trame**

La ricezione delle trame è diversa a secondo del tipo di indirizzo:

. unicast. La stazione riceve la trama se l'indirizzo è uguale a quello della scheda di rete, altrimenti ignora la trama.

. broadcast. La stazione riceve sempre i pacchetti che hanno per destinazione un indirizzo broadcast.

. multicas. La stazione riceve la trama solo se l'interfaccia è stata preventivamente abilitata. La scheda ha una lista di indirizzi abilitati. e' possibile dire alla scheda di ricevere indirizzi multicast qualsiasi. Esiste addirittura, ma non usata, una modalità

promiscous mode che dice alla scheda di ricevere tutto.
Quando la trama viene ricevuta essa è salvata nella memoria della scheda. il payload è passato al livello superiore, tramite la generazione di un interrupt alla CPU.

Le trame IEEE 802.3 sono molto simili a quelle Ethernet v2.0. c'è una differenza nel terzo campo dell'intestazione, che è un campo di 2 byte (max 65535) e in IEE 802.3 è la lunghezza del payload, ovvero la lunghezza del contenuto del campo dati, mente in ethernet v2.0 è un campo detto type (o header type) ed è un valore che specifica cosa è contenuto nel campo dati e non quanto è lungo nel senso che indica quale è il protocollo di livello superiore il cui pacchetto, ovvero il PDU in OSI, è trasportato nel campo dati. Il protocollo di livello superiore è implementato via software e il campo type dice alla scheda quale è il modulo software del livello superiore da andare ad interpellare. La scheda genera un interrupt alla CPU che porterà all'esecuzione del modulo software, che dovrà interpretare il contenuto del campo dati.
In IEEE 802.3 l'informazione non è di questo tipo e non è quindi conosciuto il livello di destinazione dei dati. A questo punto interviene il livello Logical Link Control (LCC). Una trama IEEE 802.3 deve sempre contenere una trama LLC.

Lezione n. 6

Reti Ethernet e IEEE 802.3. - Logical Link Control Layer, Physical Layer, Dimensionamento della rete

- **Logical link control**

E' un livello superiore al livello MAC (Medium Access Control) e, nell'IEEE 802.3, una trama MAC contiene sempre una trama LLC.

Nella trama LLC abbiamo una intestazione MAC, in cui il terzo campo è la lunghezza a cui segue l'intestazione LLC che ha 3 campi di 1 byte. Il primo è detto DSAP (Destination Service Access Point), il secondo è detto SSAP (Source Service Access) e il terzo è detto Control.

L'LLC è un protocollo progettato per fornire servizi di diverso tipo, anche servizi affidabili come fornire funzionalità di controllo del flusso e di controllo dell'errore. Ma nelle reti IEEE 802.3 queste funzionalità non vengono usate e l'LLC viene usato in una modalità molto semplice in cui fornisce un servizio "best-effort". In questo caso, il campo control che identifica il tipo di trama LLC, ha un valore esadecimale 0h03 e questo significa un servizio non "best-effort", che è quindi il valore usato nelle reti IEEE 802.3.

Nei due campi precedenti troviamo lo stesso valore esadecimale pari a 0hFE

Questi due campi ricordano la terminologia OSI e servono per identificare l'identità protocollare di livello superiore a cui il contenuto della trama LLC deve essere consegnato e da cui il contenuto della trama LLC arriva. Questi due campi hanno lo stesso scopo del campo Type di Ethernet v2.0.

Poiché sono supportati solo protocolli Standard IEEE e IP non è un protocollo standard, allora è stata creata l'estensione SNAP, Subnetwork Access Protocol, per ovviare al problema della mancata esistenza di riferimento a IP nei campi DSAP e SSAP.

In pratica se si deve imbustare all'interno di una trama LLC un pacchetto di un protocollo di livello superiore che non è standard IEEE allora nel DSAP e nel SSAP sarà scritto il valore FE esadecimale e dopo il campo Control saranno inseriti altri 5 byte (3+2) di estensione SNAP che servono per specificare quale è il protocollo trasportato all'interno del campo dell'LLC (Information). I primi 3 byte sono per identificare eventuali protocolli proprietari ed il campo è detto OUI, Organization

Unique Identifier. I due successivi byte formano il campo Protocol Type e identificano lo specifico protocollo di livello 3, che un costruttore può averne di diversi tipi. Ci si assicura in questo modo di avere un valore univoco da mettere in questi 5 byte per identificare qualsiasi protocollo di livello superiore, anche se non è un protocollo standard. questo rimane problematico nei confronti del protocollo IP che non è né standard né proprietario. Per protocolli di questo tipo è stato deciso di usare il valore0 nei primi tre byte e il valore 0h800, che è lo stesso valore usato per una trama Ethernet v2.0, che è il formato generato dalle schede di oggi.

- **Standard di livello fisico**

Gli standard di livello fisico sono 3 basati su rame:

. 10BASE5, cavo coassiale spesso in rame, max 500 m.

. 10BASE2, coassiale sottile, max 200 m.

. 10BASE-T, un doppino attorcigliato, lo stesso usato per reti telefoniche ... fine pag. 4

Il 10BASE-T è oggi ancora usato perché poco costoso, facile da posare e gestire e robusto.

Se si vogliono raggiungere distanze maggiori si possono usare dispositivi detti hub, o ripetitori, che hanno diverse porte, e ricevono i bit dalle interfacce inviandoli su ognuna delle altre interfacce.

Il ripetitore è una sorta di Intermediate System di livello 1 che riceve bit secondo uno standard, ad esempio 10BASE-T e trasmette i bit secondo lo stesso standard.

Per la fibra ottica ci sono 4 standard di livello fisico.

. FOIRL. Collegamento di ripetitori

. 10BASE-FL. Collegamento di ripetitori e stazioni.

. 10BASE-FB. Applicazioni tolleranti ai guasti (fault tolerant)

. 10BASE-FP. Stella ottica con accoppiatore passivo (optical star coupler). Il centro stella replica un ingresso su tutte le porte.

La soluzione in rame utilizzata è la 10BASE-T ed è basata su cavo UTP, doppino attorcigliato non schermato, facili da realizzare, flessibile e robusto. E' un cavo che consente la miglior soluzione per il cablaggio di uffici. E' facile da installare e da testare su connettori e prese RJ45.

Con il cavo 10BASE-T si realizzano soluzioni punto-punto con le stazioni collegate a ripetitori (hub), che è il centro stella. La topologia è a stella ma si ha un canale di tipo bus, dove una stazione trasmette e tutti ricevono. Si noti che il canale fisico è punto-punto ma il canale logico è a bus, dove non più di una stazione alla volta può trasmettere. Il MAC assicura che una sola stazione alla volta possa trasmettere.

Del doppino, che contiene 4 coppie, se ne utilizzano 2, una per la trasmissione e una per la ricezione.

Lo standard 10BASE-T è compatibile con lo standard di cablaggio strutturato per cui i cavi da utilizzare sono cavi UTP di lunghezza 90 m. dalla postazione di lavoro a un armadio che contiene un ripetitore. Dal"altra parte la stazione è collegata con un cavo di 10 m. In sostanza stazione → patch cord → cablaggio → patch cord → ripetitore; lunghezza max 100 m., compatibile con lo standard 10BASE-T.

La codifica di linea del 10BASE-T è la codifica detta Manchester che ha buone proprietà in termini di transizioni e quindi permette al ricevitore di sincronizzarsi sul segnale del trasmettitore.

- **Progettazione e dimensionamento della rete**

Dati alcuni parametri, è possibile calcolare l'estensione di una rete, come sotto riportato.

Vale la relazione:

$2 \cdot D / p \leq p_{min} / R$ (2D / p, tempo di percorrenza andata e ritorno), da cui

$D \leq (p \cdot P_{min}) / (R \cdot 2) \approx 2 \cdot 108 \cdot 512 / 10 \cdot 106 / 2 = 5760$ m,

in una rete Ethernet 10 Mb/s

In cui:

D è il diametro della rete; p è la velocità di propagazione del segnale nel mezzo: p_{min} è la dimensione minima del pacchetto (64 byte = 512 bit); R è la velocità di trasmissione, il bit rate

Nel progettare e dimensionare una rete non devono essere superati i limiti massimi degli spezzoni, non devono inoltre essere superati i limiti imposti dal livello MAC e cioè che il tempo massimo dei segnali sulla rete sia tale da permettere alle stazioni i rilevare collisioni. Nella pratica il dimensionamento delle reti Ethernet è molto semplice e basta su due fondamenti: quello di rispettare i limiti fisici di ogni tratta, per cui con

l’uso di 10BASE-T essa è l massimo 100m; poi di non avere più di 4 ripetitori in cascata, cioè un pacchetto, per passare da una stazione all’altra, non deve passare per più di 4 ripetitori.

Il numero massimo delle stazioni nella rete dipende da quante devono trasmettere. Le collisioni creano una perdita di efficienza, tipicamente del 30%-40%. Ogni volta che c’è una collisione non si usa il canale. Non ci devono essere troppe stazioni nella rete, per cui essa va spezzata e il dominio di collisione deve contenere poche stazioni.

- **Chiave del successo**

Il peggior progetto immaginabile, ma di gran successo, è la tecnologia di rete locale più usata e l'unica sopravvissuta. La ragione sta nella semplicità: bassa complessità e quindi basso consumo, facile da realizzare con componenti poco costosi. E' facile da utilizzare, ha un basso costo di proprietà e di esercizio. Questo è un tratto comune nelle reti di calcolatori, cioè pure in presenza di bassa complessità e basso costo, si ha un successo della tecnologia, anche in caso di bassa efficienza.

Lezione n. 7

Interconnessione di LAN tramite bridge trasparenti. Espandere la rete oltre il dominio di collisione

- **Bridging**

I bridge trasparenti sono apparati che consentono di estendere la rete oltre il singolo dominio di collisione.

- **Evoluzione nell'utilizzo - dai bridge agli switch**

Quello che si intende per bridging sono le tematiche riferite al funzionamento di questi tipi di apparato. Il contesto in cui ci troviamo è una rete locale in tecnologia IEEE 802, basata su un canale condiviso sul quale solo una stazione alla volta può comunicare ed è per questa ragione che la rete locale è detta collision domain. Infatti se più di una stazione comunica, ad esempio usando il MAC di IEEE 802.3 o di Ethernet, allora ci sarà una collisione che la rete deve rilevare e questo implica che ci sia una estensione massima del dominio.
Ci deve anche essere un limite al numero totale di stazioni presenti.

Dunque, se occorre una LAN con molte stazioni, che abbia una estensione maggiore di quella permessa dal Medium Access control CSMA/CD usato in IEEE 802.E, allora occorre usare gli apparati detti bridge (talvolta chiamati switch) che permettono la creazione di ponti di collegamento tra domini di collisione diversi, ovvero tra LAN diverse. la rete locale può crescere a piacere usando molti bridge senza subire le limitazioni dovute al fatto che il mezzo trasmissivo è condiviso.
Il bridging è detto anche trasparente e le sue modalità di funzionamento sono riferite allo standard IEEE 802.1D. E' definito come un bridge possa effettuare un inoltro selettivo di trame, il che permette di creare connettività tra le LAN. La LAN estesa che comprende bridge è detta anche bridged LAN. Il fatto che l'inoltro sia selettivo permette di creare un livello di isolamento del traffico, tra una LAN e l'altra. Questo consente a due stazioni che si trovano su spezzoni (segmenti) condivisi, ma diversi, di trasmettere allo stesso tempo. Le due stazioni saranno le uniche a trasmettere sul segmento condiviso su cui si trovano, ma non le uniche a trasmettere nella bridged LAN. I bridge si chiamano trasparenti in quanto le stazioni non sanno della loro presenza.

L'operazione che i bridge compiono è quello di "Store and Forward", ovvero memorizza ed inoltra, cioè vengono memorizzate le trame e poi esse sono inoltrate. Il bridge lavora a livello 2, a livello MAC, riceve trame MAC, le memorizza e poi le inoltra su un altro spezzone usando le regole che valgono su quello spezzone, quindi usando il CSMA/CD.

La trama deve essere memorizzata in quanto non è detto che quando arriva da uno spezzone, sia possibile trasmetterla subito ad un altro. Questo perché su quest'altro spezzone potrebbe trasmettere una stazione.

In questo modo è possibile che avvengano contemporaneamente trasmissioni su spezzoni condivisi diversi. Lo Store and forward introduce però un ritardo che è il tempo per ricevere completamente una trama e rimandarla. Ci sono dei bridge che funzionano in una modalità detta cut-Through che non è conforme allo standard IEEE 802.1D, per la quale, quando un bridge inizia a ricevere una trama da un'interfaccia, se l'interfaccia su cui deve inviare la trama non ha trasmissioni in quel momento allora il bridge comincia immediatamente a trasmettere la trama. In questo modo non viene introdotto un ritardo equivalente a ricevere completamente una trama per poi poterla ritrasmettere.

con collisioni presenti, questa soluzione diventa meno efficiente, possono essere anche inoltrate trame con errori.

Si preferisce l'uso dei switch in quanto essi separano i domini di collisione.

I reti con molte collisioni, tale sostituzione è la soluzione adottata dagli amministratori di rete.

Addirittura si possono fare collegamenti di una stazione ad uno switch, in questo caso le collisioni potranno continuare ad esserci, ma saranno limitate e capiteranno se sia la stazione che lo switch stanno trasmettendo.

Gli switch possono avere da poche porte a centinaia, con eventuali interfacce in fibra.

- **Principi di funzionamento**

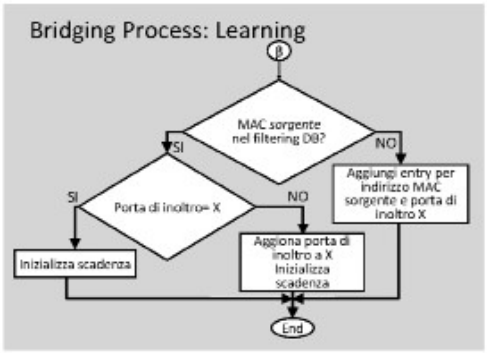
Lo switch ha come elemento centrale il Filtering Database, o forwarding, che è una struttura dati che usa per fare inoltro selettivo (selecting forwarder) delle trame. Dato un indirizzo MAC di destinazione, al switch serve la porta di inoltro di quella trama, ovvero la porta attraverso la quale quella trama deve essere inviata. Nel Filtering Database abbiamo una serie di righe che contengono un indirizzo MAC e la porta su

cui inoltrare pacchetti che siano destinati a quell'indirizzo MAC. La porta viene in qualche modo identificata da un numero o una coppia di numeri o in altro modo. Il database permette di non inviare pacchetti su altre porte e questo rappresenta un filtro. Quando un pacchetto ha una destinazione sconosciuta lo switch manda il pacchetto su tutte le porte. Quando lo switch riceve una trama da una certa porta X, esso verificherà se ci sono stati errori. Se non ci sono errori viene acquisito l'indirizzo MAC destinazione e verificato se c'è nel Filtering Database; se non c'è la trama è inoltrata su tutte le porte e tale operazione è detta di Flooding, inondazione. Se la trama ha destinazione la porta X stessa allora viene scartata, ma se la porta di inoltro, secondo quanto dice il database, è diversa da X allora la trama viene inoltrata sulla porta di inoltro. Questo permette l'inoltro delle trame solo dove servono.

Nel caso in cui una stazione venga spostata dalla porta di inoltro le trame non raggiungeranno più la stazione. Per risolvere questo problema si dà un termine di scadenza alle righe del filtering database. Quando le entry del database, ovvero le righe del database, diventano troppo vecchie, esse vengono eliminate.

- **Un esempio di funzionamento**

Il bridge da un lato fa inoltro di trame e quindi esegue del forwarding, ma dall'altro impara e lo fa guardando il MAC sorgente. Il bridge guarda l'indirizzo MAC destinazione e lo cerca nel filtering database per capire dove inviare la trama. Questo



è l'inoltro, β in figura. A questo punto il bridge testa se nella trama c'è anche un indirizzo sorgente e se esso è nel filtering database. Se non c'è viene aggiunto, tenendo conto del fatto che ora è il mittente, ma più tardi sarà il destinatario: quindi viene aggiunta una entry nel filtering database per l'indirizzo MAC sorgente con porta di inoltro X. In questo modo avviene il popolamento del filtering database. Se il MAC sorgente è già nel filtering database allora viene testato se la corrispondente porta di inoltro è uguale a X. Se lo è si inizializza la scadenza, l'informazione che si trae è che il mittente è ancora raggiungibile. Se non lo è, cioè la porta di inoltro nel database è diversa da X allora viene aggiornata la porta di inoltro e viene inizializzata la scadenza della entry.

Questo processo di learning permette al bridge di costruirsi il filtering database.

L'aggiornamento del filtering database avviene quando una stazione manda una trama su una porta diversa da quella su cui si trovava precedentemente, modificando sia la porta di inoltro sia l'età (che parte da 0).

Se capita un inoltro sbagliato, ad esempio una stazione viene spostata e viene generato traffico verso quella stazione, il bridge inoltra la trama su una porta non più corretta e quindi la stazione non riceverà traffico. Questo succederà fino a che la entry relativa a quella stazione non sarà cancellata nel filtering database per valore elevato di age, diventa “vecchia”, concetto di ageing.

Dal momento della cancellazione della entry in poi quando qualcuno cercherà di mandare del traffico alla stazione il traffico sarà gestito come unknown e sarà mandato dappertutto con il flooding e quindi la stazione lo riceverà.

Lezione n. 8

Il protocollo Spanning tree

- **Problema con il bridging trasparente**

E' una soluzione standard al problema con il bridging trasparente che riguarda la presenza di percorsi chiusi nella rete. Poiché il traffico broadcast è inoltrato con il meccanismo del flooding e quindi non viene filtrato, succede che la rete si satura velocemente e si crea il cosiddetto broadcast storm, in cui la rete diventa in una frazione di secondo piena di copie di pacchetti; la stessa cosa succede nel caso di trame unknown. La soluzione al broadcast storm è quella di spegnere il bridge, quindi è da evitare.

- **Una soluzione standard - il protocollo spanning tree**

La soluzione di eliminare i percorsi chiusi consiste nel tagliare (non fisicamente) dei link. Vogliamo comunque avere dei percorsi chiusi, in quanto sono quelli che ci offrono ridondanza e tolleranza ai guasti. Il taglio è ricavato dalla sospensione dell'uso, tramite una soluzione standard che è il protocollo spanning tree che sospende l'uso di alcune porte.

Lo spanning tree trasforma una rete con percorsi chiusi (maglie) in un albero, un grafo.

Operativamente sia ha 1. una selezione del root bridge; 2. una selezione della porta root che sarà quella per raggiungere il root bridge; 3. una selezione di designated port, porte designate a ricevere e inoltrare pacchetti in una LAN.

I bridge devono riuscire a fare questa operazione in modo distribuito. Ogni bridge deve operare queste decisioni per conto proprio scambiando informazioni con gli altri bridge.

Occorrono dunque dei parametri di configurazione che determinano quali bridge diventerà il root bridge e quali porte verranno scelte come root porte o come designated port.

Le porte che non sono né root port, né porte designate, non verranno usate, potranno essere ripristinate in caso di guasti, ovvero in caso di cambiamento topologico.

Alla fine una rete con maglie diventa un albero.

- **Un'occhiata più da vicino**

Il protocollo spanning tree si basa sullo scambio di pacchetti BPDU (Bridge Protocol Data Unit), che sono pacchetti mandati periodicamente da ogni bridge a un indirizzo multicast predefinito, Esistono due tipi di BPDU, le configuration BPDU, usate nella

fase di creazione dell'albero e le Topology Change Notification BPDU, usate quando c'è un cambiamento topologico nella rete.

Il primo passo nella creazione dell'albero è la creazione del root bridge, che è basata sul root identifier, un identificatore della radice, che contiene la root priority. All'inizio ogni bridge assume di essere root bridge.

Esso comincia a generare delle configuration BPDU, le C-BPDU che sono mandate ad un indirizzo multicast ed arrivano ovunque nella rete, quindi a tutti i bridge. In un campo è scritto che esso è il root bridge, includendo nella Bridge PDU il proprio root identifier che contiene il proprio indirizzo MAC e la propria bridge priority. Ogni bridge riceve le C-BPDU e confronta il proprio identifier con quelli nelle C-BPDU ricevute.

Esiste un criterio per il quale il bridge capisce se ha diritto a diventare root bridge oppure no. Se non deve essere il root bridge allora include l'identifier del root bridge nelle C-BPDU. Cioè se non può essere root bridge, allora inserisce il root identifier che ha appena ricevuto. In questo caso il bridge assume che sia l'altro bridge ad essere root bridge e lo scrive nelle C-BPDU che genera.

Ad un certo punto tutti i bridge riconoscono lo stesso bridge come root e quindi tutte le C-BPDU contengono lo stesso root identifier. Il bridge che a quel root identifier è a tutti gli effetti la radice, ovvero lo sa lui e lo sanno tutti gli altri.

In sostanza, tutti si candidano, esce fuori quello con diritto maggiore e questo avviene con le Configuration Bridge Protocol Data Unit, che si propagano su tutta la rete.

A questo punto deve essere selezionata la root port.

Ogni C-BPDU contiene il costo del percorso attraversato, dalla root fino al punto in cui la C-BPDU viene ricevuta, questa informazione è contenuta nel campo root path cost. Un bridge ha diverse porte e quindi riceverà diverse C-BPDU da queste porte. Nelle C-BPDU che arrivano dalle porte c'è scritto il root path cost. Il bridge confronta tale valore ricevuto dalle sue porte e sceglie come root port quella da cui riceve C-BPDU con costo minimo, tramite criterio univoco. In questa fase ogni bridge ha una porta radice. La porta radice è quella che ha il percorso migliore verso il root bridge.

Quando è stata selezionata una root port, il bridge smette di inviare C-BPDU sulla root port, quindi le C-BPDU vengono generate dal root bridge su tutte le sue porte, gli altri bridge generano C-BPDU su tutte le porte esclusa la radice, per cui le C-BPDU viaggiano dalla radice verso le foglie.

Le trame dati, non le C-BPDU, che sono inviate dalle stazioni, vengono inoltrate dai bridge attraverso le varie porte e raggiungono la radice attraverso la root port.
Le root port fanno si che il traffico vada verso la radice. Quando il bridge radice inoltra i pacchetti ricevuti sulle sue altre porte, questi discendono attraverso la root port fino alle foglie. Il traffico si propaga quindi lungo l'albero. Dalla foglia verso la radice e dalla radice verso la foglia. La root port è quella con minimo percorso.
A questo punto occorre realizzare l'ultimo passo dello spanning tree e cioè la selezione della designated port, ovvero della porta designata a inoltrare traffico su ogni LAN.
Se c'è una LAN con più di una porta allora ci saranno informazioni che sono arrivate dalla radice e quindi esse avranno seguito percorsi diversi. Il costo del percorso dalla radice viene incluso nelle C-BPDU. Dal confronto dei costi i bridge scelgono in modo coerente quale delle porte sarà designated. Quella non designated smette di trasmettere C-BPDU. Tutte le altre sono poste in stato blocking.
Quindi i passaggi totali sono 3:
• Identificazione del bridge radice. • Identificazione della porta radice. • Identificazione delle porte designated.
Con questi passaggi si crea un albero che permette di evitare il broadcast storm e non ha percorsi chiusi.

- **Cambiamenti topologici**

C'è però la necessità di reagire a cambiamenti topologici, dovuti ad esempio ad un errore, cioè una porta o un collegamento non sono funzionanti, oppure si verifica il fallimento del Link Integrity Test, oppure una C-BPDU (che vengono generate periodicamente) non viene ricevuta entro il tempo previsto.
Un bridge che si accorge di un cambiamento topologico reagisce generando una TCN BPDU (Topology Change Notification BPDU), che è inviata attraverso la root port per raggiungere più velocemente possibile la radice; essa è una trama di servizio, diversa.
La radice imposta un bit particolare detto Topology Change Bit nelle Configuration-BPDU che genera. I bridge che ricevono tale C-BPDU dalla root port, a loro volta, impostano un altro bit, detto Topology Change Acknowledgment nella loro C-BPDU, che si diffondono. Il bride che apprende di un cambiamento topologico svuota il filtering database, in quanto esso è stato costruito usando un albero che non è più valido per cui deve essere costruito un albero diverso.

A questo punto viene messo in discussione tutto ciò che era stato scelto: si guarda se il root bridge è sempre lo stesso (nelle C-BPDU), si verifica di nuovo la porta radice e si verificano di nuovo le porte designated o non designated. Il guasto viene recuperato e un nuovo albero viene costruito e includerà anche la LAN oggetto del guasto.

- **Limiti del protocollo spanning tree**

Lo spanning tree ha dei limiti, tra cui quello delle tempistiche per le quali i vari timer che controllano le reazioni non devono essere troppo bassi in quanto i bridge reagirebbero troppo velocemente per cui si potrebbero creare dei loop temporanei e quindi delle maglie con una immediata broadcast storm,

Quindi quello che si fa è far reagire lentamente i bridge, con timer lunghi, ma in questo caso ci possono essere momenti in cui si perde connettività a seguito di un cambiamento topologico.

Inoltre efficienza nei costi e prestazioni non sono punti di forza dello spanning tree, in quanto ci sono collegamenti inutilizzati, che non possono smaltire traffico, mentre altri collegamenti diventano molto carichi creando un collo di bottiglia.

Lo spanning tree crea un albero con una sola strada verso una LAN. Questo è inaccettabile per interconnessioni geografiche. Quindi in questo caso i bridge non vengono usati. Il problema dei bridge è quello di creare un solo albero da usare per tutto il traffico. Questo albero ad un certo punto può diventare congestionato. La soluzione è quella di poter usare alberi diversi a seconda del mittente, ma i bridge non sono in grado di fare questo perché usano un protocollo molto semplice per scegliere l'inoltro dei pacchetti. A differenza degli apparati detti router che usano una soluzione più sofisticata, ottenendo prestazioni migliori.

Lezione n. 9

L’evoluzione di Ethernet/IEEE 802.3. Fast Ethernet

- **Obiettivi e considerazioni**

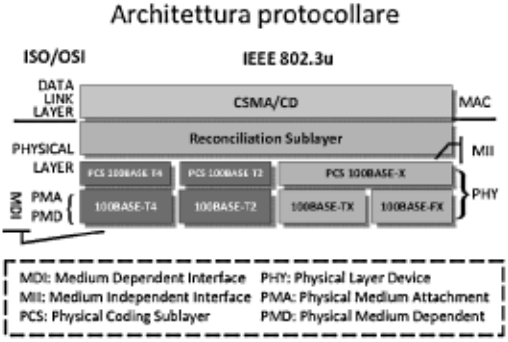
Si vogliono migliorare le prestazioni mantenendo il formato delle trame, con il MAC non modificato, aumentando il tasso di trasmissione e quindi lavorare sul livello fisico degli standard 802.3 per farne evoluzioni. Però si vuole mantenere bassa la complessità ed i costi, considerando un aumento della velocità di trasmissione di 10 volte ed avendo un incremento di costo al massimo 3-4 volte, che poi in realtà sarà minore grazie all'economia di scala e al progresso in elettronica. Aumentando 10 volte la velocità trasmissiva la dimensione massima della rete cambia e diventa 10 volte più piccola. I bit dureranno meno, saranno più brevi e quindi anche le trame saranno più brevi.

Le soluzioni potrebbero essere:

1. aumentare il tempo minimo di trasmissione di una trama allungandola (cosa che viene fatta nelle reti Gigabit Ethernet).
2. Si può evitare il CSMA/CD, come nelle reti 10G Ethernet, il che impone una minima modifica del livello MAC, fare station switching per evitare canali condivisi dove più stazioni trasmettono contemporaneamente.
3. Possiamo avere un funzionamento full-duplex, come nella fibra ottica in cui una trasmette in un verso e l'altra nell'altro. Il MAC non lo permette, ma questo modo di funzionamento è ottenuto disabilitando il CD nel CSMA/CD. Potenzialmente il throughput raddoppia, con nessun costo aggiuntivo, in quanto raddoppia la possibilità di trasmettere.

- **Fast Ethernet**

FAST ETHERNET è Ethernet che funziona a 100Mb/s. E’ cioè in grado di inviare trame a 100Mb/s.



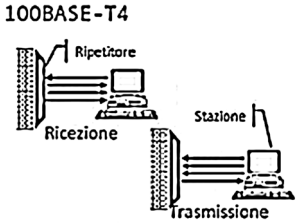
L’architettura protocollare è mostrata in figura, lo standard che definisce Fast Ethernet è IEEE 802.3u.

Lo standard non va a toccare il livello MAC, che rimane lo stesso usato in IEEE 802.3.

Esso identifica una serie di sottolivelli, dei

quali si nota la modularità.
Si notano anche moduli con X finali e con T2 e T4 finali.
Il nome inizia per 100 indicando operatività a 100 Mb/s. Base perché si opera in banda base.
I due sotto-standard di livello fisico con la lettera T nella parte finale effettuano una trasmissione su cavo UTP (Unshielded Twisted Pair) di bassa qualità trasmissiva (categoria 3), cavo telefonico.

Il 100BASE-T4 usa tutte e 4 le coppie, usando alcune coppie a turno nella stessa direzione.
Il 100BASE-T2 usa una modulazione molto complessa per riuscire a usare solo due coppie.
Questo permette il funzionamento non solo sugli stessi cablaggi di IEEE 802.3, ma anche su quelli della telefonia, prima ancora.
Purtroppo questi standard sono molto complessi e non sono stati utilizzati.



Delle 4 coppie nel doppino, una viene usata dalla stazione al ripetitore, oppure al switch; una viene usata dal ripetitore alla stazione; le restanti due coppie vengono usate o per trasmettere o per ricevere.
Quindi vengono usate tre coppie in parallelo in trasmissione ed una in ricezione, o viceversa, cioè tre in ricezione e una in trasmissione. Non si usano tutte in trasmissione in quanto una serve per ricevere informazioni.
La codifica di linea serve per ridurre la banda. Essa è del tipo 8B6T, 8 simboli binari sono codificati in 6 simboli ternari.
Abbiamo ridondanza in quanto 8 bit danno 16 combinazioni, 6 ternari ne danno 27.
La ridondanza è usata per simboli di controllo.
La codifica del segnale è su tre livelli. Con ridondanza per assicurare abbastanza transizioni (per la sincronizzazione).
Lo standard 100BASE-T2 usa due coppie, per ottenerne maggiori prestazioni la trasmissione avviene in modalità full duplex su ognuna delle due coppie. Quindi su ognuna delle due coppie si può trasmettere e ricevere contemporaneamente. Per fare

questo si usano trasformatori ibridi, come in telefonia.
Per trasmettere a 100 Mb/s, che è un alto bit rate, e per non occupare una banda troppo elevata si usa una modulazione molto sofisticata, detta PAM5 che fa una modulazione di ampiezza su 5 livelli. Crea un segnale su 5 livelli e su questo segnale vengono codificati dei simboli in base 5 (quinary symbols). Vengono presi 4 bit e mappati su 2 quinary symbols.
Gli standard a 100Mb/s utilizzati sono quelli che vanno collettivamente sotto il nome 100BASE-X.
Hanno due sottolivelli fisici, quello TX che usa cavo tipo telefonico (non telefonico), di categoria più alta con migliori proprietà trasmissive. Poi l'FX che usa la fibra ottica per la trasmissione. abbiamo dunque due standard fisici diversi con funzionalità comuni che sono definiti in un modulo, o sottolivello, comune che si chiama PCS 100BASE-X, in cui PCS sta per Physical Coding Sublayer.
Esso definisce la codifica di linea da usare che è una codifica 4B5B. Vengono presi 4 bit ed essi vengono codificati su 5. Stiamo aumentando il bit rate perché per trasmettere 100Mb/s ne dovremo trasmettere 125.
La ridondanza permette l'utilizzo di simboli di controllo e quello di creare fra essi un codice di IDLE per l'Inter Packed Gap, il momento di silenzio tra due pacchetti.
Questo è importante perché l'IPG non deve essere fatto spegnendo il trasmettitore e quindi lasciando che il ricevitore si de-sincronizzi dal trasmettitore, ma vengono trasmesse sequenze di bit che identificano un Packed Gap. Il ricevitore può dunque rimanere sincronizzato per ricevere la prossima trama.
I sotto-moduli che sono dipendenti dal mezzo fisico, ovvero i Physical Medium Dependent, si chiamano 100BASE-TX e usano cavi UTP di categoria 5, oppure cavo doppino schermato (Shielded Twisted Pair, STP), oppure 100BASE-FX che usa la fibra ottica.
La codifica di linea in 100BASE-TX è NRZI verso il transceiver, ovvero tra il livello Physical Coding Sublayer e il livello Physical Medium Dependence.
Il trasmettitore sul cavo fisico genera una codifica MLT-3, che ha una occupazione di spettro più ristretta in quanto ha una variabilità minore di una codifica NRZI.
Siccome il Physical Coding Sublayer è comune tra il Physical Medium Dependence 100BASE-TX e quello 100BASE-FX allora viene anche definita dallo standard la

codifica da usare tra i due sottolivelli. Quindi un transceiver che opera su fibra ottica è in grado di ricevere lo stesso segnale (quello a codifica NRZI) e trasmetterla sotto forma di segnale ottico.

Lo standard è talmente chiaro che il transceiver può essere pluggable, cioè si possono fare schede su cui si può staccare il transceiver per rame ed infilare quello per fibra ottica, con l'interfaccia verso il Physycal Coding Sublayer ben definita.

La ridondanza nei codici viene usata per assicurare che il segnale (quello sopra) che ha una periodicità più lunga di quella NRZI abbia comunque abbastanza transizione per consentire la sincronizzazione.

Per quanto riguarda il dimensionamento della rete usando 100BASE-TX, avendo aumentato 10 volte la velocità della trasmissione il diametro della rete si riduce di 10 volte, con un diametro massimo di 500 m. In realtà poiché dovranno essere usati dei ripetitori, in quanto i cavi rame possono essere massimo 100 m, i ripetitori introducono un ritardo che ha un impatto sul Round Trip Delay e quindi il vincolo posto dal Medium Access Control sulla dimensione massima del dominio di collisione è 205 m.

Avendo usato 100BASE-FX, occorre rispettare i vincoli del livello MAC con la dimensione massima della rete limitata dal CSMA/CD (se non si opera in full-duplex, in cui le stazioni sono collegate direttamente agli switch, con una lunghezza massima di quasi 500 m.).

Usando ripetitori con tratte in fibra, possiamo usare un solo ripetitore e possiamo fare tratte tali per cui il dominio di collisione sia al massimo 300 m. Ovvero tra due stazioni non ci devono essere più di 300 m.

Lezione n. 10

Evoluzione di Ethernet/IEEE 802.3. Gigabit speeds

- **Gigabit Ethernet IEEE802.3Z - IEEE 802.3AB**

GIGABIT ETHERNET

Gigabit Ethernet è standardizzato nei documenti 802.3Z e 802.3AB.

Uno dei problemi che Gigabit Ethernet deve risolvere è quello che deriva dall'aumentare la velocità di trasmissione dei bit alla quale si associa una riduzione del tempo di trasmissione delle trame. Questo porta ad una riduzione della dimensione della rete perché in tutti gli standard di Ethernet è importante che una eventuale collisione venga rilevata prima che la trasmissione di una trama sia terminata. E' importante che i segnali si propaghino nella rete entro il tempo minimo di trasmissione di una trama e quindi che la dimensione totale della rete, cioè la distanza massima tra due stazioni, sia tale per cui un segnale si possa propagare ed una collisione possa avvenire e propagarsi all'indietro verso il trasmettitore prima che il trasmettitore abbia finito di trasmettere la trama di dimensione minore.

In Ethernet, 10 Mb/s, il diametro della rete può essere al massimo 5 km, in Fast Ethernet, 100Mb/s, è al massimo 500 m., in Gigabit Ethernet, 1000Mb/s, è al massimo 50 mt. Questo rappresenta un problema.

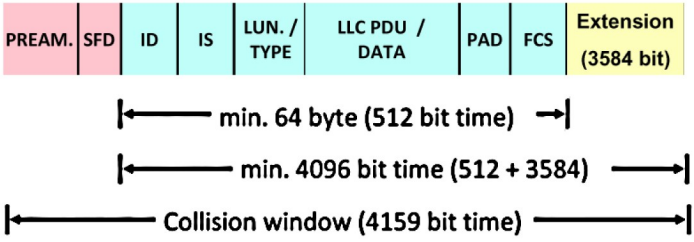
L'approccio è quello di aumentare la durata della trasmissione di una trama minima, ed è detto Carrier Extension

CARRIER EXTENSION

Significa estensione della portante.

Aumenta il tempo di trame corte, aggiungendo al fondo della trama una parte chiamata extension bit, in modo che i dati (l'effettiva trama) e la extension bit siano 4096 bit. Così facendo la collision window, ovvero il tempo necessario per trasmettere i 4096 bit a 1 Gb/s diventa 4,1 microsecondi (μs), che è paragonabile alla collision window di Fast Ethernet che è 5,1 microsecondi.

Si ha dunque un diametro della rete paragonabile a quello di Fast Ethernet, intorno a qualche centinaio di metri.



A lato un esempio della extension, in cui abbiamo una trama MAC, con il preambolo davanti, di dimensione minima di 64 byte, cioè 512 bit. In fondo a tale trama aggiungiamo un certo numero di bit affinché la trama MAC + l'extension bit sia 4096 bit, più quelli del preambolo, arrivando a 4159 bit, trasmissibili in 4159 bit time. A 1 Gb/s 1 bit viene trasmetto i 1 nanosecondo, quindi 4096 bit sono trasmessi in circa 4,1 microsecondi.

Questo tipo di approccio introduce un grande overhead, infatti per trasmettere nel caso peggiore 64 byte, o meno, occorrono 4,1 microsecondi.

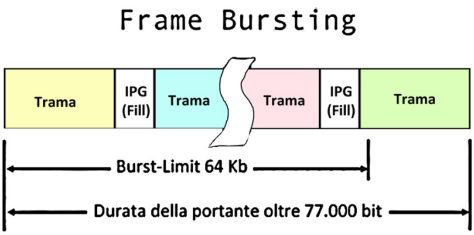
Per questa ragione Gigabit Ethernet oltre a introdurre l'extension bit introduce anche la possibilità di trasmettere in "Burst Mode".

