Introduzione

Che cos'è una rete di calcolatori

Def: una rete di calcolatori è un insieme di dispositivi di calcolo **autonomi** e **interconnessi**.

- Esempi

  Una rete telefonica NON è una rete di calcolatori:

  - I dispositivi sono interconnessi;
  - Ma **non** sono autonomi.

  Tanti computer nella stessa stanza NON fanno una rete di calcolatori:

  - Sono dispositivi autonomi;
  - Ma **non** sono interconnessi.

Perché creare una rete di calcolatori?

- Le reti forniscono servizi utili agli umani, ad esempio:
  - E-mail;
  - WWW;
  - Internet of things;
  - ecc.
- Permettono ai calcolatori di comunicare fra loro:
  - Possono condividere risorse e informazioni (ad esempio posso condividere una stampante con tutti i calcolatori di un ufficio, senza bisogno di comprare tante stampanti);
  - Posso effettuare operazioni di calcolo distribuito (per quando la potenza di un singolo calcolatore non basta, ad esempio per elaborare la foto di un buco nero);
  - Possono condividere supporti di memoria (il cloud).

Classificazione delle reti

Le reti vengono classificate in base alla dimensione geografica:

PAN « LAN « MAN « WAN « Internet

- Personal Area Network (PAN)

  Connessione fra dispositivi in una stanza, ad esempio un computer ed una stampante. Queste reti sono gestite da chi le utilizza.

- Local Area Network (LAN)

  Connessione fra dispositivi in un edificio, ad esempio il wifi di casa. Queste reti sono usate anche per campus universitari o per le reti aziendali, sono

mantenute dall'ente che le usa.

- Metropolitan Area Network (MAN)

  Reti urbane, ad esempio la rete almawifi a Bologna. Coprono aree del raggio di decine di chilometri, sono mantenute da provider di servizi di comunicazione e/o gestori telefonici.

- Wide Area Network (WAN)

  Reti nazionali e internazionali. Possono coprire distanza internazioni o planetarie, sono mantenute da vari enti multinazionali e gestori delle comunicazione. Sono reti complesse, formate da diverse tecnologie integrate fra loro (ad esempio fibra e satellite).

- Internet

  È la rete di reti, la rete globale. È formata dall'unione delle altre reti (ad esempio le WAN) che obbediscono tutte a protocolli comuni (ad esempio il protocollo email, oppure http, ecc).

Evoluzione e costi

The year: **1969**. The location: **Los Angeles**.

Viene creata la prima rete di Internet da un esperimento universitario, vengono connessi 4 calcolatori di 4 università diverse.

Una trentina d'anni dopo, nel 2003, ad Internet erano connessi oltre 172 **milioni** di calcolatori.

Oggi (2021) ci sono oltre 30 **miliardi** di dispositivi connessi (tanto che sono finiti gli indirizzi IP, che sono 4 miliardi, e si è dovuto trovare un escamotage).

Il motivo di questa crescita esponenziale è che ormai i **calcolatori necessitano** di essere **interconnessi** tra loro più di quanto ne necessitano gli umani. La maggior parte di Internet è formata da una serie di **attività** e **servizi rivolti** alle **macchine** stesse e che hanno come effetto **indiretto** quello di fornire servizi agli **umani**.

Costi

Inizialmente i costi erano coperti dai militari, in quanto solo loro avevano bisogno della rete. In seguito man mano che è cresciuta ci si è resi conto che servivano infrastrutture ad hoc (la rete telefonica non bastava più), quindi i costi sono stati coperti da consorzi internazionali, governi per la gestione su larga scala e una rete capillare di piccoli gestori locali per la piccola scala.

Gli utenti affittano l'accesso all'infrastruttura dai provider tramite contratti. All'inizio le tariffe erano a tempo, con l'aumentare della velocità delle connessioni

si è passati ad un modello tariffario a consumo. Oggi le tariffe più diffuse sono quelle di tipo “tutto incluso”.

Prestazioni

Le prestazioni sono misurate in due modi:

- Capacità di trasmissione Misura il numero di bit o byte trasmessi in un secondo. Si parla di Kbyte, Mbyte, Gbyte, ecc.
- Ritardo del collegamento È il tempo impiegato dai dati per andare da mittente a destinatario. Dipende dalla distanza fisica e altri fattori, come le regole dei protocolli di comunicazione.

Componenti

Una rete di calcolatori necessita, oltre che dei calcolatori di base, di certi componenti hardware e software.

Parte hardware:

- Dispositivo o scheda di rete Permettono di codificare, trasmettere, ricevere e decodificare i dati fra calcolatore e rete
- Mezzo di trasmissione È il supporto che permette la propagazione dei dati: cavi, fibre, fili elettrici, lo spazio nel quale si propagano le onde radio). Sono ciò che compone l’infrastruttura fisica della rete.
- Connettore di rete È l’interfaccia tramite cui si connette il dispositivo alla rete, ad esempio un antenna per il wifi o un cavo ethernet.

Parte software:

- Parti del Sistema Operativo Le schede di rete sono gestite direttamente dal SO tramite driver e devono rispettare una serie di protocolli per la gestione della comunicazione.
- Protocolli di rete Un insieme di regole per creare uno standard comune e garantire che tutti i calcolatori che si affacciano sulla rete possano comunicare fra loro.

Collegamenti e infrastrutture di rete

Un collegamento di rete è reso possibile da un mezzo di trasmissione (ad es. un cavo) che sia condiviso da almeno due calcolatori.

Un’infrasruttura è l’insieme dei collegamenti fra tutti i calcolatori connessi ad una rete. Due calcolatori qualsiasi possono comunicare fra loro se esiste un

Figure 1: Scheda di rete

cammino diretto o indiretto fra essi all'interno della rete. Vi sono diverse classi di strutture di connessione alla rete:

- **Punto a punto**: connessione diretta fra due soli dispositivi È il caso più semplice di infrastruttura ed è facile gestirlo.

Figure 2: Connessione punto a punto

- **Connessione multipla completamente connessa**: connessione diretta fra più di due dispositivi È una struttura molto ridondante, ogni dispositivo è connesso direttamente a tutti gli altri. Con l'aumentare del numero di dispositivi aumenta il casino da gestire.

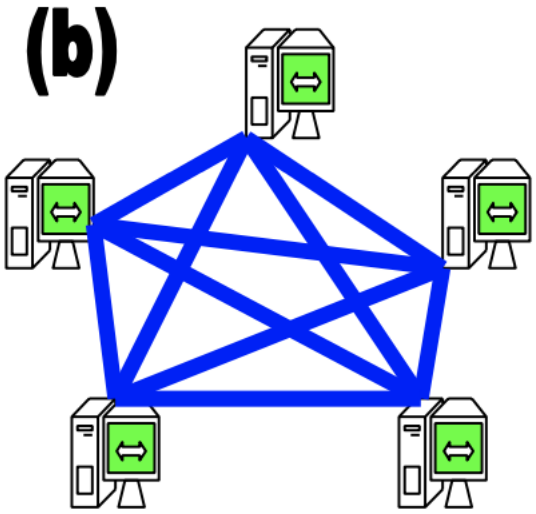


Figure 3: Connessione multipla completamente connessa

- **Connessione multipla parzialmente connessa** Permette comunque la connessione fra tutti i dispositivi ma senza che ci sia un cammino diretto

fra ogni calcolare. Il rischio è che in caso di un guasto alla rete si possa creare una partizione di reti (un insieme di componenti separato dal resto della rete).

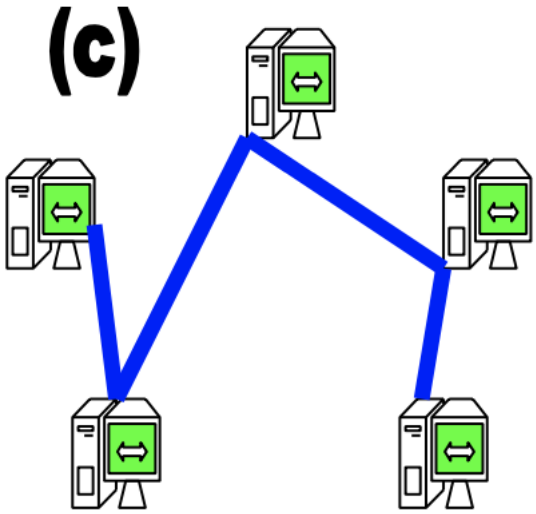


Figure 4: Connessione multipla parzialmente connessa

- **Connessione multipla con partizione di rete** Due o più gruppi di reti sono isolati fra loro. Di solito la configurazione è dovuta a guasti sulla rete o da cattive applicazioni dei protocolli.

Topologia di rete

Le infrastrutture di rete possono essere realizzate secondo vari schemi:

1. **Anello**: ogni elemento è connesso solo col precedente e col successivo
2. **Stella**: ogni elemento è connesso con un unico elemento centrale
3. **Bus**: ogni elemento è connesso al bus centrale condiviso
4. **Albero**: simile ad un albero genealogico

Come regola generale le reti più piccole adottano una delle topologie citate, mentre quelle più grandi ibridano più topologie per aumentare la ridondanza e rendere la rete meno esposta al rischio di partizionamento, in questo caso si parla di **topologia a maglia**.

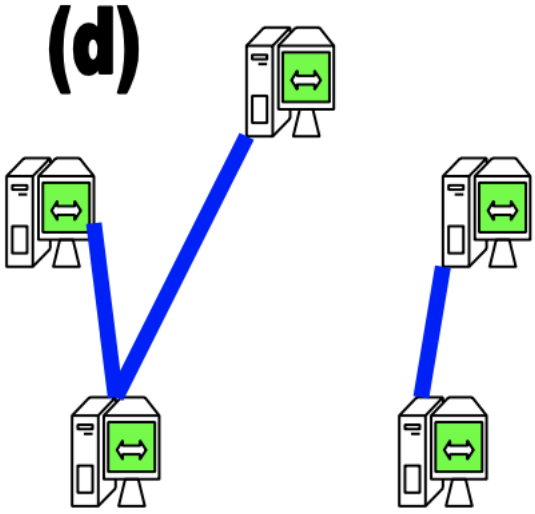


Figure 5: Connessione multipla con partizione di rete

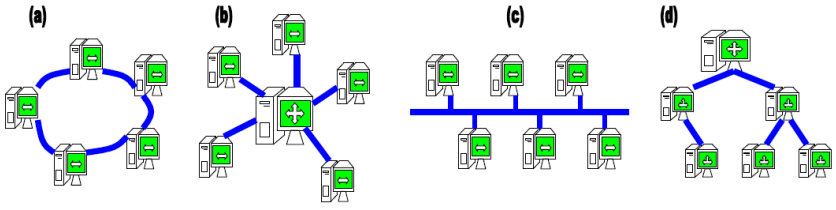


Figure 6: Tipologie di reti

Il mezzo fisico di trasmissione

È il mezzo fisico con cui si realizza la comunicazione, che supporta la propagazione del segnale fra due diverse schede di rete. Può essere essenzialmente di 3 tipi:

- Cavo metallico (di solito rame)

  L'informazione viene trasmessa come segnale elettrico (ovvero come variazione di corrente all'interno del mezzo trasmissivo). È molto usato grazie alla sua affidabilità e al suo buon rapposto costo/prestazioni.