Si ha una trasmissione continua fino a oltre 77000 bit, il trasmettitore dopo aver trasmesso una trama non si ferma, ma continua a trasmettere fino a qualcosa in più di 77000 bit.

La burst window è infatti 64 Kb.

Si ha poi il rilascio del canale, azione che in Ethernet ha due scopi, uno quello di permettere ad altre stazioni di trasmettere, uno quello di far capire al ricevitore che la trama è finita.

Per far questo c'è l'Inter Frame Gap e in Gigabit Ethernet esso è realizzato attraverso bit di riempimento (fill bit). Sono bit con una codifica particolare per cui il ricevitore capire di ricevere l'Inter Frame Gap e che quindi il pacchetto è finito.



Alternativa con elevata efficienza al carrier extension

Nel momento in cui il frame bursting viene utilizzato la stazione può trasmettere una serie di trame una dopo l'altra intervallate da degli IPG tramite fill bit, il trasmettitore può tenere il canale fino a 64 kb, a questo punto può finire di trasmettere la trama che sta trasmettendo e poi il canale deve

essere rilasciato. In totale saranno stati trasmessi un pò più di 77000 bit.
Questa modalità di operazione aumenta l'efficienza rispetto al carrier extension. Ha lo svantaggio che il trasmettitore tiene il canale più a lungo per cui le altre stazioni non possono accedere al canale. Si può usare questa modalità solo se il trasmettitore ha molte trame da trasmettere, viceversa si deve usare il carrier extension.
Gigabit Ethernet ha come modalità di funzionamento quella a canale condiviso, con accesso al mezzo CSMA/CD facendo uso di ripetitori.
Deve utilizzare carrier extension o frame bursting affinché la trama di dimensione minima richieda un tempo sufficientemente lungo.
In realtà la modalità di funzionamento a canale condiviso non viene usata, viene normalmente usata la modalità full-duplex (stazione collegata ad uno switch e non ad un ripetitore) per cui carrier extension o burst mode non necessari perché non ci sono collisioni da rilevare.
Dall'architettura protocollare notiamo che esistono 4 standard diversi di livello fisico organizzati in due famiglie: 1000BASE-T e 1000BASE-X.
Lo standard 1000BASE-T è una trasmissione full-duplex su 4 coppie, con cavo UTP categoria 5, per una lunghezza massima di 100 m.
La stazione usa tutte e 4 le coppie nel doppino contemporaneamente.
E' usata una codifica di linea PAM5, su 5 livelli, in cui 8 bit sono trasformati in 4 simboli in base-5 (quinary).
Ogni simbolo viene inviato su una coppia (quindi si hanno 125 Mbaud per coppia, cioè 125 milioni di simboli al secondo per coppia, ogni 8 bit mandiamo un simbolo sulla coppia).
La ridondanza si usa per ridurre l'interferenza, essendoci più simboli di quanto ne servano. Si usano i simboli che creano meno interferenza, quei simboli cioè che minimizzano l'interferenza tra una coppia e l'altra.
Lo standard 1000BASE-X ha 3 livelli fisici con un sottolivello comune.
Il sottolivello comune usa una codifica 8B/10B (mutuata da uno standard detto FC, Fiber Channel, usato per il collegamento di dischi esterni ai server), che prende 8 simboli binari, ovvero un byte, passato dal livello MAC e codificato in una sequenza di 10 bit, il che crea una ridondanza, anche questa usata per vari scopi.
Tra il 3 livelli fisici, uno è il 1000BASE-CX, che, una volta codificati i bit li trasmette su

un cavo in rame, ha un corto raggio di azione (25 m.). E' adatto per centri elaborazione dati, in cui ci sono collegamenti tra server a mezzo ripetitori o switch, ma più che altro switch visto che ripetitori a 1 Gb/s non si trovano commercialmente.

I connettori sono visualizzati nella immagine dedicata, e sono diversi da quelli normalmente usati. Nel cavo vengono usate 2 coppie, una per trasmettere dalla stazione allo switch ed una per trasmettere dallo switch alla stazione.

Ci sono anche connettori standard 1000BASE-X per la fibra, come mostrato.

Nell'uso della fibra ottica, si usano 2 fibre di cui una per trasmettere dalla stazione allo switch ed una per trasmettere dallo switch alla stazione. Ci sono chiavi di inserzioni.

Gli standard 1000BASE-X per fibra ottica sono 2, uno è il 1000BASE-SX e l'altro è il 1000BASE-LX, come a lato riportati.

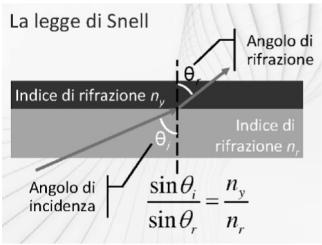
Il 1000BASE-SX, in cui SX sta per Short X, usa onde corte, cioè onde di lunghezza d'onda più corta e quindi frequenza più elevata. Si possono raggiungere tratte di 275-550 m.

Il 1000BASE-LX usa onde lunghe e la tratta può essere tra i 550 e i 5000 m.

- **Principi di comunicazione ottica**

I PRINCIPI DI COMUNICAZIONE OTTICA per capire come mai ci sono due standard per fibra ottica, che usato onde di lunghezza diversa.

La comunicazione ottica usa fibre ottiche, Laser e LED, con parametri che influenzano la comunicazione ottica.



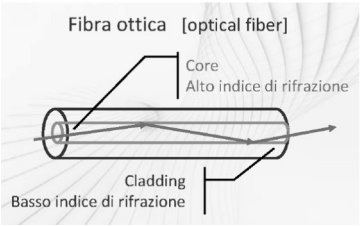
La legge di Snell si applica quando un raggio elettromagnetico incide su una superficie di separazione tra due dielettrici che hanno indice di rifrazione diversi, indice legato alla velocità di propagazione nel dielettrico. Velocità proporzionale alla velocità della luce e di tale indice.

A lato la legge di Snell, con il fenomeno della rifrazione.

Il fenomeno della riflessione, quando gli indici di rifrazione sono molto diversi.

Le fibre ottiche sono basate sul fatto di avere una superficie di separazione tra due

materiali dielettrici in cui uno ha un indice di rifrazione molto più alto dell'altro. Viene mandato un segnale all'interno del primo materiale e questo segnale si riflette sulla superficie.



La costruzione della fibra ottica avviene inserendo un materiale dentro l'altro.

Quindi la fibra ottica contiene all'interno il materiale ad alto indice di rifrazione e rappresenta il core della fibra.

L'altro materiale, a basso indice di rifrazione è detto cladding.

I segnali ottici si propagano nella fibra ottica senza uscire e questo diventa un modo per trasferire segnali ottici, anche a lunga distanza in quanto la fibra, se costruita bene ed ha poche impurità, attenua poco il segnale. Il segnale si propaga a lungo senza perdere troppa potenza e quindi è ricevibile.

Ci sono diverse problematiche, ad esempio il fenomeno per cui un segnale nella fibra si propaga in modi diversi, e quando arriva dall'altra parte ha subito una certa distorsione.

Quanto sia accentuato questo fenomeno dipende dagli indici di rifrazione e dalla dimensione del core.

Ci sono due grosse famiglie di fibre ottiche:

fibre multi-modali e fibre mono-modali.

Le prime hanno un core, relativamente, più grandi e sono misurati in micron, micrometri.

La produzione delle fibre è un aspetto molto importante, la loro produzione deve essere fatta in modo molto opportuno e devono essere molto pure, in quanto le impurità hanno impatto sulle prestazioni.

Sono fatte in silicio, con produzione anche in plastica, più grandi, usate per applicazioni automotive.

La trasmissione avviene tramite LED, light emitting diode, oppure con LASER.

La modulazione che si adotta nella trasmissione è di tipo on-off, quindi si accende il segnale per trasmettere un 1, lo si spegne per trasmettere uno 0.

La ricezione avviene mediante un photo detector e la trasmissione avviene sempre in modo sincrono, cioè il ricevitore deve sincronizzarsi sul trasmettitore.

Attenuazione

Le fibre ottiche introducono una attenuazione, che non è costante. Essa è proporzionale alla distanza, in decibel per chilometro.

Al variare della lunghezza d'onda (l'inverso della frequenza), effettuata oltre il campo visivo, l'attenuazione decresce all'aumentare della lunghezza d'onda ma soprattutto si nota che essa ha alcuni minimi.

Nella trasmissione ottica si cerca dunque di usare segnali che abbiamo attenuazione in questi minimi. Si dice dunque che esistono tre finestre per la trasmissione ottica, incentrate sulle lunghezze d'onda, ovvero 850 nm, 1310 nm, 1550 nm.

Dispositivi diversi usano finestre diverse. Nella prima finestra si possono usare LED, nelle altre si usano di norma solo laser. La distanza sarà maggiore per una finestra più alta perché l'attenuazione è minore.

Wavelenght Division Multiplexing (multiplazione a divisione di lunghezza d'onda)

E' un concetto importante e consiste nell'idea di inserire più segnali in una stessa fibra, segnali che usano frequenze diverse.

Ci sono due grosse famiglie di division multiplexing: coarse WDM che ha granularità poco fine e usa diverse finestre e Dense WDM che ha una granularità molto piccola dei canali, con un numero di centinaia di canali e, sperimentalmente anche migliaia, nella stessa finestra.

Differenze tra i vari standard di livello fisico

1000BASE-SX, short wavelength, usa lunghezze d'onda più basse, quindi usa la prima finestra e usa fibre multi-modali, con un core ad esempio di 62,5 micron e che permette di raggiungere distanze di 200 m. Oppure può usare fibra multi-modale, ma con core di 50 micron e raggiungere una distanza massima di 500 m.

1000BASE-LX, long wavelength, può usare fibra multi-modale con core da 50 a 62,5 micron e raggiungere distanze massime di 550 m, oppure usare fibra mono-modale, con core a 10 micron e raggiungere una distanza massima di 5000 m.

Prodotti non standard

Ci sono prodotti non standard, che usando la seconda finestra a 1310 nm di lunghezza d'onda possono raggiungere una distanza massima di 10 km. Oppure usando la terza finestra a 1550 nm possono raggiungere una distanza massima di 100 km.
La fibra usata in entrambi i casi è quella mono-modale.

- **10 Gigabit Ethernet IEEE 802.3AE**

10 GIGABIT ETHERNET
Standardizzata in IEEE 802.3AE.
Caratteristiche
Usa il solo funzionamento full-duplex, senza l'uso di ripetitori e senza l'uso di CSMA/CD, per cui non c'è bisogno del carrier extension né del burst mode.
E' la prima tecnologia Ethernet che fa breccia nella MAN e nelle WAN.
L'architettura è standardizzata in ben 7 standard di livello fisico.
Ce ne sono 3 accomunati dalla dicitura 10GBASE-W, in cui W sta per wide area network, WAN.
WAN PHY - 10BASE-W
Basate su SONET/SDH, cioè da un lato permettono di trasportare trame Ethernet su una rete SONET/SDH. Quindi da un lato usano le tecniche standard di trasmissione per trasmettere i bit che costituiscono le trame Ethernet definiti da SONET ed SDH.
L'effettiva velocità di trasmissione è 9,6 Gb/s.
Si può trasmettere su MAN e WAN esistenti, senza posare nuove fibre.
Usando gli stessi standard di livello fisico possiamo riutilizzare componenti già progettati e realizzati per MAN e WAN, cioè le reti metropolitane e geografiche.
Il livello fisico ha solo standard su fibra ottica, sia multi-modale che mono-modale.
Usa tutte e 3 le finestre di trasmissioni.
10GBASE-LX4 usa DWDM, ovvero 4 canali (corsie, lane) ottici sulla stesa fibra.
Le lunghezze dei segmenti vanno da 20 m. (datacenter) a 40 km.

- **40/100 Gigabit Ethernet 802.3BA**

Standard a velocità ancora più alta, accenni.
Prevede diverse velocità trasmissive nello stesso standard.
Caratteristiche generali
Solo funzionamento full-duplex, quindi niente ripetitori.
Livello fisici sia a 40Gb/s che a 100Gb/s.

Anche standard per backplane, cioè pensati per funzionare nel collegamento di schede all'interno di un apparato.

Ha standard su rame, su un cavo di rame apposito, detto twinax, che contiene due conduttori, ovvero due piccoli cavi coassiali. Si usa per trasmissione multi-lane, cioè si trasmette allo stesso tempo su più canali.

Per la trasmissione ottica si usa sia fibra multimodale che monomodale, sfruttando più corsie allo stesso tempo (multi-lane) e usando diverse lunghezza d'onda (multi-wavelength). Questo per ridurre la velocità su ognuna di esse.

Le lunghezze raggiunte vanno da 100m a 40 km.

Lezione n. 11

Reti Wireless - IEEE 802.11

Le reti wireless sono reti non cablate, specificate dallo standard IEEE 802.11

- **Caratteristiche e vincoli**

Caratteristiche e vincoli

La prima caratteristiche è che le reti wireless devono avere a che fare con un mezzo inaffidabile in quanto soggetto ad interferenze, con presenza di ostacoli o cambiamenti nell'ambiente circostante, per cui la rete deve variare la velocità di trasmissione e quando il canale non è in grado di operare a velocità elevate allora i terminali riducono la velocità di trasmissione e quindi i protocolli di livello superiore, come il livello MAC, deve essere anch'esso in grado di adattarsi. Nelle reti wireless avremo anche una variazione di copertura. La copertura non è fissa ed è difficile stabilirla a priori, essendo data da quanto distanti possono essere distanti un ricevitore ed un trasmettitore e l'ambiente che cambia.

I protocolli delle reti wireless hanno dei meccanismi per il controllo dell'errore fatto a livello data-link, ad esempio con meccanismi di ritrasmissione, come ARQ (Automatic Retransmission Request) oppure come FEC (Forward Error Correction).

Tutto questo fa sì che anche i livelli superiori, come quello di trasporto che si occupano dell'affidabilità, devono essere in un qualche modo wireless aware, cioè devono avere delle funzionalità particolari che si usano quando il livello 1 e 2 usino reti wireless.

Scalabilità

E' una caratteristica molto importante delle reti wireless a consiste nel fatto che il numero di terminali può cambiare facilmente.

Però una elevata densità di terminali riduce le prestazioni, in quanto abbiamo un mezzo condiviso che è l'etere usato per trasmettere. Avere tanti terminali, nonostante la rete possa crescere facilmente, diventa un problema, con problemi di prestazioni.

Ecco perché le reti wireless si usano e si vedono come estensioni di reti cablate.

Un altro aspetto importante nelle reti wireless è quello della sicurezza in quanto la connessione alla rete è semplice, non è necessario il contatto fisico.

Diventa fondamentale negli standard per le reti wireless disporre di meccanismi per la riservatezza delle informazioni, ovvero di soluzioni di cifratura per nascondere i dati e di soluzioni per l'autenticazione per verificare quando una stazione cerca di usare la rete, se essa è abilitata a usare la rete.

- **Standardizzazione**

STANDARDIZZAZIONE, il percorso.

Nomi e soprannomi nelle reti wireless

IEEE 802.11: Working Group del IEEE 802 LAN/MAN Standards Committee.

Cioè lo standard IEEE 802.11 fa parte del gruppo di lavoro 802, che è un comitato che si occupa della standardizzazione delle reti locali e metropolitane (LAN e MAN). Le reti wireless sono tipicamente reti locali.

Invece di IEEE 802. 11 si parla di wireless LAN, e si parla spesso di Wi-Fi (Wireless Fidelity) che è un consorzio di costruttori che, quando non erano ancora pronti gli standard, si erano accordati per assicurare e certificare l'interoperabilità dei loro apparati.

L'architettura 802.11 è uno degli standard di livello MAC e fisico che si possono usare nelle reti locali e metropolitane.

Lo standard più famoso, quello che standardizza Ethernet è 802.3 che è basato su CSMA/CD come meccanismo di accesso al mezzo.

Lo standard 802.11 stabilisce un suo MAC, un suo meccanismo di accesso al mezzo (in realtà 2) e dei suoi livelli fisici.

Livello fisico 802.11

Ci sono diversi livelli fisici che si differenziano per le bande di frequenza che utilizzano, Le bande a 2,4 GHz sono bande non regolate per le quali non c'è un ente regolatore che dice chi può trasmettere usando quella banda in una certa zona geografica, come succede per le bande televisive o per le bande usate dalle reti mobili, dalle reti cellulari di tutte le varie generazioni (dalla prima alla quinta). Ovvero che in quella certa zona ad una certa banda c'è un solo operatore abilitato.

Lo standard 802.11 opera in bande non regolate e quindi in una certa zona geografica

ci possono essere tanti operatori che trasmettono nella stessa banda. Dunque lo standard deve prevedere l'interoperabilità di questi "tanti" operatori, intesi come installazioni di reti locali diverse e non coordinate che devono poter coesistere senza far si che l'una impedisca all'altra di funzionare. La banda, ovviamente è quella e dovrà essere condivisa con caduta di prestazioni causata dall'una nei confronti dell'altra, però possono coesistere.

Alcuni standard di livello fisico usano la fascia di frequenze intorno ai 2,4 GHz, uno intorno ai 5 GHz ed uno che opera nella banda di frequenza dell'infrarosso.

Gli standard si differenziano per il tipo di codifica dei bit che utilizzano, il che ha un impatto sulla velocità trasmissiva. Quindi abbiamo standard di livello fisico che operano tra 1 e 2 Mbps, uno tra 5,5 e 11 Mbps, uno tra 8 e 54 Mbps e uno tra 5,5 e 54 Mbps. Quindi le velocità trasmissive sono diversificate perché gli standard fisici a cui si riferiscono sono stati progettati per un canale non affidabile e con caratteristiche che cambiano. Al variare delle caratteristiche del canale, il trasmettitore ed il ricevitore, rendendosi conto che c'è un elevato tasso di errore a una certa velocità trasmissiva, riducono la velocità trasmissiva. Facendo questo il segnale è più robusto ed è più facile per il ricevitore capire quale è il valore dei bit e i tassi di errore si abbassano. D'altro canto, se un trasmettitore non trasmette alla massima velocità ma si rende conto che la trasmissione è affidabile, in quanto vengono rilevati pochi errori di trasmissione, allora aumenta la velocità di trasmissione.

Per ottenere bitrate diversi si usano codifiche diverse.

Lo standard FHSS usa la codifica "Frequency Hopping Spread Spectrum", che è una codifica robusta alle interferenze, che possono essere ambientali oppure causate tra trasmettitori che trasmettono nella stessa banda e si deve minimizzare la loro interferenza. In questa codifica il trasmettitore salta spesso da una frequenza ad un'altra, usandola per poco tempo, secondo una sequenza pseudocasuale.

Lo standard DSSS usa "Direct Sequence Spread Spectrum", che è anch'essa una modalità trasmissiva molto robusta nei confronti delle interferenze.

L'HR-DSSS usa "Hi Rate Direct Sequence Spread Spectrum", che raggiunge una maggiore velocità trasmissiva. E' oggi (alla data della videolezione) uno degli standard più utilizzati.

I due standard OFDM si basano sulla codifica "Orthogonal Frequency Division

Multiplexing"; questi standard permettono di raggiungere velocità molto più alte. Quello che opera nella banda intorno ai 5 GHz non è molto comune data proprio la differenza nella frequenza rispetto agli altri. Quello a 2,4 Gb è invece molto comune.
A lato è mostrata la cronologia dello standard IEEE 802.11.

```
IEEE 802.11e (2000)
→ Qualità di servizio
IEEE 802.11i (2001)
→ Sicurezza
IEEE 802.11F (2003)
→ Inter-access
  point protocol (IAPP)
```

A lato alcuni sotto-standard importanti.
L'IEEE 802.11e, che riguarda la qualità del servizio, differenzia il traffico in base, appunto, alla qualità del servizio ("traffico migliore di altri"). Quindi ad esempio il trasmettitore può decidere di trasmettere prima una trama MAC che appartiene ad una telefonata prima di una trama che contiene dati di un file transfer.
Lo standard IEEE 802.11i riguarda la sicurezza.
Lo standard IEEE 802.11G definisce un protocollo detto Inter-access point protocol, che permette ad una stazione di muoversi da una rete wireless ad un'altra.
Vedremo cosa è un Access Point.

- **Scenari di utilizzo**

L'elemento base nelle wireless LAN si chiama Basic Service Set (BSS). Due terminali che comunicano nella stessa wireless LAN costituiscono un Basic Service Set.
Ce ne sono due tipi:
. Indipendent BSS, detto anche Ad hoc network, in cui le stazioni comunicano direttamente tra loro e non hanno bisogno di infrastruttura. Si chiama Ad hoc perché le reti Ad hoc sono quelle che si fanno per uno scopo particolare.
. BSS, basato su Access point, che è un dispositivo per comunicare, in cui la stazione non comunica direttamente con un'altra stazione, ma passa per l'Access Point. Le stazioni non ricevono i segnali da altre stazioni, ma dall'Access Point, pur essendo, esse, in grado di farlo. In questo caso serve una infrastruttura. Le stazioni possono essere distanti da qualche metro a qualche centinaio di metri a seconda delle condizioni del canale (influenzato da condizioni atmosferiche, ostacoli ecc.).

Extended Service Set (ESS)
Si ha un Extended Service Set quando si collegano due BSS, attraverso un

Distribution System che collega i due Access Point i quali faranno da bridge. Un Access Point prende le trame dalla stazioni del suo BSS e le propagano sul Distribution System e l'altro Access Point le propaga nel suo BSS. Si crea una unica rete, pur avendo due BSS separati.

I Basic Service Set possono essere completamente separati, a causa ad esempio della distanza fra Access Point, ma i Basic Service Set possono anche essere parzialmente sovrapposti e questo ad esempio al fine di supportare lo spostamento delle stazioni senza che queste perdano connettività.
Il terminale nella zona di sovrapposizione può decidere se collegarsi ad uno piuttosto che ad un altro Access Point e quindi far parte del BSS1 piuttosto che del BSS2.
Il terminale in movimento si accorge che un Access Point si sta allontanando in quanto il segnale è più debole e quindi può decidere di collegarsi ad un altro e continuare a muoversi nel nuovo BSS.
Serve un protocollo tra i due Access Point affinché questo possa avvenire ed è quello di cui parlavamo prima, ratificato dallo standard IEEE 802.11F.
Questo ci permette di avere copertura e servizio ininterrotto anche se la stazione si muove.

BSS colocate
Le BSS possono essere colocate, cioè completamente sovrapposte. Una stazione, in qualsiasi momento, può decidere quale BSS utilizzare, ovvero quale Access Point utilizzare.
Si usa questo, ad esempio, per la tolleranza ai guasti, si rompe un Access Point e quindi la stazione si collega immediatamente all'altro.
Si tenga presente che una scheda di rete (che identifica una stazione) usa sempre un solo Access Point; ci sono applicazioni particolari in cui è possibile avere schede di rete di rete doppie, che hanno due trasmettitori, due ricevitori ecc., che possono usare due Access Point diversi allo stesso tempo.
Le BSS colocate possono essere fatte per migliorare le prestazioni.
Questo per quanto riguarda il livello fisico.

- **Servizi del livello MAC**

Autenticazione

E' il primo servizio che il livello MAC offre. Esso è il primo passo per comunicare. La stazione deve dimostrare all'Access Point che è abilitata ad usare la rete wireless.

In IEEE 802.11 l'autenticazione viene richiesta dal terminale, a cui segue l'eventuale conferma dall'Access Point.

E' possibile configurare gli Access Point ad accettare un tipo di configurazione che si chiama "Open system authentication" per cui l'Access Point è aperto e non fa nessun tipo di verifica.

Un tipo di autenticazione più restrittiva è la "Shared key authentication" in cui la chiave (segreta) è precedentemente condivisa tramite un canale sicuro.

L'Access point lascerà l'accesso alla rete solo ai terminali che possiedono quella chiave.

L'altro aspetto importante dello standard è quello della riservatezza della rete, la privacy dei dati.

Riservatezza (privacy)

IEEE 802.11 definisce un meccanismo di riservatezza che si chiama "Wired equivalente privacy", WEP. Questo è un fare in modo che il canale wireless dal punto di vista della privacy sia equivalente ad un canale cablato.

Esso è basato sulla cifratura simmetrica, cioè il terminale e l'Access Point hanno una chiave condivisa segreta che usano per cifrare i dati. Questa soluzione è molto debole, quindi in seguito è stato ratificato lo standard IEEE 802.11i che prende il nome di "WiFi Protected Access", WPA. Esso realizza dei meccanismi di cifratura e di autenticazione più sofisticati e più robusti.

Associazione (disassociazione)

E' un altro servizio importante del livello MAC, l'associazione o la dissociazione di una stazione all'Access Point. La stazione che vuole usare un certo Access Point deve mettersi d'accordo con l'Access Point e quindi creare una associazione tra terminale e access point e di conseguenza anche con il distribution system, per cui gli altri Access

Point verranno a saperlo. L'associazione diventa un meccanismo fondamentale per supportare il roaming, cioè il movimento di una stazione che era prima collegata ad un Access Point e poi si collega ad un altro. Esiste una zona in cui due BSS sono sovrapposti ed una stazione è in grado di usare un Access Point piuttosto di un altro e quello che userà dipende da quello a cui si associa. Ci sarà dunque un protocollo per cui un Access Point dice ad una stazione che la sta usando ed un altro Access Point che dice che non la sta usando più. Quindi quest'ultimo non farà più nulla sulle trame della stazione con cui non c'è più associazione, pur potendole ricevere.

Divenire parte di una rete

Per divenire parte di una rete si effettua l'operazione di "Channel scanning".

Lo standard prevede, all'interno della banda, ad esempio quella a 2,4 GHz, diversi canali di comunicazione, la stazione li prova tutti per vedere se c'è qualche altra stazione e li può provare in due modi diversi.

Il primo in modo passivo, semplicemente ascoltando se qualcuno trasmette oppure in modo attivo provando a generare un segnale per vedere se ci sono altre stazioni.

Quindi quando una stazione deve diventare parte di una rete wireless prima di tutto si deve autenticare poi si deve associare ad un Access Point, e a questo punto adotta i vari parametri di livello MAC e di livello fisico che si usano in quella rete e poi può cominciare ad operare.

- **Medium Access Control (Controllo di accesso al mezzo)**

Il funzionamento dell'algoritmo di accesso al mezzo, avendo un mezzo condiviso e decidere quale stazione può comunicare.

Le modalità di controllo di accesso al mezzo sono due, distribuito o centralizzato.

L'accesso distribuito è detto "Distribution control function (DCF).

L'accesso centralizzato è detto, nello standard, "Point coordination function" (PCF).

Distribution control function

E' basata su un meccanismo di carrier sense multiple access, simile a Ethernet, in cui le stazioni prima di trasmettere ascoltano il mezzo. Diversamente da Ethernet, dove si aveva collision detection, il DCF s usa "Collision avoidance", cioè si cerca di evitare le collisioni attendendo un tempo casuale prima di ritrasmettere (backoff time). La

stazione ascolta e sente che non c'è nessuno sul mezza, ma prima di trasmettere aspetta questo tempo detto backoff time, cercando di evitare che, se un'altra stazione nello stesso tempo ha ascoltato e trovato il mezzo libero si metta anch'essa a trasmettere contemporaneamente provocando una collisione.

In alternativa si può usare un meccanismo di richiesta e di attesa di permesso (RTS/CTS), scambiando due messaggi, Request To Send e Clear To Send.
La ragione per cui si fa questo è che non si può verificare se ci sono collisioni, perché il trasmettitore nel momento che trasmette satura il ricevitore che non può sentire collisioni.
Nelle reti cablate trasmettitori e ricevitori sono collegati a canali fisici, doppini, diversi.

Per questa ragione ci vuole un meccanismo di conferma, cioè di Acknowledgment, dopo la trasmissione.
Quando il ricevente riceve una trama MAC conferma sempre dopo l'avvenuta ricezione.

Point coordination function
Essa prevede un coordinamento centrale, normalmente fatto dall'Access Point.
Si hanno tempistiche controllate, con l'Access Point, o il coordinatore, che usa un meccanismo di "poll" per dire chi può trasmettere. Questo meccanismo può coesistere con il DCF.

Lezione n. 12

Internet e Internet Protocol Versione 4 (IPV4)

- **Un po' di storia**

Internet nasce come un progetto di ricerca negli anni ‘70.
Tale progetto è finanziato da una agenzia americana, la DARPA, Defense Advanced Research Project Agency).
A questo progetto hanno partecipato diverse istituzioni, come a lato indicato.

A quei tempi Internet era una rete molto innovativa, in quanto basata sulla commutazione di pacchetto.
Poteva collegare computer eterogenei, quando tutti avevano protocolli proprietari.
Doveva essere una rete usata dalle istituzioni di ricerca americane.
Fu creata una prima rete, Arpanet network, all’inizio una rete di ricerca che in seguito diventa Internet.

Internet Protocol Suite
E’ la famiglia di protocolli di Internet, già completa alla fine degli anni ‘70. I protocolli di base erano standardizzati già a quel tempo.

I protocolli di base sono due: IP e TCP.
IP: Internet Protocol
TCP: Transmission Control Protocol

Spesso è indicata come architettura protocollare TCP/IP.

Internet è diventata la più grande rete di calcolatori ed è la “rete delle reti”.
Ha avuto un altissimo tasso di crescita.

- **L'architettura protocollare TCP/IP**

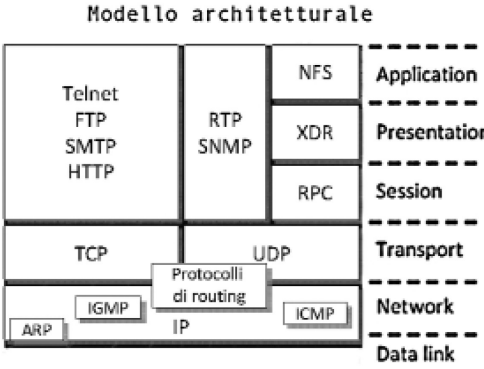
Vengono usati altri protocolli, come UDP (User Datagram Protocol), molto importante; NFS (Network File System), molto usato; ARP (Address Resolution Protocol); ed altri.

L’architettura TCP/IP è uno “standard“ di dominio pubblico, le specifiche sono

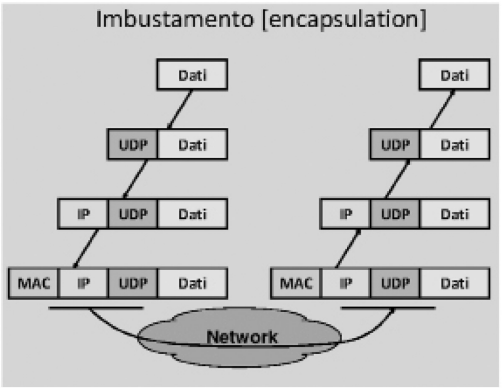
pubbliche. Essa non è in vero e proprio standard, che lo fa un ente di standardizzazione. Lo è diventato de facto.

E' indipendente da costruttori.

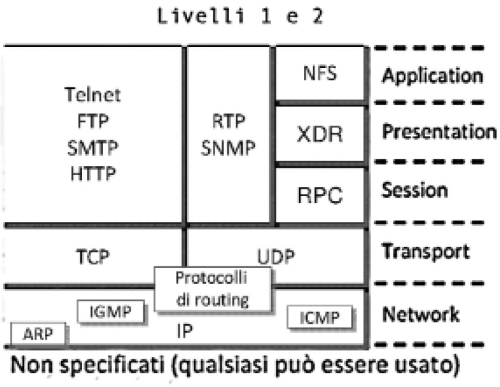
I documenti che descrivono i vari protocolli e le loro specifiche si chiamano RFC (Request For Comment).



Il modello architetturale a lato mostra come ci siano vari protocolli. Esso è messo in confronto con il modello OSI a destra, in cui non è rappresentato il livello fisico.
Alcuni protocolli dell'architettura protocollare TCP/IP corrispondono abbastanza fedelmente al modello OSI nelle loro funzionalità, altri no. Questo deriva dal fatto che TCP/IP è una architettura protocollare nata indipendentemente dal modello OSI.



Come tutti i modelli architetturali a strati, in Internet si usa il meccanismo di imbustamento (encapsulation) per cui i dati dell'utente vengono imbustati, confezionati, con l'intestazione di un protocollo di un certo livello e imbustati in pacchetti IP che vengono trasmetti inserendoli in pacchetti di livello data link, quindi aggiungendo ad esempio una intestazione MAC per poi trasferirli nella rete ed essere ricevuti dall'altra parte per essere "de-imbustati".
Le buste vengono aperte e i vari livelli protocollari vengono elaborati fino a che si arriva ai dati che vengono passati alle applicazioni che usano i servizi di rete.



Livelli 1 e 2

Dall'architettura protocollare si nota che i livelli 1 e 2 mancano. Questo perché la rete Internet deve funzionare con calcolatori omogenei, di qualsiasi marca e di qualsiasi tipo, ognuno con la sua scheda di rete.

L'idea è che il protocollo IP e la rete Internet devono funzionare indipendentemente dalle specifiche schede di rete usate. Quindi, considerando questo dal punto di vista protocollare, devono funzionare indipendentemente dallo specifico protocollo di livello 2 e di livello fisico che vengono usate.

Nell'architettura protocollare vengono definiti protocolli che funzionano dal livello 3 in su.

Il protocollo principale, quello per trasportare i dati, è il protocollo IP, Internet Protocol, e l'architettura specifica come il protocollo IP specifica come può usare i servizi di moltissimi protocolli di livello 2. Ad oggi i protocolli di livello 2 usati sono pochi: Ethernet, 802.11 (reti wireless), PPP (Point to Point Protocol).

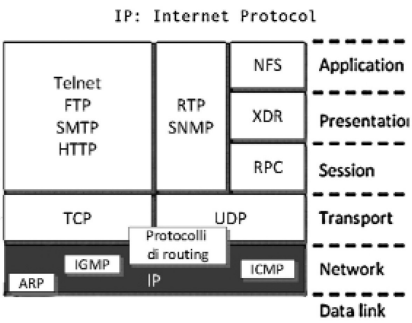
Poiché l'architettura protocollare è indipendente dal livello 2 fa si che la rete sia organizzata in modo gerarchico. La rete Internet è una rete di piccole reti di livello 2, cioè di reti che usano protocolli diversi di livello 2. I dispositivi indicati nei cerchi (R1, R2, R3 e R4) , che sono i router, prendono i pacchetti da una rete di livello e li inoltrano su un'altra rete di livello 2.

I router sono degli intermediate system (in terminologia OSI) che sono responsabili di inoltrare i pacchetti dal mittente al destinatario, da una rete ad un'altra, ad esempio da H2 a H4 in figura.

- **Caratteristiche generali di IPV4**

Il protocollo principale della rete Internet è IP, di cui analizziamo le caratteristiche generali della versione 4.

CARATTERISTICHE GENERALI DI IPV4



Esso è un protocollo di funzionalità di livello network, nella pila OSI, che trasferisce pacchetti attraverso una rete da un intermediate system ad un altro dalla sorgente alla destinazione attraverso, appunto, una serie di intermediate system.

IP: Internet Protocol

Protocollo a pacchetti, la rete è basata sulla commutazione di pacchetto, packet switching.

IP fornisce un servizio non connesso (connectionless) detto anche servizio di tipo datagram. Questo vuol dire che ogni pacchetto , dal punto di vista del protocollo IP, viaggia per conto proprio ed è indipendente dagli altri.

Il servizio è connectionless perché non richiede che la rete o chi manda un pacchetto e chi lo riceve si mettano d'accordo in precedenza, prima di trasferire i pacchetti. Quando una stazione ha un pacchetto da mandare, lo prende e lo manda nella rete.

La rete a questo punto fa del proprio meglio (concetto di best effort) per portare il pacchetto a destinazione.

Per questa ragione il servizio fornito non è affidabile, in quanto non si può sapere a priori se un pacchetto ce la farà ad arrivare a destinazione o meno.

Saranno i protocolli di livello superiore, il livello 3 o le applicazioni, a preoccuparsi di verificare se i pacchetti arrivano a destinazione ed eventualmente chiedere la ritrasmissione.

IP è un protocollo vecchio, ma non obsoleto. Stiamo usando la versione 4 ma sta subentrando la versione 6.

Datagram rispetto a servizio connesso

Ogni pacchetto attraversa la rete indipendentemente dagli altri pacchetti, quindi due pacchetti che appartengono alla stessa comunicazione possono eventualmente

seguire un percorso diverso.

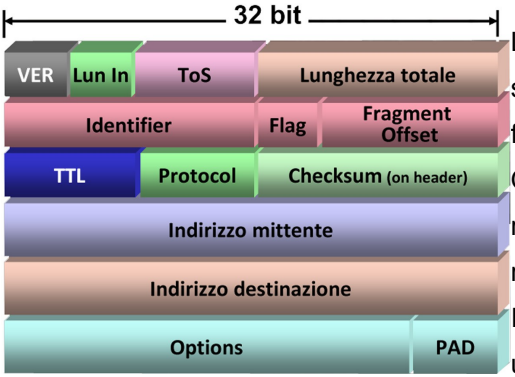
Questo ha delle implicazioni, ad esempio è possibile una consegna fuori ordine dei pacchetti, il che complica la vita al ricevitore.
Inoltre, in un servizio di tipo datagram la gestione delle risorse (p.e. banda) è complessa.
Però tale servizio ha minore complessità, prerogativa delle reti moderne che hanno successo.
Usare un servizio datagram lo rende più robusto, cioè si ha una adattamento “naturale” a cambiamenti nel traffico e nella topologia (guasti). In caso di guasti i pacchetti passeranno da un’altra strada. Cosa che non avviene in una soluzione di tipo connesso, in cui le stazioni, i nodi, si mettono d’accordo e il guasto è gestito rimettendosi d’accordo.
Il servizio datagram è adatto al traffico “dati” (bursty), con treni di pacchetti seguito da silenzio, questo tipo di dati bursty è diverso da quello tipo voce.

Il servizio datagram a pacchetti è problematico quando si vogliano realizzare servizi “carrier grade”, cioè quei servizi che gli operatori vendono per cui è necessario controllare la qualità del servizio e per cui è necessario un recupero guasti veloce.
In telefonia un guasto è recuperato in tempi dell’ordine di 50 millisecondi, con un servizio di tipo datagram tale tempo è di secondi, decine di secondi se non minuti.
Dunque IP va bene per recuperare guasti a patto che non si debbano recuperare troppo velocemente. Va bene per trasportare traffico dati ma in modo best effort, quando si deve garantire una certa qualità del servizio diventa più complicato perché la gestione delle risorse è più difficile.

Il protocollo IP, Internet Protocol, specifica:
Il formato dei pacchetti.
La funzionalità di frammentazione/riassemblaggio [segmentation and reassembly]
Il formato degli indirizzi [“addressing”] e l’assegnazione. Gli indirizzi identificano una stazione.
Specifica il routing [instradamento].

Come si possono realizzare classi di servizio diverse.

- **Formato dei pacchetti**



L'intestazione del pacchetto è organizzata su righe di 32 bit (4 byte), come mostrato in figura.

Questo perché sarebbe difficile rappresentare tutti i campi su una unica riga.

I 32 bit sono poi adattabili all'architettura di un elaboratore che opera a 32 bit, per mezzo dei registri della sua CPU.

Spostamenti dati, ad esempio dalla memoria, per multipli di 4 byte rendono tutto più efficiente.

I campi:

• VER, la versione = 4;

• Lun In, la lunghezza della intestazione; questo perché la lunghezza della intestazione IP è variabile, multiplo di 4 byte; ci sono sempre le prime 5 righe, poi ci sono le Options, campi opzionali e poi riempimento;

• ToS, Type of Service, serve per implementare classi di servizio, per distinguere i tipi di pacchetto, è un campo di 8 bit;

• Lunghezza totale, data dall'intestazione + il campo dati, non riportato in figura, i dati sono dopo l'intestazione;

• Identifier, specifica se effettuare l'operazione di frammentazione o riassemblaggio;

• Indirizzo mittente;

• Indirizzo destinatario;

• Checksum, per il rilevamento errori sull'intestazione, 2 byte;

• Protocol, contiene una indicazione di quale è il protocollo di livello superiore che è contenuto nel pacchetto, questo permette a chi riceve il pacchetto di saperlo interpretare ed elaborare (equivalente al campo Inter Type nell'intestazione Ethernet);

• TTL, Time To Leave, rappresenta il tempo di vita del pacchetto, è un campo molto

importante, è un byte, quindi ha valore tra 0 e 255. Ogni router che inoltra il pacchetto verso un altro router decrementa il valore, se il valore è zero, il router non inoltra il pacchetto e lo butta via. Questo perché i router non conoscono la strada del pacchetto e capita che li mandino in percorsi circolari detti loop o loop di routing, senza questo campo i pacchetti girerebbero nella rete per sempre;

• Options, di lunghezza variabile

Formato TLV (type-length-value), cioè il primo byte di Option è il tipo, il secondo è la lunghezza in byte e poi segue il valore;

Esempi di tipo: Source Routing, Route Recording, Timestamp;

PAD: padding [riempimento];

Una stazione che trova un campo options non conosciuto nel pacchetto, lo elabora comunque.

FRAMMENTAZIONE E RIASSEMBLAGGIO

Frammentazione

E' una funzionalità necessaria quando nella rete ci troviamo con dei collegamenti, dei link di livello 2, che hanno MTU (Maximum Transmission Unit) diverse. La MTU è la dimensione massima, in byte, dei dati che il livello 2 può trasportare.

Nella figura relativa abbiamo una stazione che è collegata ad un router con una tecnologia di livello 2 che ha una MTU di 1500; la stazione può quindi creare dei pacchetti IP di 1500 byte, imbustarli nelle trame di livello 2 e spostarle verso il router. Ma il router R1 è collegato al router R2 attraverso una rete diversa (la rete Net2) che ha un MTU diverso, pari a 620. Quindi Net1 e Net2 sono due reti con tecnologie diverse. Il router R1 riceve quindi un pacchetto IP di 1500 byte e lo deve trasferire al router R2 su una rete che permette solo pacchetti di 620 byte. Il router R1 può fare una cosa del genere solo se è in grado di frammentare il pacchetto, di cui vediamo un esempio in figura a lato, in cui un payload di 1480 è frammentato in tre pacchetti, due da 600 byte ed uno da 280.

Nella frammentazione ogni frammento deve avere un'intestazione (quasi) uguale a quella originale.

I frammenti possono arrivare a destinazione fuori ordine e quindi la destinazione deve avere della complessità aggiuntiva nel riassemblaggio, cioè nel rimettere insieme i frammenti.

Riassemblaggio

Il riassemblaggio può essere fatto o alla destinazione finale o in un router intermedio (cosa non comune).

Nella destinazione finale, alla ricezione del primo frammento di un pacchetto viene attivato un timer e man mano che arrivano i frammenti cerca di ricomporli.

Frammenti dello stesso pacchetto hanno lo stesso identifier, che è un campo, e mittente.

Tutto scartato se non si finisce entro lo scadere del timer

Campi per la frammentazione

I campi sono l'identifier, alcuni flag ed un fragment offset.

• Identifier, uguale per tutti i frammenti dello stesso pacchetto.

• Flag, 4 campi da 1 bit

O: impostato a 0, cioè il primo bit è sempre 0

DF: Don't Fragment, significa che il pacchetto non deve essere frammentato

MF: More Fragments, indica se ci sono altri frammenti se vale 1, 0 nell'ultimo frammento

• Fragment Offset, informa in che ordine il frammento sta rispetto agli altri

In multipli di 8 byte

Questo è quello che succede al pacchetto che ha un dimensione maggiore della MTU permessa.

I principi di base della rete Internet sono stati progettati recentemente tenendo in considerazione le applicazioni moderne e i loro requisiti?

Lezione n. 13

Indirizzi IP

- **Architettura di rete**

Architettura di rete

La rete è organizzata come una rete di reti, si hanno cioè delle reti in cui è possibile comunicare direttamente a livello 2 e tali reti sono collegate tra di loro attraverso dei dispositivi.

Le stazioni, nelle reti IP, si chiamano “Host”.

I dispositivi, identificabili come intermediate system nella terminologia OSI, sono detti “Router”, oppure “Gateway” che è un termine un pò obsoleto, tale termine è usato per intermediate system che operano non a livello 3 ma a livello superiore, mentre i Router lavorano a livello 3.

Il compito di un router è quello di prendere pacchetti IP dalle sottoreti e portarli su un’altra.

Le reti sono di livello 2, all’interno di esse è possibile scambiare trame di livello 2 tra tutte le interfacce collegate ed in terminologia IP si chiamano reti fisiche (physical networks).

IP introduce anche il concetto di rete logica o di sottorete logica (logical subnet).

Logical IP Subnet (LIS) - Sottorete logica IP

E’ un insieme di interfacce a cui è stato dato dato un indirizzo con lo stesso prefisso.

Prima di tutto occorre precisare che nella rete IP gli indirizzi non vengono associati agli host ma alle interfacce. Questo vuol dire che un host, o più probabilmente un router, ha tante interfacce esso avrà un indirizzo per ogni interfaccia.

Inoltre l’indirizzo IP è diviso in due parti, un prefisso ed un’altra parte. Il prefisso (prefix) deve essere lo stesso per tutte le stazioni che fanno parte della stessa rete logica. Una rete logica è un insieme di interfacce i cui indirizzi iniziano nello stesso modo, hanno lo stesso prefisso.

A una rete fisica deve corrispondere una rete logica: questa è una regola molto importante nelle reti IP per il funzionamento della rete.

La rete fisica è una rete di livello 2 in cui le interfacce collegate possono scambiarsi trame di livello 2 e deve corrispondere ad una rete logica, e viceversa. Questo vuol

dire quello che è spiegato di seguito

Reti logiche e fisiche

Tutte le interfacce con lo stesso prefisso nell'indirizzo devono essere collegate alla stessa rete fisica (a una rete logica deve corrispondere una rete fisica).

Tutte le interfacce della stessa rete fisica devono avere lo stesso prefisso nell'indirizzo (a una rete fisica deve corrispondere una rete logica).

Questo ha diverse implicazioni, la prima delle quali è di seguito riportata.

Un identificatore di rete

Un prefisso nell'indirizzo è univoco per una data rete (fisica e logica). Quindi un prefisso diventa un identificatore della rete. La parte iniziale dell'indirizzo lo chiamiamo network part, che identifica la rete, la parte restante dell'indirizzo identifica la stazione all'interno della rete e quindi prende il nome di host part.

Quindi un indirizzo IP è composto da due parti, una parte di rete ed una parte di host, ovvero network part e host part.

Si fa tutto questo per una ragione di scalability, ovvero la capacità di crescere della rete.

La rete IP è fatta di sottoreti e aver fatto corrispondere ad ogni rete fisica una rete logica e che quindi ci sia un prefisso nell'indirizzo che la identifica vuol dire che i router che in giro per la rete inoltrano i pacchetti, per capire dove deve andare il pacchetto, non devono sapere dove si trova ogni singola stazione, come succede per uno switch Ethernet o un bridge Ethernet, essi devono sapere solo dove si trovano le reti. Questo implica che ci sia una gerarchia nell'inoltro dei pacchetti che vengono inoltrati all'interno di ognuna delle reti fisiche usando i servizi della rete fisica. I router servono per inoltrare i pacchetti da una rete all'altra. I router devono conoscere i vari prefissi e come raggiungerli nell'intera Internet, non ogni singola stazione. Questo vuol dire che l'utilizzo degli indirizzi (il modo in cui gli indirizzi vengono assegnati alle stazioni) ed il routing sono strettamente legati.

La scalability non si ottiene gratuitamente, si ottiene al prezzo di uno spreco di indirizzi, cioè al prezzo una bassa efficienza nell'uso degli indirizzi, che, normalmente è meno

del 25%, cioè di tutti gli indirizzi a disposizione se ne riescono ad usare il 25%. Quindi non si riesce ad usare il 75%, c'è uno spreco a vantaggio di una maggior scalability nella rete.

- **Formato degli indirizzi**

Gli indirizzi IP

- Hanno lunghezza 32 bit (4 byte)
- Sono rappresentati in notazione decimale puntata(dotted decimal notation)
- Ogni byte è espresso come numero decimale separato da un punto, ad esempio 12.4.56.38 oppure 193.129.3.215
- Ogni elemento assume un valore tra 0 e 255

L'indirizzo è lungo 32 bit, esso deve essere organizzato come un prefisso che identifica la rete, seguito da una parte di host che identifica l'host all'interno della rete. Come facciamo a sapere quali di questi 32 bit costituiscono l'identificativo di rete (prefisso) e quali l'identificativo di host. Quello che ci chiediamo è quanto è lungo il prefisso: avere una dimensione fissa sarebbe troppo limitativo. Ad esempio con 2 byte per l'uno e due byte per l'altro, identifichiamo 65536 possibilità, ma se la rete fisica ha 100 host sprechiamo tantissimi identificatori di host che non possiamo usare da nessuna altra parte. Se il prefisso è troppo lungo abbiamo pochi identificatori di host. Non potendo decidere a priori, si stabiliscono tre dimensioni di prefisso, come segue:

Class A: 1 byte	Indirizzi di Class A
Class B: 2 bytes	Indirizzi di Class B
Class C: 3 bytes	Indirizzi di Class C

Per identificare se il prefisso è di Classe A, B o C si deve verificare il primo byte. Il valore del primo byte ci permette di capire se l'indirizzo è di Classe A, B o C. In particolare si dovrebbero guardare i primi bit.

Classe A

Il primo bit ha valore 0, con il primo byte che ha un valore tra 0 e 127, per esempio 84.240.20.1;

Max 128 prefissi di rete (Network)

Max 16M indirizzi per host;

Classe B

L'indirizzo inizia per 10, quindi il primo byte ha valore tra 128 e 191, per esempio, 153.240.20.1

Max 16K prefissi di rete (Network)

Max 64K indirizzi per host

Classe C

L'indirizzo inizia per 110, quindi il primo byte ha valore tra 192 e 223, per esempio, 203.240.20.1

Max 2M prefissi di rete (Network)

Max 255 indirizzi per host

Si può continuare

Classe D

L'indirizzo inizia per 1110, con un valore possibile tra 224 e 239, per esempio, 225.240.20.1

Usati per multicast, una stazione manda un pacchetto che ha per destinazione un gruppo di stazioni.

Classe E: anycast

Tutte le classi successive alla C hanno indirizzi non assegnati a interfacce, mentre gli indirizzi delle classi A, B e C identificano le interfacce, con una parte che identifica la rete ed una parte che identifica l'host. La parte che identifica l'host può assumere alcuni valori particolari.

Valori particolari del campo host

Tutti 1: è detto directed broadcast

Per esempio, 203.240.20.255, di classe C, con l'ultimo byte con tutti i bit a 1. Il pacchetto viene inoltrato dai router fino alla rete di destinazione sulla quale il pacchetto è destinato ad essere ricevuto da chi è disposto. Il punto è che il particolare identificativo di host con tutti i bit a 1 è riservato, ed è riservato a questo scopo e quindi non può essere assegnato ad una certa interfaccia.

Tutti 0: si usa per identificare la LIS, ovvero la rete logica, Logical IP Subnet.
Per esempio, 203.240.20.0 è il cosiddetto indirizzo della rete. La rete che ha indirizzo 203.240.20 è la rete 203.240.20.0. Non è usato come indirizzo destinazione, quindi, in linea di principio, può essere assegnato ad un'interfaccia.

Identificatori di host disponibili
Data una parte di host di n bit, ci sono 2n - 2 identificatori disponibili.
Eventualmente 2n - 1 se l'indirizzo di rete è assegnato ad un'interfaccia, cioè il .0 finale, anche se normalmente non si fa in quanto esso convenzionalmente è quello che serve per dare un nome alla LIS e, nota bene, non mandare pacchetti.

Indirizzi particolari

Tutti 1: limited broadcast
255.255.255.255
Esso non è ricevuto da tutte le stazioni ed inoltre non è neppure routed [inoltrato, instradato] dai router. Il pacchetto, quando inviato, viene propagato nella rete fisica e viene ricevuto da stazioni "interessate" a riceverlo. Non va a tutte le stazioni di Internet e non va a tutte le stazioni collegate alla stessa rete fisica. E' un pacchetto che può essere ricevuto da un certo numero di stazione della rete fisica.

Altro indirizzo particolare e fatto da tutti 0, ovvero 0.0.0.0 che rappresenta questo host, che non ha un indirizzo IP e, volendo mettere un indirizzo del mittente, metterà tutti 0.

Altro indirizzo particolare, che un insieme di indirizzi, è quello detto di loopback, nella forma 127.*.*.* ovvero con 127 nel primo byte. Se una stazione manda un pacchetto

all'indirizzo che inizia con 127, e normalmente si usa 127.0.0.1. Quello che succede è che il livello IP prepara il pacchetto, mette dentro l'indirizzo destinazione, poi, invece di passare il pacchetto al livello inferiore, il livello data link affinché venga mandato via, auto-riceve il pacchetto e lo elabora. Questo serve ad esempio per ragioni di testing, con due applicazioni sulla stessa stazione che possono simulare una comunicazione attraverso la rete esattamente come avverrebbe con il livello IP.

- **Netmask**

Serve a superare le limitazioni ed i problemi che si hanno con le classi di indirizzo, ovvero i problemi con il Classful Addressing, appunto i prefissi basati sulla classe.

I problemi sono poca flessibilità, che portano ad una bassa efficienza nell'uso dello spazio di indirizzi. La classe C dà 254 identificativi, la classe B ne dà 16K, quindi se ho una rete con 500 host, non posso usare la classe C, devo usare la classe B con un enorme spreco.

La soluzione sarebbe quella di poter avere una parte di host di 9 bit piuttosto che di 8 bit, per cui avrei 512 identificativi di host con 9 bit.

Il secondo problema è che l'assegnazione degli indirizzi deve essere fatta in modo centralizzato per assicurarsi che non ci siano due organizzazioni nel mondo che usano gli stessi indirizzi.

Questo si fa avendo un ente centralizzato che assegna i prefissi e, nell'assegnare gli indirizzi, assegna un prefisso naturale, di una certa lunghezza, cioè di un certo valore.

Dal prefisso naturale, ad esempio 130.192, dovrebbe essere possibile per ogni nuova sottorete che vorremmo fare usare un prefisso derivato da quello, ad esempio il prefisso 130.192.5.0. In un altro 130.192.6.0.

Vorremmo poter creare dei prefissi più lunghi a partire dei prefissi naturali brevi.

Per questo ci viene in aiuto la netmask che ci permette di fare una identificazione dei prefissi non basata sulle classi.

La netmask è una sequenza di bit associata ad un indirizzo IP.

Essa serve per demarcare il confine tra la parte di rete e quella di host nell'indirizzo IP.

Dato un indirizzo IP, ad esempio 192.168.10.69, questo è un indirizzo di classe C, per cui il prefissi è dato dai primi 3 byte, 24 bit. Però vorremmo trovare un modo affinché il prefisso non sia dato dai primi tre byte, ma dai primi 26 bit. Per questo si una netmask che ha i primi 26 bit ad uno e gli ultimi a 0. Quelli a zero sono i bit che nell'indirizzo

identificano l'host. Quindi dove ci sono gli 1, l'indirizzo è la parte di rete, dove ci sono gli 0 è la parte di host.
La netmask è scritta in notazione decimale puntata.

La parte di rete/host può avere qualunque lunghezza.

La netmask non può avere valori qualsiasi, di seguito i valori ammissibili per i byte della netmask:

0 0000 0000
128 1000 0000
192 1100 0000
224 1110 0000
240 1111 0000
248 1111 1000
252 1111 1100
254 1111 1110
255 1111 1111

Netmask / Prefissi naturali
I prefissi corrispondenti alla classe si chiamano prefissi naturali e per usarli quando si usano le netmask, si definiscono le netmask cosiddette naturali che hanno un numero di bit a 1 pari alla lunghezza del prefisso specificato dalla classe.
I prefissi naturali sono quelli che si ricavano da una classe. Un indirizzo di classe A ha un prefisso naturale che è un byte. La netmask naturale del prefisso di classe A è 255.0.0.0.
Classe A -> 255.0.0.0
Classe B -> 255.255.0.0
Classe C -> 255.255.255.0

Con la netmask possiamo fare quello che si chiama subnetting o quello che si chiama supernetting.

Subnetting e Supernetting

Subnetting: si prende un certo prefisso naturale per creare un prefisso più lungo di quello naturale, questo tramite la creazione di una netmask con un numero di bit a 1 superiore alla netmask naturale.

Supernetting: si crea un prefisso più corto di quello naturale.

Si fa questo quando si vuole avere un prefisso che riassume in sé una serie di altri prefissi più lunghi utile come informazione ai router per inoltrare i pacchetti. Riduce il numero di informazioni con cui i router hanno a che fare e aumenta la scalability della rete.

Negli esempi di subnetting, si noti come sia evidenziato nell'indirizzo che si ha che vogliamo dividere in una parte di rete e una parte di host. La parte di rete contiene un prefisso naturale (detto anche Net) e l'estensione del prefisso naturale, il campo della rete, detta anche Subnet, come dire che c'è una rete che viene divisa in sottoreti. Dal punto di vista della logica dell'IP abbiamo un identificativo di rete che è il campo della rete e un identificativo di host.

Per dire che il prefisso sarà lungo 26 bit si usa una netmask con 26 bit a 1 e i restanti a 0.

A questo punto, dal prefisso naturale è possibile creare tante subnet, tanti prefissi più lunghi, per esempio il prefisso che ha nella subnet il valore 01 e quindi che ha identificativi di host in cui gli ultimi 6 bit vanno dal valore 000001 al valore 111110, ovvero da 65 a 126 in decimale.

Scrivendo tutto in notazione decimale puntata otteniamo 192.168.10.[da 65 fino 126].

Si può anche definire altre subnet, dove gli identificativi di host hanno sempre lo stesso formato (da 000001 a 111110), ma il prefisso della subnet vale 10 invece di 01 come prima e allora, scrivendo il tutto in notazione decimale puntata abbiamo l'indirizzo 192.168.10.[da 129 a 190].

Dalla notazione decimale non si capisce bene dove finisce il prefisso e dove inizia l'estensione di host, se non si scrive il tutto i binario, per lo meno la parte finale.

Subnetting e assegnazione centralizzata degli indirizzi

Gli indirizzi sono assegnati alle organizzazioni in prefissi naturali, al Politecnico di Torino è stato assegnato il prefisso naturale 130.192. Questo è l'identificativo della Net, della rete del Politecnico di Torino.

Questo viene visto come un "grosso" insieme di indirizzi e la singola organizzazione può usare il subnetting per definire prefissi per ogni rete (ogni subnet all'interno della sua rete aziendale) in modo indipendente dall'ente che assegna gli indirizzi.

Lezione n. 14

Routing dei pacchetti IP

- **Decisione di routing degli host**

La decisione di routing che gli host devono fare quando si trovano ad inviare il pacchetto, cioè la decisione che un mittente di un pacchetto deve fare.

Architettura della rete Internet

Ci sono degli host, collegati alle reti fisiche, e dei router, che inoltrano pacchetti da una rete fisica all'altra.

Le reti fisiche corrispondono a reti logiche, dove una rete logica è un insieme di stazioni che hanno indirizzi IP con lo stesso prefisso, quindi tutte le stazioni nella rete fisica hanno lo stesso prefisso e tutte le stazioni con lo stesso prefisso sono parte della stessa rete fisica.

Rete fisica

Dal punto di vista del routing, una rete fisica è caratterizzata dal fatto che i pacchetti IP possono essere consegnati direttamente.

Teniamo presente che, anche se la chiamiamo una rete fisica, in genere si tratta di una rete data link, dove ciò che si può consegnare direttamente sono trame di livello data link, di livello 2.

Si tratta, per esempio, di una rete Ethernet o di una rete WiFi.

Poichè all'interno di una rete fisica è possibile scambiare trame di livello 2, allora i pacchetti IP all'interno della rete fisica possono essere inviati dal mittente direttamente al destinatario, come mostrato nella Rete #1 a lato, mettendoli in una trama di livello 2.

Quando però il mittente deve raggiungere il destinatario in un'altra rete fisica, esso consegnerà il pacchetto ad un router che è il suo default gateway e il router lo inoltrerà alla destinazione inviando il pacchetto, mettendolo in una trama di livello 2, sull'altra rete fisica.

Il router sa se la destinazione è nella sua stessa rete fisica o no attraverso il prefisso, questo perché nelle reti IP una rete logica (LIS, Logical IP Subnet) corrisponde ad una rete fisica e viceversa. Quindi tutte le stazioni sulla stessa rete fisica avranno lo stesso prefisso e tutte le stazioni con lo stesso prefisso saranno sicuramente sulla stessa rete

fisica. Quindi l'host deve verificare il prefisso, il proprio e quello della destinazione.
Nella decisione di routing dell'host viene controllato il proprio indirizzo da parte dell'host e se esso è lo stesso prefisso della destinazione allora vuol dire che si è collegati alla stessa rete fisica e quindi il pacchetto si può mandare direttamente.
Il caso invece per cui si deve passare per un router è il caso in cui i prefissi del mittente e del destinatario sono diversi. In questo caso l'host consegna il pacchetto al router, quello identificato come default gateway, quello che compare nella configurazione dell'host.

- **Prefissi e reti fisiche**

Prefissi classful e reti fisiche
In questo caso la lunghezza del prefisso si vede dalla classe; nell'esempio ci sono due reti "fisiche", i realtà data link, di livello 2. Una rete fisica è collegata all'interfaccia superiore del router, che potrebbe essere una rete Ethernet alla quale sono collegate un certo numero di stazioni. Poi c'è una rete "fisica" collegata all'interfaccia inferiore, che contiene anche un bridge, attraverso il quale può passare una trama di livello 2 inviata dal router, il quale può inviare trame di livello 2 a tutta questa rete.
Si usano indirizzi naturali, il cui prefisso è determinato dalla classe, quindi nella rete fisica superiore è stata definita una LIS, Logical IP Subnet, con prefisso 200, classe C, e l'indirizzo di questa rete è 200.2.1.0.
Il router ha un suo indirizzo, che è 200.2.1.254. Si tende a dare alle interfacce dei router gli indirizzi che hanno come parte di host o l'indirizzo più alto (254, si ricorda che 255 è il broadcast diretto non utilizzabile per una interfaccia), oppure l'indirizzo più basso, ad esempio 200.2.1.1.
Nella rete inferiore si è scelto un prefisso naturale 205.1.4, con il router che ha indirizzo 205.1.4.253.

Prefissi classless e reti fisiche
Gli indirizzi in esempio iniziano con 131 per cui di per sé sono indirizzi di classe B e quindi un prefisso naturale di 2 byte, 131.1; però il prefisso 131.1 è usato sia nella rete fisica superiore che in quella inferiore. Si noti come la topologia della rete sia come quella precedente, ma usando indirizzi diversi. Dall'uso degli indirizzi sembrerebbe un errore, ma in realtà non stiamo usando prefissi naturali, per cui è necessario avere

associato agli indirizzi una netmask e, quella usata, è 255.255.255.0. Essa indica che il prefisso è costituito dai primi 3 byte, l'ultimo è l'host. Questo si fa quando si configura l'interfaccia, alla quale viene assegnato un indirizzo a cui è associata una netmask.
Quindi la rete sopra ha prefisso 131.1.1 e viene chiamata 131.1.1.0, quella inferiore ha prefisso 131.1.4 e viene chiamata 131.1.4.0.
La sopra è dunque la rete 131.1.1.0 con netmask 255.255.255.0: la netmask deve essere fornita altrimenti non si può sapere quale è il prefisso.
La netmask è la stessa per tutta la LIS.
In un successivo esempio, sempre classless, si nota l'uso di un prefisso di classe C, con una netmask 255.255.255.248, che non è un multiplo di 8 bit. In realtà non cambia nulla perché la netmask è 29 bit; sono stati lasciati 3 bit per l'estensione dell'host.
La rete sotto è 203.1.1 più 5 bit a 0. Il nome della rete è 203.1.1.0.
Quella sopra ha prefisso 203.1.1.8 con la stessa netmask in cui i .8 vuol dire 203.1.1 con 4 bit a 0 e 1 bit a 1.
Per verificare occorre scrivere l'ultimo byte della netmask in binario e si potrà notare come gli indirizzi siano stati assegnati per rispettare la corrispondenza tra rete logica e rete fisica.
Nell'ultimo caso di esempio si parte dallo stesso prefisso naturale, 203.1.1.0, di classe C e abbiamo fatto due sottoreti usando il subnetting per creare prefissi più lunghi con netmask di lunghezze diverse. Ogni rete ha la sua netmask, in un caso 255.255.255.248 e nell'altro 255.255.255.240. Si usa in questo caso il termine di variable subnetting.
Per l'host non è complicato verificare se gli indirizzi sono gli stessi e l'operazione per verificare se due indirizzi hanno lo stesso prefisso o no si chiama prefix matching.

- **Prefix matching (confronto dei prefissi)**

Stessa LIS: comunicazione diretta
Si suppone di avere un host che deve mandare un pacchetto ad una destinazione che è nella stessa LIS e quindi ha lo stesso prefisso.
Abbiamo un indirizzo dell'host (192.168.10.65, scritto anche in binario) che deve mandare un pacchetto. L'host ha la propria netmask, fornita alla configurazione ed essa è 255.255.255.192, che indica un prefisso di 26 bit, scritta in binario. Quello che fa l'host è un AND bit a bit tra il suo indirizzo e la sua netmask. Questo avviene in un

ciclo di clock. Il risultato è una sequenza di 32 bit, con valore decimale 192.168.10.64, che ha in sostanza il prefisso nei primi 26 bit ed ha una serie di bit a 0 nell'estensione dell'host.
Quindi questa operazione azzera l'estensione dell'host.
Abbiamo poi un indirizzo destinazione, ad esempio 192.168.10.101, l'host dovrebbe in teoria capire quale è l'indirizzo della destinazione, ma non lo sa perché non ha la netmask della destinazione. Inoltre quello che realmente gli serve è sapere se l'indirizzo di destinazione è uguale al suo. Se quindi estrae dall'indirizzo destinazione un numero di bit pari alla lunghezza del proprio prefisso e lo confrontasse con il proprio prefisso, allora saprebbe se è uguale o no. Se è uguale hanno lo stesso prefisso, se non lo è, esso non è necessariamente il prefisso dell'host che potrebbe essere in realtà più lungo o più corto.
L'host prende l'indirizzo destinazione, ne fa un AND bit a bit con la propria netmask e ottiene una sequenza di bit, che è della stessa lunghezza del proprio prefisso estratta dall'indirizzo originale della destinazione più gli altri bit a 0. A questo punto confronta il risultato ottenuto dalla stessa operazione con il proprio indirizzo e verifica se sono uguali. Se sono uguali i prefissi sono uguali, la sorgente e la destinazione hanno lo stesso prefisso, e dunque la LIS è la stessa e quindi il pacchetto può essere consegnato direttamente. Se non lo sono, vedi "Different LISes: Involve Router", il fatto che i primi 3 byte siano uguali non vuol dire nulla. Alla fine dell'operazione vengono fuori due sequenze di bit diverse, sono uguali i primi 24 bit, man non i successivi due bit. Quindi i prefissi sono diversi, le stazioni appartengono a LIS diverse e sono in reti fisiche diverse, le stazioni non possono mandare il pacchetto direttamente, la comunicazione deve avvenire mediante un router.

- **Principi di funzionamento dei router e scenari di uso di indirizzi**

I router fanno ciò per cui sono "famosi": "ROUTE" I PACCHETTI, ovvero scegliere un percorso per far arrivare alla destinazione i pacchetti.

Prefix matching per ogni interfaccia

Quando un router riceve un pacchetto da un mittente guarda l'indirizzo destinazione del pacchetto e deve fare la stessa operazione fatta dal mittente (prefix matching) per tutte le sue interfacce. In pratica deve essere verificato se la destinazione è in una

delle LIS a cui il router è connesso e su quale si trova. Tramite l'operazione di prefix matching il router riesce a capire quale è l'interfaccia su cui deve inoltrare il pacchetto. Il router in figura in alto ha due interfacce e quindi per ogni interfaccia fa una operazione di bitwise AND tra l'indirizzo che ha sull'interfaccia e la netmask che ha sull'interfaccia con l'indirizzo della destinazione e la netmask che ha sull'interfaccia. Facendo questa operazione con l'interfaccia di sopra troverà che i prefissi sono diversi, facendolo con quella di sotto troverà che i prefissi sono uguali e allora sa che può consegnare direttamente il pacchetto usando il servizio Ethernet, se la rete è Ethernet, mettendo il pacchetto IP in una trama Ethernet e mandandolo a destinazione.

Nel fare prefix matching il router troverà al più una corrispondenza, ma ci può non essere corrispondenza, come mostrato nella figura centrale. In questo caso il router si avvarrà della sua routing table, la tabella di routing.

La routing table contiene una serie di righe in cui ogni riga contiene una destinazione e un next hop. Una destinazione è una sotto-rete logica IP, una Logical IP Subnet, una LIS. Quindi è un prefisso che identifica quella particolare sotto rete logica per cui è una coppia

indirizzo/netmask.

Se stiamo usando indirizzamento classless, che è quello che si fa oggi nelle reti IP, allora per capire quanto è lungo il prefisso serve una netmask. Quindi in ogni riga, detta entry, della routing table il router ha una coppia

indirizzo/netmask che rappresenta una destinazione e un next hop, che è il prossimo router a cui i pacchetti, per quella particolare destinazione, intesa come LIS a cui la destinazione finale del pacchetto appartiene, devono essere inoltrati. Il next hop è sempre direttamente collegato, in quanto il router deve essere in grado di consegnare il pacchetto. Il next hop ha lo stesso prefisso di una delle interfacce del router.

Ad esempio la destinazione 190.3.1.0 che è su tre byte, è raggiungibile mandando pacchetti ad un next hop che è 190.3.3.2, che è il router R2. Esso è direttamente collegato ed è parte della stessa rete fisica ed ha lo stesso prefisso di R5. Se la destinazione è 190.3.9.0 allora R5 invia pacchetti a 190.3.6.8 che è l'indirizzo (formato sia da 190.3.6.8 sia da 255.255.255.0) che R3 ha sull'interfaccia collegata alla rete Ethernet 190.3.6.0 che è una rete a cui anche R5 è collegato. R5 guarda con quale delle sue interfacce fa prefix matching e la usa per inoltrare il pacchetto verso R3.

Nel successivo esempio, destinazioni che hanno lo stesso prefisso naturale (190.3, indirizzo di classe B), vediamo che ci sono 3 sotto-reti create dallo stesso prefisso naturale con lunghezze di prefissi diversi, determinati dalle netmask. Nella rete dove c'è H4 si è usato un prefisso di 24 bit, sulle due reti che collegano i due router R3 e R4, che hanno molte meno destinazioni e quindi hanno meno necessità di identificatori di host, si usa una netmask di 30 bit, che è la netmask più lunga che si può usare che ci dà due identificativi. L'identificativo della rete risulta dunque 190.3.9 e 6 bit a zero.

Il router usa la tabella di routing e le entry nella tabella di routing facendo, con un indirizzo di destinazione dato, un AND logico bit a bit tra indirizzo destinazione e la netmask e vede se il risultato è uguale all'indirizzo associato.

La netmask viene messo in AND bit a bit con l'indirizzo destinazione. Il risultato viene confrontato con l'indirizzo associato: se sono uguali vuol dire che si deve usare il next hop, se non sono uguali, cioè non c'è un matching, allora si passa alla riga successiva. E così via finché il router non trova una riga che fornisca matching.

Se nessuna riga risulta in un matching il pacchetto viene scartato.

Se più righe fanno un matching, come per le prime due righe dell'esempio in figura, il router deve scegliere un next hop e sceglierà, sempre, quella riga che ha il prefisso più lungo. Il router farà l'operazione detta longest prefix matching. Cerca cioè nella tabella una entry che ha prefisso più lungo possibile che offra un matching e quindi in questo caso di esempio il router sceglierà la seconda riga per cui inoltrerà il pacchetto a 190.3.6.8 (R3), con destinazione 190.3.9.0.

La ragione di questa scelta sta nel fatto che l'informazione più specifica, ovvero la destinazione più specifica, è quella con il prefisso più lungo.

C'è un caso particolare in cui si ha una route (entry nella tabella di routing) più specifica ed è quella che si chiama default net route.

Questo avviene quando si ha un prefisso naturale 190.3 con un next hop e poi dei prefissi, delle subnet ricavate dal quel prefisso naturale con un next hop diverso e quindi una route specifica per quelle subnet, come in figura, la prima riga.

C'è anche una route di default più generale, detta appunto default route, che prende la forma mostrata in ultima riga, con una entry in cui la netmask è fatta di tutti 0 e l'indirizzo è fatto di tutti 0.

Questa è una entry molto particolare, per cui qualsiasi indirizzo di destinazione farà matching.
In sostanza questo indica al router di guardare tutte le entry e, nel caso in cui nessuna entry dà un match allora l'entry da usare è l'ultima, la default route.
In questo modo il pacchetto non sarà buttato via e 192.3.6.8 sarà il next hop.

Sequenza di routing, per riassumere
Reti direttamente collegate
Entry [righe] più specifiche
Meno specifiche (aggregate)
Default router

Lezione MOLTO importante
Il piano d'uso degli indirizzi (addressing) ed il routing, in particolare le prestazioni del routing, sono strettamente legati.
Quanto buono sarà il percorso che i pacchetti faranno e quanto grandi saranno le tabelle di routing sono strettamente legati.
Il routing e le sue prestazioni sono strettamente legati al formato degli indirizzi ed al modo in cui gli indirizzi vengono assegnati alle stazioni e quindi alle varie reti logiche.

Lezione n. 15

ARP e ICMP

Sono due protocolli di servizio della architettura di protocolli della rete Internet.

Sono protocolli che non vengono usati per trasportare dati dell'applicazioni dell'utente ma per aiutare il funzionamento della rete dei dispositivi.

- **Address Resolution Protocol**
  (Protocollo per la risoluzione degli indirizzi)

Il protocollo ARP è collocato al livello 3 subito sopra il livello 2 del modello protocollare. E' un protocollo di servizio e non trasporta dati.

Non ha le tipiche funzionalità del livello trasporto di trasportare dei dati attraverso più hop dal mittente al destinatario ma è un protocollo che aiuta il livello trasporto a funzionare a dovere ed ecco la ragione per cui è collocato a livello trasporto. Nell'immagine è messo subito sopra il livello 2 per indicare che i messaggi vengono imbustati direttamente dentro trame di livello 2.

Caratteristiche generali

→ Protocollo di tipo solicitation basato su broadcast; solicitation vuol dire che una stazione richiede ad altre di fornire informazioni, a differenza di altri protocolli dove una stazione fornisce informazioni direttamente senza che siano esplicitamente richieste. E' basato su broadcast in quanto i messaggi verranno mandati a tutte le stazioni di una rete usando i servizi di broadcast del livello 2.