- Ethernet

  Più prestante del singolo cavo di rame, è formato da 4 fili di rame attorcigliati fra loro in un coating isolante, in questo modo permette la trasmissione di più informazioni a parità di tempo.

- Fibra ottica

  Fibre di vetro attraverso cui viaggia la luce, offre prestazioni altissime e necessita di meno amplificatori di segnale lungo la strada ma ha un costo molto elevato.

- Onda radio (wireless)

  Il mezzo di trasmissione in questo caso non esiste (è "l'etere"). Possono essere usate onde radio, o raggi infrarossi. Permettono grande flessibilità nello spostamento dei calcolatori, ma non sono molto affidabili e raggiungono velocità ridotte.

Scheda di rete

La scheda di rete si occupa di interfacciare un calcolatore alla rete codificando, trasmettendo, ricevendo e decodificando le informazioni. Essa dipende dal mezzo di trasmissione usato ed è dotata di un proprio indirizzo MAC (Medium Access Control) univoco che identifica la scheda di rete nel mondo.

Canale di comunicazione di rete

Un canale di comunicazione è una visione astratta del mezzo di trasmissione, ovvero è come scelgo di usare quel mezzo per trasmettere informazione. Ci sono fondalmentalmente due categorie di canali di comunicazione: canale punto a punto e canale broadcast.

Il canale **punto a punto** è un canale riservato fra due dispositivi in comunicazione; il vantaggio è che non c'è bisogno di includere mittente e destinatario, in quando la comunicazione è fra due soli calcolatori.

Steso un canale trasmissivo, possiamo fare più comunicazioni punto a punto in parallelo: un gruppo comunica sul canale A, uno sul B, ecc.

Gli svantaggi sono che bisogna riconfigurare la rete nel momento in cui si voglia parlare con qualcun altro e che il canale è riservato: se sta venendo usato non si può fare altro (ad es. quando ci si connetteva ad internet e non si poteva usare il telefono).

Un canale **broadcast** è un canale dove tutti i dispositivi trasmettono e ricevono (se mando un messaggio tutti lo ricevono). In questa configurazione è richiesto l'utilizzo di protocolli per evitare collisioni e perdita di messaggi. I vantaggi sono che con un solo canale possiamo mettere in comunicazione più calcolatori e che un dispositivo può comunicare con altri dispositivi della rete senza bisogno di riconfigurare quest'ultima.

Reti a commutazione di circuito

L'idea è quella di comporre a cascata una serie di canali di comunicazione punto a punto in modo da poter mettere in comunicazione i due calcolatori agli estremi della rete.

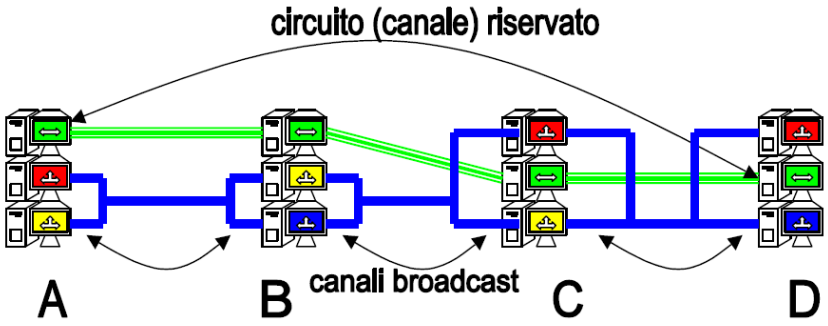


Figure 7: Rete a commutazione di circuito

Un esempio classico è la linea telefonica.

Nelle reti a commutazione di circuito il jitter è molto basso. Solitamente si paga per il tempo di utilizzo. Lo svantaggio è che viene riservato un canale intero per una comunicazione, quindi ogni momento in cui non si comunica è uno spreco di risorse del sistema.

Reti a commutazione di pacchetto

A differenza delle a commutazione di circuito, questa è una metodologia molto usata nelle reti a canali broadcast. Dal momento che internet usa reti broadcast, tutta la comunicazione di internet avviene a pacchetto.

L'idea è di scomporre la comunicazione in più pacchetti sequenziali e ognuno diventa una trasmissione sulla rete. Ogni pacchetto viene spedito indipendente-

mente sul canale broadcast e dovrà includere l'indirizzo del destinatario e quello del mittente, oltre che la comunicazione vera e propria.

Pro:

- È richiesto un minor numero di risorse di rete;
- Questo minor numero di canali richiesto è anche usato meglio.

Contro:

- Ogni singolo pacchetto subirà una latenza di rete maggiore;
- Singoli pacchetti possono andare perduti o arrivare in ordine sparso.

Il vantaggio maggiore (per il marketing) è che si può far pagare l'utente per la quantità di dati consumati invece che a tempo.

Ulteriore vantaggio: nel momento in cui voglia comunicare con un altro dispositivo potrei farlo subito semplicemente indicando il suo indirizzo come destinatario, a differenza di quanto dovremmo fare con i canali punto a punto.

Servizi orientati alla connessione e non

Dal momento che i singoli pacchetti possono percorrere strade diverse per arrivare al destinatario, non è detto che arrivino nello stesso ordine di invio o che qualche pacchetto non si perda per strada. Per questo entrano in gioco i servizi di trasmissione, che determinano le proprietà di mittente e destinatario in base ai protocolli usati.

I servizi di trasmissione possono essere orientati alla connessione (connection-oriented) oppure no (connectionless).

Servizi orientati

Questi servizi garantiscono che la trasmissione sia affidabile e corretta: si assicurano che i pacchetti arrivino nel giusto ordine e, tramite un sistema di feedback dal dispositivo destinatario, rispediscono i pacchetti che non arrivano entro un certo intervallo di tempo.

Tali servizi sono implementati attraverso opportuni protocolli alternativi: ad esempio i pacchetti potrebbero essere numerati, oppure spediti attraverso un circuito virtuale riservato, o altre soluzioni ancora.

Servizi non orientati

Non si preoccupano di garantire l'ordine corretto dei pacchetti e nemmeno la ricezione di tutti i pacchetti; si occupano solo di spedire i pacchetti che ricevono, sono molto simili alla spedizione di una sequenza di lettere tramite poste italiane.

Reti locali

Protocolli di rete organizzati a livelli

Un protocollo è un insieme di regole di comportamento e procedure di gestione dei processi di comunicazione (semantica) e di regole di tipo sintattico, ovvero come devono essere strutturati i messaggi da scambiare.

Quindi in sostanza i protocolli stabiliscono cosa fare quando e cosa fare come, se si vuole comunicare nella rete.

Grazie ad essi è possibile la compatibilità di comunicazione fra i dispositivi che sono connessi alla rete, che sono eterogenei fra loro.

I protocolli solitamente hanno nome dello standard e nome commerciale:

- **802.11** è il nome standard, **Wifi** è il nome commerciale;
- **802.3** è il nome standard, **Ethernet** è il nome commerciale.

Architettura di protocolli

Una Architettura dei protocolli di rete è un insieme di protocolli che funziona in maniera coordinata e sincrona, dal momento che un solo protocollo non basta per gestire una rete complessa come Internet.

I protocolli sono strutturati a livelli. La regola generale dice che, dato un livello x, esistono protocolli alternativi che possiamo interscambiare. I livelli superiori e inferiori possono interagire col livello x solo con le opportune API.

Dal momento che i protocolli possono cambiare e aggiornarsi (magari per sistemare falle, magari per introdurre migliorie) è necessario che, qualunque sia il loro funzionamento interno, si comportino sempre allo stesso modo quando ricevono input dai protocolli del livello sottostante e quando mandano output ai livelli sovrastanti.

Ed è qui che interviene l'architettura:

> Un'architettura di protocolli definisce il template sul quale vanno realizzati i protocolli di rete.

Alla fine è sempre solito principio di astrazione/implementazione alla base di tutta l'informatica.

Esempio di architettura di protocolli

Vediamo un esempio di architettura di protocolli tramite il classico esempio dell'innamorato Italiano che vuole parlare alla sua innamorata Giapponese.

Figure 8: Innamorato Italiano e innamorata Giapponese

Architettura standard di protocolli di rete

Lo standard di tutte le comunicazioni di rete è formato da 7 livelli ed è chiamato **Standard ISO/OSI RM** (Open System Interconnection Reference Model).

È un insieme di livelli *completo* (non esiste un livello mancante) e *rigoroso* (formalmente definito in quel modo e o lo rispetti o non lo rispetti, non ci sono vie di mezzo).

Ogni livello gestisce una classe di problematiche di rete mentre riceve una versione semplificata del resto della rete per cui tutte le problematiche degli altri livelli sono già state risolte.

Man mano che si sale nella rete si nota che dal punto di vista del livello più alto la rete è una scatola magica dove le cose semplicemente funzionano.

I 7 livelli sono (dal più alto al più basso):

- Livello Applicazione
- Livello Presentazione
- Livello Sessione
- Livello Trasporto
- Livello Rete
- Livello MAC/LLC
- Livello Fisico

Ecco una rappresentazione schematica dei livelli dell'architettura:

Figure 9: Rappresentazione schematica dei livelli

Architettura dei protocolli di Internet

L'architettura dei livelli dei protocolli di Internet utilizza solo 5 dei 7 livelli menzionati precedentemente.

Ogni volta che un dato viene trasmesso, a partire dall'origine la trasmissione viene **incapsulata** in "buste" con dati aggiuntivi.

Al momento della ricezione poi ogni livello prende la busta, ne estrae il contenuto e lo passa al livello sovrastante (quindi è tipo una matrioska).

Nello specifico su internet i livelli usati sono:

- Livello applicazione: genera il dato da spedire;
- Livello Trasporto: imbusta i dati inserendo info sul modo di riordinare i pacchetti e controlla la velocità di invio delle buste;
- Livello Rete: spezza i dati in pacchetti e sceglie il cammino sul quale inviarli;
- Livello MAC/LLC: esegue la consegna finale dei dati alla rete locale;
- Livello Fisico: dove transitano i dati.