→ Il protocollo ARP serve per trovare la corrispondenza tra un indirizzo di livello 3 e uno di livello 2

→ Un indirizzo di qualsiasi protocollo di livello 2 (Ethernet o 802.3, in generale indirizzi MAC) e di livello 3 (protocollo IP), ma ARP funziona su ogni tipo di protocollo

→ Il protocollo di livello 3 e di livello 2 a cui si fa riferimento viene specificato in ogni singolo messaggio

Principi di funzionamento

→ I messaggi ARP vengono imbustati direttamente dentro trame di livello 2, nel caso più come in trame Ethernet, ed in questo caso si usa il valore esadecimale sottostante nel campo Ethertype, per specificare che la trama contiene un messaggio ARP

→ Ethertype 0x0806

→ Il protocollo, nel caso di reti TCP/IP, permette di trovare una corrispondenza tra indirizzi MAC e indirizzi IP; tale corrispondenza viene memorizzata nella cosiddetta ARP cache

<MAC address> <IP address> memorizzata in cache

→ ARP cache

→ Quando c'è un pacchetto di livello 3 da inviare, da parte di una stazione

→ Se la corrispondenza tra l'indirizzo della destinazione e il corrispondente indirizzo MAC di quella stazione è nella cache, il pacchetto viene inviato usando quella informazione

→ Altrimenti, si genera una ARP Request, un messaggio ARP request per scoprire quale è l'indirizzo MAC corrispondente all'indirizzo IP di destinazione del pacchetto IP da inviare

Si noti l'esempio di Address Resolution, in cui la stazione con indirizzo IP H deve mandare un pacchetto ad una stazione con indirizzo IP G. La stazione mittente deve scoprire l'indirizzo MAC del destinatario, questo perché sappiamo che i pacchetti IP inviati alle stazioni direttamente collegate si mandano usando i servizi di livello 2, mettendo quindi il pacchetto IP dentro una trama di livello 2, ad esempio una trama Ethernet, nella quale bisogna scrivere l'indirizzo MAC della destinazione che si vuole raggiungere. La destinazione sarà nella stessa rete fisica perché in IP le stazioni mandano i pacchetti direttamente solo a destinazioni che sono solo nella stessa rete fisica. Se la destinazione non è nella stessa rete fisica, il pacchetto viene mandato ad un router che è nella stessa rete fisica del mittente e ne è il default gateway. Del destinatario, che è una stazione nella stessa rete fisica, occorre stabilirne l'indirizzo MAC.

Fatto questo sarà possibile inviare pacchetti a destinazione (vd. esempio a pagina precedente). Inoltre sarà memorizzata la corrispondenza tra indirizzi IP e MAC, della stazione destinataria, in una tabella, detta ARP Cache (cache perché è una memoria nascosta, gestita in modo trasparente e con validità limitata, per cui non ricevendo più pacchetti da G con indirizzo MAC g, l'entry della cache viene cancellata. L'indirizzo MAC può cambiare, ad esempio nel cambio di scheda). L'informazione contenuta

nell'ARP cache (G → g) deve avere validità limitata e quando scade deve essere eliminata. Quindi quando H deve mandare un pacchetto a G deve rifare una richiesta ARP.

Nello stesso tempo quello che può succedere è che anche G ed R possono aggiungere una loro corrispondenza nella tabella ARP Cache perché quando hanno visto la richiesta ARP da parte di H hanno visto quale è la corrispondenza tra indirizzo IP e indirizzo MAC di H. A G serve prendere nota di questa corrispondenza perché se H sta chiedendo l'indirizzo MAC di G è perché gli sta per mandare un pacchetto e se H manda un pacchetto a G allora molto probabilmente ad un certo punto G ne dovrà mandare uno indietro, in quanto la comunicazione è bidirezionale. Sfruttando la richiesta per apprendere l'indirizzo MAC di H non dovremo fare una richiesta ARP. Questo fa comodo sia al router come a qualsiasi altra stazione collegata alla stazione locale che ha ricevuto la richiesta ARP, l'ha elaborata, in sostanza inutilmente; ma ne può trarre un beneficio facendo una memorizzazione nella propria ARP cache la corrispondenza, in caso di bisogno. In quest'ultimo caso l'operazione di caching può o non può essere implementato.

In una richiesta ARP è implicata una elaborazione nella CPU (da interrupt) il che può causare rallentamenti di elaborazione se ci sono migliaia di stazioni collegate alla rete locale e che non sono target della richiesta. ARP è dunque un protocollo semplice, ma che può avere un impatto sulle prestazioni dei sistemi collegati alla rete e non solo sulle prestazioni della rete in quanto è mandato in broadcast.

Per questa ragione il protocollo IPv6 non usa ARP, cioè non usa il broadcast per fare la risoluzione degli indirizzi. C'è da dire comunque che la quantità di traffico ARP su una rete è molto limitata, in quanto le stazione fanno una richiesta ARP solo la prima volta che vanno a contattare una stazione che non hanno mai contattato prima. Oggi come oggi, inoltre, le stazioni non contattano stazioni sulla stessa rete locale, ma contattano stazioni che si trovano da qualche parte in Internet.

Formato dei messaggi ARP

Si vede, dalla figura a lato, come effettivamente il protocollo ARP sia un protocollo generico pensato per qualsiasi protocollo di livello 2 e qualsiasi protocollo di livello 3. I nomi dei campi sono quelli originali in inglese. Il messaggio inizia subito con un campo

“Hardware Type” e “Protocol Type”, in cui per Hardware si intende quale è il protocollo di livello 2 e per Protocol Type quale è il protocollo di livello 3 di cui si vogliono risolvere gli indirizzi. Gli indirizzi possono avere lunghezze di verse, per cui i successivi campi, HLEN e PLEN ne riguardano il valore. Operation specifica se abbiamo una richiesta o una risposta, poi abbiamo l’indirizzo di livello 2 del richiedente, che è 6 byte nell’esempio riportato, assumendo un livello 2 Ethernet, ma la lunghezza è determinata dal campo HLEN. Poi abbiamo l’indirizzo di livello 3 del mittente, quello del Protocol Type, di lunghezza PLEN., e nell’esempio 4 byte pensando ad IP. Poi ci sono gli indirizzi di livello 2 e di livello 3 del target.

- **Proxy ARP**

E’ una estensione dell’ARP, un meccanismo aggiuntivo all’ARP.
Serve per rilassare la necessità di corrispondenza 1 a 1 tra rete logica e fisica. Permette di avere una singola LIS (logical IP subnet) suddivisa su due o più reti fisiche. Abbiamo detto che questa cosa non deve essere fatta, in quanto nel protocollo IP ci deve essere una corrispondenza 1 a 1 tra reti logiche e reti fisiche. Rete logica vuol dire tutte le stazioni con lo stesso prefisso, corrispondenza 1 a 1 vuol che dire che tutte le stazioni che hanno lo stesso prefisso devono essere sulla stessa rete fisica. Il proxy ARP è un meccanismo che permette di avere lo stesso prefisso (naturale, quindi 192.168.1 nell’esempio) su due reti fisiche diverse. Una configurazione del genere è sbagliata e non funzionerebbe perché quando H1 vuole mandare un pacchetto ad H2 la richiesta ARP si propaga solo sulla rete dove si trova H1, nessuno risponde ed H1 non riesce a mandare il pacchetto ad H2. Questa è dunque una cattiva configurazione,che non funziona, ma il proxy ARP risolve questo problema.

Ma, se questa è una cattiva configurazione, perché la accettiamo, perché qualcuno si è dato la pena di creare un meccanismo che permette alla rete di funzionare comunque in questa situazione?

Si ha la necessità di avere una LIS che si espande su più reti fisiche quando c’è una crescita ed una espansione non pianificata della rete; supponiamo di avere una rete piccola che non ha tanti host collegati e dunque decidiamo di collegare tutti gli host alla stessa rete fisica e quindi diamo a tutti questi host lo stesso prefisso. Nel crescere, un

elevato numero di stazioni che cercano di trasmettere in una rete Ethernet porta ad un degrado delle prestazioni. Quindi si spezza la rete, che inizialmente era unica, in due pezzi di rete diversi separati. Abbiamo ora due reti fisiche separate, ma abbiamo la stessa configurazione di prima, con lo stesso prefisso a sinistra e a destra, il che è problematico. E' però importante far crescere le reti ed ad un certo punto, quando le prestazioni non sono più quelle volute poter spezzare la rete fisica, la rete di livello 2, in due parti, senza dover cambiare la configurazione.
Una situazione del genere, cioè avere una LIS su più reti fisiche, si può anche avere anche in caso di errore: nell'esempio a destra si è inteso usar un prefisso di 25 bit, quindi non più un prefisso naturale quindi sulla rete di destra abbiamo due reti fisiche diverse; sulla rete di destra si è deciso di usare il prefisso 192.168.1.128/25, quindi dando gli identificativi di host .129 e .130 a H2 e al router. Sulla rete di sinistra si è deciso di usare 192.168.1.0/25, quindi un prefisso di 25 bit. La notazione /25 indica, invece di specificare la netmask, la lunghezza del prefisso. Nella rete a destra il prefisso è 192.168.1. il primo bit a 1; in quella di sinistra è 192.168.1. il primo bit a zero. Sono due prefissi diversi, il che non è un problema. Ma se, configurando la stazione H1, l'amministratore di rete fa un errore usando il prefisso naturale e dice che il prefisso è 24 bit. Questo è un errore di configurazione con il risultato che H1 "pensa" che la LIS sia unica e quando cerca di mandare un pacchetto ad H2 guarda l'indirizzo IP di H" (192.168.1.129), poi guarda il proprio indirizzo e la propria netmask, fa l'operazione di AND bit a bit e gli risulta che i due prefissi siano gli stessi. Quindi questo lo porta a pensare di essere collegato direttamente per cui farà una richiesta ARP per scoprire l'indirizzo MAC di H2, ma non riceverà nessuna risposta. La rete quindi non funziona perché si ha una situazione equivalente ad avere la stessa LIS su due reti fisiche diverse.

Principio di funzionamento del proxy ARP
→ Il proxy ARP è una funzionalità che normalmente viene espletata dal router che risponde al posto di un host.
Ad esempio, H1 che vuole mandare un pacchetto ad H2 e vede che ha lo stesso prefisso, genera una richiesta ARP chiedendo chi ha l'indirizzo IP di H2. Nessuna ha quell'indirizzo per cui non c'è nessuna risposta. Allora H1 genera ancora ed ancora la

richiesta, ed a questo punto entra in gioco il meccanismo di proxy ARP: il router, che ha visto tutte queste richieste, e non ha risposto perché quella non era il suo indirizzo, “capisce” che c’è qualcosa che non va e che la stazione 192.168.1.129 sembra non essere sulla rete. Quindi il router risponde e fornisce il proprio indirizzo MAC.
A questo punto H1 riceve la risposta, ma non sa che la risposta viene dal router, e comunque invia il pacchetto IP mettendolo in una trama MAC indirizzata a quel particolare indirizzo MAC. La trama MAC si propaga ed arriva al router che, facendo il suo mestiere di inoltrare i pacchetti, farà la sua richiesta ARP per inoltrare il pacchetto sulla rete di destra. A questo punto H2 risponderà ed il router sarà in grado di inoltrare il pacchetto verso H2.
Questo meccanismo di rispondere per conto di qualcun altro si chiama proxy ARP.
Proxy in inglese è il procuratore, qualcuno che fa qualcosa al posto di un altro.

- **Reverse ARP (RARP) - ARP inverso**

E’ un diverso modo di funzionamento dell’ARP.
Caratteristiche generali:
→ Dato l’indirizzo di livello 2 di una stazione, scoprire quello di livello 3;
→ Stesso formato di messaggio del’ARP, con un valore diverso nel campo Operation;
→ Protocollo di tipo solicitation basato sul broadcast;
→ Un tempo usato da stazioni senza disco all’atto dell’avviamento [boot] per conoscere il proprio indirizzo IP, conoscendo il proprio indirizzo MAC;
→ EtherType per RARP inoltrato su trame Ethernet: 0x8035
→ Meccanismo rimpiazzato dal DHCP, per la configurazione automatica delle stazioni
→ Più flessibile

Principio di funzionamento
La stazione A che vuole scoprire quale è la corrispondenza tra un indirizzo MAC e un indirizzo IP, in questo caso il proprio indirizzo MAC, genera una richiesta RARP, la mette in una trama MAC mandata ad un indirizzo broadcast, indicando sia come indirizzo mittente sia quello destinatario il proprio. Non specifica il proprio indirizzo IP perché non lo conosce. Questa richiesta in broadcast arriva a tutti quanti e se sulla rete c’è qualche stazione configurata a rispondere, come un server configurato a

fornire un indirizzo IP alla stazione in base a quello che è il suo indirizzo MAC, può rispondere con una risposta RARP (RARP Reply) che nell'esempio parte dalla stazione E e quindi in una trama MAC va direttamente da E ad A e che contiene l'indirizzo IP di E, ma soprattutto l'indirizzo IP di A per cui A scopre il proprio indirizzo IP.

- **Internet Control Message Protocol (ICMP)**

Nel modello protocollare è posizionato al livello 3, non perché trasporta i dati (è un protocollo di servizio), ma perché serve al livello 3, facendolo funzionare meglio verificandone il funzionamento. E' volutamente rappresentato più all'interno del livello 3 rispetto all'ARP per dire che i messaggi ICMP non vengono direttamente imbustate in trame MAC ma dentro pacchetti IP.

Caratteristiche generali

→ Protocollo di servizio (non trasporta dati per le applicazioni)

→ Imbustato in IP

    → Protocol: 0x01

→ Si usa per notifica di errori e anomalie

→ Non specifica reazioni, che è lasciato alla specifica implementazione del protocollo IP

→ L'invio di messaggi ICMP non è obbligatorio

→ I messaggi possono essere ignorati

→ Casi d'uso

    → Verificare il funzionamento della rete

    → Notificare anomalie

    → Scoprire la netmask

    → Migliorare il routing

Formato dei messaggi

Campo Type, 8 bit

Campo Code, 8 bit

Campo Checksum, 16 bit

Campo con Dati specifici del tipo, 16 bit
Campo Intestazione + primi 64 byte del pacchetto con il problema notificato

Messaggio Echo
→ Usato per verificare se un host è raggiungibile
→ Sequence Number è usato per correlare messaggi Reply e Request. Quando una stazione manda un messaggio Echo Request ad un'altra stazione questa risponde con un messaggio Echo Reply.
→ Usato nell'applicazione PING, applicazione molto importante per vedere se una destinazione è raggiungibile.
Questo avviene generando dei messaggi Echo Request e vedendo se arrivano delle risposte Echo Reply. Se arrivano delle risposte la destinazione è raggiungibile, cioè i pacchetti IP possono andare fino a quella destinazione e tornare indietro. Se i pacchetti ICMP non tornano indietro non vuol dire necessariamente che la stazione non è raggiungibile, ma può voler dire anche che la stazione non sta generando risposte Echo Reply perché non è obbligatorio che la stazione li generi. Nelle reti moderne può anche voler dire che queste risposte siano filtrate da qualche dispositivo intermedio.
Un altro messaggio importante è il messaggio Destination Unreachable, che viene usato da un router quando non riesce ad inoltrare un pacchetto verso la destinazione e nel sotto-campo di Code può specificare perché non riesce a raggiungere la destinazione.

Messaggio Destination Unreachable
0 Network unreachable (il router non conosce la rete)
1 Host unreachable (il router non riesce a raggiungere l'host)
2 Protocol unreachable
3 Port unreachable
4 Fragmentation needed and DF set
6 Destination network failed
7 Destination host failed
8 Source host isolated

9 Comm. with dest. network administratively prohibited
10 Comm. with dest. host administratively prohibited
11 Network unreachable for type of service
12 Host unreachable for type of service

Un altro importante messaggio è quello di Redirect che serve per suggerire un diverso next hop verso la destinazione. Si usa questo quando una stazione deve mandare pacchetti ad un'altra e usa il suo default gate. Una stazione che non è collegata direttamente manda i suoi pacchetti al default gateway. Il default gateway si accorge che per raggiungere quella destinazione deve inoltrare i pacchetti ad un altro router. A questo punto il default gateway si accorge che la stazione può mandare i pacchetti direttamente a quel router e può mandare un messaggio ICMP di tipo Redirect alla stazione per informare la stazione che i pacchetti li può inviare direttamente al router. Questo tipo di messaggio serve solo per notificare il mittente, non può essere usato tra router.
Messaggio Redirect
→ Non per notificare un router (non è il mittente)

L'importante Messaggio Time Exceed può essere mandato da un router quando questo scarta un pacchetto perché il Time To Leave è zero.
→ Il TTL in un pacchetto IP è zero
→ Usato nell'applicazione TRACEROUTE, che è usato per verificare il percorso che i pacchetti seguono nella rete. L'applicazione TRACEROUTE mostra quali sono tutti i router attraversati. L'applicazione funziona generando pacchetti con Time To Leave prima uguale a zero, quindi il primo router lo scarterà e manderà una notifica e così l'applicazione impara l'indirizzo del primo router, poi a 1 e si impara l'indirizzo del secondo router e così via.
→ Il reassembly timer arriva a zero

Lezione n. 16

Transport layer (il livello trasporto)

- **Transport layer in Internet**

Nell'architettura di protocolli TCP e UDP hanno funzionalità tipiche del livello trasporto. Questi protocolli saranno imbustati dentro pacchetti IP, ed essi forniranno servizi a protocolli di più alto livello che in alcuni casi sono specifici di certe applicazioni (Telnet, FTP, SMTP, HTTP e RTP, SNMP) , in altri casi protocolli più sofisticati nel livello Session e Presentation del modello OSI.

TCP e UDP

→ Sono protocolli del livello trasporto e sono in alternativa l'uno all'altro;

→ Servizi diversi perché hanno caratteristiche molto diverse;

→ TCP fornisce un servizio affidabile, connesso, a byte, in quanto manda delle sequenze di byte a chi offre il servizio, di lunghezza variabile e non manda messaggi;

→ UDP fornisce un servizio di tipo Best-effort, fa del proprio meglio, ma non garantisce nulla per cui non è affidabile, fornisce un servizio di tipo datagram e quindi genera messaggi che vengono portati verso la destinazione. E' non connesso.

- **Multiplexing e demultiplexing**

→ Consente a svariate applicazioni di usare i servizi di comunicazione, nel livello trasporto. Multiplexing e demultiplexing è un concetto generale.

Il meccanismo che il livello trasporto ha per capire quale è l'applicazione di livello superiore che deve ricevere i dati e quindi consente di fare la demultiplazione si chiama Port.

La Port è un valore di due byte che si trova nell'intestazione dei messaggi di livello trasporto. Sia i messaggi TCP che UDP cominciano con 4 byte che contengono una Port sorgente (2 byte, valori da 0 a 65536) ed una Port destinazione (2 byte). Il resto dell'intestazione è diverso per TCP e UDP.

Il valore di Source port identifica quale è l'applicazione di livello trasporto che sta generando i dati contenuti nel messaggio di livello trasporto.

Il valore della Destination port identifica a quale delle applicazioni che stanno usando il

servizio di livello trasporto i dati vanno consegnati.

→ 0 ... 1023 Porte Statiche, per server, sempre attivi
→ 1024 ... 65535 Porte Dinamiche, per client, ad esempio il browser web, che usano le porte in modo dinamico

Well Known Port [note], ad uso dei server che forniscono servizi particolari. Sono definite in varie RFC.

Servizio	port	TCP	UDP	
ftp	21	X		Porta 21, servizio ftp
smtp	25	X		
http	80	X		
pop	110	X		
SNMP	161	X		
DNS	53	X	X	n.b.: 2 servizi di livello trasporto

La quintupla magica identifica nelle reti IP a quale applicazione appartengono i pacchetti, ad esempio pacchetti dello scaricamento di una pagina web, oppure di una telefonata.
Una comunicazione è univocamente identificata da:
→ Indirizzo IP sorgente
→ Indirizzo IP destinazione
→ Protocollo di livello 4 (Trasporto)
→ Porta mittente
→ Porta destinazione

- **UDP (User Datagram Protocol)**

Protocollo di livello trasporto (insieme al TCP, di cui è alternativo). Specifica originale nella RFC 768

Un protocollo datagram
→ Non connesso [connectionless, non serve instaurare una connessione prima della comunicazione, i pacchetti vengono inviati e basta]

→ Non c'è necessità di negoziazione iniziale

→ Ogni messaggio è indipendente dagli altri messaggi che appartengono alla stessa comunicazione, non sono numerati, non ce n'è uno prima di un altro

Servizio best-effort, per cui

→ I messaggi possono

- → Andare persi
- → Essere recapitati fuori ordine

L'applicazione deve tener conto di questo, a fronte della minor complessità a livello trasporto.

Le caratteristiche di UDP sembrano essere quelle del servizio IP. Il protocollo UDP sembra fornire lo stesso servizio del protocollo IP. UDP ha un valore aggiuntivo dato innanzitutto dal multiplexing, per cui molte applicazioni possono usare il servizio.
UDP non aggiunge nulla al servizio offerto da IP in termini di affidabilità e di ordine dei pacchetti, è importante averlo in quanto introduce la possibilità di fare multiplexing e di multiplexing del traffico di applicazioni diverse e quindi di avere le porte.

Valore aggiunto di UDP

→ Multiplexing

- → Port

→ Checksum per verificare l'integrità dei dati

- → Opzionale
- → Non ci sono contromisure

Se la checksum è stata implementata (può non esserlo) ed il pacchetto contiene errori, UDP scarta il pacchetto (messaggio sarebbe più corretto nel contesto UDP, in sostanza sono datagram, un insieme di byte che vengono mandati insieme e hanno una intestazione) e non lo inoltra al livello superiore.

Niente controllo di flusso e congestione

→ Non c'è adattamento alle condizioni di rete

- → Può portare alla congestione dei router

(svantaggio), di cui non si accorge; saranno le applicazioni a rallentare il traffico di dati

→ Non si tira indietro, invia sempre (vantaggio, ad uso ad esempio, di applicazioni che devono comunque inviare dati)

Perché UDP?

→ Non stabilisce connessione
  → Non c'è ritardo
  → Non c'è overhead
→ Protocollo semplice → usa meno risorse
  → Non c'è controllo dello stato della connessione
  → Piccola intestazione, per la semplicità del protocollo
→ Non c'è controllo congestione

Formato intestazione

Ci sono 4 campi, ognuno di 2 byte.

Di essi due sono opzionali.

→ Checksum e porta mittente sono opzionali
  → A zero se non usati

Casi d'uso

→ Rete affidabile
  → NFS (Network File System), condivisione dischi, non troppo lontani (Unix, Linux)
→ Affidabilità non richiesta
  → Consegna periodica di dati (un sensore)
  → Media (audio e video, le cui codifiche sono

robuste ad un certo livello di perdita)

→ Se il recupero errori può essere problematico
  → SNMP (Simple Network Management Protocol, raccoglie informazioni dagli apparati di rete; serve anche quando la rete è congestionata, serve a maggior ragione in questo caso)

→ Dati in un solo messaggio, contenuti in pochi byte
  → DNS (Domain Name Service, che un protocollo il cui servizio è quello di scoprire un nome associato ad un indirizzo)

→ Il tempo di consegna è fondamentale
  → Per applicazioni real-time [tempo reale]
  → Media (dati audio e video), le applicazioni multimediali sono quelle che hanno portato un incremento notevole del traffico UDP. Le applicazioni tradizionali che usavano UDP prima della multimedialità erano poche e generavano poco traffico.
  → Interattività, i ritardi devono essere bassi

→ Non elasticità
  → Media (dati audio e video), ci può essere tolleranza negli errori, ovvero ci può essere una qualche perdita, ma non un abbassamento del bitrate (tot Mbit/sec).

Traffico elastico, al contrario di quello multimediale, generato da applicazioni elastiche, è ad esempio il trasferimento file, per cui varierà il tempo di consegna se ci sono problemi, ma nulla più, il file alla fine sarà utilizzabile, al contrario di un video che viene trasmesso alla metà di quanto dovrebbe essere trasmesso il quale non sarà utilizzabile.

- **TCP (Transport Control Protocol)**

Protocollo di livello trasporto, alternativo all'UDP.

Caratteristiche generali

→ Protocollo di tipo connesso, si richiede l'apertura di una connessione prima di poter trasferire i dati ed è di tipo full-duplex

→ Full-duplex, una volta che la connessione è aperta, i dati possono essere trasferiti nelle due direzioni

→ Protocollo byte-oriented [a byte], il mittente manda una sequenza di byte, non organizza i dati in messaggi, come nell'UDP; in TCP l'applicazione dice "mandami questa sequenza di byte, poi quest'altra ..." ed il TCP deciderà come organizzare quei byte, che l'applicazione vuole mandare, in messaggi, in quanto il TCP userà, chiaramente, il servizio del protocollo IP e dovrà raccogliere quei byte dentro dei messaggi che vengono messi in pacchetti IP. Quindi la divisione del flusso delle informazioni in messaggi viene fatta dal protocollo TCP stesso, non dall'applicazione che usa il TCP. E questo ha implicazione sulla realizzazione stessa delle applicazioni. Se un'applicazione invia, ad esempio , prima 100 byte, poi 200 byte, il TCP può fare benissimo un unica trasmissione di 100 byte, al che in ricezione non ci si possono aspettare 100 byte e 200 byte spediti all'origine, ma ci si deve aspettare una sequenza di byte di dimensione non conosciuta a priori

→ Il protocollo fornisce un servizio affidabile

→ Byte ricevuti tutti e nel giusto ordine

Casi d'uso

Applicazioni che necessitano di affidabilità

→ FTP: File Transfer Protocol

→ SMTP, POP, IMAP: trasferimento di e-mail

→ HTTP: world wide web, pagine web

Funzionalità

→ Controllo dell'errore

→ ARQ: Automatic Retransmission Request

→ Controllo di flusso, evitare di trasmettere troppo, cioè più di quanto il mittente o il ricevente è in grado di elaborare

→ Controllo di congestione, evitare di trasmettere più di quanto la rete riesce a trasferire

→ TCP solo per applicazioni elastiche

→ Fornisce funzionalità di gestione connessioni

→ (De)multiplexing

→ Segmentazione del flusso dati, è byte oriented

→ No correlazione tra invii e ricezioni

Complesso → Costoso

→ Stato della connessione

→ Memoria

→ Elaborazione

→ Intestazione più grande

→ Overhead di trasmissione

→ Occorrono messaggi di conferma

→ Overhead di trasmissione

→ Ritrasmissioni, anche a sproposito

→ Memoria, per i paccheti ritrasmessi

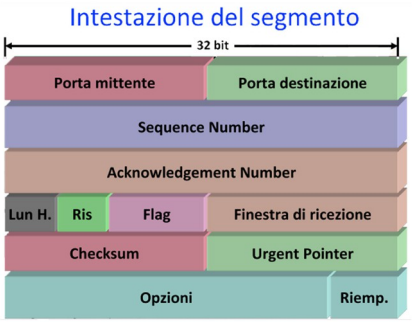
→ Ritrasmissioni inutili

→ Overhead di trasmissione

Lezione n. 17

TCP (Transport Control Protocol). Maggiori dettagli

- **Formato dei segmenti**



Intestazione del segmento

La porta mittente e la porta destinazione (2 byte ciascuna), che abbiamo in tutti i messaggi di livello trasporto, visto anche in UDP.

Seguono due campi di 4 byte, che sono numeri di sequenza: Sequence Number e Acknowledgment Number, che servono per fare il controllo dell'errore.

I numeri di sequenza, Seq e Ack, sono mostrati in figura dedicata.

I numeri di sequenza sono numeri di byte, nella figura di esempio, Seq=12 indica il numero del primo byte trasmesso da L nella sequenza di byte che trasmetterà, TCP è byte oriented. Il Ackowledgement number, Ack=15, che R mette nel suo messaggio è il prossimo byte che R si aspetta. Se L ha mandato i byte 12, 13 e 14, allora R si aspetta il 15. Questo vuol dire che ha ricevuto correttamente e nella giusta sequenza fino al 14-esimo incluso.

I numeri possono non essere gli stessi e le stazioni si mettono d'accordo su quali numeri usati e da quali partire.

L metterà in Ack in numero del prossimo byte che si aspetta di ricevere, Ack=37.

Si noti che la connessione è bidirezionale, entrambe le stazioni sono in grado di trasmettere e di ricevere.

Per questa ragione esse possono realizzare il cosiddetto ACK Piggybacking, tramite il quale manda gli ACK negli stessi pacchetti che mandano i dati.

L invia i dati ed R li riceve; R manda i dati e popola il campo sequence number con il numero di sequenza del primo byte e il campo ACK number con il numero di sequenza

del prossimo byte che si aspetta da L.
Se L non ha nulla da inviare manderà un messaggio senza dati in cui ciò che è rilevante è l'ACKnowledgment number. Anche se non ci sono dati, la stazione L deve mettere sempre un valore valido nel campo sequence number, che sarà il prossimo dato che, in questo caso, trasmetterà.

L'intestazione del messaggio TCP contiene anche dei flag, come riportato di seguito, che sono di 1 bit
Flag
→ ACK: valore valido nel campo Acknowledgement
→ PSH: i dati vanno passati al livello superiore (push), informazione per il ricevitore
→ URG: valore valido nel campo Urgent Pointer, si dice al ricevitore che nei dati c'è una porzione che è urgente e bisogna immediatamente prendere e passare al livello superiore. La differenza con il Push è che con esso si indica la ricevitore di passare tutti i dati ricevuti fino a quel momento al livello superiore. Con Urgent si deve passare solo il sottoinsieme di byte che ci sono nel pacchetto. L'urgent point dice all'interno del campo dati dove sono localizzati all'interno del campo dati. Siccome questo si usa in casi particolari il flag serve per indicare al ricevitore quando usarlo o meno.
→ Ris: campo di bit riservati
→ Lun H.: quanto è lunga l'intestazione, poichè l'intestazione ha lunghezza variabile, data da 20 byte fissi più byte di "options", con campi di riempimento
→ SYN: sincronizzare i sequence number, per decidere da quale sequence number partire, fa all'atto dell'apertura della connessione
→ Apertura connessione

→ FIN: il mittente ha finito la sequenza di byte, quindi FIN è un bit settato a 1, questo comporta la

chiusura della connessione, per la quale servono 4 messaggi

→ Chiusura di connessione

→ RST: reset di connessione, serve quando una applicazione deve chiudere immediatamente

una connessione (si pensi alla chiusura di un browser, per cui non serve più scaricare nulla,

oppure nel caso in cui nella rete ci sono molte perdite per cui il trasmettitore ed il ricevitore

non sono più sincronizzati ed allora si necessita di un reset della connessione, per cui tale

flag è settato da una stazione che si rende conto del problema)

→ Checksum: per verificare se ci sono stati errori di trasmissione

→ Finestra di ricezione: campo per realizzare il controllo di flusso

- **Gestione delle connessioni**

Cioè l'operazione di apertura e chiusura.

L'apertura della connessione avviene tramite il cosiddetto "Three-way Handshake", detto così perché richiede lo scambio di tre pacchetti.

La chiusura della connessione comporta quattro messaggi, due in una direzione e due in un'altra, con un significativo overhead.

- **Controllo dell'errore (error control) del TCP**

Ritrasmissione di tipo Go-back-N, cioè quando si perde il byte numero N, si va indietro e si ritrasmette tutto dal byte N in poi. Si fa questo perché è molto probabile che i segmenti siano persi in burst, in gruppi. E' raro perdere pochi byte.

→ Go-back-N

→ Segmenti persi in raffiche [burst]

→ Ci possono essere ritrasmissioni inutili

→ Quindi sovraccarico

→ Ritrasmissione selettiva opzionale (dipende dalla

implementazione)

Per capire se i dati sono andati persi, un metodo è quello dell'uso di un time-out, tramite un timer.

- **Controllo di flusso (Flow control)**

E' basato su una finestra detta sliding window.

I byte in sospeso sono quelli trasmessi, ma di cui non si sa se sono stati ricevuti o meno.

Il numero di byte trasmessi sono al massimo quelli che stanno in una finestra, la sliding window.

Alla fine della trasmissione dei dati della finestra, il trasmettitore si ferma finchè non riceve degli ACKnowledgment; se ha meno dati da trasmettere vengono trasmessi tutti quelli da trasmettere.

Alla ricezione di un ACK, che attesta fino a quale byte è stato ricevuto, la finestra viene spostata facendola scorrere in avanti, fino al primo byte confermato. A questo punto ci sono nuovi byte da trasmettere, per poi fermarsi di nuovo. Per una finestra sufficientemente grande il trasmettitore non si fermerà mai perché gli ACK arriveranno prima che il trasmettitore sia riuscito a trasmettere tutto.

Dimensionamento della finestra

→ Buffer del mittente

→ Byte da confermare

→ Buffer del destinatario

→ Controllo di flusso

→ Rete (buffer dei nodi)

→ Controllo congestione

Il dimensionare la finestra in base al carico della rete, cioè in base allo stato di riempimento dei buffer dei nodi della rete è il cosiddetto controllo di congestione.

Il destinatario, quando riceve i segmenti TCP deve:

→ Consegnare al livello superiore la sequenza di byte completa e ordinata, contenuta in un

buffer, di cui c'è bisogno

→ Buffer

→ Riordinare i segmenti

→ Tenere i dati fino a che sono prelevati

Il controllo di flusso è importante perché il buffer ad un certo punto si può riempire, ed a questo punto eventuali altri byte ricevuti devono essere scartati.

→ Il buffer si può riempire

→ Il campo receiver window serve per comunicare lo spazio disponibile del buffer al trasmettitore

→ La finestra di trasmissione è sempre tenuta più piccola della finestra di ricezione

- **Controllo di congestione (congestion control)**

Meccanismo importante del TCP, che serve per reagire quando c'è congestione nella rete.

Quando c'è congestione nella rete, succede che ci sono segmenti persi, quindi si hanno ritrasmissioni Go-back-N, per cui viene ritrasmesso molto, per cui si avrà ancora più congestione e più segmenti persi e più ritrasmissioni. Questo stato si perpetuerebbe, come nel 1986 quando Internet si è completamente bloccata, ed è successo quanto TCP non aveva il controllo delle congestione.

Il controllo della congestione consiste in

→ Ridurre la dimensione della finestra in caso di congestione, con il trasmettitore che cerca di indovinare se c'è congestione e lo fa attraverso le presenza di duplicate ACK o di time-out. Quando qualcosa è andato perso si tratta di congestione in quanto gli errori di trasmissione sono molto improbabili.

→ Duplicate ack

→ Time-out

→ Quando la finestra è stata ridotta, essa viene aumentata gradualmente dal trasmettitore, di un valore detto MSS, cioè Maximum Segment Size, cioè un numero di byte pari alla dimesione massima di un segmento.

L'aumento è fatto quando non c'è congestione, per cui quando la finestra è piccola, l'aumento è fatto velocemente, a livello esponenziale; quando la finestra comincia a

diventare grande, l'aumento continua pian piano e si parla di una fase di "Congestion avoidance", in cui si cerca di evitare la congestione evitando di far crescere troppo la finestra.

Quando c'è un Duplicate Acknowledgment il trasmettitore reagisce; il Duplicate ACK indica una moderata congestione, per cui la finestra viene dimezzata e si fa quindi una diminuzione moltiplicativa (multiplicative decrease); poi viene aumentata di 1 Maximum Segment Size per ogni finestra completamente ricevuta (additive increase).

Quando invece c'è un time-out, la congestione è severa e, invece di dimezzare la finestra, si porta la finestra a 1 Maximum Segment Size, cioè si rende la finestra molto piccola. A questo punto si vuole aumentare in modo aggressivo la finestra, aumentandola di 1 MSS ad ogni segmento ricevuto. Quindi per ogni segmento si ha un aumento esponenziale della finestra (exponential increase). Si aumenta la finestra fino a che la dimensione è metà di quella che era al momento del time-out. Poi si fa un aumento additivo (additive increase) per non crescere troppo.

Tutto inizia con un meccanismo detto slow start, in cui la finestra è posta ad 1 MSS, si incrementa velocemente in modo esponenziale fino alla prima perdita. A questo punto si incrementa in modo additivo per fare congestion avoidance.

Esistono svariati problemi con il controllo di congestione, ne esistono svariate varianti, di cui nessuna perfetta, specialmente su long fat pipes, cioè su reti molto veloci e molto lunghe perché i tempi di reazione sono lunghi.

Lezione n. 18

Domain Name System (DNS)

- **Principi di base**

In Internet si usano Nomi e Indirizzi

→ I nomi sono più facili da usare per gli utenti

→ Gli indirizzi sono usati per “instradare” i pacchetti

→ Inteso per essere usati dai calcolatori, dai mittenti e dai router par far arrivare il pacchetto alla destinazione. Il DNS fornisce un meccanismo per capire quale è l’associazione tra il nome che si da ad una stazione e il suo indirizzo, che verrà usato dalla rete

→ Un indirizzo ↔ più nomi, indirizzo associato a più nomi

→ Più servizi su un server (FTP, WWW)

→ Un nome ↔ più indirizzi, un nome associato a più indirizzi

→ Bilanciamento di carico [load balancing]

→ Content caching, i contenuti di un server vengono duplicati e tenuti vicini a dove si trova l’utente

La corrispondenza tra nomi ed indirizzi può essere mantenuta in locale, ad esempio in un file.

→ File /etc/hosts

stazioni di tipo unix

→ 127.0.0.1 localhost

→ 223.1.2.1 alpha

→ 223.1.2.2 beta

→ 223.1.2.3 gamma delta

→ Metodo non pratico su reti di grandi dimensioni

→ Non è scalare, cioè non è in grado di

operare su larga scala

DNS fornisce una soluzione gerarchica, che è in grado di operare in situazioni di grosse dimensioni, come TCP.
Quindi DNS funziona su larga scala
→ E' gerarchica dal punto di vista della sintassi dei nomi
→ Gerarchica nell'assegnazione dei nomi, delegata ad autorità responsabili, per cui si crea una gerarchia di autorità a livello territoriale che assegnano i nomi
→ Gerarchica nella risoluzione dei nomi: i server che si usano per risolvere i nomi sono organizzati in una gerarchia. I server costituiscono un database distribuito

Un database distribuito
→ La gerarchia di nomi è usata per trovare l'informazione nella gerarchia dei server
→ L'organizzazione gerarchica dei server e dei nomi è slegata dalla gerarchia di rete (indirizzi e routing). La rete ha una gerarchia di indirizzi, nel modo in cui i pacchetti vengono portati in giro per la rete, il modo in cui viene fatto il routing. La gerarchia del routing nella rete è indipendente da quella del DNS. In altre parole i server in un certo livello gerarchico del DNS non devono essere collocati con lo stesso livello gerarchico degli indirizzi.

- **Gerarchia dei nomi di dominio e dei server**

Un nome completo è detto Fully Qualified Domain Name (FQDN)
→ E' un nome con un numero variabile di livelli gerarchici, con il punto che separa domini gerachici diversi, che possono avere un significato, come mostra a lato. Quindi www.dauin.polito.it è un nome di dominio, con una sua gerchia

Sintassi dei nomi di dominio
→ Basata su codifica ASCII
→ La sintassi può essere in una versione internazionalizzata, dove per rappresentare quei caratteri più sofisticati si usa una codifica con due caratteri ASCII insieme

Top Level Domain (TLD) è la parte che sta più a destra del nome di dominio. Ci sono due tipi di Top Level Domain, il Country Code TLD ed il Generic TLD

→ Country Code TLD (ccTLD), riferito ad una nazione

→ .it, .fr, .uk, …

→ Generic TLD (gTLD)

→ Non vincolati

→ .com .net .org .info

→ Vincolati (sponsorizzati)

→ .gov .edu .aero .coop

Dominio di secondo livello è la successiva parte del nome di dominio, dopo il Top Level Domain

→ Può rappresentare organizzazioni o aziende in alcuni TLD

→ .polito.it, apple.com

→ Può fornire una caratterizzazione (di tipo organizzativo) ulteriore per altri TLDs

→ .bt.co.uk ucl.ac.uk, in questo caso .co differenzia organizzazione commerciali da .ac che sono organizzazioni accademiche; ucl è University College of London

La registrazione dei nomi, o assegnazione dei nomi è inizialmente gestita in modo centralizzato dallo IANA

→ IANA: Internet Assigned Numbers Authority

→ Esso decide TLDs, poi delega altre operazioni

→ Esso delega

→ Assegnamento (registrazione)

→ Gestione Server di dominio; il server di dominio conosce la corrispondenza tra nomi ed indirizzi per tutti quei nomi che appartengono a quel dominio, cioè per tutti quei nomi che finiscono, ad esempio, per .it, parlando del server responsabile del dominio .it. La stessa cosa varrà per il dominio di secondo livello, ad esempio per il dominio di secondo livello apple.com ci sarà un server responsabile della corrispondenza tra nome ed indirizzo di tutti quei nomi che finiscono per apple.com.

Gerarchia dei server

→ C'è un server per ogni dominio di secondo livello ed in qualche modo c'è una relazione tra il server del dominio di primo livello corrispondente e quelli di secondo

livello

→ Ogni server di dominio di secondo livello conosce gli indirizzi per gli host con nomi nel dominio

→ Ogni server di dominio di secondo livello conosce gli indirizzi dei server che sono responsabili dei domini di livello inferiore

→ Quindi i server sono organizzati nella stessa gerarchia con cui sono organizzati i nomi

→ I server sanno quali sono queste corrispondenze grazie ad una configurazione manuale. Il dNS, cioè, è fortemente basato sulla configurazione manuale

Oltre ai vari server dei singoli domini, esiste un server che si chiama il Root Server, che, essendo di particolare importanza, non è unico, ma ce ne sono diversi, con nomi tipo a.root-server.net, b.root-server.net, c.root-server.net, fino a m.root-server.net.

Root Server, sono a livello più alto della gerachia DNS e conoscono gli indirizzi di tutti quei server che sono responsabili dei domini di livello Top.

→ I root server hanno nome [a-m].root-server.net

→ Sono gestiti da IANA

→ Conoscono gli indirizzi dei server dei TLD

→ ccTLDs: it fr uk

→ gTLDs: com gov aero

Server dei TLD

→ Conosce l'indirizzo dei server del prossimo livello

→ rai.it co.uk nwu.edu

Server del secondo livello, server che conoscono le stazioni che hanno un nome nel dominio di secondo livello; inoltre, siccome alcune organizzazioni potranno avere ulteriori livelli di dominio, i server di secondo livello contengono anche gli indirizzi dei server di livello successivo

→ Nomi delle stazioni

→ www.rice.edu, ftp.nasa.com

→ Server del livello successivo

→ cs.rice.edu, technion.ac.il

Dunque, come spiegato finora, i server sono organizzati in una gerarchia, di cui viene mostrato un esempio nella figura a lato.
La configurazione è fatta a mano, aggiungendo le informazioni nei server quando c'è un nuovo nome di dominio.

La gerarchia è una gerarchia logica, infatti

→ Lo stesso server può essere ospitato su più calcolatori, ci possono essere più copie di uno stesso server, c'è una sola copia logica; in pratica occorre ridondanza dei dati

→ Un calcolatore può ospitare più server DNS, ad esempio del dominio .com e del dominio .net.

→ Sincronizzazione con il server primario

DNS Hosting

→ Il calcolatore che ospita un server DNS non deve essere "vicino" agli host del dominio. Il calcolatore che ospita il server responsabile di rai.it, ad esempio, non deve necessariamente trovarsi sulla rete della Rai, dove c'è l'host www.rai.it, ftp.rai.it eccetera. Esso può essere ovunque. E questo da la possibilità di realizzare servizi di DNS Hosting, per cui un service provider può fornire supporto per la registrazione dei domini

→ Service provider fornisce supporto per la registrazione dei domini. Volendo registrare il dominio pippo.it, se non lo usa nessuno, occorre creare un server, installarlo, configurarlo e dire quale è il suo indirizzo IP affinché sia messo nel database del dominio .it. Tale server con pippo.it deve sempre essere raggiungibile affinchè il nome possa essere risolto, il che può essere difficoltoso in alcune situazioni, ad esempio "in casa". Quello che si fa è quindi andare da un service provider che permette la registrazione dei nomi; il service provider si fa carico di notificare il gestore del dominio .it e a questo punto il service provider mi fornisce un servizio di DNS Hosting e installa il server per il dominio pippo.it., di cui fornisce l'indirizzo al gestore del dominio .it in modo che quando serve trovare un indirizzo corrispondente ad un

nome che finisce per pippo.it il server è pronto a fornire l'indirizzo.

→ Un service provider offre DNS hosting

- **Risoluzione dei nomi**

A tale fine, la stazione (client), deve avere le seguenti informazioni:

→ Indirizzo di uno o più server DNS

→ Possono essere ovunque

→ Normalmente sono vicini

→ Normalmente il server è responsabile del dominio della stazione, ovvero del dominio in cui la stazione si trova

→ Dominio di default (opz.)

Risoluzione dei nomi

Supponiamo che dalla stazione tilie.polito.it si voglia accedere ad un server di nome ftp.technion.ac.il.

La stazione potrebbe avere una cache interna degli indirizzi, ma se non ha l'indirizzo corrispondente a quel nome, allora la stazione manda una richiesta DNS al proprio server, server che è contenuto nella configurazione della stazione. Probabilmente questo server di riferimento non saprà l'indirizzo corrispondente a ftp.technion.ac.il, allora il server è stato configurato manualmente a sapere quale è l'indirizzo del root server per cui farà la stessa richiesta al root server. Il root server guarda il nome che si deve risolvere, guarda quale è il dominio Top Level, e, sapendo quale è il server corrispondente al dominio Top Level "il", manda la richiesta a quello. A questo punto il server del dominio "il" guarda il dominio di secondo livello "ac", di cui conosce l'indirizzo grazie alla configurazione, per cui passa la richiesta al server responsabile di "ac" che vede che il prossimo dominio è technion e passa la richiesta al server responsabile del dominio "technion". A questo punto, il server responsabile del dominio "technion" saprà sicuramente quale è l'indirizzo associato al nome ftp perché quello è il suo dominio, per cui, quando qualcuno ha deciso di chiamare una stazione ftp.technion.ac.il, ha scritto nel server DNS di quel dominio quale è l'indirizzo della stazione. Quindi il server può fornire l'indirizzo a chi lo ha chiesto a lui, quindi al server del dominio "ac", il quale lo passa indietro e così via fino a quando la stazione ha l'indirizzo IP del server che l'utente vuole raggiungere e può mandare dei pacchetti IP.

Questa modalità di risoluzione dei nomi si chiama ricorsiva.
Ne esiste una detta iterativa, dove il primo server chiede quale è l'indirizzo dell'altro server, si ha dunque una specie di rosa di richieste che partono tutte dallo stesso server.

La modalità ricorsiva si usa perché tutti i vari server che ricevono le risposte possono memorizzare temporaneamente le informazioni che hanno ricevuto e quindi fare una operazione di caching della rcorrispondenza tra nome ed indirizzo. L'operazione di caching viene fatta anche nelle stazioni, in modo da avere già l'indirizzo in caso di ulteriore richiesta da parte dell'utente.

Caching
→ Memorizzazione temporanea [caching] di nomi/indirizzi
→ Anche nelle stazioni
→ Velocizza le interrogazioni da parte dei server, che partono dalla cache
→ Risposte non-authoritative, cioè il server che da la risposta non ha autorità per quella risposta, ma viene fornito l'indirizzo di un server che ce l'ha
→ Indirizzo di un server authoritative
→ Risoluzione iterativa, è una modalità che non permette di sfruttare il meccanismo del caching

Tipi di record DNS (il DNS è un database)
→ A → il tipo di record A (Address) contengono la corrispondenza tra nome e indirizzo
→ MX → da dominio di posta a indirizzo di mail server, occorre dunque conoscere quale è l'indirizzo del server di posta responsabile di quel dominio di posta. E' una problematica diversa che può comunque essere gestita dal DNS, in quanto database contiene anche questo tipo di informazioni
→ CNAME → da alias a nome canonico, canonic name
→ NS → da nome di dominio a indirizzo del suo server
→ Le richieste (query) specificano il tipo di record voluto, ad esempio di può fare una query per record di tipo A oppure di tipo MX ecc.

Inverse Resolution [risoluzione inversa]

→ Dato un indirizzo IP, trovare il nome canonico

→ Stessa procedura e gerarchia di server

→ Si usa un Record PTR (pointer), legato ad un nome fittizio CHE HA UN NOME CANONICO

→ Si costruisce un Nome di dominio fittizio

x.y.z.t.IN-ADDR.ARPA

→ Per esempio, 130.192.3.24 query (PTR)

24.3.192.130.in-addr.arpa

→ Registrazione di polito.it

→ DNS server (130.192.3.21)

→ Address range (130.192.0.0)

- **Formato dei messaggi**

E' molto articolato; ci sono una serie di campi a singolo bit.

Campo parametri

→ 0-operation [operazione]

(0=query, 1= response)

→ QT-query type [tipo rich.]

→ 0: standard

→ 1: inverse

→ A-autorithative answer

→ T-Truncated [troncato]

→ D-Ricursione desiderata

→ R-Ricursione disponibile

→ G-Non usato

→ RT-Response type [tipo risp.]

→ 0: no errore

→ 1: errore di formato nella richiesta

→ 2: fallimento server

→ 3: nome non esistente

Lifetime [durata]

→ Associato ad ogni informazione

→ Usato per gestire l'invecchiamento nelle cache

- **Estensioni del DNS**

Dynamic DNS (DDNS)

→ Aggiornamento automatico dei server DNS

→ Utile in associazione alla configurazione dinamica degli host

→ La stazione comunica il proprio indirizzo al server

→ Possibile rischio di sicurezza

DNS Security Extensions (DNSSEC)

→ Per proteggersi da attacchi basati sul DNS

  → P.e., cache poisoning [inquinamento cache]

  → Dati DNS fasulli

→ Autenticazione risposte

  → Niente cifratura

→ Firma digitale dei record

  → Crittografia a chiave pubblica

  → Certificati digitali

→ Record specifici DNSSEC

Lezione n. 19

Protocolli di livello applicativo e posta elettronica

- **Livello applicativo**

Nell'architettura di protocolli, il livello applicativo può utilizzare direttamente i servizi del livello TCP o UDP, ed avere funzionalità di quelli che sono i livelli OSI Session, Presentation e Applicazion, oppure appoggiarsi a dei protocolli che forniscono funzionalità specifiche di livello Session (RPC) o Presentation (XDR) ed avere funzionalità specificatamente di livello Application (NFS, protocollo Network File System, per la condivisione dei dischi).
Per quanto riguarda i protocolli di posta elettronica, essi sono basati direttamente sui servizi del TCP e forniscono sia funzionalità di livello Session e Presentation, sia Application.

Le applicazioni nell'architettura TCP sono basate sul paradigma detto client-server
→ Il server [servitore] è un programma sempre in esecuzione
→ Il server aspetta richieste
→ Il client [cliente] inizia la comunicazione, contattando il server
→ Il server ha indirizzo IP (nome) e porta noti
→ Porta statica
→ Porta di tipo standard, nota a priori
→ Paradigma che è un modello tradizionale in Internet
→ FTP, WWW, e-mail

Si è andato affermando un ulteriore paradigma, il paradigma peer-to-peer (P2P)
→ In esso non c'è un ruolo predefinito, pur essendo l'applicazione fatta da tanti programmi, processi su macchine diverse che comunicano, ma non con un ruolo predefinito, tipo server che viene sempre contattato e client che contatta
→ Ogni host può contattare o essere contattato
→ Cioè, opera sia da client sia da server
→ Le soluzioni P2P usa no uno o più server (o super peer) che sono necessari per conoscere gli altri
→ Modello più nuovo

→ VoIP, emule (file sharing), Skype (voce e videoconferenza, basato su protocolli proprietari)

Una caratteristica comune ai protocolli di livello applicativo è che essi sono protocolli testuali.
Questo vuol dire che i protocolli testuali usano una codifica testuale dei messaggi, i messaggi sono sequenze testuali di caratteri
→ Codifica inefficiente
→ Facile ricerca guasti (i messaggi sono leggibili)
→ Non è necessario che siano supportati da un analizzatore di protocollo

- **Architettura per il recapito dei messaggi di posta elettronica**

La posta elettronica è basata sul fatto che un utente riceve i propri messaggi di posta elettronica su un server, detto "Mail server" o anche "Post office". I messaggi permangono sul server finché l'utente, tramite un programma di lettura della posta elettronica (un client), non va a contattare il server, per recuperare il messaggio (a destra nell'immagine).
Serviranno dunque dei protocolli per permettere ad un utente di recuperare i propri messaggi di posta elettronica sul server.
Quando un utente vuol mandare un messaggio di posta elettronica (a sinistra nell'immagine), il programma di posta elettronica contiene un client di posta che va a collegarsi ad un Mail server per trasferire il messaggio di posta al server, il quale aiuterà l'utente a distribuire opportunamente i suoi messaggi di posta elettronica, andando a cercare quale è il mail server che mantiene la casella postale del destinatario del messaggio di posta e trasferire tale messaggio.
Il protocollo per spostare il messaggio dall'utente al server si chiama SMTP, Simple Mail Trasnfer Protocol.
I protocolli per recuperare i messaggi dal server sono diversi, uno è detto POP, Post Office Protocol, IMAP, Internet Message Access Protocol, HTTP, Hyper Text Transfer Protocol, che è anche il protocollo che si usa per il web, ma è anche rilevante per la posta elettronica.

- **Protocolli per il trasferimento dei messaggi**

Si tratta di trasferire messaggi dalla stazione dell'utente ad un server e da questo

verso il server di destinazione, il protocollo è il protocollo SMTP.
Si basa sul protocollo di livello trasporto TCP sull'uso porta 25, per cui il client apre una connessione TCP sul server che è in attesa sulla porta 25. Il client deve solo sapere quale è l'indirizzo del server, informazione fornita al client in fase di configurazione della posta elettronica. Il server per la posta in uscita è detto anche outgoing mail server.

SMTP: Simple Mail Transfer Protocol
→ Testuale
→ Client-server
→ TCP - porta 25, di default
→ Aperta dal client
→ Command-response (il client manda dei messaggi ed il server risponde)
→ Status code, che dice al client se il server sà soddisfare la richiesta oppure no.

Sessione SMTP, come funziona
Il client apre una connessione TCP alla porta 25 del server, che manda un messaggio al client, tramite uno Status Code, ad esempio il codice 220, che è OK; il client manda un messaggio "HELO" seguita dal nome di dominio della stazione mittente, HELO perché i messaggi sono di 4 byte. Il serve manda un messaggio 250, che, cominciando per 2, indica esito positivo e segue una frase leggibile, dipendente dall'implementazione del server. Quindi la stazione manda il comando MAIL FROM indicando il mittente e il serve può dire 250 che indica OK. Poi il mittente banda il destinatario come messaggio RCP TO seguito dall'indirizzo. Questa fase è detta fase di "handshaking".
A questa fase segue quella di trasferimento dati: il client manda un messaggio di comando DATA, il client risponde ok con il messaggio 354, seguito letteralmente dalla frase "Enter mail, end with "." by itself". A questo punto il client invia un messaggio che è formato da una sequenza di caratteri ASCII (127 valori), dice al server di aver finito il messaggio con "." e un a capo. A questo punto il server può dire 250 message accepted. Il client manda un messaggio "QUIT" per chiudere la connessione alla quale

il server manda un messaggio 221 a chiusura connessione.
Il programma telnet permette l'apertura di una connessione TCP ad una certa porta e permette di vedere quali caratteri transitano sulle connessioni TCP.

MISURE ANTISPAMMING
Messaggi mandati in modo non richiesto.
Innanzitutto il server non accetta messaggi da chiunque per chiunque.
Vengono effettuati dei controlli.
Il server risponde di gestire messaggi del dominio polito.it, con il messaggio 571 di destinatario proibito.

Il server deve permettere di mandare fuori messaggi del proprio dominio.

Il server può fare dei controlli, ad esempio l'indirizzo del client, che sta in un certo range di indirizzi.
Oppure ci sono meccanismi di autenticazione del client, a mezzo username e password.

Il server mittente troverà il server destinatario usando il DNS, facendo una risoluzione del DNS del dominio di posta del destinatario, per un record di tipo MX, Mail Exchange.

Formato dei messaggi di posta elettronica
→ Sequenza di caratteri ASCII, separati in righe
→ Eventualmente righe di lunghezza limitata
Ci sono campi di intestazione.

Le immagini
→ Possono essere inviate, ma devono essere codificate come sequenze di caratteri
→ Per esempio base64
→ Il destinatario deve sapere che il messaggio contiene una immagine codificata, deve leggere le informazioni sapendo la codifica, il tipo e decodificare il tutto

Per gestire tutto questo, si usa lo standard MIME.

Multipurpose Internet Mail Extensions: MIME

→ Esso prevede intestazioni aggiuntive, la versione, il tipo, il nome immagine, la codifica ed il contenuto

MIME Version: 1.0

Content Type: image/png; name=”image001.png”

Content Description: image001.png

Content Transfer Encoding: base64

iVBORw0KGgoAAAANSUhEUgAAAKgAAABDCA

xAAADsQBlSsOGwAAABl0RVh0U29mdHdhcm

EEQXBBxR19URQF3OEzRo0rKvoeCYm4xJen

Content-Type, contenuti di tipo diverso

→ text

→ plain, html

→ image

→ jpeg, gif, png

→ audio

→ video

- **Protocolli per accedere ai messaggi**

Per accedere ai propri messaggi si può usare Webmail, a lato google mail.

Per fare questo quello che serve è

→ Web server in esecuzione sul calcolatore che ospita il mail server

→ Fornisce accesso ai messaggi tramite interfaccia web

→ I messaggi restano sul server

Pro e contro di webmail

→ Ideale quando non si usi un proprio PC

→ Disponibile ovunque

→ Utilizzabile solo se si ha una connessione Internet

Post Office Protocol: POP

→ Utenti di un singolo PC

→ I messaggi sono spostati sul client

→ Disponibile off-line

→ Protocollo testuale

→ TCP alla porta 110

Sessione POP

+OK implica qualcosa che va a buon fine

-ERR implica un avvenuto errore

Ai messaggi segue sempre qualcosa di leggibile.

Essendo in una situazione command response, si ha una serie di comandi, di cui USER serve ad autenticarsi, insieme al comando PASS di password

A questo punto, ad autenticazione avventuta, parte una sessione SMTP, a cui seguirà una fase di chiusura.

Sessione SMTP (vd. appunti)

Internet Message Access Protocol: IMAP

→ Utilizzatori di più PC

→ Per esempio 1 PC al lavoro, 1 PC a casa

→ Protocollo testuale

→ TCP alla porta 143

Unisce il meglio dei due mondi, POP e webmail

→ Disponibile off-line

→ I messaggi rimangono sul server

→ In gerarchia di cartelle

→ Sincronizzazione con copia locale

Lezione n. 20

World Wide Web

- **La ricetta del successo del World Wide Web, WWW**

“Ragnatela”su scala mondiale di documenti, in cui le pagine web contengono riferimenti ad altre (links). Il riferimento può essere ovunque in Internet.
Il web può essere utilizzato da chiunque.

Gli ingredienti
→ Server
→ Client (browser)
→ Formato dei documenti (Linguaggio HTML)
→ Identificatori (URI, Universal Resource Identifier)
→ Un protocollo (HTTP)

Pagine web
→ Sono oggetti vari composti in una struttura
→ Sono pagine multimediali
→ Scritte nel linguaggio Hyper-Text Markup Language (HTML), che ci permette quali oggetti sono sensibili e come la pagina deve essere visualizzata; il browser fa il rendering della pagina
→ Alcuni oggetti sono “sensibili”

Universal Resource Identifier (URI), è un identificatore di oggetti
→ Identifica ogni oggetto (risorsa), nel web
→ Dice anche dove trovare l’oggetto
    → URL: Universal Resource Locator [localizzatore]
→ Come recuperare la risorsa tramite il sup protocollo
    → Protocollo da utilizzare
Il protocollo http usa come standard la porta 80.

Browser web
→ “Visualizza” pagine web

→ Ne scarica una nuova a seguito di un click

→ Può usare vari protocolli

→ HTTP, FTP, SIP (per fare una telefonata)

→ Oggetti di vario tipo

→ Images, video, sound, plug-in

Fattori del successo

→ Intuitivo

→ “Colorato”

→ Multimediale

- **Hypertext Transfer Protocol (HTTP)**

E’ il protocollo usato per tarsferire gli ipertesti delle pagine web, le descrizioni scritte in HTML.

Caratteristiche

→ Testuale, sequenze di caratteri

→ Basato sul paradigma Client-Server

→ Basato su TCP

→ Apertura da parte client

→ Normalmente porta 80 (server), specificata nella URI

Il protocollo HTTP è Client-Server, in particolare è di tipo Request (Client) - Response (Server)

Il client prima di tutto apre una connessione TCP, poi fa una richiesta HTTP, usando il protocollo HTTP, e il server fornisce una risposta. Poi il client può fare una nuova richiesta, riceve una nuova risposta e così via. Dopo una serie di interazioni può visualizzare la pagina web.

Il protocollo HTTP è stateless, cioè è senza memoria; il server risponde ad ogni richiesta indipendentemente dalle richieste precedenti. Quando al server arriva la seconda richiesta non sa se è legata alla precedente, anche se è la stessa connessione TCP. Ogni richiesta è una storia a parte, la richiesta chiede un oggetto, il server prende l’oggetto e lo passa, l’oggetto è specificato dalla URI.

Questo approccio ha degli svantaggi se si vuole tenere traccia di quello che l’utente ha

fatto prima, ad esempio in caso di shopping on-line. L'HTTP nella sua versione base non permette di fare questo, lo fa in versioni successive, con l'aggiunta di qualche meccanismo.

Il formato dei messaggi HTTP

Ci sono delle richieste e delle risposte, in sequenze di caratteri, organizzate in righe. La prima riga si chiama request line nelle richieste oppure status line nelle risposte. Le righe sono terminate da un a capo, dato dai caratteri CR LF.

Dopo la prima riga, abbiamo una serie di campi intestazione, nel formato <nome>:< valore> e terminato con un a capo. L'intestazione termina con una riga vuota in cui c'è solo un a capo, CR LF.

Segue il corpo del messaggio, ad esempio in una richiesta di una pagina web il corpo è vuoto, in una risposta ci sono dei contenuti, che sono la pagina web.

In dettaglio la request line:

Request Line

<metodo> <URL> <versione>

→ Metodi

→ GET, POST, PUT, HEAD

→ Versione

→ HTTP/1.0

→ HTTP/1.1

In particolare GET per prendere un oggetto, POST per dare al server un oggetto nuovo, PUT per modificare un oggetto esistente sul server.

Mettere oggetti sul server significa ad esempio mandare i dati al server di un form di una pagina web.

Si può prelevare solo l'intestazione di un oggetto, con il metodo HEAD, per acquisire informazioni sull'oggetto.

Dopo il metodo abbiamo la URL (o URI) che identifica l'oggetto che vogliamo chiedere al server o che vogliamo dare al server, e poi la versione del protocollo.

Ci sono attualmente due versioni HTTP, una differenza è quella che nella versione 1.0 ogni richiesta e successiva risposta usano una connessione TCP che veniva chiusa

alla risposta. Nella versione 1.1 permette di mantenere la connessione aperta, detta connessione persistente.

Esempio di messaggio Request
GET /baldi/pubs/index.htm HTTP/1.1
Host: staff.polito.it
User-Agent: Mozilla/5.0
Accept: text/html,application/xhtml
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive

La GET chiede un oggetto al server.
“Connection: keep-alive” significa di chiedere al server di mantenere la connessione aperta. Il server deciderà se è il caso o no.

Status Line, del server
<versione> <status code>
→ 200 OK
→ 301 Moved Permanently
→ 400 Bad Request
→ 404 Not Found
→ 500 Internal Server Error

Esempio di messaggio Response, dal server
HTTP/1.1 200 OK
Date: Sat, 15 Jun 2013 21:17:27 GMT
Server: Apache
Accept-Ranges: bytes
Content-Length: 93589
Keep-Alive: timeout=15, max=97
Connection: Keep-Alive

Content-Type: text/html
<html><head>

“Content-Type: text/html “ è un campo molto importante, perché dice al client come gestire il contenuto che sta arrivando nel body.

Categorie di status code
→ 2xx successo
→ 3xx redirezione
→ 4xx problema con il client
→ 5xx problema con il server

- **Caratteristiche avanzate**

Autenticazione
Il browser chiede di inserire utente e password; questo succede quando il server richiede una autorizzazione a seguito di una richiesta del client per utilizzare una certa risorsa.
Il client manda un messaggio GET al quale il server risponde con uno status code “401 Unauthorize WWW-Authenticate: <chig>”. In questo caso il server ha incluso anche una autenticatione “WWW-Authenticate“ con una “Challenge”, che è una sequenza di caratteri casuale che il client può usare per autenticarsi.
A questo punto il client fa apparire la mascherina di autenticazione e poi ripete la richiesta GET con le sue credenziali di autenticazione, messe in un campo che si chiama “Authorization“. Per rendere intellegibile le credenziali si usano algoritmi crittografici che operano usando la password sulla “Challenge”. Se l’autorizzazione ha successo il server risponde “200 OK” e invia l’oggetto richiesto.
Da quel momento il client quando comunica con il server manda sempre le credenziali di autenticazione.
Questo tipo di autenticazione è debole, con la variante https di http si usano meccanismi di autenticazione più robusti, più difficili da attaccare.

Cookie
→ Meccanismo per consentire al server di identificare un client prima servito

→ Rende interazione stateful [con memoria] possibile (l'http è di per sè stateless, senza memoria)

→ Si può realizzare il carrello della spesa

→ Preferenze dell'utente

A lato si nota che il cliente chiede al server l'oggetto abc, a cui il server risponde OK (e lo invia), ma invia anche un cookie di valore "xyz", una sequenza di caratteri, che viene memorizzato dal client.

Ad ogni nuova richiesta (GET) da parte del client viene aggiunta una intestazione con la sequenza del cookie.

Il server correla la richiesta a quella di prima e risponde con l'oggetto. Se il cliente facesse una richiesta senza il cookie, probabilmente otterrebbe una risposta con un nuovo cookie.

- **Miglioramento delle prestazioni**

Caching [memorizzazione temporanea]

→ Oggetti memorizzati sul client

→ E se cambiano?

→ Metodo HEAD

→ Campo dell'instazione

If-modified-since: <data>

→ 304 Not Modified

Server proxy

La caching ha vantaggio solo se l'utente accede più volte allo stesso oggetto, invece, usando dei server proxy, il proxy può fare il caching e quindi se un utente chiede un oggetto, il proxy lo va a chiedere al server, quando lo ottiene ne memorizza una copia e se un client chiede lo stesso oggetto viene restituito direttamente.

Questa idea può essere ampliata creando copie dei contenuti vicino agli utenti, grazie al Content Delivery Network (CDN).

Content Delivery Network (CDN)

Quando un utente fa una richiesta, riceve una risposta da un server vicino a lui che ne ha una copia.

Lezione n. 21

Assegnazione degli indirizzi e indirizzi privati

- **Assegnazione di indirizzi alle organizzazioni**

Principi di base

→ L'indirizzo IP di ogni stazione deve essere unico

→ Il coordinamento deve essere centralizzato, consiste nel decidere sull'utilizzo degli indirizzi, chi può usare e quale indirizo

→ IANA: Internet Assigned Numbers Authority

→ Meccanismo di delega, affinchè tutti non debbano contattare lo IANA

Anche se abbiamo un ente che ha responsabilità su un solo continente, questi enti delegano gli ISP, attaverso una ulteriore delega:

→ Internet Service Provider (ISP)

→ Centro sistemi informativi

→ Gestore informatico di dipartimento/laboratorio

Tutto questo risulta pesante per ogni nuova sottorete (LAN), per cui lo IANA ha definito l'esistenza di indirizzi privati.

- **Indirizzi privati**

Qual è l'idea?

→ Chiunque li può utilizzare senza chiederne conto

→ Ci saranno duplicati

→ Non possono essere usati "su" Internet

→ Solo dove si è sicuri che siano univoci

I dispositivi comunicano tra di loro in un ambito privato.

Qual è il problema nel comunicare con il resto del mondo? Il problema è che i router inoltrano i pacchetti sul percorso più breve, quindi, avendo due stazioni B, come in figura, che hanno lo stesso indirizzo IP, nel momento in cui A vuole mandare un pacchetto a B, i router inoltreranno il pacchetto sul percorso più breve, per cui sarà raggiunta solo la stazione B più vicina.

Indirizzi privati

→ 10.0.0.0/8

→ 1 prefisso di classe A

→ 172.16.0.0/16 - 172.31.0.0/16

→ 16 prefissi di classe B

→ 192.168.0.0/24 - 192.168.255.0/24

→ 256 prefissi di classe C

Volendo usare un indirizzo in ambito ristretto, in ambito locale, perché non ne utilizziamo uno qualsiasi? Cioè, per collegare le stazioni di una rete casalinga, perché non posso usare un indirizzo qualsiasi? La ragione sta nel fatto che, se per qualche ragione, da qualche parte, in Internet, c'è un host pubblico che ha lo stesso indirizzo di un host privato, i pacchetti spediti da un host locale all'host locale con stesso indirizzo di una qualche host pubblico nella rete Interne, quest'ultimo non potrà mai essere raggiunto.; questo è il concetto di occultamento della destinazione. E' quindi importante aver definito degli indirizzi da usare a livello locale che nessuno cercherà di usare per dei server pubblici.

La struttura di reti private è definita come intranet, che può essere sia pubblica che privata.

- **Intranet pubbliche e private**

Intranet

→ Rete IP privata, cioè una rete privata, in tecnologia IP, privata in quanto appartenente ad una azienda o ad una organizzazione (una università ad esempio)

→ Di proprietà di una azienda o organizzazione

→ Host privati, cioè l'intranet ha host che possono comunicare solo all'interno dell'organizzazione, che hanno indirizzi privati

→ Indirizzi privati

→ Host pubblici, in quanto devono poter comunicare con tutto il resto del mondo

→ Indirizzi pubblici

Una intranet è di norma organizzata in due parti, una parte pubblica ed una parte privata. Queste due parti sono topologicamente collegate come mostrato in figura, cioè l'intranet pubblica sarà collegata ad Internet, quindi alla rete pubblica in tecnologia IP,

alla rete che non è di proprietà della particolare organizzazione in oggetto; la intranet privata sarà collegata alla intranet pubblica ed i confini tra la parte privata e la parte pubblica della intranet saranno ben definiti.
Nella intranet privata, avendo stazioni che comunicano solo tra di loro, possiamo usare indirizzi privati.
La intranet privata è molto più grande della intranet pubblica. Le poche stazioni della intranet pubblica, con indirizzi pubblici, sono di norma dei server.
Quindi una stazione della intranet privata che volesse comunicare con il resto del mondo (B della intranet privata con A), non ci riuscirebbe, ma non nel senso di non poter mandare pacchetti, in quanto i pacchetti li può mandare e li potrà mandare ad un indirizzo pubblico di un certo server che sta da qualche parte, ma la stazione B non vedrà nessuna risposta dal server per cui la comunicazione fallirà. Quello che succede è che quando il server A, con il suo indirizzo pubblico, risponderà a B, che ha un indirizzo privato, ci saranno bune probabilità che da qualche parte, vicino ad A ci sia una stazione con lo stesso indirizzo privato B, che riceverà la risposta da A.
Le stazioni di una rete locale hanno indirizzi privati, ma ci interessa che esse possano, ogni tanto, poter comunicare con l'esterno. La soluzione a questo è di seguito riportata.

Comunicare su Internet con indirizzi privati
→ Possiamo usare temporaneamente un host pubblico, possibile con dei meccanismi attuati all'interno della rete
→ Possiamo "cambiare" temporaneamente indirizzo IP (con uno pubblico); non cambiando la configurazione, ma usando un meccanismo interno della rete
→ La modalità di permettere ad un utente privato, con una stazione ad indirizzo privato, di comunicare, ad esempio di poter scaricare pagine web da un server pubblico, è un meccanismo che ha permesso di prolungare di 20 anni la vita di IPv4

Uno dei modi per permettere ad un utente con indirizzo privato di comunicare su Internet è quello di usare proxy server.
Nella intranet privata ci sono stazioni con indirizzo IP privato e una di questa vuole scaricare una pagina web: invece di contattare direttamente il server web www.netscire.it contatta un server proxy, per il protocollo particolare per il server web,

quindi un server proxy dell'HTTP, detto anche web proxy. E' dunque un proxy per una specifica applicazione.

Il proxy ha un indirizzo pubblico, manderà una richiesta al server che gli risponderà fornendo la pagina richiesta e il proxy http la fornirà al client, che quindi ha comunicato in Internet.

Poichè i proxy sono di tipo applicativo, nel caso un cliente richiedesse di scaricare un file con ftp occorrerebbe un server proxy ftp.

Per ogni applicazioni occorre un server proxy, ma poichè un server è un software in esecuzione su un host, più server possono coesistere sulla stessa macchina, sullo stesso host.

In questo tipo di soluzione, la stazione privata deve essere configurata esplicitamente ad usare il proxy. La soluzione non è trasparente, a differenza dell'altra soluzione, cioè quella di cambiare temporaneamente indirizzo alla stazione, che è fatta con una funzionalità detta Network Address Translation (NAT).

- **Network Address Translation (NAT)**

Una stazione con indirizzo privato può comunicare su Internet tramite NAT, Network Address Translation [traduzione di indirizzi], che una funzionalità che sta in un router di accesso tra la intranet pubblica e la Internet. Pur non essendo obbligatoria una tale posizione è comunque importante che stia sul percorso che i pacchetti faranno per andare verso il server pubblico che deve essere raggiunto.

Questa funzionalità modificherà gli indirizzi che si trovano nei pacchetti

Funzionamento

Dato un host A con indirizzo privato, esso vuole comunicare con un server B di indirizzo pubblico, quindi l'host A genera un pacchetto IP che vuole andare da A a B. Il pacchetto è inviato sulla intranet privata e attraversa la intranet privata e la intranet pubblica verso l'host con indirizzo B, con i vari router che indirizzano in modo opportuno il pacchetto. Sulla strada il pacchetto incontra la funzionalità di NAT che modifica gli indirizzi presenti nel pacchetto. In figura l'indirizzo mittente è modificato in X; l'indirizzo X, che può essere un pool di indirizzi pubblici, è un indirizzo associato al dispositivo che ha la funzionalità di NAT. Quando il pacchetto arriva a destinazione e B

risponde, B risponderà mandando la risposta all'indirizzo X, al dispositivo con la funzionalità NAT che, ricordandosi dell'operazione di aver modificato l'indirizzo A in X, farà l'inverso, modificando il pacchetto che arriva da B con destinazione X mettendo destinazione A, inoltrando il pacchetto sulla intranet privata per cui il pacchetto arriverà ad A, completando la richiesta.
Il risultato è che A, con indirizzo privato, ha comunicato con B ad indirizzo pubblico.

Proxy e NAT
Essi funzionano in modo diverso e non hanno nulla in comune se non per il fatto che i pacchetti che arrivano a B con un indirizzo X del mittente che è un indirizzo diverso dal vero mittente A.
Questo sia nell'uso del NAT, sia nell'uso del proxy.
Con il NAT la stazione manda richieste in modo trasparente, con il proxy la stazione mittente lo deve conoscere.
I due metodi sono diversi e permettono scenari diverse. Il proxy può ad esempio memorizzare le risposte, facendo una operazione di caching.

Altre applicazioni del NAT
→ Si usa NAT quando ci sono spazi di indirizzamento privati sovrapposti, stazioni di reti private vicine con stessi indirizzi, ad esempio private da fusioni e acquisizioni aziendali
→ Fusioni e acquisizioni aziendali
→ Si usa nelle extranet
→ Federazioni di intranet

L'Address Expansion è una delle situazioni più comuni.

Address Expansion
→ In essa più indirizzi locali (privati) sostituiti con un solo indirizzo globale (pubblico). Si noti l'uso dei termini locale e globale al posto di privato e pubblico. Lo stesso indirizzo globale è usato per più di un indirizzo locale in modo contemporaneo, questo ha permesso di non esaurire gli indirizzi pubblici. Quando tornano le risposte NAT

riuscirà a distinguerle dalle porte, sia dalle porte mittenti che dalle porte dei server, dagli indirizzi dei server
→ Distinguere in base alle porte

Un esempio

Il NAT si basa sulla presenza di una Mapping Table [tabella di corrispondenza], che contiene gli indirizzi locali (Address) e i numeri di porta (Port) che vengono usati dalle stazioni interne (Inside) e dalle stazioni esterne (Outside).

Nello scenario di esempio la funzionalità di NAT ha due indirizzi IP pubblici , 3.1.1.5 e 3.1.1.6.