  I protocolli fanno il loro meglio ("best at all") ecco perché i provider nel proporre gli abbonamenti forniscono la velocità massima raggiungibile, ma nessuna garanzia sulla velocità minima, perché non è tecnicamente possibile garantire che la trasmissione vada a buon fine.

Livelli e integrazione delle reti

Vediamo nel dettaglio i vari livelli di rete.

Livello fisico

Il livello più basso è quello che si occupa fisicamente della trasmissione. I protocolli implementati a questo livello sono molto semplici e si riassumono in:

- Codifica e spedisce il pacchetto appena ricevuto dal livello MAC/LLC;
- Decodifica e manda al livello MAC/LLC il pacchetto appena ricevuto.

Questo livello trasforma i segnali digitali in analogici.

La velocità di una connessione non dipende dalla velocità con cui transitano i dati, siccome questa è sempre la stessa ed è la velocità della luce, ma dalla quantità di dati che posso mandare in un certo intervallo di tempo.

Più dati mando e più brevi saranno i segnali, rendendo quindi più difficile il compito di captarli e decodificarli, per questo per aumentare la "velocità" di una rete sono necessarie tecnologie più sofisticate, che siano in grado di captare e decodificare senza sbagliare un certo numero di bit per secondo.

Livello MAC/LLC

Il livello più semplice di una rete a commutazione di pacchetto è il **segmento di rete locale**.

Esso è formato da un mezzo di trasmissione condiviso con un canale ad accesso multiplo (ricordi i canali?). Tutte le schede video connesse al segmento ricevono le trasmissioni effettuate e trasmettono a tutte le altre schede video.

Per riconoscere il destinatario di una trasmissione viene dato ad ogni scheda video un **indirizzo MAC**, che è unico ed è assegnato in fase di produzione, è come un'impronta digitale della scheda di rete.

Il Livello MAC/LLC si occupa di fornire l'accesso al mezzo trasmissivo (Livello Fisico), quindi si occupa di trasmettere i dati, di specificare a chi è destinata la trasmissione e di ricevere i dati destinati alla propria scheda video.

Tipicamente un Livello MAC/LLC si comporta così per trasmettere dati:

- Attendi che sia il tuo turno di comunicare;
  - Nel caso di una rete di topologia bus o star significa vedere se la rete è libera e, nel caso sia occupata, aspettare un tempo t casuale prima di tentare di nuovo l'invio;
  - Nel caso di una rete token ring significa aspettare di avere il token;
- Spedisci il messaggio;
- Aspetta l'acknowledgement, se non arriva prova a rispedire il messaggio.

E viceversa quando li riceve:

- Controlla che il dato in transito abbia il mio indirizzo MAC;
- Se sì ricevi i dati e mandali ai livelli superiori;
- Manda indietro l'ack per i pacchetti arrivati.

Quando ricevo un pacchetto mando immediatamente l'acknowledgement.

Il protocollo infatti prevede che il mezzo trasmissivo sia da considerarsi occupato anche immediatamente dopo l'invio del pacchetto, in questo modo l'ack è protetto da collisioni.

Come rendo il segmento affidabile?

Rendere il canale affidabile significa eliminare (o ridurre al minimo) gli errori di trasmissione dovuti a collisioni o interferenze.

Come abbiamo visto il protocollo MAC/LLC prevede l'invio di un acknoledgement per confermare l'avvenuta ricezione del frame; il mittente quindi dopo aver spedito i dati resta in attesa di questo ack e, se dopo un certo tempo di timeout non ha ricevuto nulla, rispedisce il frame incriminato.

Questo funzionamento è solido poiché:

- Non potrebbe funzionare il contrario: il destinatario non può mandare ack di pacchetti che non ha ricevuto perché non sa se deve effettivamente ricevere un pacchetto o no.
- Se anche l'ack si perde e non torna al mittente, worst case scenario il mittente rispedisce il frame e il destinatario lo scarta; si usa un po' di più la rete ma si mitiga il rischio di mancata consegna di frame.

Ovviamente questo comportamento si presta a diversi tipi di attacchi da parte di malintenzionati, ad esempio non mandare volutamente gli ack e costringere quindi il mittente a riempire la rete di frame e occuparla per tutti gli utenti connessi.

> **Denial of acknoledgement**: attacco hacker che consiste nel non mandare l'ack e porta il mittente a intasare la rete mandando infiniti pacchetti.

A livelli più alti questo protocollo non può garantire l'affidabilità, può solo farlo fra i nodi di un segmento locale. Se ad esempio lungo il cammino dovesse esserci un ostacolo alla trasmissione sarà il livello 3 a reindirizzare i dati seguendo un'altra strada.

Tecnologie per schede di rete: 3 esempi

Come abbiamo detto l'architettura dei protocolli di rete è realizzata per permettere di usare un protocollo piuttosto che un altro senza che questo cambiamento vada a rompere il funzionamento dei livelli sotto e sovra stanti.

A titolo di esempio vediamo 3 protocolli diversi per il livello 2 (MAC/LLC) che possono essere usati su diverse schede di rete che si affacciano sullo stesso segmento locale:

- **Ethernet** (o IEEE 802.3):

È lo standard delle reti cablate; la scheda di rete può ascoltare il canale e trasmettere se questo è libero, altrimenti sceglie un tempo random dopo il quale riprovare. L'ethernet è in grado di rilevare le collisioni e nel caso ritentare in seguito la trasmissione.

- **Wifi** (o IEEE 802.11):

  È lo standard delle reti senza fili. Anche in questo caso la scheda ascolta il canale e trasmette se lo trova libero. Non è però in grado di capire se ci sono state collisioni, quindi per ridurre il rischio che esse avvengano si dilazionano le trasmissioni nel tempo.

- **Token ring**:

  In realtà è ormai usata pochissimo, richiede che la rete sia a topologia ad anello. Ogni scheda di rete attende di ricevere il token e, una volta ricevutolo, può trasmettere entro un certo tempo t prestabilito. È una rete che si presta molto ad attacchi (ad esempio distruggo il token, oppure genero token in più di tanto in tanto, ecc), quindi è usata solo in contesti chiusi in cui si sa di potersi fidare del buon rispetto delle regole dei calcolatori connessi; in compenso è molto efficiente e garantisce alte prestazioni.

Comporre segmenti in reti locali

Il passaggio successivo è unire i vari segmenti di rete locale in una rete locale vera e propria (quindi una LAN).

Per comporre i vari livelli si procede sempre a livello Fisico (1) e livello MAC/LLC (2), nello specifico vengono usati:

- **Repeater**

  Ripetitori. I segnali trasmessi sul mezzo fisico degradano con la distanza, nella fattispecie l'ethernet degrada dopo circa 200 metri, quindi ogni 200 (o, meglio, 100) metri si aggiunge un ripetitore. Il ripetitore **amplifica** e **rigenera** il segnale ricevuto. Il repeater collega due segmenti di rete che hanno la stessa tecnologia MAC; non legge i dati, vede semplicemente che sta arrivando un segnale e lo amplifica e passa oltre. Con l'allungarsi delle distanze chiaramente le trasmissioni impiegano un po' di tempo in più per viaggiare da mittente e destinatario, quindi nel caso di reti LAN più grandi diventa necessario allungare i tempi di timeout.

- **Hub**

  È un repeater multiporta. È il nodo centrale di una rete a stella. Quando riceve un segnale da una porta lo copia e lo manda a tutti gli altri elementi della rete. È usato pochissimo in quanto costa poco meno dello switch, che però offre più funzionalità e porte.

- **Bridge**

Funziona come uno switch (quindi lavora a livello 2) ma connette tecnologie a livello locale con MAC protocol **diversi** (ad es. Ethernet e Wifi). Se riceve trasmissioni da un protocollo (ad es. Ethernet) e deve mandarle ad un altro protocollo (ad es. Wifi) prende il pacchetto, lo smembra e lo ricostruisce in un frame compatibile con l'altro protocollo (i dati cambiano la "busta gialla").

- **Switch**

  Lavora a livello 2. A differenza del bridge permette di connettere molti più segmenti (oggi gli switch hanno anche 96 porte). Interconnette tecnologie **omogenee**, non diverse, e filtra i pacchetti da inoltrare a seconda della loro destinazione. Manda i dati in broadcast solo se non sa dov'è il destinatario. - **Buffered switch:** ha un buffer di memoria che ha la funzione di memorizzare tutti i frame che arrivano. Questo siccome la trasmissione deve essere smistata e non è automatica, ma segue le regole del protocollo MAC, può essere necessario che una trasmissione debba attendere il completamento di un'altra precedente.

Esempio completo di rete locale

Un esempio di diversi segmenti di rete locale collegati attraverso Bridge, Hub e Repeater.

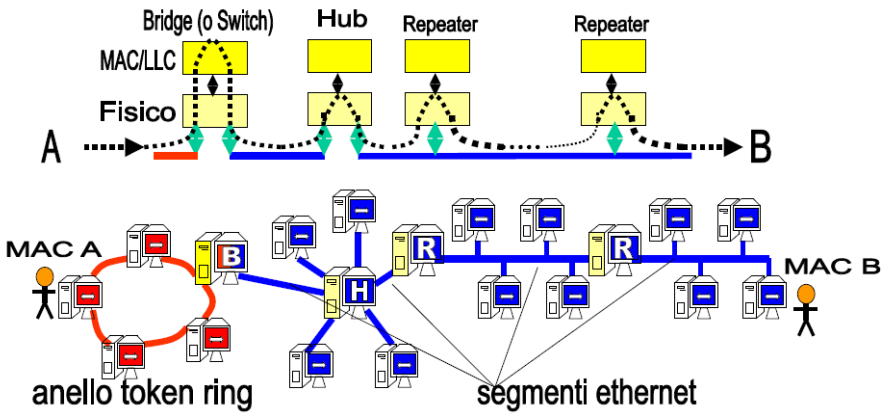


Figure 10: Esempio completo di rete locale

Reti di reti

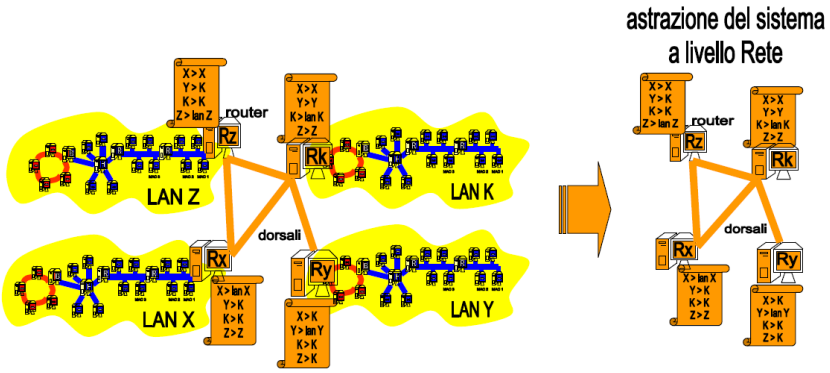
Questo è un concetto **fondamentale** per tutto il corso, non avere chiaro questo livello significa non saper fare granché.