Nella NAT Mapping table, prima riga, è supposto che la prima stazione, di indirizzo privato 10.1.1.5 e porta 2345 (Inside, Local) mandi un pacchetto al server 2.1.1.1 sulla porta 80 (Outside, Local). Nella funzionalità di NAT l'indirizzo privato 10.1.1.5 viene sostituito con un indirizzo pubblico, 3.1.1.5 (Inside, Global). La funzionalità di NAT deve costruirsi la riga per ricordarsi che ha sostituito l'indirizzo 10.1.1.5 con 3.1.1.5. per pacchetti che vanno verso l'indirizzo 2.1.1.1, cioè l'indirizzo del server, che è già quello giusto e non deve essere toccato dalla funzionalità NAT. Anche le porte rimangono inalterate. Quando il pacchetto torna indietro, viene riconosciuto dall'indirizzo del server e NAT rimodifica l'indirizzo, in questo caso destinazione, affinchè arrivi a 10.1.1.5.
Quando la seconda stazione, di indirizzo privato 10.1.1.7, manda un pacchetto al server 4.3.2.1, porta 21, l'indirizzo locale (privato) viene sostituito con l'altro indirizzo globale (pubblico) e viene creata una entry (seconda riga).
Quando tornano i pacchetti, essi tornano per l'indirizzo 3.1.1.6, che va cambiato con 10.1.1.7.
Se la stazione, a questo punto, fa una nuova richiesta, allo stesso server, ma su una porta del server diversa da quella di prima, allora si noti come l'indirizzo globale assegnato dal NAT possa essere uno dei due che il NAT "possiede".
Il NAT distingue, in fase di ritorno dei pacchetti, in base all'indirizzo del server e da

quello Inside, globale.
Situazione critica, in cui due stazioni voglio mandare ognuna un pacchetto allo stesso server, sulla stessa porta e che il NAT abbia un solo indirizzo globale; tutte le informazioni che servono a distinguere i pacchetti di ritorno dal server sono uguali, quindi i valori che inserirebbe in Inside, Global, della seconda riga sarebbero le stesse della prima riga, per cui ci sarebbe una situazione non risolvibile.
La soluzione a questo tipo di problema è quella di cambiare la porta anche al mittente, usando il cosiddetto PAT, detto anche NAT overload.

PAT: Port Address Translation
→ Anche chiamato NAT overload
→ La porta (locale interna) è sostituita con un numero casuale
→ E' problematico se serve una porta specifica, come nel caso dell'uso di IPSec o di DNS

Esiste anche il concetto di NAT statico.
→ Il NAT dinamico funziona per comunicazioni iniziate dal lato interno, e va bene per client privati
→ E per server interni, ad esempio "Pubblici" con indirizzo privato? La soluzione è: righe inserite manualmente nella tabella

Lezione n. 22

Configurazione delle stazioni

- **Configurazione manuale di stazioni IP**

Configurazione delle stazioni IP

→ Informazioni indispensabili

→ Indirizzo IP

→ Netmask, per capire la lunghezza del prefisso e sapere se le destinazioni sono nella setssa rete fisica o no

→ Necessari in pratica, quando si deve comunicare con stazioni al di fuori della rete fisica, in pratica nella quasi totalità dei casi

→ Default gateway (1 o più), è un router da usare, più di uno da usare in caso di guasti; la stazione non sa se il pacchetto va a destinazione o meno, però quando cerca di risolvere, tramite il protocollo ARP, l'indirizzo IP del default gateway e non riceve una risposta, se ha un secondo default gateway, proverà ad usare quest'ultimo

→ DNS server (1 o più)

→ Informazioni opzionali

→ Nome

→ Dominio di default

→ Server WINS

Inizialmente la configurazione delle stazioni avveniva in modo manuale, ad uso di tecnici.

E' il sistema operativo a fornire le funzionalità.

Problemi

→ Per utenti non tecnici

→ Terminali mobili, da un edificio ad un altro, oppure una stazione wireless

Nel sistema operativo è possibile specificare che si vuole utilizzare una configurazione dinamica.

- **Configurazione dinamica degli indirizzi (DHCP)**

Dynamic Host Configuration Protocol

Ci sono anche altri metodi per la configurazione dinamica, di cui uno, in disuso, utilizzato per il boot dalla rete di stazioni senza disco fisso, è il metodo con richieste multiple, ovvero richieste con protocolli diversi; in questo non c'è la netmask

→ Richiesta RARP per ottenere un indirizzo IP

→ Messaggio del protocollo ICMP, detto Address Mask Request, a mezzo di un router sulla rete

→ Messaggio ICMP, Gateway Discovery

→ Eventualmente più risposte

DHCP: Dynamic Host Configuration Protocol

Un server DCHP sulla rete fisica su cui si trova la stazione che ha bisogno della configurazione. La stazione effettua una richiesta di configurazione a cui il server risponde. Il server ha un database di indirizzi IP, precisamente configurazioni complete da assegnare alle stazioni. Il server sceglie una stazione e la offre alla stazione, quindi passa alla stazione indirizzo, netmask, indirizzo del default gateway, indirizzo dei server DNS. Questo in un unico protocollo, con i suoi messaggi, progettati apposta per questo scopo, con la presenza di un server, sul quale si basa DHCP.

Le caratteristiche di DHCP

→ Imbustato in UDP

→ Porta 67, alla quale vengono mandati i messaggi UDP, quindi il server DHCP è in attesa sulla porta 67 usando l'UDP e questo vuol dire che i messaggi DHCP vengono imbustati dentro pacchetti IP, per cui il client deve specificare il proprio indirizzo per metterlo nel campo indirizzo mittente e l'indirizzo del server. Il client non conosce né il proprio indirizzo, né quella del server per cui. La soluzione è quella di mandare i messaggi in broadcast, sia a livello IP che a livello MAC; l'indirizzo IP di destinazione è un indirizzo IP broadcast, il pacchetto IP è imbustato in una trama MAC mandata in broadcast, il client usa come indirizzo sorgente 0.0.0.0 e per identificare il server, il client usa l'indirizzo broadcast locale 255.255.255.255

→ Messaggi in broadcast

→ A livello MAC e IP

→ Client usa 0.0.0.0

→ Il server usa 255.255.255.255

Questo pacchetto IP, che contiene un messaggio UDP, che contiene la richiesta DHCP viene mandato a questo indirizzo IP, messo in una trama MAC e mandato all'indirizzo MAC broadcast. Questo si propaga su tutta la rete locale e quindi arriva a tutte le stazioni della rete locale. Le varie stazioni scarteranno il pacchetto perché le stazioni locali vedono un pacchetto mandato all'indirizzo broadcast locale e lo ignorano oppure ci guardano dentro e vedono che è un pacchetto UDP alla porta 67; non avendo, la stazione locale un processo in attesa di pacchetti sulla porta 67, il pacchetto sarà scartato.

Il server DHCP può invece rispondere proponendo una configurazione, così come altri server. Il client ne sceglie una e la richiede.

Negoziazione

→ Il server propone una configurazione IP

→ Ci potrebbero essere più server

→ Più proposte

→ Il client ne sceglie una e la richiede

Allocazione degli indirizzi che il server offre al client, ci sono tre tipi di allocazione

→ Allocazione dinamica, che va bene per la situazione descritta avanti

→ Lo stesso indirizzo IP è assegnato a stazioni diverse in diversi momenti, ad esempio negli hot spot di aeroporti, ecc., con stazioni che usano un indirizzo per un tempo relativamente breve e, quando la stazione non ne ha più bisogno, tale indirizzo è riassegnabile dal server ad un'altra stazione che farà richiesta

→ Una stazione può ricevere indirizzi diversi nel tempo

→ Allocazione automatica, valida in un ambito più controllato

→ Una stazione riceve sempre lo stesso indirizzo IP dal server DHCP, grazie all'indirizzo MAC, detto anche indirizzo hardware, della stazione

→ Non è noto/deciso in precedenza, il primo è scelto a caso dal server DHCP

→ Allocazione manuale, situazione diversa dalla configurazione manuale degli

indirizzi; l'amministratore di rete deve indicare per ogni indirizzo MAC quale indirizzo IP usare; la configurazione avviene sul server DHCP ed è più semplice della configurazione manuale sulla stazione

→ Una stazione riceve sempre lo stesso indirizzo

→ Indirizzo assegnato manualmente dall'amministratore di rete

Principali campi del messaggio DHCP

→ op: op code/tipo di mess., tipo di operazione

→ 1 = BOOTREQUEST

→ 2 = BOOTREPLY

→ htype: HW type, tipo di indirizzo di livello 2, che sarà scritto nel messaggio

→ hlen: HW address len, lunghezza di indirizzo di livello 2

→ chaddr: client HW address

→ xid: Transaction ID, identificatore di transazione

→ yiaddr: indirizzo IP assegnato dal server al client

Opzioni, è il campo che contiene le varie informazioni che serviranno per la configurazione. La codifica è di tipo Code - Length - Value, con un primo campo di un byte che identifica il codice del campo opzionale, un campo lunghezza (che indica il numero di byte) ed un campo valore. Nel caso di un codice non conosciuto, l'opzione, che ha un valore che non può essere compreso, viene scartata sapendo che tale campo ha una lunghezza specificata. Questo rende il protocollo facilemente estensibile. Alcuni dei codici più comuni sono

→ Tipo di messaggio (codice 53)

→ Subnet mask (Codice 1)

→ Router (3)

→ Nome di dominio (15)

→ Server DNS (5)

In particolare, per il tipo di messaggio, con codice 53

Value	Message Type
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNAK
7	DHCPRELEASE
8	DHCPINFORM

Scambio di messaggi, avviene quando il client vuole una configurazione. Il cliente richiede una configurazione inviando un messaggio “DHCPdiscover”, quindi il campo 53 di valore 1. Questo messaggio serve per scoprire i server DHCP. Sulla rete locale ci sarà un server DHCP, con un suo indirizzo IP ed un suo indirizzo MAC. Il serve, alla ricezione del messaggio “DHCPdiscover” risponderà con un messaggio “DHCPoffer“, in cui è inclusa una configurazione che vuole offrire al client; questo messaggio del server andrà all’indirizzo 0.0.0.0 e sarà mandato in broadcast. Il client riceve l’offerta dal server, anche da altri server, fa una scelta e deve comunicare quale scelta ha fatto affinché quelli non scelti usino la configurazione offerta, ma rifiutata, ad altri client.
Il client a questo punto manda un messaggio “DHCPrequest” sempre dall’indirizzo 0.0.0.0 all’indirizzo broadcast 255.255.255.255 e questo perché il client non può ancora usare la configurazione in quanto il server ha per ora solo fatto l’offerta e deve essere il server a confermare tale configurazione. Inoltre, mandando questo messaggio in broadcast, anche gli altri server sapranno quale scelta ha fatto il client. A questo punto il server manda un messaggio acknowledgment, il messaggio “DHCPack”, per cui il client mette sulla sua interfaccia la configurazione.
La configurazione fornita ha una durata limitata che si chiama lease (affitto).

Lease [affitto]
→ L’allocazione di indirizzo IP ha durata limitata

→ Il client può chiederne il rinnovo prima che scada, quindi manda di nuovo un messaggio DHCP Request

→ DHCP Request–DHCP Ack

→ Può esserne offerta una nuova, con un messaggio DHCP Offer dal server

→ DHCP Request–DHCP Offer

Rinnovo della lease

→ Se il rinnovo fallisce, si deve rifare l'allocazione

→ Per esempio, il server non risponde a DHCP Request

→ Da DHCP Discover in poi

Quando una stazione fa il reboot

→ Rinnovo lease (DHCP Request)

→ Nuova configurazione (DHCP Discover)

Limitazioni del DHCP

→ Client e server devono essere nella stessa rete fisica, perché il client manda la richiesta in broadcast ed il server la può ricevere solo se è nella stessa rete fisica

→ Non praticabile in reti con tante sottoreti, cioè in reti molto grandi, con tante piccole sottoreti in quanto, reti con molti host hanno prestazioni che decadono

DHCP Relay, è un meccanismo, implementato nei router, che serve ad implementare la funzionalità DHCP.

→ Normalmente realizzato nei router

→ Il DHCP Relay inoltra messaggi DHCP Request, che sono inviati sulla rete locale, a un server DHCP remoto

→ L'indirizzo del server remoto è fornito manualmente, da opportuna configurazione

→ In questa richiesta il DHCP Relay include il proprio indirizzo IP. Specificatamente l'indirizzo IP che il router ha sull'interfaccia da cui ha ricevuto la richiesta DHCP. L'indirizzo del DHCP Relay sulla rete del client è incluso nel campo giaddr, gateway IP address, del messaggio DHCP

→ Indirizzo assegnato in base alla LIS del client, deducibile dal campo giaddr, che contiene il prefisso dell'indirizzo

→ Nel campo giaddr

→ A questo punto il server manda il messaggio DHCP Reply al DHCP Relay

→ Il DHCP Relay lo inoltra nella rete del client

DHCP e DDNS

Quando si usa DHCP risulta conveniente usare il DDNS, il DNS dinamico. Questo perché l'host riceve un indirizzo dinamico per cui occorre che il suo nome venga associato all'indirizzo giusto che riceve ogni volta.

Nei vari sistemi operativi questo funziona in modo diverso, a lato un client DHCP in Windows 2000 e altri clients DHCP (Win9x e NT, in cui tutto è fatto dal server).

Si noti come siano aggiornati i record DNS di tipo A e di tipo PTR.

- **Configurazione automatica**

Risolve alcuni problemi della configurazione dinamica. E' basata su indirizzi riservati, detti indirizzi automatici.

Motivazione

→ Potrebbe non esserci un server

→ Dentist office [ufficio del dentista], per identificare un luogo privo di competenze tecniche

→ Indirizzi riservati: 169.254.0.0/16

→ Automatic Addresses

→ Indirizzi link-local, in IPv6, che si usano solo sulla rete locale

→ Per comunicazione sulla sola rete fisica

→ Niente default gateway, cioè non si ha un default gateway

Principi di funzionamento

Quando una stazione diventa operativa, genera la parte di host dell'indirizzo, in modo casuale. Per assicurarsi che stazioni diverse generino la stessa parte di host, si usa qualcosa di univoco, come l'indirizzo MAC o il tempo attuale, come seme di generazione. La stazione, per verificare che non ce ne sia un'altra con lo stesso

indirizzo, usa una richiesta ARP e guarda se qualcuno risponde. In questo modo la stazione ha un indirizzo che può usare solo nella comunicazione locale sulla rete fisica.

→ La stazione genera la parte di host

- → Generazione casuale
- → Indirizzo MAC o tempo reale usato come seme

→ ARP per verificare l'unicità

Lezione n. 23

Algoritmi di routing

- **Routing e forwarding**

Routing

Determinare un percorso attraverso la rete per i pacchetti.

Forwarding [inoltro]

→ Far avanzare i pacchetti attraverso la rete verso la destinazione

→ Include una decisione di routing, cioè sul percorso da fare

Routing proattivo, cioè capire, prima ancora che arrivino i pacchetti, quale è la strada migliore per raggiungere le destinazioni; i nodi effettueranno il forwarding, l'inoltro del pacchetto alla destinazione, a fronte delle informazioni acquisite dal routing proattivo. Per quanto riguarda il routing proattivo

→ Esso è indipendente dall'effettivo traffico

→ Determina quali stazioni siano raggiungibili, e non solo da che percorso

→ Obiettivo è calcolare la strada migliore, dipendente da criteri specifici

→ Normalmente lo si chiama semplicemente "routing", aggiungendo "proattivo" lo si distingue dall'azione di scelta del percorso che viene fatta all'atto dell'inoltro e detta routin al volo

Routing "al volo", che serve quando si fa forwarding

→ Viene operato quando si trattano i singoli pacchetti

→ Si fa usando informazioni locali preconfezionate, che si trovano nella cosiddetta routing table o forwarding table

→ Routing/forwarding table

→ Prodotta da routing proattivo o, in alcuni tipi di architettura di rete, da una operazione di segnalazione fatta prima di mandare i pacchetti

Algoritmi per il routing “al volo”

→ Routing by Network Address, usato nelle reti IP, dove il pacchetto contiene l’indirizzo IP di destinazione e la scelta di routing è fatta in base all’indirizzo nel pacchetto, che normalmente è un indirizzo di rete

→ Label Swapping, ogni pacchetto contiene una etichetta, cambiata ad ogni nodo

→ Source Routing, il mittente decide la strada dei pacchetti

Ogni architettura di protocolli utilizza uno o più algoritmi.

Fasi del forwarding

→ Routing (“al volo”) dei pacchetti

→ Occorre selezionare la porta di uscita

→ Eventualmente anche il next-hop, il prossimo router cui inviare il pacchetto

→ Switching [commutazione], operazione da fare una volta capita la porta dalla quale il pacchetto deve uscire: il switching è il trasferimento alla porta di uscita

→ Trasmissione del pacchetto

Una classificazione degli algoritmi di routing proattivi

→ Algoritmi non adattativi (statici), non si adattano ai cambiamenti della rete

→ Algoritmi adattativi (dinamici), adattano le scelte di routing anche quando cambia qualcosa nella rete

- **Routing non adattivo**

Algoritmi non adattativi

→ Fixed directory routing, il più comune

→ Noto come routing statico, basato su tabelle fisse

→ Configurazione manuale delle tabelle ad opera dell’amministratore di rete, scritte nel router; in pratica il routing lo fa il gestore della rete

→ Flooding [inondazione] selettivo e derivati, questo è un altro algoritmo non adattativo; si manda il pacchetto dappertutto o quasi, quindi si parla di flooding selettivo; ci sono anche dei derivati

Pro e contro del fixed directory routing

→ L'amministratore di rete ha pieno controllo

→ Si presta ad errori

→ Non si adatta a cambiamenti topologici

A lato (vd. appunti) è evidenziato come sia possibile specificare anche delle route di backup (se si rompe un link), o delle route da usare come load balancing, per distribuire il carico su percorsi diversi. In partica l'amministratore di rete dice che se c'è da mandare un pacchetto ad A o a B, mandalo a sinistra; per mandarlo a D mandalo a destra, se devi mandarlo a E, mandalo in basso. Ma, se vuoi, i pacchetti per A e per B puoi mandarli in basso. Si tratta di una piccola reazione ad un malfunzionamento.
Ci può essere il rischio che un pacchetto venga rimbalzato tra due router (B e C nell'esempio), creando un loop di routing e la rete non funziona (A non è più raggiungibile).

Statico o dinamico?
Quello che si fa è che si usa il routing statico nelle zone periferiche della rete, dove da un lato c'è una unica strada.
Nel core della rete sarà fatto routing dinamico.

- **Routing dinamico**

Basato su algoritmi adattativi, di tre categorie

→ Routing centralizzato

→ Routing isolato

→ Routing distribuito

  → Distance Vector

  → Link State

Routing centralizzato, vuol dire che da qualche parte della rete c'è un Routing Control Center che è un nodo di rete che calcola e distribuisce le tabelle di routing per tutti gli altri nodi della rete (si tratta di routing proattivo per preparare le tabelle che saranno usate nel forwarding dei pacchetti per fare il routing al volo). Tutte le tabelle vengono calcolate da un unico centro di calcolo

→ Routing Control Center (RCC)
→ Calcola e distribuisce le tabelle di routing
→ Ha bisogno di informazioni da tutti i nodi
→ Ottimizza le prestazioni
→ Semplifica troubleshooting, la ricerca guasti
→ Traffico di rete sostenuto nelle vicinanze dell'RCC
→ L'RCC è un singolo punto di fallimento del sistema
→ L'RCC è un collo di bottiglia
→ Non adatto per reti altamente dinamiche

Routing isolato, alternativa al routing centralizzato
→ Ogni nodo decide i percorsi indipendentemente
→ Non c'è scambio di informazioni con gli altri nodi; questa soluzione non genera traffico, non c'è point of failure, non c'è un collo di bottiglia nella rete
→ Per esempio, algoritmo Backward Learning
→ Usato dai bridge del protocollo IEEE 802.1D

Routing distribuito
Unisce i vantaggi di routing isolato e centralizzato
→ Router collaborano nello scambiarsi informazioni sulla connettività nella rete, sui collegamenti, sul funzionamento dei router
→ Ogni router decide in modo indipendente, ma in modo coerente, il che non è banale perché la decisione viene fatta in modo distribuito; a tale scopo sono stati progettati opportuni algoritmi per scambiare informazioni e permettere ai router di decidere in modo distribuito ma coerente

- **Algoritmo di routing Distance Vector**

I nodi imparano per mezzo dei distance vector, che sono elenchi di distanze, quali sono i nodi raggiungibili attraverso le loro interfacce e a che distanza.

Principi di base, come a lato riportato

Distance Vector

→ Lista di destinazioni raggiungibili (tutte!), generata da ogni nodo; in una rete grande questa lista sarà enorme
→ La lista contiene la distanza dal router che manda l'annuncio, secondo una misura detta metrica
→ Ogni router genera il suo distance vector
→ Ogni router manda il suo distance vector a tutti i vicini

Scenario esemplificativo, in cui A riceve i distance vector dai nodi B e D; con queste informazioni, A fa una operazione di fusione e genera il proprio distance vector.
La fusione permette ad A di generare la routing table.

Fusione e generazione dei distance vector
La prima riga si legge come A avente l'informazione che può raggiungere A a distanza 0, oppure a distanza 1, ricevuta da B e D; quindi nella routing table ci viene scritta l'informazione migliore e viene generata l'informazione di distance vector per questa informazione migliore.
In seconda riga si hanno due informazioni di quale sia la distanza per raggiungere B e viene presa l'informazione migliore, notare come sia modificata la distanza in quanto A deve mandare i suoi pacchetti a B. Nel caso di E, A sceglie una informazione a caso, essendo entrambe valide.
Analogamente per le altre righe, per le quali A guarda tutte le destinazioni annunciate nei distance vector e fa l'operazione di scegliere il distance vector del vicino con un costo minore. Questa è l'operazione di fusione dei distance vector. Da questa operazione nasce la tabella di routing, dalla quale nasce la distance vector di A, prendendo le destinazioni e le destinazioni alle quali A le sa raggiungere. La distance vector viene mandata a tutti i router vicini, che a loro volta faranno la fusione. Quando, dalla operazione di fusione viene generata un tabella di routing uguale a quella esistente allora il router capisce che può smettere di inviare distance vector poichè è stata determinata la topologia della rete.
Il router, in sostanza, manda il suo distance vector, solo quando esso cambia.

Cambiamento topologico, ad esempio si rompe il link tra A e B.

Quindi tutto quello che A aveva appreso dal link B è inservibile, quindi A deve fare nuovamente l'operazione di fusione (merge), per ricostruire la tabella di routing andando a prendere l'informazione migliore che è soltanto quella di D. Poi A genera il distance vector che sarà ricevuto dai nodi che, a loro volta faranno le operazione viste. Questo mostra l'aspetto dinamico, adattativo, dell'algoritmo.

Il cambiamento topologico causa parecchi problemi

→ Black Hole [buco nero]
→ Count to infinity [conteggio a infinito]
→ Rimbalzi (loop)

Sono problemi di instabilità causati da cambiamenti topologici, per i quali i router cominciano a fare pasticci e non raggiungono una configurazione stabile delle loro tabelle di routing e a capire come inoltrare i pacchetti sulla rete.

Esistono soluzioni, che sono solo parziali
→ Split Horizon
→ Path Hold Down
→ Route Poisoning

In sostanza
Il problema di base è che i router non conoscono la topologia della rete. Ad esempio, in base ai distance vector che B riceve, B non può distinguere i casi, cioè B non distingue se la rete è fatta in un modo o nell'altro.

Vantaggi del distance vector
→ Facile da implementare
→ I protocolli sono facili da utilizzare
→ Richiedono minima configurazione

Limitazioni
→ Complessità e tempo di convergenza esponenziali (nel caso peggiore)
→ Da $O(n^2)$ a $O(n^3)$, con n il numero di nodi

→ I router e collegamenti più lenti determinano il tempo di convergenza di tutti i router in rete

→ Ottimizzazione complicata

→ Ricerca guasti complicata

→ Molto traffico di routing (e dati memorizzati)

Il distance vector non è adatto a reti grandi e complesse.

L'alternativa al distance vector sono gli algoritmi di routing link state.

- **Algoritmi di routing Link State**

Principi di base , come a lato riportato, ovvero il nodo E manda in giro una informazione del tipo "io, nodo E, sono connesso a A e F", cioè E dice quale è la topologia intorno a lui, ovvero lo stato dei suoi link (link state). Questa informazione viene mandata a tutti i router della rete, compreso C. I nodi che ricevono tale informazione "capiscono" la topologia della rete. Quindi i router si costruiscono una mappa della rete.

Link state

→ Vengono create informazioni sullo stato dei collegamenti (link), dette Link state

→ Le informazioni devono essere mandate da ogni nodo a tutti gli altri nodi, ovvero viene fatta una operazione di Selective flooding [inondo selettivo], che è un punto critico, di non facile realizzazione

→ Ogni nodo si crea una mappa della rete

→ La stessa mappa su ogni nodo (importante!!!)

→ Ogni nodo calcola le "route" sulla mappa, per mezzo di un algoritmo, detto di Dijkstra

→ Algoritmo di Dijkstra, un matematico, (shortest path first), che cerca il percorso più breve

perché tutto funzioni, i link state devono arrivare a tutti i router, affinchè la mappa che i router creano sia quella corretta. Se i link state arrivano a tutti i router, essi possono creare percorsi ottimali e coerenti.

Link state database, che ogni nodo crea.

Ad esempio quello di A dice che A è collegato a B e a D, con un link di collegamento,

quindi in questo caso la metrica è il numero di link, ma può essere la distanza o altro, la capacità dei link.
Il router genera la sua tabella di routing.

Convergenza rapida
→ L'algoritmo di Dijkstra ha una bassa complessità di calcolo
→ L • log (N)
    → L: numero di link
    → N: numero di nodi
→ I link state si propagano velocemente
→ I link state non richiedono nessuna elaborazione prima dell'inoltro
L'algoritmo lavora anche su reti grandi.

Traffico di routing e uso di memoria limitati
→ I link state sono piccoli
→ Neighbor greeting veloce ed efficiente
    → Protocollo per scoprire i vicini

Altri vantaggi
→ Raramente genera loop
    → Percorsi di inoltro circolari
→ Semplice da capire e per fare ricerca guasti
    → Tutti i nodi hanno basi dati identiche

Limitazioni
→ Alta complessità di implementazione
    → Selective flooding
    → Prima implementazione ha richiesto diversi anni, 5
→ Protocolli con complessa configurazione

Lezione n. 24

Architettura e protocolli di routing in internet

- **Protocollo e dominio di routing**

Alcune definizioni di base

Nell'architettura di protocolli, posti tra il livello 3 ed il livello 4 in quanto i protocolli di routing, che servono per far capire ai nodi su quale percorso inoltrare i pacchetti (definito in precedenza come routing proattivo, per capire quale è la strada migliore per raggiungere le destinazioni sulla rete indipendentemente che ci sia o meno traffico), servono a livello 3 perché possa funzionare bene, ma alcuni di questi protocolli di routing generano messaggi che sono imbustati in pacchetti di livello 3 mentre altri generano messaggi che sono imbustati in pacchetti di livello 4. In realtà i protocolli di rounting non dovrebbero nemmeno stare nella pila protocollare perché non servono per trasportare dati dell'utente, ma servono per trasportare dati dei router, dati di servizio. L'architettura protocollare spiega come sono gestiti i dati dell'utente.

Protocollo di routing

Protocollo usato dai router per scambiare informazioni sulla rete e determinare il percorso migliore per ogni destinazione.

→ E' basato su un algoritmo di routing, per definire quali sono le informazioni che devono essere scambiate e come devono essere utilizzate

→ Definisce le metriche (numero di link attraversati, la banda dei link, il tasso di perdita dei link)

→ Il modo in cui codificarle nei pacchetti

→ Tempistiche specifiche

→ Definiscono parametri configurabili

Dominio di routing

→ Un insieme di router che usano lo stesso protocollo di routing

→ Sono una parte connessa di rete, una parte di rete in cui è possibile far trasferire delle informazioni da un qualsiasi nodo del dominio di routing ad un qualsiasi altro

nodo senza uscire dal dominio di routing. In un dominio di routing essi si devono scambiare informazioni usando un certo particolare protocollo

Ridistribuzione
→ Un router può appartenere a più domini di routing
→ Quindi può usare più protocolli di routing
→ E quindi può ridistribuire informazioni apprese con un protocollo tramite un altro.
In altre parole, se il router usa due protocolli di routing vuol dire che fa parte di due domini ed allora può ridistribuire le informazioni apprese in un dominio, nell'altro. Cioè ridistribuire col protocollo B le informazioni che ha appreso con il protocollo A. Questo è detto ridistribuzione, normalmente regolata da politiche di ridistribuzione.

Politiche di ridistribuzione, configurabili
→ Definite dall'amministratore del router
→ Filtri sugli annunci (cioè sulle informazioni di routing che vengono distribuite). Si usa il termine annunci perché le informazioni di routing annunciano l'esistenza, raggiungibilità di una destinazione. Le implementazioni di router mettono a disposizione dei meccanismi all'amministratore di rete per specificare quali degli annunci ricevuti come un protocollo devono essere distribuiti con un altro protocollo. Quindi l'amministratore può stabilire dei filtri e può anche stabilire come convertire le metriche da un protocollo all'altro. Ad esempio può convertire il numero di link nella banda minima per passare da un router all'altro
→ Conversione di metriche
→ Definire una priorità tra le sorgenti di informazione, con metriche di protocolli diversi

- **Autonomous system (sistemi autonomi)**

Di cosa si tratta
→ Un insieme di sottoreti IP (vicine) raggruppate per
→ Topologia e per criteri organizzativi; amministrate da una stessa amministrazione
→ Per esempio le sottoreti di un grosso internet service provider
Quindi si dice che un insieme di sottoreti del service provider X sono un autonomous system

Perché si fa questo?

→ Gestione indirizzi e routing strettamente coordinati

→ Eventualmente ci saranno più domini di routing

→ Interfacce tra autonomous system sono tenute sotto controllo

→ Dati, non tutti i dati non sono scambiati

→ Informazioni di routing, non sono tutte scambiate

Cioè un router di un autonomous system non dice necessariamente tutto ad un router di un altro autonomous system, questo perché possono benissimo appartenere ad organizzazioni diverse. Dunque il flusso di dati ed il flusso di informazioni di routing sarà controllato.

Gli AS si creano quindi per ragione di amministrazione, in modo che all'interno dell'autonomous system le scelte di routing possano essere autonome (il service provider deciderà di usare i protocolli che vuole nel modo in cui vuole e con la configurazione che vuole). Le scelte del routing esterno, cioè tra un autonomous system ed un altro dovranno essere negoziate.

Gli autonomous system si creano per ragioni di scalability, affinchè la rete possa crescere e possa funzionare come rete di grande dimensioni, questo perché le informazioni di routing non sono propagate ovunque, cioè la frontiera tra un autonomous system ed un altro è il posto giusto dove ridurre la quantità di informazione di routing che viene propagata. Si riduce filtrando le informazioni oppure aggregando delle informazioni, prendendo un certo numero di annunci diversi ed aggregandoli in un annuncio unico come se esso ne fosse il riassunto. La perdita di informazione bilancia la minor propagazione dell'informazione.

Da questo si può notare che pur essendo la rete Internet enorme, tra i vari autonomous system vengono propagate informazioni di routing con tutti i dettagli, ma tra autonomous system si eliminano dettagli per cui i router non vengono sopraffatti da informazioni.

→ Amministrazione

→ Scelte sul routing interno autonome

→ Scelte sul routing esterno negoziate

→Scalability

→ Informazioni non sono tutte propagate ovunque

Si deve in qualche modo riuscire a controllare lo scambio di informazioni tra autonomous system, rendendo ogni autonomous system autonomo in qualche modo.
I router interni agli autonomous system sono detti Interior gateway, questi sono router collegati solo ad altri router interni all'autonomous system e scambiano informazioni tra di loro con dei protocolli detti IGP, Interior Gateway Protocol.
I router collegati a router di un autonomous system diverso (gestiti quindi in modo diverso dal punto di vista amministrativo) sono detti Exterior gateway, Border gateway o Boundry router. Questi scambiano informazioni di routing con protocolli che chiamiamo EGP, Exterior Gateway Protocol. La famiglia di protocolli EGP avrà caratteristiche diverse dalle caratteristiche della famiglia di protocolli IGP.
Gli autonomous system sono identificati da un numero, come si nota in figura.

L'identificazione di un autonomous system

→ Tramite un numero di 2 o 4 byte

→ Assegnato da IANA (Internet Assigned Numbers Authority)

→ Numeri privati di autonomous system

→ 64512-65534 (originali, su due byte)

→ Per scambi controllati di informazioni di routing

I numeri privati degli autonomous system sono analoghi agli indirizzi privati. Chunque può usare tali numeri di autonomous system senza dover chiedere una autorizzazione allo IANA, ma non ci sarà garanzia che siano univoci. Si usano in contesti di reti private, per creare zone in cui lo scambio di informazioni di routing sia controllato.

Aspetti amministrativi

Gli autonomous system sono importanti anche per aspetti di tipo amministrativo tipo

quello di decidere che tipo di percorso faranno i pacchetti nella rete tra autonomous system.
In figura a lato è supposto che nell'AS85 ci sia una destinazione D. In una rete i router troveranno la strada migliore per inoltrare le informazioni nella stessa rete; in uno scenario di più autonomous system è importante il modo in cui i dati fluiscono, dipendente dagli annunci.
Gli annunci che vengono generati in un certo modo hanno un impatto sui percorsi, quindi annunci fatti in un certo modo permetteranno il percorso migliore.

Exterior Routing, routing tra autonomous system
→ Routing non necessariamente sui percorsi più brevi
→ Le scelte sono basate su politiche, politiche configurabili che riflettono gli accordi tra i gestori degli autonomous system
Il router cerca sempre il percorso più breve, il percorso migliore secondo la metrica del particolare protocollo. Nel caso di protocolli da usare per l'exterior routing, routing tra autonomous system, invece la scelta deve essere basata su delle politiche configurabili dall'amministratore della rete, quindi il protocollo di routing e la sua implementazione devono prevedere questo.

Gli autonomous system si fanno anche per ragioni di scalability, per poter operare su grosse reti.
Per ottenere scalability con l'uso di autonomous system si cerca di aggregare le destinazioni

Scalability [capacità di operare su grosse reti]
→ Le destinazioni possono essere aggregate
→ 195.1.2.0/24 e 195.1.3.0/24 annunciate come 195.1.2.0/23 dal border gateway, cioè il router di bordo (router che sta al bordo). Le due destinazioni vengono annunciate in modo aggregato e quindi esso annuncia una sola destinazione. Nell'annuncio il /23 indica il prefisso di 23 bit che le annuncia tutte e due per cui viene usato il supernetting
I router esterni all'autonomous system vedranno un solo annuncio e quindi dovranno

elaborare, memorizzare, propagare un solo annuncio e faranno meno lavoro, perdendo un pò di informazione, come ad esempi quanto possono essere lontane le due destinazioni.

Questo riprende il concetto iniziale per cui vogliamo avere gli autonomous system per far operare i router in una rete gigantesca riducendo la quantità di informazioni e questo consiste nel creare i punti di aggregazione all'uscita dagli autonomous system ed il confine, il perimetro, dell'autonomous system ci fornisce il punto ideale dove fare questa operazione.

- **Architettura di routing di Internet**

Come viene organizzato il routing nella rete Internet, che è tutta organizzata in Internet Service Provider. Gli ISP non sono tutti uguali, alcuni hanno reti molto grandi che raggiungono diversi continenti e sono interconnessi tra di loro per mezzo di border gateway. Questi ISP (Tier 1 ISP in celeste, Internet Service Provider di Livello 1) scambiano informazioni di routing e scambiano traffico dati; all'interno avranno i propri router che useranno protocollo IGP per capire come raggiungere ogni destinazione. A questi ISP si collegheranno dei Service Provider più piccoli (Tier 2 ISP in giallo, Internet Service Provider di Livello 2), che sono collegati a quelli di livello 1 tramite router boorder gateway ed anche ogni ISP di livello 2 costituisce un autonomous system ed usa un protocollo IGP per scambiare informazioni di routing e di dati. Se un ISP di Tier 2 vuole mandare un pacchetto ad un altro ISP di Tier 2, il pacchetto passerà da un ISP di Tier 1, per cui ci sarà una operazione di transito attraverso un service provider di tier 1. I service provider di tier 2 possono essere collegati a più service provider di tier 1. Poi abbiamo anche Service Provider di Livello 3, cioè Tier 3 ISP (in verde), che sono collegati a quelli di livello 2. Gli ISP Tier 3 possono avere anche più collegamenti con gli ISP Tier 2, per ragioni di load balance e di affidabilità. Si noti come i collegamenti possano essere di tipo diverso: quelli di colore rosso sono i collegamenti detti di private peering, perché collegano Service Provider dello stesso livello e di norma questi Service Provider hanno un interesse mutuo ad essere collegati tra di loro in quanto vogliono che i propri clienti possano mandare pacchetti ai clienti dell'altro Service Provider. Entrambi hanno una rete molto estesa che dà raggiungibilità su grosse aree geografiche e vogliono essere collegati. Il collegamento di private peering è sicuramente fra i Service Provider di livelli 1, ma può anche essere fra quelli di livello

2 e anche fra quelli di livello 3. Questo può avvenire per non far fare ai pacchetti strade molto lunghe, in quanto la copertura di service provider, ad esempio quelli di livello 3, può essere molto vicina, nella stessa città. La ragione del peering è quello di aver interesse mutuo nel far passare traffico che, in quest'ultimo caso sarà più veloce e gli utenti vedranno prestazioni migliori.

I collegamenti di colore bianco sono collegamenti di tipo Client-provider, in cui un provider compra un servizio dall'altro. Ad esempio un Service provider di livello 2 può comprare un servizio (in pratica chiedere un collegamento ed il servizio comprato è quello di connettività, oppure detto di transito) a quello di livello 1 in quanto quest'ultimo ha pacchetti che possono andare negli Stati Uniti, od un altro per raggiungere la zona Asia-Pacifica.

I router prenderanno decisioni che riflettono gli accordi. I router di bordo, usando i protocolli di routing per i router esterni (External Gateway Protocol) devono essere in grado di implementare gli accordi commerciali tra i Service Provider.

Saranno quindi regolati i vari annunci che i router di bordo mandano verso l'altro router di bordo e, ad esempio, l'ISP Tier 1 manderà annunci solo per destinazioni americane all'ISP Tier 2 con cui ha preso accordi in tale senso. Queste sono quindi politiche di routing e riguardano quali annunci mandare e quali non mandare, ed in quale direzione. Il routing tra anonimous system è tutto basato su questi criteri. E tutto questo vale anche per collegamenti di tipo private peering, ad esempio fra service provider di livello 3, in alto a destra nella figura. Questo si traduce nel non utilizzare un collegamento per far passare traffico oltre quello stabilito dall'accordo. Quindi anche sui collegamenti di peering si devono implementare le politiche di routing che riflettono gli accordi commerciali.

I collegamentii sono detti Private peering perché i service provider per avere quel collegamento devono avere un collegamento tra loro due router, che non saranno necessariamente vicini, per cui sarà necessario acquistare, o affittare, l'uso di un mezzo trasmissivo (fibra ottica, ad esempio) , quindi questo sarà un collegamento privato, tra due service provider.

L'alternativa al dover creare dei collegamenti dedicati, cioè privati, tra due router è quello di creare i cosiddetti NAP o IXP, Neutral Access Point (NAP), Internet eXchange

Point (IXP).
Si tratta di un locale a cui i service provider possono collegarsi per mezzo, ad esempio, di una fibra che va dal loro router in un loro locale a questo locale che si può dire essere pubblico, gestito da un gestore di terza parte. I service provider metteranno nel locale del NAP/IXP un loro router. Avendo nel NAP/IPX un punto di accentramento, i service provider possono creare un loro collegamento di peering (tratto rosso tratteggiato), per cui possono scambiare dati ed informazioni di routing non attraverso un link dedicato a loro due, ma attraverso il punto pubblico, pubblico perché ci possono essere altri service provider che hanno un collegamento verso il NAP/IXP. Questo elimina la necessità di collegamenti privati dedicati, così come collegamenti client-provider e questo aumenta la connettività.

Neutral Access Point (NAP), Internet eXchange Point (IXP) è una LAN che collega router di vari AS (ISP), con coppie di router scambiano informazioni di routing, eventualmente usando BGP.

Lezione n. 25

Protocolli di routing e servizi di consegna “speciali”

- **Due famiglie di protocolli**

Ci sono due famiglie di protocolli nella rete Internet, due tipi di protocolli

→ Interior Gateway Protocol (IGP)

→ Usato per routing intra-dominio, cioè all’interno di un autonomous system, da non confondersi col fatto che un protocollo di routing funziona sempre nel dominio di routing viene utilizzato, per definizione

→ Exterior Gateway Protocol (EGP)

→ Inter-domain routing, cioè usato nel routing tra domini

Ci sono due famiglie di protocolli perché essi hanno obiettivi diversi.

Obiettivi diversi → Diversi criteri di progettazione

Caratteristiche degli IGP

→ Obiettivo è distribuire informazioni sulla topologia di rete

→ Scegliere route [percorsi per l’inoltro di pacchetti] in base a tali informazioni topologiche.

Nei protocolli IGP il router cerca di trovare la route “migliore”, che dipende dalla definizione che vogliamo dare. I vari protocolli definiscono delle metriche, ovvero dei modi per misurare i percorsi ed un criterio per dire quale percorso è migliore dell’altro. Il router userà il protocollo per raccogliere informazioni e poi per calcolare i percorsi migliori per tutte le destinazioni in base a queste informazioni. Una volta calcolati i percorsi il router può costruirsi la tabella di routing che dice quale è il next hop a cui mandare i pacchetti per una certa destinazione

Caratteristiche dei protocolli della famiglia EGP

→ Servono per distribuire informazioni su Autonomous System

→ Servono a distribuire costi amministrativi, cioè dei costi che rappresentano le scelte degli amministratori di rete su quali siano i percorsi migliori o preferibili per il traffico, tra gli anonimous system

→ Questo permette ai router di decidere in base a politiche configurate dagli

amministratori. L'obiettivo dei router è dunque trovare la route "preferita" (non la "migliore"), in base alle indicazioni di chi configura il router, in base agli accordi fatti con i gestori degli altri autonomous system

Vediamo dunque quali sono oggi i protocolli di routing di queste due famiglie di protocolli, iniziando dall'IGP, dai protocolli che usano l'algoritmo del distance vector e poi da quelli che utilizzano l'algoritmo link state e poi passando a quelli della famiglia EGP.

IGP – Distance Vector
→ RIP: Routing Information Protocol
→ IGRP: Interior Gateway Routing Protocol
→ E-IGRP: Enhanced IGRP, versione successiva

IGP – Link State
→ OSPF: Open Shortest Path First, più famoso
→ Integrated IS-IS, molto usato

EGP
→ BGP: Border Gateway Protocol
→ IDRP: Inter Domain Routing Protocol, utilizzato pochissimo
→ Il routing statico è anche una possibile opzione, nel routing inter-dominio, per cui è l'amministratore che definisce il percorso che i pacchetti faranno. Questo per la ragione di scelta del percorso "preferibile" e non "migliore". Per ha il problema che non è adattativo, cioè non si adatta a cambiamenti topologici

- **Interior Gateway Protocol**

RIP
→ Originariamente sviluppato per un'architettura diversa
→ RFC 1058 (1988), che ha adattato il protocollo all'uso di Internet e RFC 1388 (1993), che fornisce una prima nuova versione. E' basato sull'algoritmo distance vector
→ Implementato anche in stazioni Unix/Linux, per apprendere gateway; una stazione Unix è funzionante anche come gateway

Caratteristiche

→ Hop count [numero di hop]

→ Al più 15 hop, per non entrare in modalità count to infinity

→ Messaggi di aggiornamento sono periodici

→ Distance vector

→ Ogni 30 s, questo vuol dire che un router manda, ogni 30 secondi, l'elenco di tutte le destinazioni che sa raggiungere, che è molto traffico; questo serve a dimostrare ai vicini che è attivo; questa soluzione è inefficienza

→ Convergenza: in 3 min, nei casi peggiori; è il tempo che ci mette la rete a riconfigurarsi, per cui in tale tempo ci potrebbero essere stazioni non raggiungibili, il che è problematico, ma meglio di una rete rotta

IGRP, protocollo molto usato

→ Protocollo proprietario della Cisco Systems, sempre basato sul distance vector

→ Supera alcuni dei limiti del RIP

→ Per un po' era l'unica alternativa a RIP

Caratteristiche

→ Metriche articolate

→ Ritardo sui link

→ Banda dei link

→ Affidabilità dei link

→ Carico dei link

→ Massima lunghezza dei pacchetti, che i link lasciano trasmettere, detta MTU, Maximum Transmission Unit

→ Multipath routing, mentre di norma i router IP scelgono un percorso e mandano tutti i pacchetti su quel percorso. In questo caso, se ci sono due o più percorsi alternativi, il protocollo dice di usarli tutti, in modo inversamente proporzionale al loro costo, che è calcolato sul peso dei pacchetti. Il carico è dunque diviso sui percorsi

Il percorso migliore è scelto in base ad una combinazione pesata delle informazioni sopra. Il protocollo ha una configurazione di default, ma l'amministratore può

specificare quale metrica è più importante, ad esempio tra basso ritardo ed alta affidabilità

OSPF, basato sull'algoritmo link state, più complicato da implementare, in particolare la distribuzione dei link state. E' stato utilizzato dopo a tutti gli altri protocolli. E' un protocollo molto ben fatto e molto potente, più difficile da configurare

→ RFC 1247 (1991) e RFC 1583 (1994)

→ Routing gerarchico

→ Il dominio di routing è diviso in aree

→ Aggregazione di informazione tra aree

I router di un'area avranno tutte le informazioni, quelli esterni avranno informazioni meno dettagliate, in quanto sono state aggregate, ma in questo modo i router non si trovano ad avere a che fare con troppe informazioni.

L'area 0, o backbone area, è un'area particolare, che deve collegare tutte le altre come da figura a lato, architettura di rete con OSPF. In questo protocollo bisogna dire ai router quali sono le aree; i router possono avere funzionalità particolari: quelli interni alle aree si chiamano internal router, quelli interni alla backbone area si chiamano backbone router, che sono "normali", poi ci sono gli area border router che sono configurati ad essere parti di più aree che fanno una funzione particolare, quella di vedere tutti i dettagli delle aree per cui sono configurati (e collegati), poi ne fanno un riassunto e lo propagano nelle varie aree. Ci sono dunque dei link state particolari che servono per riassumere ciò che c'è in un'area e dirlo in un'altra area. Questo riassunto fa in modo che i router non debbano avere a che fare con una mappa dettagliata della rete: essi hanno una mappa dettagliata della loro area con informazioni sull'esterno. Questo rende il protocollo gerarchico e fa in modo che funzioni su reti molto grandi, cioè fornisce al protocollo una grossa scalability.

Integrated IS-IS, un altro protocollo che usa l'algoritmo link state

→ Estensione di un protocollo progettato per funzionare su reti OSI, IS-IS sta per Intermediate System to Intermediate System. Fatto per avere un protocollo oltre al RIP (con dei limiti) e al IGRP (proprietario)

→ Routing gerarchico

→ Router di diverso livello gerarchico

→ Molto utilizzato prima che OSPF fosse disponibile

→ Utilizzato su grosse reti

→ utilizzato da grossi ISP

→ Tutt'ora utilizzato

→ perché non si cambia ciò che funziona

- **Exterior Gateway Protocol**

BGP, in sostanza l'unico protocollo usato nella famiglia di protocolli Exterior Gateway Protocol (EGP)

→ Attualmente alla versione 4

→ RFC 1654 (1994)

→ Path vector

→ Sequenza di AS fino alla destinazione, protocollo di tipo path vector; la sequenza dice come raggiungere la destinazione per attraversamento di AS

→ Politica di scelta delle route configurabile

InterDomain Routing Protocol (IDRP)

→ Evoluzione di BGP per OSI

→ Adattato a TCP/IP

→ Sarebbe dovuto essere "la" scelta di EGP per IPv6

→ Non molto usato, causa nuove versioni di BGP

- **Content Delivery Network**

Reti molto usate, per fornire servizi web, quindi per assicurarsi che l'accesso al web da parte degli utenti sia veloce ed abbia buone prestazioni, ma soprattutto per servizi tipo il video on demand.

L'idea di base è che, dovendo portare contenuti all'utente, saranno create copie dei contenuti vicino all'utente, con server che hanno delle cache, ovvero copie temporanee dei contenuti, oppure con repliche dei contenuti. Il client scaricherà i contenuti da una replica locale (il server prende il nome di "replica server"). Le aziende (ad esempio CAMAI) quindi installano in giro per il mondo repliche di server sulle quali vengono fatte copie di contenuti di qualsiasi genere a cui l'utente può accedere senza accedere

al server originale, con prestazioni migliore, cioè con maggiore velocità. L'azienda vende questo come un servizio.
Quando il client effettua una richiesta (GET www.netscire.it) esso deve accedere al server a lui più vicino, con i server che hanno indirizzi diversi e si troveranno in posti diversi. Gli approcci sono diversi, alcuni non standard. Uno è il DNS, oppure riscrivere le URL, oppure usare anycast.

Come funziona?
→ DNS, che fornisce un indirizzo di una replica locale
→ Riscrittura delle URL, che funziona per le pagine web
→ Anycast, un tipo di consegna pacchetti particolare per cui i pacchetti non vengono mandati all'indirizzo specifico di una stazione ma ad un indirizzo anycast e poi la rete, per mezzo dei router, si occupa di fare la consegna; l'indirizzo corrisponde ad un gruppo di server ed i router si fanno carico di consegnare il pacchetto al server che appartiene a quel gruppo e che è più vicino al mittente. L'anycast non è molto usato.

- **Multicasting IP**

E' un altro tipo di consegna particolare.

Concetto alla base
I pacchetti sono portati dalla sorgente a più destinazioni
→ E' la base per comunicazioni di gruppo, da uno a molti oppure da molti a molti
    → Per esempio videoconferenza, video broadcasting
→ L'indirizzo identifica il gruppo, di stazioni

Indirizzi multicast
→ Indirizzi di classe D
    → Cominciano per 1110
    → 224.0.0.0 - 239.255.255.255
→ L'indirizzo identifica un gruppo di host e non una particolare stazione, cioè non una particolare interfaccia
→ Il pacchetto è consegnato a tutti gli host del gruppo
→ Il gruppo selezionato di host può essere ovunque nella rete Internet, cioè ovunque

nel mondo

Questo si traduce in due fasi principali: come consegnare un pacchetto ad un gruppo di host all'interno di una stessa rete fisica, ad esempio di una rete IEEE 802, ovvero una rete Ethernet. Poi di farlo oltre la rete.

In una rete IEEE 802

→ La consegna di gruppo è delegata al livello inferiore (MAC)

→ Cioè all'indirizzo IP multicast (che rappresenta un gruppo) è fatto corrispondere un indirizzo MAC multicast

→ 01-00-5E-0 (1 bit), indirizzo destinatario MAC, a questo indirizzo seguono i 23 bit meno significativi dell'indirizzo IP multicast

→ I 23 bit meno significativi dell'indirizzo IP multicast; quindi, dato un indirizzo IP multicast, siamo in grado di costruire un indirizzo MAC multicast, in cui la corrispondenza non è 1 a 1 perché l'indirizzo IP multicast è 32 bit, di cui i primi 4 di valore fisso per essere di classe D. Quindi ci sono 28 bit per 228 possibili indirizzi, che sono mappati su 223 possibili indirizzi MAC. Ci possono dunque essere sovrapposizioni, ma questo non crea problemi

→ Per essere sicuri che una stazione riceva i pacchetti, la scheda di rete della stazione deve essere configurata a ricevere trame (Ethernet) per tale indirizzo MAC multicast, corrispondente agli indirizzi IP multicast che gli interessano

→ Associazione al gruppo iniziata dal destinatario

→ I pacchetti mandati all'indirizzo multicast sono ricevuti da tutti gli host associati al gruppo; quindi il fatto di ricevere i pacchetti inviati ad un certo indirizzo dipende da chi riceve e non da chi manda

Oltre la singola rete

→ I router scoprono host group su ogni LAN, ogni singola rete, per mezzo del protocollo IGMP

→ Internet Group Management Protocol (IGMP)

→ I router annunciano host group agli altri, usando protocolli di routing multicast. I router a questo punto sanno dove sono in giro per la rete Internet quelli interessati ad

un certo gruppo multicast, si costruiscono le loro route, i loro alberi di distribuzione e quando arriva un pacchetto per un certo gruppo multicast, su una rete locale, ricevono il pacchetto e lo inoltrano verso tutte quelle reti dove ci sono delle destinazioni

Stato dell'utilizzo

→ Non supportato in modo diffuso

→ Non si sposa con la pratica comune per il controllo e ingegnerizzazione di traffico

→ Per lo più è limitato ad ambienti controllati

→ Per esempio soluzioni di video broadcasting su IP

Lezione n. 26

Sicurezza delle informazioni

- **Sfide nella sicurezza delle informazioni**

La prima riguarda la segretezza, o privacy; la seconda è l'integrità; la terza è l'autenticazione e l'ultima è la ripudiabilità.

Segretezza/privacy

Un osservatore non può accedere all'informazione

→ Per esempio qualcuno che intercetti i pacchetti mentre transitano nella rete

→ Particolarmente semplice violare la privacy in comunicazioni wireless perché consiste nel sintonizzarsi alle giuste frequenze

Integrità

Assicurarsi che i dati non siano stati manipolati

→ Per esempio pacchetti modificati (sia intestazione, sia contenuto) mentre transitano attraverso la rete

Autenticazione

L'autore è quello che ci si aspetta

→ Un utente o un sistema invia dati facendo finta di essere un altro

→ In alcuni casi l'autenticazione include anche integrità

Non ripudiabilità

L'autore non può negarlo

→ Dopo aver fatto un'operazione un utente possa dichiarare che sia stato qualcun altro

→ Per esempio firma on-line di un contratto

- **Crittografia**

→ Offre soluzioni per tutti i casi sopraelencati

→ Letteralmente: scritto nascosto

→ Ha a che fare con tecniche e protocolli per proteggere le informazioni e per rendere sicuro lo scambio di informazioni

Criptazione [encryption], su cui è basata la crittografia

→ L'informazione è rappresentata da un codice

→ La versione codificata non svela l'informazione

→ Sono necessari specifici tecniche e parametri (segreti, più che altro i parametri, che vedremo saranno le cosiddette chiavi) sono necessari per rivelarla

Il funzionamento di base, come rappresentato in figura a lato, è che, data l'informazione, ad esempio un testo in chiaro, si applica un algoritmo normalmente noto che usa un parametro (chiave) che è normalmente segreto. Quello che si ottiene è un messaggio criptato, cioè codificato in un modo che non è intellegibile.

Per rendere il sistema di criptazione più robusto devono essere usate chiavi lunghe, in termini di sequenza di bit.

Quando si è ottenuto un messaggio criptato, ci sarà la possibilità di decriptarlo, applicando un algoritmo di decriptazione, che usa la stessa chiave (che deve rimanere segreta tra chi vuole scambiare tali dati), che permetterà di ottenere il messaggio originale in chiaro.

Decriptazione [decryption]

Chiave condivisa/segreta

→ La chiave deve essere condivisa in modo sicuro

→ Per esempio off-line

→ Richiede una relazione preesistente

→ Se la chiave è compromessa da uno dei partecipanti, nessuno la può più usare

Questo tipo di criptazione che usa la stessa chiave per criptare e per decriptare è detta crittografia a chiave simmetrica, perché si usa la stessa chiave; è detta anche a chiave segreta perché essa deve rimanere segreta e deve essere conosciuta solo tra le persone, o le stazioni, che devono scambiare informazioni.

Crittografia a chiave asimmetrica

Nel voler trasferire un messaggio in modo sicuro con questa tecnica, cifriamo il

messaggio con un algoritmo e una chiave ottenendo un messaggio cifrato che può essere trasferito attraverso la rete. Il ricevente userà un algoritmo di decriptazione e una chiave diversa da quella usata per criptare il messaggio; il fatto di usare chiavi diverse sta alla base del nome crittografia a chiave asimmetrica. Un algoritmo molto famoso è l' RSA (chiave di 2048 bit).

Questo tipo di crittografia è detto anche a chiave pubblica, in quanto una delle due chiavi può essere distribuita pubblicamente e usata per criptare, questa è la chiave pubblica. C'è poi la seconda chiave, detta chiave privata, che serve per decriptare (ad esempio la smart card è un piccolo processore con un pò di memoria che contiene una chiave privata ed il processore è in grado di eseguire operazioni di crittografia asimmetrica).

Crittografia a chiave pubblica

→ Una delle due chiavi può essere distribuita pubblicamente

→ Usata per criptare, essa è la chiave pubblica

→ Solo chi ha la chiave privata corrispondente alla chiave pubblica può decifrare il messaggio

→ La chiave privata non deve mai essere condivisa

→ Più facile tenerla al sicuro

→ Le chiavi sono complementari

→ Si può fare un deposito (pubblico, un repository) per le chiavi pubbliche

→ Associate ai loro possessori

- **Come affrontare le sfide della sicurezza delle informazioni**

Segretezza/ Privacy

A vuole mandare un messaggio cifrato a B; A va a prendere la chiave pubblica di B dal deposito di chiavi, cifra il messaggio e lo manda a B. Quando B riceve il messaggio, usa la sua chiave privata per decriptare il messaggio ed ottenere il messaggio in chiaro.

Crittografia simmetrica e asimmetrica

→ (De)criptazione asimmetrica richiede più capacità di calcolo

→ La criptazione asimmetrica è usata per condividere in modo sicuro una chiave

segreta

→ La chiave segreta condivisa è usata per cifrare i dati con un algoritmo simmetrico, che richiede meno potenza di calcolo

→ La chiave segreta è cambiata periodicamente, perché nessun algoritmo è completamente robusto.

Le soluzioni di protocolli per la sicurezza, come IPsec, oppure SSL, sono basate sul concetto di usare la crittografia asimmetrica per negoziare chiavi che vengono cambiate periodicamente e così si ha una soluzione molto robusta.

Autenticazione

E' risolta anch'essa tramite la crittografia asimmetrica.

A vuole comunicare con B che vorrebbe essere sicuro dell'identità di A. Dunque A usa la sua chiave privata per cifrare il messaggio e lo manda a B, che vede che il mittente è A. A questo punto B va al deposito di chiavi e preleva la chiave pubblica di A. Le chiavi sono intercambiabili, ciò che viene cifrato con una può essere decifrato solo con l'altra. Quindi B prova a decifrare il messaggio usando la chiave pubblica di A. Se il messaggio che viene fuori è il messaggio corretto, cioè è un messaggio che ha senso, allora vuol dire che effettivamente quello cifrato era stato cifrato con la chiave privata associata alla chiave pubblica di A. In questo modo B ha accertato l'identità di A ed inoltre che il messaggio non è stato cambiato, proprio per il fatto che il messaggio che esce fuori sia sensato.

In realtà, per fare l'integrità del messaggio si usa la firma digitale, la firma digitale fornisce anche l'autenticazione del mittente del messaggio.

Firma digitale (integrità-autenticazione)

A genera un riassunto molto succinto del messaggio che deve inviare a B, quindi viene generato quello che si chiama un message digest e che ricorda il codice di rilevamento dell'errore nei protocolli di basso livello ed anche di livello trasporto. Il codice di rilevamento dell'errore è una sequenza limitata di bit o byte che si può calcolare automaticamente da tutto il pacchetto (quindi da una sequenza più grande di bit o byte) tale per cui anche se cambia anche solo un bit nel messaggio, il codice di rilevamento dell'errore cambia. Il message digest è lo stesso concetto, però progettato

non per dei pacchetti che sono messaggi di una certa dimensione ma sono dei documenti più grossi. Due messaggi leggermente diversi produrranno message digest diversi, per mezzo di algoritmi dedicati.
Fatto il message digest A prende la sua chiave privata e cripta il message digest e poi associa il message digest al documento. Questo è il documento firmato. Il message digest cifrato con la chiave privata di A è la firma. Infatti questo ci permette di identificare che il contenuto del documento è quello originale, ma anche chi ha scritto il documento è quello originale, ed è il possessore della chiave privata che ha cifrato il message digest.
A questo punto B deve saper verificare la firma quando riceve il messaggio. Quindi B recupera la chiave pubblica di A e decifra il message digest. Poi, siccome il message digest era stato creato con un algoritmo noto, B crea localmente il message digest di quello che ha ricevuto. A questo punto B deve confrontare il digest ricevuto decifrando quello che ha andato A e quello che ha generato lui stesso. Se sono uguali allora questo vuol dire due cose, che il messaggio non è cambiato e che chi aveva cifrato quel messaggio ha la chiave privata corrispondente alla chiave pubblica che B ha usato per decifrare il digest, quindi il mittente A. Quindi è stato autenticato il mittente ed è stata appurata l'integrità del messaggio. Se qualcuno avesse cambiato il messaggio strada facendo, quando B calcola il digest risulta diverso. C'è da notare che chi cambia il messaggio non può cambiare il digest perché il digest è cifrato con la chiave di A. Se qualcuno cambia il messaggio, e ricalcola il digest con una chiave diversa da quella di A, al momento della verifica, si avrà una diseguaglianza. Questo è un meccanismo molto potente e molto utile.

Cryptographic Digest (Hash)
Quello che si fa realmente è generare dei digest crittografici, chiamati spesso anche hash, per ci sono algoritmi che, dato un messaggio ed una chiave pubblica, generano direttamente un cryptograhic digest, quindi il digest già cifrato che è una firma elettronica (o digitale) del messaggio.
La firma elettronica dipende dunque dal contenuto stesso del messaggio: questa soluzione è molto potente.
Algoritmi molto noti per la generazione del cryptographic digest sono MD5 e SHA. Il

primo genera dei digest di 128 bit, il secondo di 160.

Un problema di autenticazione lo abbiamo se la chiave pubblica del firmatario non appartiene al firmatario. C'è quindi un problema di autenticazione ed integrità delle chiavi pubbliche stesse. Questo si può risolvere con la firma digitale.

- **Certificati digitali**

Detti anche certificati a chiave pubblica, perché basati sulla crittografia a chiave pubblica.

Di cosa si tratta? Una chiave con una etichetta, firmate.
Quindi il certificato digitale contiene sostanzialmente una chiave, il possessore della chiave, una firma che garantisce l'integrità del tutto.
Si tratta di verificare chi firma i certificati digitali: essi sono firmati da una autorità di certificazione, CA.

Certification Authority [autorità di certificazione]
→ La CA verifica l'identità del possessore della chiave prima di firmare
→ Per esempio il possessore della chiave deve presentarsi di persona con un documento; questa è una operazione che deve essere fatta off-line e non può essere fatta on-line.
→ Il certificato digitale è utilizzabile per assicurare il non ripudio delle informazioni. Questo perché quando qualcuno firma le informazioni con la chiave che è nel certificato, l'identità legale è confermata in quanto la CA ne ha effettuato la verifica in modo fisico.

A lato un esempio di certificato, con dettagli vari.
La firma è sotto, in fondo, di 256 byte. La firma garantisce che nessuna delle informazioni riportate sopra può essere cambiata.
Se viene cambiata una sola informazione, allora la firma non è più verificabile.
Si noti l'algoritmo usato per la firma, algoritmo usato dalla CA del Politecnico di Torino per fare la firma. C'è scritto chi ha fatto la firma, la validità e una chiave pubblica , su 256 byte, che si certifica essere posseduta da Mario Baldi.

La chiave pubblica può essere usata con un particolare algoritmo di decriptazione, che è indicato.
Il punto chiave è che è legata una identità ad una chiave pubblica, per mezzo di una Certification Authority che ha firmato il tutto.
Il certificato deve anche poter essere certificato. Si dovrebbe prendere la chiave pubblica della CA e verificare il certificato. Ma il problema è che non c'è una sola CA, per cui quando sono stati definiti i certificati digitali è stato definito anche il Public Key Infrastructure, che risolve il problema del non poter avere una sola CA, così come lo sarebbe se ci fossero più CA indipendenti, con la difficoltà che deriva tra il confrontare i certificati. La soluzione è che ci possono essere più CA, ma esse sono organizzate in una gerarchia di Ca.

PKI: Public Key Infrastructure
→ Sarebbe irrealistico avere
    → Una singola CA
    → Più CA indipendenti
Gerarchia di certification authority
Il che porta al problema di come verificare un certificato.

Verifica dei certificati
Occorre verificare la firma della CA. Quindi occorre avere la chiave pubblica della CA, per cui c'è bisogno del certificato della CA. Tale certificato sarà firmato da un'altra CA, e quindi si va a recuperare il certificato di quest'ultima Ca, e così via lungo la gerarchia, fino al livello più alto che è la Root Certification Authority ed essa firma i certificati da sola. Poichè questa non è verificabile allora quando si richiede un certificato si ottiene, in modo sicuro, anche per la propria chiave pubblica, il certificato della Root CA. Questo ci permette di verificare i certificati di tantissime CA che appartengono alla stessa gerarchia.

Rilascio dei certificati
→ La CA verifica l'identità legale del possessore
    → Eventualmente attraverso una Registration Authority

→ La CA crea la coppia di chiavi

→ La CA firma il certificato, che valida la chiave pubblica

→ Il certificato con la chiave pubblica viene messo in un deposito [repository]

→ La chiave privata è consegnata al possessore

→ Il certificato della root CA è dato al possessore del certificato

Tutto il mondo si fida di una stessa Root CA?

→ Sfortunatamente no

→ Ci sono diverse Root CA

Come otteniamo i loro certificati?

Inclusi nei sistemi operativi e browser → Ci fidiamo tutti dei produttori di software!?! (forse solo perché non lo sappiamo)

Lezione n. 27

Sicurezza di rete

- **Tipi di attacchi**

Snooping [spiare]

→ Sui collegamenti

→ Wireless

→ Sonde, su collegamenti fisici

→ Può essere fatto anche nei nodi

→ Il traffico può essere dirottato verso un punto di osservazione

Perturbazione del servizio

→Perturbazione del routing

→ Messaggi di routing fasulli immessi nella rete

→Informazioni DNS fasulle, immesse nel DNS

→ DNS poisoning

→Sovraccarico router o host

→ (Distributed) Denial of service

Exploit [debolezze sfruttate]

→ Normalmente software bug

→ Accesso non autorizzato ad un nodo

→ Furto di informazioni

→ Perturbazione servizio

→ Causare “crash” del nodo

→ Attivare percorsi di esecuzione poco comuni, che sono quelli testati di meno, per mezzo di pacchetti inusuali

→ Invio di pacchetti inusuali

→ Per esempio frammenti IP sovrapposti

→ Bug nella pila di rete, implica “crash” nel sistema

→ Richieste inusuali alle applicazioni

Furto di “identificatore” di rete

→ Address spoofing [appropriazione di indirizzo]

→ Server falsi

→ Modifica di pacchetti “al volo”

Esecuzione inconscia di codice dannoso

→ Computer virus

→ Eventualmente in e-mail

→ Trojan horse [cavallo di Troia]

→ Nascosto in un programma

→ Worm [verme]

→ Si propaga tramite rete, quando vanno in esecuzione

Obiettivi

→ Accesso (backdoor), al sistema tramite apertura di porte

→ Furto di dati

→ “Crash” del sistema

→ Uso dell’host (botnet)

→ Attacchi, anonimizzazione

→ Spiare (video, tastiera)

Inoltre

→ “indovinare” password

→ Per esempio usando dizionari

→ Phishing

→ Frodi tramite e-mail

Soluzione: educazione utente

- **Difese all'interno della rete**

Molte difese sono basate su tecniche crittografiche

Tecniche crittografiche

→ Criptazione dei contenuti

→ Autenticazione

→ Integrità dei contenuti

→ Identità del mittente

Queste tecniche possono essere utilizzate a diversi livelli protocollari. Servono dei protocolli.

Protocolli crittografici

→ Negoziare algoritmi

→ Condividere e cambiare chiavi

→ Ottenere certificati

Tecniche di filtraggio

→ Traffico

→ Firewall (personale, eseguito non nella rete, ma nell'host, oppure sulla rete). Il firewall scarta il traffico non autorizzato

→ Applicabile a livello rete o a livello applicativo

→ Antivirus (personale o nella rete)

→ Tecniche di filtraggio nell'esecuzione, in modo da controllare in fase di esecuzione che il codice sia approvato, non modificato; si possono usare tecniche crittografiche per firmare il codice o di verifica del codice durante l'esecuzione, ad esempio la trusted computing platform, che fa in modo di eseguire su un calcolatore codice approvato e non malevolo

Il monitoraggio è un altro metodo di protezione. Consiste nel controllare continuamente quello che accade nella rete.

Monitoraggio

→ Intrusion detection systems (IDS)

→ [sistemi per identificazione di intrusioni]

→ Identificazione malware

→ Identificazione anomalie

- **IPsec IP Security**

Esso crea un framework (una infrastruttura) per permettere a due entità in comunicazione di mettersi d'accordo su quali protocolli usare, per autenticarsi, per scambiarsi le chiavi, e per capire quali sono gli algoritmi da usare, sia per la criptazione, sia per l'autenticazione.

Caratteristiche

→ Criptazione e autenticazione

→ Un paio di chiavi di sessione per ogni direzione, quindi in totale ci sono quattro chiavi

→ Una per criptazione

→ Una per autenticazione

→ Cambiate periodicamente

IPsec usa uno schema detto IKE, Internet Key Exchange.

Internet Key Exchange (IKE)

Usato per accordarsi su

→ Protocolli da usare per scambiare le chiavi

→ Algoritmi che si usano per la criptazione e per l'autenticazione

→ Criptazione (DES, 3DES, RC5)

→ Autenticazione (MD5, SHA1)

→ Chiavi, cioè IKE è usato per i protocolli che servono per accordarsi sulle chiavi

IKE

→ Include svariati protocolli, esso è un framework

→ Per esempio ISAKMP: Internet Security Association and Key Management Protocol

→ Autenticazione dei comunicanti

→ Scambio chiavi tra i comunicanti

Chiavi iniziali, possono essere basate

→ su un segreto condiviso, devono avere una relazione preesistente

→ su certificati digitali

Scambio di chiavi

→ Diffie-Hellman, è un algoritmo

→ Crittografia a chiave pubblica per autenticare i comunicanti da un lato e poi dall'altro la criptazione delle chiavi scambiate

→ Criptazione delle chiavi scambiate

→ DES

Una volta che le due entità di comunicazione hanno le chiavi per cifrare, i dati dovranno essere cifrati e comunicati.

IPsec lavora a livello IP e ha due modalità di funzionamento differenti: transport mode encapsulation e IPsec tunneling

Transport Mode Encapsulation

Prevede la cifratura delle informazioni di livello trasporto ed è basata su un formato di pacchetto detto ESP, per il quale, se abbiamo un pacchetto che deve andare da S a B, ed il pacchetto ha l'intestazione IP, contiene un messaggio TCP, con l'intestazione ed i dati, viene aggiunto tra l'intestazione IP e l'intestazione TCP una intestazione aggiuntiva ESP; ESP definisce dunque una intestazione, ma anche una coda al messaggio, come da figura. L'intestazione ESP permette di autenticare tutto il campo payload del pacchetto IP originale, e quindi in questo caso il messaggio TCP. Da questo il nome Transport Mode Encapsulation, in quanto va ad incapsulare il livello trasporto e a proteggere e ad autenticare il livello trasporto e a cifrare il livello trasporto. Qualcuno che guarda il pacchetto (campo dati) non riesce a vedere cosa contiene il pacchetto. Quello che riesce a vedere è che esso è un pacchetto IP che va da S a D, vede che c'è una intestazione ESP, in cui ci saranno informazioni per il destinatario su quali algoritmi usare per decifrare e verificare l'autenticità. Si noti che l'autenticazione prevede anche l'intestazione ESP ed una parte di coda ESP, questo affinche nessuno possa modificare tale informazione senza che il ricevente se ne accorga.

Authentication header

C'è poi un'altro tipo di header, detto authentication header, che non fornisce una

soluzione di cifratura, ma solo di autenticazione, diversa da quella dell'ESP, in quanto questa copre anche l'intestazione IP. La conseguenza è che usando l'authentication header, nessuno può modificare l'intestazione IP, ma può solo vederne i contenuti. Si ha dunque una funzionalità di integrità, e anche di autenticazione in quanto esse sono strettamente legate.

L' authentication header non fornisce funzionalità di cifratura e, per questo, si può usare insieme all'ESP.

Da notare che con una soluzione del genere si avranno problemi con il NAT, in quanto l'indirizzo mittente e/o quello destinazione potrebbero essere cambiati e, vedendo che il pacchetto non è integro, la destinazione scarterà il pacchetto.

Riassumendo il Transport Mode Encapsulation

→ Usata per comunicazioni tra host

→ ESP: Encapsulation Security Payload

→ Authentication header

Esiste un'altra modalità dell'IPsec, detta IPsec Tunneling

IPsec Tunneling

→ VPN: Virtual Private Network, in cui si cerca di collegare reti aziendali attraverso una rete pubblica come Internet. Si hanno dei dispositivi detti VPN gateway o, IPsec gateway se si usa IPsec. Questi dispositivi prendono un pacchetto, ad esempio che transita da A a B e lo mettono in un altro pacchetto che va da X a Y. Quando il pacchetto arriva a Y, Y vede che il pacchetto è per lui e vede che dentro c'è un altro pacchetto, toglie il pacchetto interno e lo immette nella rete aziendale. Questa operazione è detta, in generale, tunneling. IPsec prevede di funzionare in questa modalità, quindi fare il tunneling, ed inoltre cifrare il contenuto del pacchetto.

Quindi, così facendo, si parla di tunnel mode encapsulation

Tunnel Mode Encapsulation

Nel tunnel mode incapsulation si ha il pacchetto originale di prima, che va da S a D, imbustato in un pacchetto che va da X a Y, protetto con le varie tecniche viste prima:

1. con una sola intestazione ESP
2. con una intestazione AH che proteggerà sia l'intestazione interna che quella esterna

3. con un ESP che nasconde completamente il pacchetto interno ed una intestazione AH che protegge tutto quanto.

Nell'usare il tunnel mode, un osservatore non riesce a leggere gli indirizzi del mittente e del destinatario, cioè il pacchetto originale è completamente nascosto all'osservatore, mentre con il transport mode encapsulation gli indirizzi originali sono sempre visibili.

IPsec è una soluzione abbastanza robusta, ma complicata da utilizzare, in quanto ha tanti parametri di configurazione e tante modalità di funzionamento diverse, esiste dunque un'altra soluzione, molto più utilizzata, che si chiama Secure Socket Layer, SSL.

- **SSL Secure Socket Layer**

Nell'architettura di protocolli, i socket sono una interfaccia di programmazione che i livelli superiori possono usare per accedere ai servizi del livello trasporto. Normalmente le applicazioni, i protocolli di livello applicativo, usano i socket per chiedere l'apertura di connessioni TCP, l'invio di messaggi TCP, l'invio di messaggi UDP. L'SSL si preoccupa di cifrare e di autenticare le informazioni che le applicazioni cercano di mandare su una connessione TCP o in un messaggio UDP prima che questi vengano inseriti nel messaggio

Caratteristiche

→ Autenticazione delle due entità in comunicazione

→ Permette la creazione di una sessione di trasporto sicura, che è criptata ed autenticata

→ TLS: Transport Layer Security, che è la versione standard di SSL, che nasce nella comunità Internet, ma non è standard

SSL e TSL sono attualmente implementati in librerie che sono utilizzate dalle applicazioni. Le due soluzioni sono molto simili ed hanno differenze molto piccole. Esse sono largamente utilizzate, per rendere sicuri protocolli che normalmente non lo sono e che, per mezzo di SSL lo diventano, subendo anche una modifica nel nome:

→ POPS, Secure IMAP, Secure SMTP, HTTPS, SFTP

→ Normalmente diversa porta

→ HTTP: 8O, HTTPS: 443

→ Eventualmente la stessa: STARTTLS

SSL assicura uno “strato sicuro” al di sopra del livello trasporto.