Reti di reti e internet working

Nel momento in cui bisogna collegare fra loro più reti locali, i protocolli di Livello 1 e 2 diventano inefficaci, in quanto renderebbero la comunicazione complessa e soggetta a ritardi ed errori.

Si passa quindi alla necessità di costruire una gerarchia con cui organizzare le reti.

Al **Livello Rete** (3) appare la figura del **Router**, che funge da rappresentante della rete locale a lui associata (1 router = 1 rete locale). I router sono a loro volta collegati da linee dati veloci chiamate **dorsali**.



- Esempio di rete di reti locali

Il router controlla l'accesso alla rete locale:

- Smista i pacchetti che transitano per la rete:
  - Nel caso di un pacchetto nato nella rete e destinato alla rete stessa il router resta a guardare.
  - Nel caso di un pacchetto destinato all'esterno il router applica il protocollo di livello 3 e si interfaccia con le dorsali di collegamento.

Il protocollo di livello Rete (3) prevede che il router fornisca il frame di informazioni sugli **indirizzi IP**.

Questo perché il MAC non è un indirizzo dotato di struttura gerarchica, nello specifico le differenze fra MAC e IP sono:

- Il MAC è univoco e fisso per il dispositivo ma non ha una struttura gerarchica;
- L'IP ha una struttura gerarchica, è univoco ma non è esclusivo per il dispositivo: può cambiare (generalmente ogni giorno vengono riassegnati).

L'indirizzo IP ci fornisce **due** informazioni importanti:

1. Indica la rete di appartenenza del destinatario.
2. Indica, all'interno di questa rete, chi è il destinatario.

Può capitare che si formino pacchetti zombie, ovvero instradati male e che per questo finiscano in un loop senza destinatario finale; per rimediare il protocollo IP implementa un contatore che diminuisce ogni volta che il pacchetto viene trasmesso alla rete successiva, quando questo contatore raggiunge lo 0 il pacchetto viene **ucciso**.

Livello rete: Internet protocol (IP)

Come abbiamo accennato, al livello di Rete (3) nasce il bisogno di un nuovo tipo di indirizzo, l'indirizzo IP (**Internet Protocol**, IPv4 o IPv6 che sia).

Questo indirizzo funziona per tutto internet ed è globale, strutturato e gerarchico:

- **Globale** perché non esiste rete connessa ad Internet che non abbia indirizzo IP;
- **Strutturato** perché è diviso in due parti, una per l'indirizzo della rete locale e l'altro per l'indirizzo della scheda di rete interna alla rete locale;
- **Gerarchico** perché appunto le due parti di indirizzo sono in ordine gerarchico (rete locale > scheda di rete).

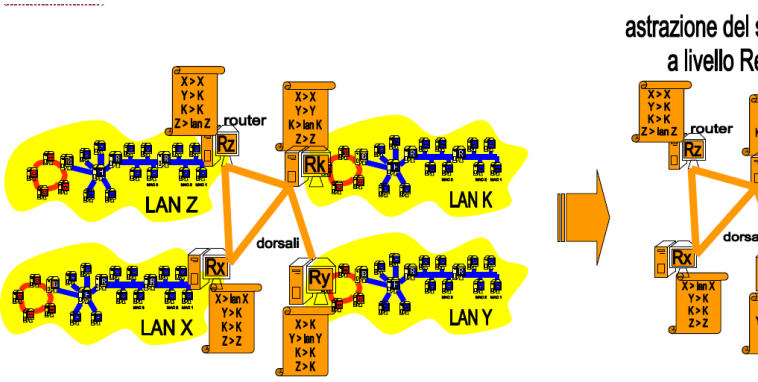
L'indirizzamento IP è detto anche logico proprio poiché si tratta di un indirizzo virtuale che può essere riassegnato a piacere (a differenza del MAC che è fisico e proprio della scheda di rete).

Ogni volta che io mi sposto fisicamente (con uno smartphone o un portatile) in una nuova rete cambio indirizzo IP. Ciò pone un grande grattacapo a livello di rete, con gli indirizzi che cambiano continuamente diventa difficile capire dove spedire i pacchetti. Di queste problematiche si occupa il Router.

> La busta del pacchetto: il protocollo inserisce il frame in un ulteriore busta che contiene gli indirizzi IP di mittente e destinatario.

Il Router

I router quindi sono dispositivi che lavorano a livello 3: nel momento in cui bisogna inviare e ricevere dati guardano solo la parte di IP che indirizza la rete, ignorando (momentaneamente) la parte che indirizza l'host, creando in questo modo una visione fortemente semplificata di Internet (come esemplificato nell'immagine).



- Astrazione della rete di reti a Livello 3.

Protocollo di Instradamento e tabelle

Ogni router possiede una **tabella di instradamento**, ovvero una tabella che mappa la topologia della rete (almeno a una distanza di 1: conosco i miei host e il mio router superiore). I router sono dispositivi molto semplici, non possiedono quasi mai più di 4 uscite, di cui uno rivolta verso la rete interna. Di solito ogni router conosce solo un solo altro router, ovvero il suo router "padre", comunemente chiamato il **default gateway**.

Il **protocollo di instradamento** è quello che si occupa di aggiornare le tabelle di instradamento del router, questo è necessario poiché le "strade" di internet non sono immutabili ma cambiano spesso e per i più svariati motivi: guasto alla linea, congestionamento, ecc.

Protocollo di routing

I cammini possibili per spedire un pacchetto sono diversi e trovare il più efficiente è compito dei **protocolli di routing**, che usando le tabelle cercano il cammino più breve e rapido possibile.

> Attenzione: protocollo di instradamento e protocollo di routing sono due cose diverse: il primo aggiorna le tabelle, il secondo usa le tabelle per scegliere la strada su cui mandare i dati.

Frammentazione

Per rispettare i limiti imposti dalla rete (o dal mezzo fisico di trasmissione) e per ridurre il limite di errori il router si trova spesso a **frammentare** i dati da inviare: ad esempio devo spedire un pacchetto da 128kb ma la rete mi consente di mandare pacchetti di massimo 32kb, quindi il router mittente frammenta il pacchetto e lo spedisce in vari frame, viceversa il destinatario deve riunire il tutto.

Dal momento poi che tra l'invio di un frammento ed il successivo il protocollo di routing potrebbe trovare una strada diversa, può avvenire che i frammenti

arrivino disordinati al destinatario (o che alcune parti non arrivino proprio).

> La comunicazione a livello rete è quindi di tipo **connectionless**: il destinatario forse riceve il pacchetto, se non lo riceve non è un problema del router, che si occupa solo di spedirlo (questa problematica si risolverà a livello 4).

Indirizzamento IPv4

Come abbiamo detto, un indirizzo IP viene associato a una e una sola interfaccia (o scheda) di rete.

Se così non è ci saranno due destinatari per lo stesso pacchetto.

I router di questo se ne accorgono subito: due macchine con lo stesso IP stanno necessariamente nella stessa rete locale (perché la parte network è uguale), quindi il router della rete locale se ne accorge non appena riceve due ack per un pacchetto.

Non appena il router si accorge del problema lo risolve internamente alla rete, senza mai farlo notare all'esterno: solo lui saprà che c'è stato il problema e lo avrà risolto.

Una sola macchina però può avere più indirizzi IP, poiché potrebbe avere più schede di rete (quindi attenzione alla differenza in caso di domanda all'esame).

Indirizzo IP Statico e Dinamico

L'indirizzo IP può essere statico: fisso un indirizzo IP ad un indirizzo MAC, così chiunque su internet sa che se vuole parlare alla mia macchina deve parlare a quell'indirizzo IP.

L'indirizzo IP dinamico invece cambia continuamente e quindi non mi rende sempre raggiungibile dall'esterno, rende più difficile essere sempre accessibile da internet.

Com'è fatto un indirizzo IPv4

Un indirizzo IPv4 è composto da 32 bit o 4 byte.

Di solito è indicato con 4 valori decimali separati da punti, dove ogni valore va da 0 a 255.

Useremo continuamente l'aritmetica binaria per manipolare gli indirizzi IP e capire cosa sta succedendo, quindi conviene saperla a menadito.

Ripasso di aritmetica binaria.pdf

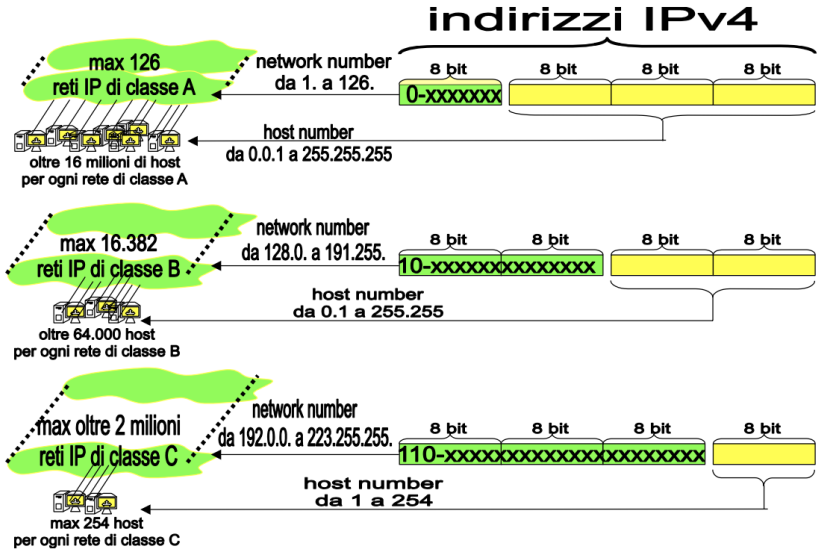
Un indirizzo IP è sempre composto di due parti (vedi immagine nel paragrafo dopo):

- Il **Network Number** della rete rappresentata dal router.
- L'**Host Number** dell'interfaccia di rete dentro a quella rete.

Classe di rete

Il valore dell'indirizzo IP determina la **classe della rete**. Le classi definiscono tipologie di reti logiche diverse.

- Classi di rete



Rete di Classe A: 0

Una rete di classe A è una rete in cui il bit più significativo vale 0, tradotto in decimale sono reti di classe A tutte le reti con indirizzo IP che va da `1.x.x.x` a `126.x.x.x`.

> L'indirizzo con tutti 0 e gli indirizzi con tutti 1 (dopo i bit che indicano la classe di rete) sono sempre riservati, quindi in ogni conteggio gli indirizzi disponibili sono 2 in meno di quelli fisicamente possibili.
>
> Attenzione: l'indirizzo con nella parte host tutti i bit a 0 è usato per esprimere la rete su Internet, mentre quello con tutti i bit della parte host a 1 serve per indicare le trasmissioni broadcast per i pacchetti destinati a tutti gli host che appartengono a quella famiglia logica.

Quindi ci sono al massimo 126 reti di classe A in tutto internet: queste reti sono molto rare e sono utilizzate dai grandi provider di internet, poiché permettono di disporre di ben 2^{24} host diversi (16.777.216 di host). Si tratta quindi di reti dal potenziale enorme.

Rete di Classe B: 10

Se i primi due bit più significativi valgono 1 e 0 siamo in una rete di classe B. Sono reti di Classe B tutte quelle i cui indirizzi IP vanno da `128.0.x.x` a `191.255.x.x`.

Una rete di classe B è identificata dai primi due byte dell'indirizzo IP, quindi tolti i due bit fissi e gli indirizzi riservati, abbiamo un totale di 16.382 reti. Restano a disposizione due byte per indirizzare gli host, che saranno quindi più di 64.000.

Rete di Classe C: 110

Una rete di classe C ha i primi tre bit più significativi che valgono rispettivamente 1, 1 e 0. In decimale parliamo di tutti gli indirizzi da `192.0.0.x` a `223.255.255.x`.

Facendo sempre i soliti conti abbiamo che esistono 2^{21} reti di classe C (2.097.152 di reti), ognuna delle quali potrà disporre al più di 254 host (256 meno i soliti due riservati rispettivamente a indicare la rete e a indircare le comunicazioni broadcast).

Subnetwork e netmask

È possibile creare sottoreti totalmente stagne all'interno di una stessa rete attraverso le maschere di rete.

L'idea è che posso rubare uno o più bit a partire dal più significativo della parte host dell'indirizzo IP per indirizzare una sottorete.

Se rubo un bit dalla parte host e lo "regalo" alla parte rete sto facendo subnetting. In base a quante sottoreti voglio creare scelgo il numero di bit da rubare: voglio creare due sottoreti? Mi basta rubare solo il primo bit, per crearne 3 posso rubare due bit e così via.

Per fare questa operazione si usa una **netmask** (maschera di rete), che appunto si occupa di "convertire" i bit della parte host.

La netmask spezza in due componenti logiche l'indirizzo IP:

- Indirizzo di subnetwork, che ha i bit uguali a 1 nella netmask;
- Host number dell'host appartenente alla sottorete: bit corrispondenti ai bit uguali a 0 nella netmask.

Con questi due componenti è possibile creare una gerarchia di sottoreti ciascuna amministrata dal suo **default router**. I default router fanno tutti capo al router che poi effettivamente rappresenta la rete su internet.

Quello che la netmask fa essenzialmente è dire quali sono i bit che compongono la parte rete e sottorete, questo è possibile perché il router utilizza l'indirizzo IP e la netmask per fare un & bit-a-bit al termine del quale l'indirizzo che ho

ottenuto è quello della rete/sottorete, mentre la parte che ho azzerato poi la posso andare a riprendere ed è l'indirizzo host.

Di solito non si usa un intero byte per creare una sottorete poiché una volta raggiunto il numero massimo di host, eventuali altri host finiranno in altre sottoreti che non avevamo intenzione di creare. Si prendono solo i bit necessari a creare il numero di sottoreti che voglio.

Separare in sottoreti mi permette di impedire che i frame arrivino ovunque nella rete. Il paradosso è che possiamo virtualmente staccare degli host che sono fisicamente attaccati semplicemente cambiando gli indirizzi IP dei due host.

Al contrario del subnetting c'è il supernetting, ovvero raggruppare più reti in una più grande. Non è così semplice come il subnetting, vedremo meglio in futuro.

Esercizi da esame

Subnetting e netmask

Notazione CIDR - Classless Inter Domain Routing

È un'alternativa per riferirsi in maniera semplice alle maschere di rete quando si parla fra colleghi.

Prendiamo una rete di classe C, ad esempio 193.48.32.x, impostiamo una netmask che divide la rete in due: `255.255.255.128`, questa netmask la possiamo indicare con la notazione `/25`, che è un modo compatto di dire che impostiamo a 1 i primi 25 bit della netmask.

Questa notazione funziona solo tra "umani", nelle macchine dobbiamo comunque usare la notazione decimale.

Routing, ICMP, ARP, DHCP, IPv6

Forwarding

Il forwarding è il processo di instradamento dei pacchetti dal mittente verso la destinazione.

Lo abbiamo affrontato tramite un esempio:

Forwarding dei pacchetti

Routing

Il problema del routing altro non è che il problema di dover aggiornare le tabelle di instradamento dei pacchetti.

Gli algoritmi di routing provano a fare un'istantanea della rete (anche se è impossibile avere un'immagine completa, è troppo dinamica) e a delineare la tendenza della rete in quel momento, così da poter costruire delle mappe stradali più efficaci possibili (l'obiettivo è sempre trasportare tanti dati, nel minor tempo possibile e con il maggior success rate possibile).

Ci sono due motivi per cui i collegamenti fra le reti di internet possono cambiare:

- Si trova una strada più rapida, oppure si congestiona una strada o cade la linea, quindi delle modifiche fisiche sulla rete;
- Cambiano gli accordi e le politiche economiche tra i provider di grandi porzioni della rete (AS, i Sistemi Autonomi, grosse collezioni di reti soggette ad una politica di amministrazione comune).

La seconda tipologia di cambiamenti porta a modifiche più lente, al più la rete viene aggiornata ogni ora, mentre le modifiche del primo tipo cambiano la rete ogni frazione di secondo.

Questo continuo cambiamento della rete è anche il motivo per cui i pacchetti non percorrono tutti la stessa strada e possono arrivare in disordine.

Algoritmi di Routing

Gli algoritmi di routing adottano una serie di protocolli in modo da poter avere una sintomatologia della rete in quel momento.

Una prima stima la fanno già nel momento in cui inviano i pacchetti e aspettano gli Acknowledgements: se questi non arrivano probabilmente la strada è interrotta.

A questi si aggiungono i protocolli appositi per assistere i router nel lavoro da fare. Gli algoritmi devono essere il più semplici possibile, non ha senso usare algoritmi complessi ed esosi di risorse per aggiornare le tabelle di instradamento di una rete modesta, mentre non bastano algoritmi leggeri per aggiornare le tabelle di grandi reti nazionali.

La soluzione quindi è che sono anche essi gerarchici: a livello locale usiamo algoritmi statici (praticamente dei non algoritmi, dato che se la rete è statica quelle poche volte che cambia possiamo aggiornare le tabelle a mano), poi salendo fino a livello ad esempio della rete GARR si usano algoritmi come BGP (Border Gateway Protocol), RIP (Routing Information Protocol) o OSPF (Open Shortest Path First).

Rete GARR

Piccolo interludio più a scopo informativo sulla rete GARR.

È la rete che dà i servizi web all'intero Ateneo e a tutti gli enti di ricerca italiani.

Consortium GARR

Qui si può vedere una mappa della rete GARR, è un ottimo esempio di una grande rete nazionale.

GINS Weathermaps

Dalla weathermap si può vedere in tempo reale lo stato di utilizzo delle reti connesse alla rete GARR.

> *"I router sono macchine spettacolari: immaginate un rack di server che trita pacchetti in entrata e uscita. Un rack di server, di calcolatori single-purpose."*

Protocollo ICMP

L'Internet Control Message Protocol è quel protocollo che gestisce i messaggi di controllo su Internet. Si tratta in poche parole di uno standard per codificare le informazioni necessarie alla gestione di Internet stesso. Viene utilizzato continuamente da host, router e gateway per scambiare informazioni di livello rete.

I messaggi del protocollo ICMP viaggiano dentro le "buste arancioni" del protocollo IP, quindi a livello 3 (ICMP non arriva mai al livello 4).

Il protocollo scambia informazioni tramite codici messaggio; ad esempio il codice 000 corrisponde ad una *echo reply*, mentre il codice 008 corrisponde ad una *echo request* (e si aspetta una reply in cambio). Di seguito alcuni tipi di messaggi che ICMP può scambiare:

- Rete di destinazione non raggiungibile (possibile interruzione di rete?);
- Rete di destinazione sconosciuta (abbiamo sbagliato l'indirizzo?);
- Host destinazione non raggiungibile (host spento o scollegato?);
- Host destinazione sconosciuto (abbiamo sbagliato indirizzo?);
- Protocollo richiesto non disponibile;
- Ricerca di un cammino alternativo per la destinazioni (se esiste).

Ogni messaggio ICMP ha un campo type e uno code, la coppia dei due determina il contenuto del messaggio.

Un ulteriore campo contenuto in un messaggio ICMP è il campo TTL (Time To Live): ovvero il counter che decidere se uccidere il pacchetto perché sta rimbalzando troppo nella rete. Il router che uccide il pacchetto recupera l'indirizzo

IP del mittente e gli manda un messaggio ICMP per avvisarlo dell'uccisione del pacchetto, in modo che se il mittente lo desidera può ritentare la trasmissione impostando un TTL più grande o cambiando instradamento.

Vediamo ora due applicativi che fanno uso del protocollo ICMP.

Ping

Serve per verificare la connessione fra due host.

Nel fare un ping da un host-1 ad un host-2 ciò che accade è che l'host-1 manda una echo request a host-2 e questo risponde con un echo reply. Nel fare ciò viene calcolato anche il tempo di andata e ritorno delle richieste (RTT - Round Trip Time).

```
ping csgw-3-0-5.cs.unibo.it
```

Ping ci fornisce anche il valore TTL impostato, in questo modo capiamo che ci sono stati al più x router sul percorso, altrimenti il pacchetto sarebbe morto prima.

Al programma Ping posso passare sia un indirizzo IP che un nome logico. Nel secondo caso Ping recupera da solo l'indirizzo IP necessario facendo una richiesta al server DNS.

Traceroute

Nel cmd di windows si chiama tracert, nel terminale Unix traceroute.

```
traceroute csgw-3-0-5.cs.unibo.it
```

L'applicativo Traceroute serve per tracciare la strada che percorre il nostro pacchetto per arrivare da noi all'host destinazione.

Per fare ciò Traceroute emette dei Ping in cui il valore di TTL è impostato a 1 e incrementato sequenzialmente. Così facendo il pacchetto viene ucciso ad ogni passo ed ogni volta viene mandato un messaggio ICMP all'applicazione. Il messaggio ICMP contiene anche l'indirizzo del router che lo ha mandato e quindi così facendo Traceroute traccia tutti i router sul cammino.