Normalmente, quando il server usa il protocollo sicuro, aspetta su una porta diversa.

La porta può essere la stessa se si usa il TLS, in quanto esso prevede un protocollo che si chiama STARTTLS che consente al server ed al client di accordarsi sull’usare la modalità sicura o no, mentre quando il client usa l’SSL si aspetta che il server usi l’SSL, ecco perché usa una porta diversa.

Di seguito vediamo un meccanismo generale dell’SSL, di come inizia la comunicazione (SSL Handshake).

SSL Handshake [presentazione]

Quando il client apre una connessione, ad esempio TCP, al server, e la connessione è stata aperta, il client manda un messaggio ClientHello al server, che è un messaggio dell’SSL. Il server risponde fornendo al client il proprio certificato digitale che contiene la chiave pubblica del server. Il client a questo punto può verificare l’autenticità del certificato e questo lo fa avendo una serie di certificati di certification authority messi sul sistema dal costruttore del sistema operativo del client; se non ce l’ha segue la procedura vista a suo tempo. Dopo la verifica del certificato il client genera una chiave simmetrica, quindi una chiave da usare con crittografia simmetrica, e la manda al server criptandola usando la chiave pubblica del server. In questo modo un osservatore non può capire quale è la chiave. A quasto punto il client ed il server hanno una chiave privata che possono usare per comunicare.

In realtà, in questa fase, il client ed il server negoziano tutta una serie di parametri.

Negoziazione parametri

→ Il client offre

→ Lista di “cypher”, che è una lista di algoritmi di criptografia che è in grado di usare con relativi parametri

→ Parametri

→ Il server

→ Sceglie i cypher

→ Può richiedere il certificato del client, per verificare l’identità del client

Caratteristiche legate alla sicurezza, esse dimostrano che questo protocollo è parecchio sicuro

→ Solo hello e cert del server sono “in chiaro”, tutto il resto viene cfrato

→ Il client ed il server usano una coppia di chiavi di sessione per direzione

→ Criptazione

→ Autenticazione

→ Cambiate periodicamente

- **Firewall**

Un filtro di pacchetti

Regole basate sui campi dei protocolli

→ Regole del tipo: indirizzi mittenti A-F possono comunicare con server S

→ Porta P usata su server Q

→ HTTP ammesso su server R

→ Scarta ogni altro pacchetto

Diversi tipi di firewall

→ Regole a livello di singolo pacchetto (stateless)

→ Regole a livello flusso, cioè del tipo “ammetti le connessioni TCP aperte da un cliente interno alla rete aziendale”

→ Stateful, osservazioni di un certo periodo per mettere in correlazione informazioni diverse

→ Application firewall, vanno a guardare i dati a livello applicativo

→ Firewall in grado di identificare malware e virus

I firewall devono essere in posizione strategica.

Dal punto di vista fisico si realizzano sottoreti diverse, quella interna privata ed una pubblica, detta de-militarized zone, DMZ, in cui l’accesso è permesso dall’esterno, ma in modo controllato, solo verso server predefiniti.

In quella interna l’accesso è estremamente controllato ed è permesso solo dai server che si trovano nella zona demilitarizzata.

Lezione n. 28

IP versione 6 (Ipv6) - Prima parte

- **Una nuova versione di IP: come mai?**

Perché un nuovo IP?

Una sola VERA risposta:

uno spazio di indirizzi più grande

Altre risposte

→ Più efficiente sulle LAN, vero, ma non [una differenza cos' importante

→ Multicast e anycast

→ Sicurezza maggiore

→ Policy routing, inoltrare pacchetti secondo dei criteri diversi che non siano l'indirizzo della destinazione

→ Plug and play, le stazioni non hanno bisogno di essere configurate

→ Differenziazione traffico, traffico trattato in modo diverso dipendente dalle sue caratteristiche

→ Mobilità, cioè supportare stazioni che si muovono

Queste funzionalità si hanno anche su IP4, di cui sono una aggiunta, mentre in IPv6 ne fanno parte "dalla nascita".

Perché gli indirizzi IPv4 scarseggiano?

Sono lunghi 32 bit, quindi circa 4 miliardi di indirizzi, tuttavia ..

... essi sono sono usati in modo gerarchico

→ Il prefisso usato in una rete fisica non può essere usato in una diversa

→ Quindi ci sono tantissimi indirizzi inutilizzati

Ad esempio per un prefisso di 16 bit, restano 65535 indirizzi possibili da utilizzare per quella rete fisica, ma se la rete fisica ha solo 20000 host, 45000 indirizzi sono inutilizzati.

In IPv6 vogliamo avere più indirizzi, utilizzati in modo gerarchico (cosicché i router si devono preoccupare di dove sono le reti fisiche identificate dai prefissi), dunque ci sarà spreco di indirizzi anche in IPv6.

Allora, quanti indirizzi dovrebbe avere IPv6?

→ Un approccio scientifico, per mezzo dell'adozione di una addressing efficiency

H = log 10 (numero di indirizzi) / numero di bit degli indirizzi utilizzato, che è il numero totale di indirizzi a disposizione

Addressing Efficiency, fatto prendendo delle misure

→ In reti esistenti (allora)

→ H varia tra 0.22 e 0.26

→ Assumendo un milione di miliardi di stazioni in rete

→ 68 bit nel caso di efficienza minima, cioè 68 bit sono quelli necessari

Melius abundare quam deficere

Sono stati scelti, alla fine, 128 bit, ovvero 16 byte.

655.570.793.348.866.943.898.599 indirizzi IPv6 per metro quadro di superficie terrestre

- **Indirizzi IPv6**

Come li scriviamo e come li usiamo

Notazione: 8 numeri esadecimali separati da ":"

Gruppi di 2 byte

FEDC:BA98:0876:45FA:0562:CDAF:3DAF:BB01

1080:0000:0000:0007:0200:A00C:3423:A089

Scorciatoie

Per visualizzare indirizzi in modo user-friendly

Gli 0 iniziali di ogni gruppo di cifre possono essere omessi

→ 1080:0:0:7:200:A00C:3423

Gruppi di 0 possono essere sostituiti da "::"

→ 1080::7:200:A00C:3423

Organizzazione dello spazio degli indirizzi (2128)

→ Indirizzi Multicast

→ 1111 1111, il primo byte è fatto da tutti 1

→ FFxx:... è un indirizzo multicast

→ Indirizzi Link local/site local

→ Equivalenti ad indirizzi IPv4 privati

→ 1111 1110 1

→ Link local: FE80::/64, cioè i successivi 64 bit hanno un qualsiasi valore; con questa notazione si indica un prefisso in uso; gli indirizzi Link Local si usano solo su una rete locale e sono gli equivalenti degli indirizzi automatici IPv4, quelli che iniziano per 169.254

→ Site local: FEC::/10, sono l'equivalente degli indirizzi privati in IPv4 come 10.0.0.0

Gli indirizzi rimanenti prendono il nome di Global Unicast, cioè indirizzi unicast globali e servono per dare indirizzi alle interfacce delle stazioni. Di seguito come è organizzato lo spazio global unicast.

- **Indirizzi rimanenti (Global Unicast), Unicast Globali**

Organizzazione dello spazio Global Unicast

→ Indirizzi per IPv4 interoperability, sono indirizzi dedicati alla possibilità di avere stazioni IPv4 e IPv6 che coesistono ed operano sulla stessa rete; sono indirizzi che hanno i primi 80 bit a zero; di questi indirizzi ce ne sono due tipi: IPv4-mapped e IPv4-compatible. Gli indirizzi IPv4 interoperabilty includono un indirizzo IPv4 in un indirizzo IPv6

→ 0...0 (80 bit) → 0::/80, tipica notazione IPv6

→ Per la fase di transizione da IP4 a IPv6

→ Indirizzi IPv4-mapped, gli 80 bit a 0 iniziali sono seguiti da 16 bit a 1, per cui sono "impegnati" 96 bit, con 32 bit finali in cui metteremo un indirizzo IPv4

→ 16 bit a 1 → 0:0:0:0:0:FFFF::/96

→ Indirizzi IPv4-compatible, gli 80 bit iniziali a 0 sono seguiti da ulteriori 16 bit a 0

→ Altri 16 bit a 0 → 0::/96

Per esempio 0::0::0::0::0::0::A00::1

→ Notazione compatta, degli indirizzi IPv4-compatible

→ ::A00:1

→ Notazione speciale, adottata per indirizzi IPv4-compatible

→ ::10.0.0.1

Aggregatable Global Unicast [aggregabili]

Sono indirizzi che identificano le stazioni globali su tutta la rete, quindi non locali. Poter aggregare significa avere un prefisso comune a tutte le reti che stanno, ad esempio in Europa in modo da poter avere nei router una unica informazione di routing: per raggiungere l'Europa "vai da quella parte". Affinché questo sia possibile occorre che in Europa tutti gli indirizzi abbiamo lo stesso prefisso, cioè che siano assegnati in modo da avere un prefisso comune, cosa che in IPv4 non si faceva.

→ Tali indirizzi iniziano per bx001

→ Prima cifra in esadecimale è 2 o 3

→ Assegnazione topologica, con la gerarchia che esiste tra i service provider; facendo così otterremo una efficace aggregazione

I service provider di Tier 1 riceveranno un prefisso, quelli di Tier 2, che ad essi sono collegati, riceveranno un prefisso ricavato da quello dei service provider di Tier 1, un pochino più lungo, e così via.

Stesso principio di routing di IPv4

La rete è divisa in sottoreti, le stazioni che si trovano nella stessa sottorete comunicheranno direttamente usando i servizi del livello 2; le stazioni che si trovano in sottoreti diverse comunicheranno atraverso i router. I router a loro volta inoltreranno i pacchetti tra di loro per raggiungere la destinazione. Tutto questo è dunque uguale a IPv4, con l'idea che i router dovranno conoscere solo un prefisso che identifica ognuna delle sottoreti. Quindi in ogni sottorete tutte le stazioni avranno lo stesso prefisso.

Questo va ad avere un impatto sulla struttura degli indirizzi.

Struttura degli indirizzi

Gli indirizzi sono divisi in due parti, una è il prefisso e l'altra è l'identificatore dell'interfaccia.

Stesso principio di assegnazione di IPv4
(diversa terminologia)
→ Subnetwork: insieme di host con lo stesso prefisso (LIS in IPv4)
→ Link in IPv6 (rete fisica in IPv4)
Subnetwork ≡ link, cioè in IPv6 come in IPv4 una subnetwork deve corrispondere ad un link e viceversa, ovvero tutti gli host sullo stesso link (host on-link) devono avere lo stesso prefisso, per cui capiranno di poter comunicare direttamente
→ Host on-link [sul link] hanno lo stesso prefisso
→ Comunicano direttamente
→ Host off-link hanno prefissi diversi, quindi comunicano attraverso router
→ Comunicano tramite router

Prefisso
E' identificato da una coppia indirizzo/netmask in IPv4, mentre in IPv6 la coppia indirizzo/netmask è sostituita da un prefisso indirizzo/N, dove N è la lunghezza di prefisso (in bit), come sotto riportato
→ FEDC:0123:8700::/36
→ 1111111011011100
00000001001000111000
In IPv6 non ci sono classi di indirizzo, per cui la lunghezza del prefisso non si capisce guardando i primi bit. Il prefisso è esplicitamente specificato nella notazione indirizzo/lunghezza del prefisso.

Assegnazione di indirizzi
Come vengono assegnati gli indirizzi ha un impatto anche su come viene organizzato il prefisso.
Negli indirizzi agregatable global unicast è deciso che il prefisso abbia lunghezza 64 bit e l'identificatore di interfaccia è di 64 bit, che è un numero di bit elevato, ma la ragione per cui si scelgono 64 bit per l'identificatore dell'interfaccia è che, se si vuole, è possibile usare l'indirizzo Ethernet in versione estesa come identificatore di interfaccia.
Questa è comunque solo una parte dello spazio di indirizzamento, con gli indirizzi che

iniziano per 001 in binario.
L'assegnazione deve essere fatta in dipendenza della gerarchia che c'è nella topologia di rete, ad esempio la gerarchia dei service provider.

Plug and Play, non è necessario configurare le stazioni

Scenari
→ Ufficio del dentista, senza amministratore di rete
→ Mille calcolatori allo scarico merci, da configurare
Soluzione: autoconfigurazione
→ Stateless: senza server
→ Statefull: con DHCP, basato su server DHCP

- **Protocolli modificati**

Ci sono cambiamenti nell'architettura di protocolli in IPv6 rispetto a Iv4.
→ IP
→ ICMP
→ ARP, che scompare
    → Funzionalità di ARP integrate in ICMP
→ IGMP, che scompare
    → Integrato in ICMP

Altri protocolli sono solo aggiornati
→ DNS (nuovo tipo di record di tipo AAAA)
→ RIP e OSPF
→ BGP e IDRP , cambia il modo in cui si chiamano le destinazioni
→ TCP e UDP, in TCP cambiano gli indirizzi per cui bisogna cambiare l'implementazione del protocollo, non il formato dei messaggi TCP o UDP, ma la implementazione dei protocolli
→ Interfaccia socket, che si usa per accedere ai servizi

- **Interfaccia di programmazione (socket)**

Che cos'è?

→ Interfaccia di programmazione per accedere ai servizi TCP/IP dall'interno di un programma, che vuole usare la rete

→ Usata nell'implementazione di applicazioni

→ per inviare messaggi UDP

→ per inviare byte su connessioni TCP

Principi fondamentali

→ Nata in ambiente Unix

→ I/O come accesso a file

→ Socket descriptor: equivalente, per l'uso della rete, ad un file descriptor

Per usare la rete si ottiene un socket descriptor per poter fare operazioni di scrittura (mandare dati attraverso la rete) e operazioni di lettura

Socket

→ E' il punto di accesso ai servizi di rete

→ Viene associato a una connessione TCP o sessione UDP

Operazioni sui socket

→ Aspettare richieste a una porta

→ Server

→ listen(), funzione dell'interfaccia

→ Accettare richieste (usate nei server)

→ Collegare alla porta di un host remoto (server)

→ Usata da un client

→ Richiede di specificare indirizzo e porta

Lezione n. 29

IP versione 6 (IPv6) - Seconda parte

- **Formato dell'intestazione dei pacchetti**

Il formato dell'intestazione IPv6 è più semplice di quella di IPv4.

E' più grande (40 byte, rispetto a minimo 20 byte per IPv4, a causa degli indirizzi più lunghi, a 16 byte). I campi sono di uguale lunghezza.

I pacchetti IPv4 e IPv6 sono imbustati in modo diverso nel protocollo di livello 2. IPv6 è organizzato come se fosse un protocollo diverso da IPv4 e non una sua versione. Questo permette di avere sulla stazione la pila protocollare di protocolli IPv4 e la pila protocollare di protocolli IPv6 insieme e coesistenti. Questo, che si chiama approccio dual stack, è fondamentale per garantire una transizione graduale da IPv4 a IPv6.

Il campo priority è simile al campo type of service di IPv4; il campo flow label è un campo nuovo, etichetta di flusso, che serve a mettere in relazione pacchetti che appartengono allo stesso flusso; IP fornisce un servizio connectionless di tipo datagram, quindi ogni pacchetto viaggia per conto suo ed è indipendente dagli altri. Si è però visto che per certi tipi di applicazioni (Audio e video) è importante capire quali pacchetti appartengono alla stessa applicazione; in IPv4 è difficile, guardando le informazioni che ci sono nel pacchetto.

Il campo Hop Limit è identico al campo time To Leave di IPv4; il campo Lunghezza dei dati è la lunghezza del campo payload; il campo next header che è equivalente al campo protocol di IPv4, secondo certi punti di vista ed esso dice che cosa segue l'intestazione.

In generale possiamo osservare che in IPv6 sono stati eliminati dei campi

→ Campi non molto utili, come quello che contiene il checksum

→ Campi non usati in ogni pacchetto, come quelli usati per la rammentazione

→ Non più necessari, come la lunghezza intestazione, poiché in IPv6 la lunghezza è fissa

Extension Header [intestazioni di estensione], sono le informazioni che non servono sempre

→ Sono aggiunte solo quando utili

→ Quindi non sono elaborate inutilmente in ogni pacchetto

Extension Header contengono
→ Hop By Hop Option, informazioni opzionali usati da ogni singolo hop
→ Routing, informazioni di tipo source routing, ovvero scrivere nel pacchetto la sequenza di router che il pacchetto deve attraversare
→ Fragment, per gestire la frammentazione, anche se essa si cerca di evitarla in IPv6 e si fa fare al mittente
→ Authentication, l'equivalente di authentication header di IPv4, serve per autenticare le informazioni nel pacchetto
→ Encrypted Security Payload, equivalente ad IPsec di IPv4, fornisce sicurezza nel trasferimento di pacchetti
→ Destination Option, contiene informazioni opzionali elaborate solo dalla destinazione

Formato degli extension header
Tutti gli extension header iniziano con l'informazione di quale è quello successivo, un altro extension header, oppure un messaggio UDP, oppure un segmento TCP.
Alcuni extension header, quelli con lunghezza variabile, hanno un campo lunghezza

Header chaining [concatenazione intestazioni]
Si può, cioè, fare una concatenazione dei vari extension header.

- **Neighbor discovery (scoperta dei vicini)**

Meccanismo importante, che è una nuova funzione di ICMP. Esso permette non solo la scoperta di nuovi router ma di un qualsiasi vicino, ovvero una interfaccia (stazione o router) collegata allo stesso link (rete fisica in IPv4). Si tratta di scoprirne l'indirizzo.
Nuova funzione di ICMP
→ Va a sostituire ARP (Address Resolution Protocol)
→ Basato sul multicast, a differenza dell'ARP
→ Con tutta probabilità coinvolge un solo host

Solicited Node Multicast Address

Questo è un particolare indirizzo multicast, che viene sottoscritto da ogni host. Esso è un indirizzo multicast, su 128 bit; esso, quindi, inizia per FF, ha una configurazionbe fissa per i primi 104 bit, i rimanenti 24 bit, quelli meno significativi, sono l'indirizzo IP

→ Sottoscritto da ogni host

→ FF02::1:FF/104 | 24 bit meno significativi di un indirizzo IP; quando una stazione diventa operativa ed ha un indirizzo su un link sottoscrive il gruppo identificato dal "Solicited Node Multicast Address" che ha una configurazione fissa nei primi 104 bit e il proprio indirizzo nei restanti 24.

→ Probabile 1 host per gruppo, in quanto è poco probabile che ci siano due indirizzi IP uguali

Trasmissione IPv6 Multicast, cioè invio di pacchetti multicast in IPv6

→ E' basata sul multicast di livello MAC, per cui il pacchetto IP è messo in una trama MAC che deve essere inviata ad un indirizzo MAC

→ Indirizzo IPv6 multicast viene fatto corrispondere all' indirizzo MAC, costruito come sotto riportato

→ 33-33|4 byte meno significativi dell'indirizzo IPv6

Risoluzione di indirizzi

Ci si basa su due messaggi

→ ICMP Neighbor Solicitation, primo messaggio

→ Inviato a Solicited Node Multicast Address, corrispondente all'indirizzo IPv6 da risolvere

→ ICMP Neighbor Advertisement, messaggio di riposta dalla stazione che ha quell'indirizzo IP

→ Inviato direttamente all'indirizzo richiedente

Esempio di risoluzione

→ Per trovare l'indirizzo MAC dell'host

2001::ABCD:EF98

→ ICMP Neigh. Sol. a Sol. Node Mult Add:

FF02::1:FFCD:EF98

→ Imbustare in trama MAC a 33:33:FF:CD:EF:98

Quando la stazione che aveva fatto la richiesta riceve la risposta, mette la risposta in un cache detta Host Cache, che tiene la corrispondenza tra indirizzi IPv6 e indirizzi MAC.

Host Cache
→ Corrispondenza tra indirizzi IPv6 e indirizzi MAC
→ Equivalente alla ARP cache

- **Transizione a IPv6 (?)**

Transizione da IPv4 a IPv6
→ Deve essere graduale
→ Senza interruzioni di servizio
→ Indolore per gli amministratori di rete
Notare la mancanza di IPv5, che è stata abbandonata in quanto non valeva la pena fare la transizione da IPv4.

Come si può realizzare la transizione?
→ Approccio dual-stack, all'interno di una stazione si usano sia IPv6 sia IPv4
    → IPv6 gestito come un nuovo protocollo
    → Le stazioni possono generare/ricevere pacchetti v6 o v4
→ Corrispondenza tra indirizzi
→ Tunneling, la possibilità di trasportare pacchetti IPv6 dentro pacchetti IPv4 e viceversa
→ Avere meccanismi di traduzione, cioè una stazione manda un pacchetto IPv4 verso una stazione IPv6

Reti IPv6 isolate

Siamo pronti?
→ Tutti i protocolli sono specificati
    → Ormai da un po': dal 1996!!

→ Implementato nei router

→ Anche se meno stabile di IPv4

→ Eventualmente non tutte le funzionalità

→ Alcune realizzazioni hardware (Layer 3 switch)

→ Implementato negli host

→ In Windows da 2000 e XP

→ In Unix, FreeBSD, Linux

→ Parecchie applicazioni

→ Eventualmente con qualche bug

Quando succederà?

→ C'è una larga base di installato IPv4

→ Un'unica vera motivazione: esaurimento dello spazio di indirizzi

→ Il problema di esaurimento degli indirizzi è stato mitigato

→ Cauta assegnazione degli indirizzi

→ Uso diffuso di indirizzi privati

→ NAT e proxy

Dunque, IPv6 non serve?

→ NAT non va bene con alcune applicazioni

→ Problematico con meccanismi di sicurezza

→ Tracciabilità degli utenti

→ Scomodo con i server

→ Non molti → indirizzi pubblici

Limitazioni finora accettabili

Ad un certo punto capiterà il puro esaurimento dello spazio di indirizzi

→ Specialmente nella regione Asiatico-Pacifica

→ IANA ha esaurito i prefissi di classe A nel Feb 2011

→ RIPE alla fine del 2011

Eventualmente legislazione

Lezione n. 30

Mobilità nelle reti IP

- **Sfide della mobilità**

Movimenti trasparenti (ad IP)

Quando una stazione si muove, può fare dei movimenti trasparenti dal punto di vista del protocollo IP.

→ All'interno della stessa rete fisica

→ Nella cella o tra celle di una rete cellulare

→ Tra BSS di un ESS WiFi, cioè tra un Basic Service Set ed Un Extended Service Set WiFi

→ Tra le porte di uno switch

→ In tutti questi casi la mobilità è gestita dal livello 2 , cioè i protocolli e i dispositivi di livello 2 (i bridge, i switch, gli acces point, le base station) di una rete mobile si occuperanno di accorgersi che la stazione non è più, ad esempio, collegata ad una porta di uno switch, ma ad un'altra, per cui manderanno le trame di livello 2 dall'altra. E così anche gli access point di un Extended Service Set

Dal punto di vista del protocollo IP non è cambiato nulla , la stazione è sempre parte della stessa rete fisica.

La mobilità non è trasparente quando la stazione cambia rete fisica, in quanto il prefisso dell'indirizzo IP dipende dalla posizione nella rete e questo perché la logical IP subnet (LIS), cioè tutte le stazioni con lo stesso prefisso corrisponde ad una rete fisica.

Quando si cambia rete fisica si ha uno spostamento

Il prefisso dell'indirizzo IP dipende dalla "posizione"

→ La logical IP subnet (LIS) corrisponde ad una rete fisica

→ Se una stazione cambia rete fisica

→ Deve cambiare LIS (prefisso)

→ Cambiamento di indirizzo

Dare un nuovo indirizzo alla stazione è un problema, perché tutte le connessioni TCP e sessioni UDP attive vengono interrotte e quindi ogni servizio in corso si interrompe, ad esempio trasferire un file, controllare la posta elettronica, fare una telefonata ecc.

→ Gli identificativi di sessione/connessione includono l'indirizzo IP

→ La quintupla "magica" (indirizzo sorgente e destinazione, protocollo di trasporto, porta sorgente e porta destinazione) identifica univocamente i pacchetti di un certo flusso, cioè di una certa connessione TCP o di una certa sessione UDP

→ Il cambio di indirizzo IP crea problemi anche sui meccanismi di autorizzazione basati sull'indirizzo che quindi rifiuteranno la stazione

- **Mobile IP**

Una soluzione, fra altre, che risolve la sfida di poter cambiare rete fisica senza dover, in qualche modo, cambiare indirizzo e scambiare dati. Non usatissima.

Caratteristiche

→ RFC 3344 (2002)

→ Trasparente a livello trasporto e applicazioni

→ Interoperabilità con stazioni che non hanno supporto per mobile IP

→ Scalability

→ Sicurezza

→ Autenticazione per evitare impersonificazione di stazione mobile [spoofing]

→ Mobilità limitata

→ Al più un "movimento" al secondo

Utilizzo degli indirizzi

→ La stazione mobile ha un suo indirizzo permanente

→ Corrisponde alla propria posizione principale, quando cioè si trova nella home network

→ Home address, l'indirizzo permanente

→ Quando la stazione si muove in una foreign network, ovvero una rete diversa dalla rete home

→ Acquisisce indirizzo locale di nome care-of address, che è traducibile come "presso, c/o"

→ Care-of address, indirizzo locale in foreign network

Inoltro dei pacchetti

Home address è usato per l'invio e ricezione di pacchetti (come indirizzo sorgente e indirizzo destinazione), quindi nell'esempio si può notare che quando una stazione manda un pacchetto da H1 a Z3, essa costruisce un pacchetto che va da H1 a Z3.

La stazione dovrà ricevere le risposte, per cui Z3 risponderà ad H1, ma in realtà i pacchetti per H1 che sono mandati all'home address, verranno consegnati al care-of address, ovvero i pacchetti per la stazione mobile sono mandati all'home address ma consegnati al care-of address, come figura a lato

Chi sta all'estremo del tunnel? Cioè chi mette il pacchetto mandato dal server dentro un altro pacchetto all'indirizzo F3 per consegnarlo alla stazione mentre essa è mobile, ovvero è sulla foreign network? Esso sarà il cosiddetto home agent, che garantisce che i pacchetti siano alla fine consegnati alla stazione.

Per il care-of address abbiamo un paio di opzioni, di cui una, appena vista è la co-located care-of address, poi c'è il cosiddetto foreign agent care-of address

Co-located care-of address

→ Ottenuto dalla stazione sia in modo permanente che dinamico (per esempio tramite DHCP)

→ Più indirizzi IP necessari, uno per ogni stazione mobile nella foreign network e questo è limitativo

→ Host termina il tunnel, cioè l'host deve estrarre il pacchetto interno, il suo indirizzo home, dal pacchetto esterno inviato al care-of address e questo vuol dire che l'host ha un carico elaborativo maggiore

→ Non serve un foreign agent

Foreign Agent

Foreign Agent Care-of Address

→ Il care-of addrees è un indirizzo del foreign agent

→ Indirizzo può essere condiviso

→ Niente carico elaborativo nella stazione, in quanto il tunnel è terminato nel foreign agent

A prescindere dall'uso di una soluzione co-located o foreign agent care-of address, il foreign address deve essere registrato.

Registrazione

La stazione che opera su una foreign network notifica il proprio home agent del care-of address che sta usando, perché l'home agent deve inoltrare il pacchetto. La registrazione serve per comunicare il care-of address

→ Comunica il care-of address

La registrazione può essere fatta dal foreign agent, se il care-of agent è configurato su un foreign agent.

La stazione deve in qualche modo scoprire che c'è un foreign agent, far sapere al foreign agent che ha bisogno di supporto per la mobilità, comunicare il proprio home address e, a quel punto, il foreign agent può notificare l'home agent.

Messaggi di registrazione

Il protocollo Mobile IP definisce, oltre all'architettura, anche protocolli che servono per fare questi scambi di informazione, queste registrazioni. Ci saranno anche funzionalità di registrazione al fine di evitare che un host maligno faccia finta di essere parte della home network per accedervi.

→ Messaggi del mobile IP protocol

Funzionalità di autenticazione

→ Per evitare che un host maligno faccia finta di essere parte della home network per accedervi ed acquisire un indirizzo di una rete

E' dunque fondamentale che nella fase di registrazione ci siano meccanismi di autenticazione.

E' anche importante avere dei meccanismi per gli annunci degli agenti, cioè gli agenti Mobile IP devono rendersi noti alle stazioni.

Annuncio dell'agente

→ Gli agenti Mobile IP devono rendersi noti alle stazioni, questo si fa con una estensione del messaggio ICMP router advertisement

→ Estensione del messaggio ICMP router advertisement

→ Grazie a questi messaggi, la stazione mobile può capire "dov'è", se si trova nella home network o nella foreign network; la stazione mobile può rimanere in ascolto e vedere se c'è un home agent o un foreign agent che si annuncia e quindi capire che c'è supporto per la mobilità e capire se si trova nella home network o nella foreign network. Se non riceve degli annunci entro un certo tempo, la stazione li può anche sollecitare

→ Una stazione mobile può sollecitare l'annuncio del mobile agent, usando il messaggio ICMP router solicitation

La soluzione Mobile IP è stata progettata per supportare la mobilità, inizialmente quando si è cominciato a lavorare su IPv6, ma poi si è estesa anche a IPv4.

Essa ha però delle limitazioni, quindi sono sia state progettate soluzioni diverse, sia, a supporto della mobilità, soluzioni progettate per altri ambiti, di cui una è la cosiddetta Proxy Mobile IPv6

- **Proxy Mobile Ipv6**

Nasce per supportare la mobilità in IPv6, ma in realtà può essere usata anche in situazioni in cui IPv6 non è utilizzato.

Caratteristiche

→ Non richiede supporto negli host, cioè proxy mobile IP richiede che gli host siano in grado di capire i protocolli del mobile IP, che gli host siano a conoscenza della presenza di home agent, un foreign agent. In proxy mobile IPv6 l'host non deve avere nessuna modifica, ma ci sarà un elemento della rete che segue i movimenti degli host

→ Un elemento di rete segue i movimenti degli host

→ Elemento basato su protocolli standard e comunemente utilizzati

→ Un elemento di rete si occupa delle azioni legate alla mobilità

→ Segnalazione (dell'home agent o del foreign agent)

→ Tunneling

→ Protocolli specifici del proxy mobile IPv6 usati dall'elemento di rete

Architettura

la stazione ha il suo indirizzo e lo mantiene ovunque vada; sarà il MAG che si accorge che nella rete c'è una stazione in mobilità e a fare le cose che servono; il MAG (o il foreign agent nel caso precedente) e la stazione devono essere sulla stessa rete fisica; la stazione che si muove non sa nulla del supporto alla mobilità, che è supportata completamente dai dispositivi di rete. In questo tipo di architettura ci dobbiamo spostare dove c'è un MAG, nell'altro ci possiamo spostare nella rete anche in assenza di foreign agent.

In sostanza questa architettura ha il MAG, che deve avere funzionalità diverse da un foreign agent.

Architettura non necessariamente IPv6

→ Mobile node: IPv4 o IPv6 (o dual stack)

→ La rete tra il MAG (Mobile Access Gateway) e il LMA (Local Mobility Anchor, equivalente dell'home agent): IPv4 o IPv6;

→ Segnalazione basata su IPv6, ma potrebbe essere IPv4

→ LMA è un Home Agent Mobile IPv6

- **Locator/Identifier Separation Protocol (LISP)**

E' una ulteriore soluzione per gestire la mobilità.

Gli indirizzi IP hanno due funzioni

→ Identificare le stazioni

→ Localizzare le stazioni

→ Cioè assistono i router nel trovare un percorso verso gli host

LISP le separa, mentre nell'indirizzo IP tradizionale le due funzioni sono fuse.

Quindi identifichiamo da un lato un identifier a e dall'altro un route locator.

Identifier e Route Locator

→ Possono essere entrambe un indirizzo IP

→ Oppure possono essere qualcos'altro

→ Per esempio coordinate GPS, indirizzo MAC

Quello di cui abbiamo bisogno è quello di avere un valore univoco su tutta la rete che

permette di identificare la stazione e poi di un qualche altro valore che in qualche modo aiuta a trovare la stazione nella rete.
LISP non ha nessun interesse a sapere cosa siano questi due valori, ma fornirà tutti gli strumenti per fare la corrispondenza, ovvero dato un identificatore trovare il localizzatore e quindi poter raggiungere la stazione.
Le stazioni sono sempre identificate con l'identificatore e, se si spostano, avranno bisogno di un localizzatore diverso.

I campi di applicazione del LISP
→ Mobilità (LISP non era stato pensato per questo)
→ Scalability del routing
→ Attraversamento di zone IPv4 a pacchetti IPv6 e viceversa
→ Network virtualization, cloud computing; si identificano reti virtuali in un datacenter. Locazione identica per identità diverse
→ Multihoming, quando un organizzazione è cliente di due service provider
LISP è applicabile in quei casi in cui abbiamo un identificatore che non dice ai router come inoltrare i pacchetti. I campi di applicazioni sono molti.

Principi di funzionamento del LISP
→ Mapping system: identifier ↔ locator, LISP fornisce una corrispondenza tra identifier e locator
    → Dapprima basato su BGP
    → Poi ispirato al DNS
    → Uno qualsiasi va bene
→ LISP è usato dai router, dagli apparati di rete
    → Quindi gli host non sono consci

LISP e mobilità
→ Gli host mantengono l'identificatore muovendosi
→ Un nuovo locator è acquisito muovendosi
→ LISP è usato per trovare la corrispondenza tra l'identificatore, cioè l'home address ed il care-of address e anche per assicurare la consegna

- **Host Identity Protocol (HIP)**

Ulteriore soluzione per la mobilità

Cosa fa?

→ Disaccoppia identità e posizione

→ Crea uno spazio di nomi per l'identità degli host

→ Basato su crittografia asimmetrica (a chiave pubblica)

Come fa?

→ Le applicazioni usano identificatori crittografici detti host identity tag

→ Un Host Identity Tag è generato a partire da una chiave pubblica

→ I protocolli assicurano che il comunicante possieda la corrispondente chiave privata, cioè chi dice di avere un certo tag possiede la chiave privata

Perché è stato fatto HIP?

→ Creato per risolvere problemi di sicurezza, avere comunicazioni sicure in cui si può sempre verificare l'identità del comunicante, ma poiché l'identificatore non dipende dalla posizione, si può anche usare per gestire la mobilità

→ Usato per mobilità perché l'identificatore non dipende dalla posizione

→ Cambiare indirizzo non coinvolge le applicazioni, poiché le applicazioni usano questo identificatore per identificare mittente e sorgente, cambiare indirizzo non coinvolge in alcun modo le applicazioni

DOMANDE POSTE NELLE LEZIONI di RETI di CALCOLATORI:

Lezione 12

• Qual è la relazione tra l'architettura di protocolli OSI e quella TCP/IP?
• Che tipo di servizio viene offerto dal protocollo IP?
• Qual è lo scopo del campo TTL (time to live)?
• Perché è importante che i router IP siano in grado di frammentare i pacchetti e perché questo richiede supporto specifico del protocollo IP?

Lezione 13

• Cosa è una logical IP subnet (LIS)
• A cosa servono le classi di indirizzo
• Cosa è e perché è stata introdotta la netmask

Lezione 14

• Perché è importante avere una corrispondenza tra reti fisiche e LIS
• Come si usa la netmask per capire se due indirizzi hanno lo stesso prefisso
• Cosa è contenuto nella tabella di routing
• Come i router usano la tabella di routing

Lezione 16

• Come vengono identificati i pacchetti di un flusso?
• Qual è il valore che UDP aggiunge ad IP?
• Che tipo di servizio viene offerto da TCP?

Lezione 18

• A cosa serve il DNS?
• Quali sono gli scopi di avere una struttura di nomi di dominio gerarchica?
• Perché si utilizza la modalità di risoluzione dei nomi ricorsiva?
• Perché i messaggi DNS contengono una durata di validità delle informazioni portate?
• Perché c'è necessità di un DNS sicuro?

Lezione 19

• Qual è la differenza tra il paradigma client-server e quello peer-to-peer?
• Com'è organizzato il sistema di posta elettronica?
• Qual è lo scopo del protocollo SMTP?
• Qual è lo scopo del MIME?
• Qual è la differenza tra i vari protocolli per l'accesso alla posta elettronica?

Lezione 24

• Cosa è un protocollo di routing?
• Perché Internet è organizzata in Autononous System?
• Perché è conveniente avere Neutral Access Point?

Lezione 25

• Quali sono le differenze tra IGP e EGP?
• Quali sono i principali protocolli IGP?
• Qual è il protocollo per routing interdominio più usato?
• Cosa è una content delivery network?
• Come avviene il recapito di un pacchetto IP da un mittente ad un gruppo di destinatari?

Lezione 26

• Quali sono le sfide nella sicurezza informatica?
• Qual è la differenza tra crittografia a chiave simmetrica e a chiave pubblica?
• Come si usa la cifratura a chiave pubblica per autenticare il mittente di un messaggio e il contenuto del messaggio
• Perché servono certificati digitali

Lezione 27

• Su cosa sono comunemente basati attacchi alla sicurezza?
• In che modo Ipsec imbusta le informazioni per criptarle ed autenticarle?
• Cosa è alla base della sicurezza del web?
• Che funzionalità usare per proteggere una rete aziendale da attacchi informatici

portati tramite la rete?

Lezione 29

• Perché l'intestazione IPv6 è divisa in header ed extension header?
• Qual è il vantaggio del Neighbor Discovery rispetto ad ARP?
• Quali sono gli elementi chiave della transizione graduale da IPv4 a IPv6

Lezione 30

• Qual è il problema fondamentale della mobilità in una rete IP?
• Cosa è un care-of-address nella soluzione mobile IP?
• Cosa assicura la consegna dei pacchetti e il mantenimento delle comunicazioni durante la mobilità nella soluzione mobile IP?
• Cosa caratterizza la soluzione PMIPv6?
• Qual è il concetto alla base di LISP?
• In che contesto è stata progettata la soluzione Host Mobility Protocol?

.