Protocollo ARP e RARP

La ARP table è quella tabella che associa ad ogni indirizzo IP connesso alla rete il relativo indirizzo MAC della scheda di rete connessa. Questa serve per poter inviare correttamente i pacchetti, dal momento che devo specificare entrambi gli indirizzi.

Anche il protocollo ARP può essere utilizzato tramite un applicativo da riga di comando omonimo.

La principale funzione del protocollo ARP è di dire alla scheda di rete che lo ha invocato quale sia l'indirizzo MAC di un determinato indirizzo IP.

A grandi linee ARP funziona così:

- Il router genera un frame con la richiesta ARP (se sei il calcolatore con IP x, rispondi col tuo MAC) e la manda in broadcast sulla rete locale;
- Il calcolatore interessato, se esiste ed è connesso, manda un frame di risposta al router nel quale specifica il MAC richiesto.

Il protocollo RARP è il contrario (infatti sta per Reverse-ARP): dato un indirizzo MAC conosciuto, richiedo l'indirizzo IP associato.

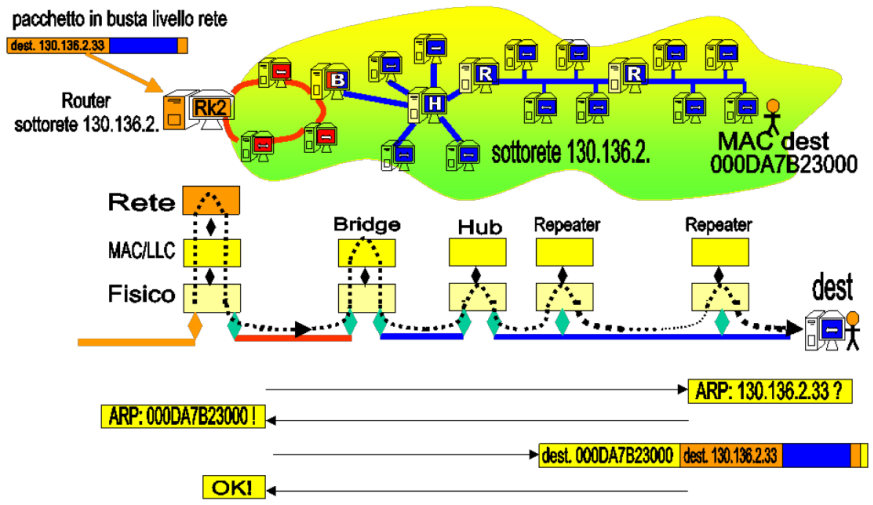


Figure 11: Esempio di scambio di messaggi ARP

> Sperimentando in classe abbiamo scoperto un trucco interessante: facendo un **ping** (con l'omonima applicazione) all'indirizzo di **broadcast** ricevo in risposta la ARP table di tutte le macchine attualmente connesse. Quindi con ping posso fare un'istantanea della rete.

Protocollo DHCP

È il protocollo che si occupa di assegnare un indirizzo IP ad ogni nuova macchina che si connette alla rete. Come per i precedenti protocolli, DHCP è anche un'applicazione che incarna le funzioni di questo protocollo.

La parte applicazione si concretizza in un server DHCP, che si occupa dell'assegnaziojne degli indirizzi IP, e in un client, ovvero colui che richiede al server di avere un IP.

Può capitare che il router faccia anche da DHCP server, non sempre ma può accadere.

In genere i numeri di rete (classe A, B e C) sono assegnati da grandi enti internazionali (come RIPE, ICANN, ARIN, APNIC).

I numeri di host possono invece essere assegnati manualmente (dall'amministratore di rete) oppure automaticamente da parte del server DHCP.

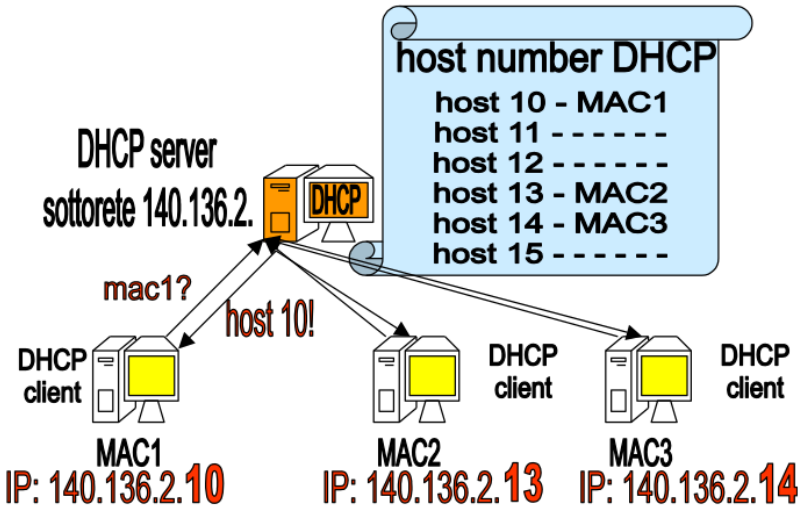


Figure 12: Esempio schematico di server DHCP

Indirizzamento IPv6

Nel 1990 furono avviati i lavori per la definizione e l'implementazione del protocollo IPv6; questo poiché si prevedeva che IPv4 avrebbe finito gli indirizzi tra 2008 e 2018 (4 miliardi di indirizzi a disposizione). La previsione fu azzeccata: gli indirizzi IPv4 sono effettivamente finiti nel 2018, questo grazie anche al fatto che negli anni precedenti si era iniziato ad usare in modo ibrido IPv6 e IPv4.

Il grande problema che si è dovuto affrontare nell'implementare IPv6 è stato che bisognava che il nuovo protocollo rispettasse le regole dello stack (dell'architettura di protocolli).

Caratteristiche salienti

Gli indirizzi IPv6 sono formati da 16 byte, il che porta il totale di indirizzi disponibili a $2^{128} = 3,4 \times 10^{38}$ (qualcosa come miliardi di miliardi di miliardi di miliardi), in questo modo ci si è tenuti abbastanza larghi per essere sicuri di non

dover ricambiare il protocollo in futuro (ci sono circa 15.000 indirizzi per ogni metro quadro della superficie terrestre).

Anche in IPv6 abbiamo il concetto di rete e sottorete e di host:

- Per la parte rete abbiamo a disposizione i primi 64 bit;
- I 64 bit meno significativi sono l'host number: 2^{64} host dentro ogni rete e sottorete, sono tantissimi (18 miliardi di miliardi, ogni rete può avere un numero infinitamente maggiore a quello di una vecchia classe A).

Host numbers

Anche in IPv6 vale il discorso che il router può assegnare a piacere un numero di host ai dispositivi connessi alla propria rete, a patto che questo numero non sia già stato usato per un altro host.

Dal momento che per il numero di host abbiamo a disposizione ben 64 bit e un MAC address è composto da 48 bit, si è deciso che il router assegnerà sempre come numero di host il MAC address dell'host stesso. In questo modo siamo sicuri non solo che il numero di host sia unico all'interno della rete, ma che sarà unico in tutta internet.

Inoltre in questo modo si semplifica anche il protocollo DHCP, dato che so già che il mio host number è uguale al mio MAC address e quindi ci saranno meno scambi di informazioni.

Nuova struttura del pacchetto

Dal momento che gli indirizzi IPv6 hanno una forma e caratteristiche diverse da quelli IPv4, bisogna che cambi anche la struttura dei pacchetti inviati tramite questo protocollo.

Innanzitutto cambia l'header del pacchetto, in modo da poter ospitare i 16 byte dell'indirizzo anziché i 4 byte di IPv4.

Inoltre nel protocollo IPv4 era previsto un campo, il ToS, che poteva essere utilizzato per garantire priorità ai pacchetti che la richiedevano. Il router avrebbe dovuto analizzare il campo ToS di tutti i pacchetti in entrata e poi li avrebbe smistati in uscita in base alla priorità richiesta dal pacchetto.

Questa funzionalità non è mai stata utilizzata poiché appesantiva di molto il carico di lavoro dei router, rallentando la connessione e annullando i vantaggi che avrebbe dovuto portare. Il capo ToS è quindi rimasto inutilizzato.

Con la definizione del protocollo IPv6 si è provato a reintrodurre questo sistema di priorità. Per ovviare alle problematiche di lentezza si è stabilito che non saranno più i singoli pacchetti ad avere o meno una determinata priorità, ma l'avranno sequenze di pacchetti intere.

Così i router si trovano a gestire per priorità dei flussi di pacchetti e non diventano troppo lenti. Best of both world I guess.

Frammentazione

In IPv4 il protocollo si occupava di dividere i pacchetti ricevuti dal livello trasporto in tranci compatibili con la dimensione richiesta per l'invio, operando quindi una frammentazione dei pacchetti.

In IPv6 questa pratica è stata eliminata, o meglio è stata spostata al livello trasporto, in modo che il livello rete riceva i pacchetti già delle dimensioni giuste e si occupi solo di spedire i pacchetti al livello di sotto.

Anatomia di un indirizzo IPv6

La rappresentazione binaria di un indirizzo IPv6 non è molto dissimile da quella di un indirizzo IPv4, si tratta di 16 byte invece che di 4.

La rappresentazione "human-readable" però è abbastanza diversa:

- I byte sono scritti non più in decimale ma in esadecimale, due cifre hex rappresentano un byte;
- Non ci sono più i `.` a fare da separatori ma i `:` ;
- Ogni gruppo dell'indirizzo rappresenta non uno ma 2 byte;
- In totale ci sono 8 gruppi separati dai `:`, che infatti sono 16 byte;
- I gruppi in cui tutte le 4 cifre valgono 0 possono essere omessi.

Esempio di indirizzo IPv6 (esteso): `BC12:FD00:3813:0000:0000:1111:2222:3333`.

Versione abbreviata: `BC12:FD00:3813:::1111:2222:3333`.

Tunneling IPv4

L'ultima grande problematica che si è dovuta affrontare è stata il dover decidere come passare dal protocollo IPv4 all'IPv6.

Una volta messo appunto il protocollo IPv6 è impensabile eseguire un semplice switch-off (ovvero in un dato momento si smette di usare in toto il procotollo v4 e si inizia ad usare solo il v6), poiché non tutti i router, non tutte le schede di rete, non tutti i calcolatori sono in grado di gestire il nuovo protocollo. Uno switch-off del genere richiederebbe anni se non decenni di preparazione e ciononostante potrebbero comunque verificarsi bug e malfunzionamenti che si protrarrebbero nella rete per altri anni ancora.

L'unica strada percorribile è quindi quella di una lenta sostituzione del protocollo IPv4: si crea pian piano una sorta di Internet parallela che funziona con IPv6 e la si integra all'internet che gira su IPv4. Le schede di rete più moderne riconoscono entrambi i protocolli ed in base allo stack che devono utilizzare si comporteranno di conseguenza.

Rimane solo da gestire il caso in cui ci siano tragitti sulla rete (o intere backbone) che ancora non sono in grado di gestire il protocollo IPv6. In questi casi i router agli estremi di questi percorsi opereranno il tunneling, ovvero incapsuleranno il

pacchetto IPv6 dentro una busta IPv4 e la spediranno lungo la backbone, per poi estrarre il pacchetto non appena questo esce dal tunnel creato.

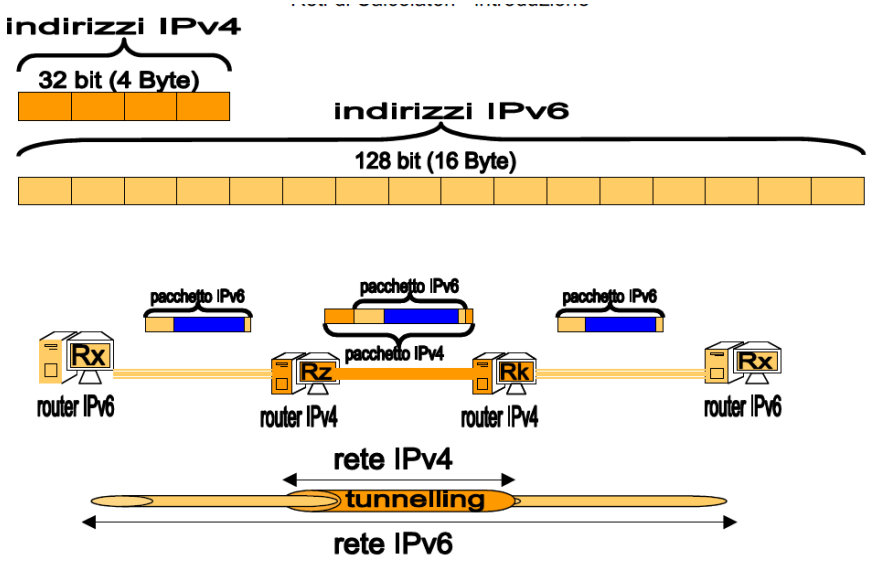


Figure 13: Schema IPv6 ed esempio di tunneling IPv4

Livello Trasporto

Introduzione

Lasciamo ora il livello rete e passiamo al quarto livello dello stack: il livello trasporto.

Questo livello ha due compiti principali:

- Rendere la comunicazione di rete - *che è connectionless* - affidabile end-to-end. I router sono dei dispositivi che lavorano a livello 3, questo vuol dire che loro mandano avanti i pacchetti, come dei passamano, non gli importa cosa succede ai pacchetti: se non arrivano, se il buffer si riempie e fanno cadere i pacchetti in eccesso. Sono tutti problemi che per i router non sono importanti, semplicemente loro vanno avanti a spedire i pacchetti successivi. Quindi il primo compito del livello trasporto è assicurarsi che i pacchetti dispersi vengano reinviati, in modo da rendere la connessione affidabile.

- Gestire la congestione. Immaginiamo questa situazione: un router riceve 300Gb di dati al secondo e ha un buffer di 300Gb, però riesce a elaborare e spedire solo 10Gb al secondo. Ne consegue che il primo secondo restano da gestire 290Gb, ma il secondo subito dopo arrivano altri dati che superano la capacità del buffer, quindi il router li lascia cadere. I router che hanno inviato quei pacchetti non riceveranno l'ack e quindi ripeteranno il processo; ben presto non solo il router centrale sarà congestionato, la congestione si espanderà a macchia d'olio verso tutti gli altri router. Quello che fa il livello trasporto è regolare in maniera rapida e responsive la pressione sui router: non appena ci si rende conto che la rete inizia a congestionarsi bisogna subito mettere in pratica dei sistemi appositi per decongestionarla immediatamente.

I protagonisti della suite di protocolli del livello trasporto sono TCP e UDP.

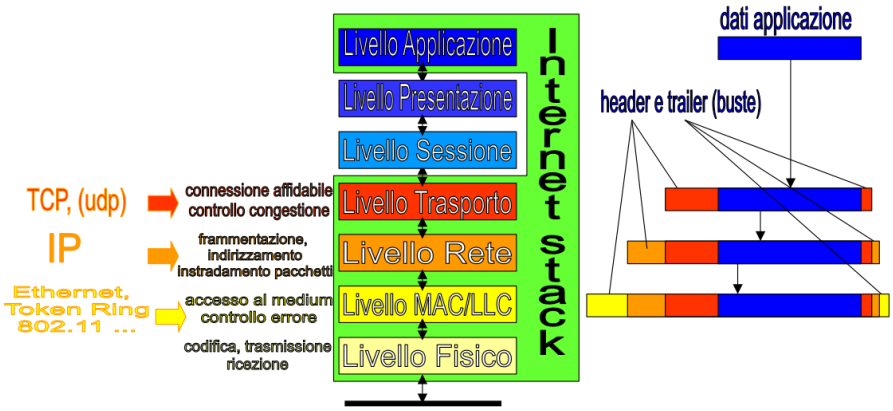


Figure 14: Schema illustrativo del livello trasporto

Ruolo del livello trasporto

I pacchetti a livello trasporto si chiamano **segmenti**.

Quando il destinatario finale riceve i segmenti provvede a fare il riordino, che è solo prerogativa sua, non di chi trasmette i dati (i router leggono solo la busta pacchetto, non quella segmento).

Il livello trasporto riguarda solo il mittente originale e il destinatario finale. Non riguarda i router nel mezzo, in questo modo la backbone di internet non viene appesantita e si mantiene la scalabilità.

Il destinatario finale deve crearsi un buffer in cui salvare i *pacchetti* e una volta ricostruito il *segmento* originale (cioè ha messo tutto in ordine) può passarlo al livello di sopra.

Socket

I socket sono gli end-point di un canale di comunicazione a due vie. Essi sono identificati dall'indirizzo IP e dalla porta usata, dato che ogni porta corrisponde ad una applicazione sul dispositivo che sta comunicando.

I protocolli che gestiscono i socket sono due: TCP e UDP.

Protocollo TCP

TCP (Transmission Control Protocol) è il protocollo di livello trasporto usato de-facto su internet. Lo standard dell'architettura ne prevede l'utilizzo in connubio con il protocollo IP, per questo si parla più propriamente di TCP/IP.

Il protocollo TCP prevede che prima di iniziare uno scambio dati, la connessione vada attivata. Il server TCP si mette in ascolto sulle porte delle applicazioni che gestisce, a quel punto se un computer vuole avviare una comunicazione inizia quella che è chiamata la **three-way-handshake**, a seguito della quale inizia la comunicazione vera e propria che poi verra chiusa al termine (quando il client ha ottenuto quello che voleva o quando il server manda in timeout il client).

La three-way-handshake

Come abbiamo detto TCP richiede che la connessione venga attivata prima di iniziare lo scambio di dati. Questa attivazione viene fatta tramite la three-way-handshake, che come il nome suggerisce consta di 3 passaggi:

- Il client richiede la connessione al server sul socket opportuno;
- Il server (se è disponibile) risponde con un ack OK;
- A questo punto il client manda un terzo messaggio con i parametri necessari per inizializzare lo scambio dati.

Questi passaggi servono a rendere la connessione affidabile, scalabile ed efficiente.

Un possibile vettore di attacco può essere il mandare la richiesta di apertura di connessione e, dopo aver ricevuto l'ack, non mandare il terzo messaggio e lasciare il server in attesa (e quindi occupato). Per prevenire questo tipo di attacchi sono stati implementati i timeout, in modo che il server chiuda la connessione dopo troppo tempo in cui non ha ricevuto nulla.

TCP richiede anche una procedura per chiudere in maniera corretta la connessione: questo procedimento serve alle due parti per accordarsi sul fatto di non avere più nulla da dirsi e quindi di poter buttare giù la connessione di comune accordo. È uno scambio necessario poiché il server quando apre la connessione alloca un buffer per la comunicazione con il client, buffer che viene rilasciato per altre connessioni solo al termine della comunicazione corrente.

Controllo di flusso e congestione della rete

I pacchetti sono grandi pochi kilobit, se ne mandassi uno alla volta userei solo una frazione della rete (se ad esempio avessi una 100 mega, userei pochissimo), quindi bisogna trovare un modo di inviare più pacchetti insieme senza però congestionare i router.

Il problema sta nel capire quali pezzi del segmento ci siamo persi e richiedere il reinvio di quelli soltanto. In pratica dobbiamo trovare un modo per far conoscere al destinatario un'informazione che ha solo il mittente.

È matematicamente impossibile avere la certezza di riuscirci, ma dobbiamo avere fiducia che prima o poi ci riusciamo. Ci basta una certezza che la rete funzioni il 99,9% delle volte e lo faccia nei tempi previsti.

Non posso sapere se il mio pacchetto farà o meno saltare un router sopraffatto dal lavoro, quindi non mi resta che mandare il mio pacchetto sperando che i router nel mezzo reggano. E come me fanno tutti i mittenti sulla rete.

Esiste però una strategia da adottare per evitare che un congestionamento si propaghi a macchia d'olio. Per sapere i miei limiti inizio inviando con un pacchetto, se torna l'ack posso provare a mandarne due, se tornano gli ack provo a mandarne 4 e così via, in questo modo sto testando la rete come fosse una scatola nera. Questo meccanismo è detto di **sliding window**, poiché rappresenta appunto la finestra di pacchetti che posso inviare.

Lo stesso fanno tutti gli altri utenti connessi. I router ricevono sempre più pacchetti e a un certo punto saltano e fanno il drop. Se ad un certo punto non mi torna l'ack di un segmento potrebbe essere un sintomo di riempimento della rete, se rispedisco il pacchetto e arriva l'ack allora era solo un problemino incidentale e nulla più, ma se inizia a perderne di più è un sintomo grave e vuol dire che il buffer di un router nel mezzo si sta congestionando, quindi bisogna rallentare con l'invio di dati.

Il rallentamento non è graduale: si decelera con **stop-and-wait**: si riparte da un pacchetto e si procede col probing: mando un pacchetto e vedo se il router ce la fa. Questo decadimento serve perché se il pacchetto è droppato vuol dire che i buffer dei router sono pieni, quindi la cosa più furba è ripartire da 1, dando così modo al router di smaltire il carico, e ricominciare il gioco.

La fase in cui raddoppio i pacchetti ogni volta si chiama **slow start**; Una volta che arrivo al punto in cui la scorsa volta si congestionò il router, non raddoppio più l'invio di pacchetti e mando solo un pacchetto in più per volta, questo comportamento si chiama **congestion avoidance**.

Nel caso di una connessione TCP il meccanismo di avvio di una conversazione contiene anche le informazioni su quanti pacchetti riesce a gestire il server, in modo che quando con il slow start arriviamo a quel numero, smettiamo di incrementare il numero di pacchetti inviati.

Ci sono anche casi in cui il protocollo fa affidamente all'utente come fosse parte dell'architettura: il protocollo UDP prevede che l'utente rimandi o rinunci a stabilire una connessione se questa non risponde o risponde lentamente (quindi se la rete è congestionata). Alcune applicazioni quindi possono essere basate su protocolli connectionless.

Se oggigiorno internet funziona molto bene è perché troviamo sempre strade spianate, non congestionate. Questo vuol dire che abbiamo un'infrastruttura capace di reggere tutto questo traffico senza entrare in congestione.

Protocollo UDP

Il protocollo UDP (User Datagram Protocol) è molto meno usato del TCP poiché è un protocollo di tipo connectionless, infatti UDP non necessita di protocolli di apertura e chiusura della connessione (niente three-way-handshake, se il client vuole comunicare con il server lo fa e basta) e non dispone di meccanismi per assicurarsi che tutti i segmenti siano arrivati a destinazione.

UDP viene usato in pochi casi, ad esempio come smistatore di un server: posso usare un socket con il protocollo UDP che riceve tutte le richieste del server e in base alle richieste e alle risorse disponibili indirizza i client verso altri socket che saranno invece gestiti con modalità TCP.

Nomi di dominio e servizio DNS

Gli utenti preferiscono navigare su internet tramite nomi di domini, ad esempio nomi di host come cs.unibo.it. Questi nomi hanno quindi lo stesso funzionamento degli indirizzi IP, difatti vengono assegnati da enti internazionali per evitare che possano esistere due nomi uguali per risorse diverse. ai router servono gli indirizzi IP. Questi indirizzi fanno comodo ma non sono parte dell'architettura di rete, che prevedere l'uso di indirizzi IP.

Il DNS è un servizio che risolve questo problema, esso prende in input un nome di dominio human-friendly e restituisce l'indirizzo IP corretto.

Organizzazione dei server DNS

Il servizio DNS è formato da una catena di server organizzati gerarchicamente.

Ogni router deve far riferimento ad almeno un server DNS, e ogni server DNS conosce almeno un DNS suo superiore nella gerarchia fino a quando si raggiunge la radice.

I server ricevono richieste tramite un protocollo che si chiama anch'esso DNS (quindi DNS indica sia il protocollo sia il server che realizza tale protocollo).

Un server DNS non è tenuto a conoscere tutti gli indirizzi di internet, quindi il protocollo prevede che il DNS locale, nel momento in cui non conosca l'indirizzo

IP della risorsa richiesta, restituisca l'indirizzo del suo superiore a cui poter inoltrare la richiesta e così via finché non si trova il server che conosce l'indirizzo IP richiesto.

Un meccanismo migliore è quello del *DNS ricorsivo*: se il mio server DNS locale non conosce la risposta è lui stesso che chiede al suo superiore e così via finché la risposta non torna all'host che aveva fatto la domanda.

DNS è un servizio che si basa su UDP, quindi è una richiesta oneshot: se si congestiona il server la richiesta cade. Per questo motivo il funzionamento del protocollo lato host è abbastanza basilare: il client manda una richiesta una volta sola e se non riceve risposta dopo un po' di tempo di downtime ripete la richiesta. La risposta funziona anche come ack, perché non ha senso mandare sia l'ack che l'indirizzo richiesto.

Il meccanismo iterativo previene il congestionamento dei server, che se sono sopraffatti di lavoro dicono al client di mandare la richiesta ad un altro server DNS, liberandosi così di lavoro quando sarebbero sul punto di bloccarsi.

Welcoming server

Il welcoming server sovrintende ad un rack di server DNS; il suo compito è controllare quanto sono carichi i server sotto di lui e inoltrare la richiesta a quello meno carico, in modo da distribuire il carico e evitare collassi.

Il welcoming server smista le richieste in arrivo ma non gestisce la risposta.

DNS root

I DNS root server (radice del mondo) conoscono gli indirizzi IP di tutti i DNS a loro inferiori. Non esiste un solo DNS root server ma qualche decina; ognuno di essi è gerarchicamente equivalente agli altri.

Tutti i domini appartengono al dominio radice del mondo, ad esempio il dominio del dipartimento, che è `cs.unibo.it`, appartiene al dominio di `unibo`, che a sua volta appartiene al dominio `it` che a sua volta appartiene al dominio *radice del mondo*. Il dominio radice andrebbe indicato con un punto finale ma siccome è unico non è necessario esplicitarlo.

I server DNS root non contengono gli indirizzi di ogni singola risorsa su internet, essi contengono gli indirizzi dei DNS che sovrintengono ai domini che stanno sotto i DNS root, quindi ad esempio conoscono gli indirizzi dei server DNS che gestiscono i domini `.it`, di quelli che gestiscono i domini `.com` e via dicendo.

Per fare un esempio: se una macchina in cs.unibo.it (informatica) vuole accedere ad una macchina di mineralogia, essa chiederà al server DNS di unibo.it chi sia mineralogia, se il server non lo sa non ha senso che inoltri la richiesta al server superiore, perché riconosce che unibo appartiene al suo dominio, quindi manda la richiesta ai server che sovraintende e richiede se qualcuno di loro sa dove si

trovi mineralogia e, una volta ottenuto l'indirizzo corretto, lo restituisce alla macchina del dipartimento di informatica che lo aveva chiesto.

Cache

Ogni server DNS ha una cache per le richieste che riceve, in modo da avere la risposta pronta per la seconda volta che la riceverà e così alleggerire il carico della rete.

Livello Applicazione

Il livello applicazione è quello che contiene le API per usare le funzioni che esso stesso mette a disposizione.

Alcune famose applicazioni di rete sono:

- Email (SMPT, POP3, IMAP)
- World wide web
- DNS
- HTTP

I protocolli fra il livello applicazione e quello trasporto, cioè i livelli sessione e presentazione, non sono usati nell'architettura dei protocolli di internet, che quindi con il livello applicazione si compone in totale di 5 livelli.

Il livello applicazione quindi si basa direttamente sul livello trasporto e per questo motivo molte applicazioni che richiedono servizi connection-oriented si basano sul protocollo TCP usando numeri di porta che nel tempo sono diventati di fatto degli standard, questi sono detti **Well-known port numbers**, ad esempio:

- HTTP usa sempre la porta 80, mentre HTTPS usa di solito la porta 443;
- SMTP usa sempre la porta 25;
- DNS usa sempre la porta 53;
- E così via.

Esempi di servizi di livello applicazione

Vediamo alcuni servizi che sono offerti dal livello applicazione.

- Posta elettronica
  - SMTP: Tutti lo usiamo per spedire messaggi di posta a qualcun altro. Il nostro client si connette ad un server con due socket, sopra ognuno di essi c'è un linguaggio (specificato dalle API) che viene usato per inviare le mail.

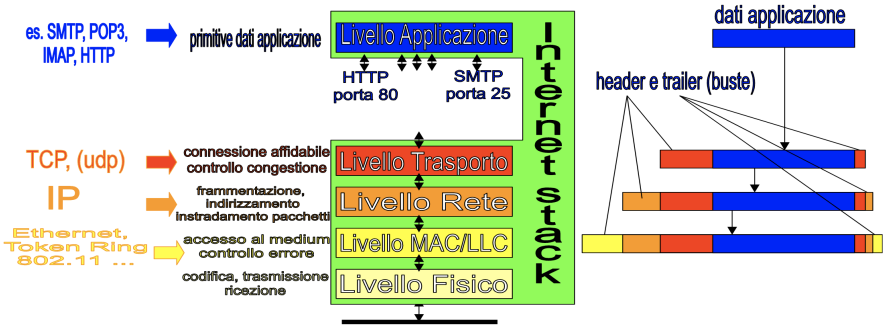


Figure 15: Schema del livello applicazione

  - POP3 o IMAP: Sono alternativi l'uno all'altro, IMAP è il più recente dei due e ha API più ricche. Questi protocolli servono a leggere i messaggi in arrivo.
- World Wide Web:
  - Basato sul protocollo HTTP, che è quello usato per trasferire pagine web. L'idea è di avere un testo sul mio computer dove determinate parole sono collegate a dei socket su internet che vengono aperti quando ci clicco sopra e mi aprono la pagina web relativa (un ipertesto). Questa è stata la scintilla iniziale del web. Improvvisamente tutto era diventato dinamico e rapido.

Servizi client-server e peer-to-peer

Le applicazioni ed i servizi su internet possono essere realizzati secondo due modalità differenti: con l'architettura client-server o con architettura peer-to-peer. In seguito sono state sviluppate anche varie forme ibride.

L'architettura client-server prevedere la presenza di un client che richiede ad un server l'accesso a servizi o applicazioni, i server per l'appunto soddisfano queste richieste. Architetture di tipo client-server sono usate ad esempio per le email, per i servizi DNS, per il WWW.

L'architettura peer-to-peer invece non prevere la distinzione netta dei ruoli, tutti gli host sono contemporaneamente client e server. Ogni host cerca di soddisfare le richieste che riceve e agisce da client quando richiede servizi agli altri host. Classici esempi di servizi P2P sono i servizi di condivisione file come Freenet.

Cybersecurity

Introduzione

Chiaramente non c'è bisogno di spiegare perché la cybersecurity è un argomento fondamentale per un corso di reti di calcolatori. Nel corso della trattazione degli argomenti di cybersecurity si farà spesso ricorso ad un esempio classico della sicurezza informatica noto come Alice e Bob.

Alice e Bob sono due amici (o amanti, come preferite, sono chiamati così perché le iniziali siano A e B) che desiderano comunicare informazioni riservate fra di loro. Solitamente nella comunicazione fra Alice e Bob interviene un terzo incomodo: Trudy (da intruder), che di volta in volta prova a sottrarre o alterare i dati o inviarne di falsi ad Alice o Bob, i quali devono farsi più furbi di Trudy per comunicare con un ragionevole livello di sicurezza.

Chiaramente non è detto che Alice e Bob siano necessariamente persone reali, potrebbero essere persone come server di online banking come anche router che si scambiano informazioni, si tratta solo di un esempio.

Ora che abbiamo ben chiari i protagonisti del prossimo malloppone di appunti, possiamo andare avanti.

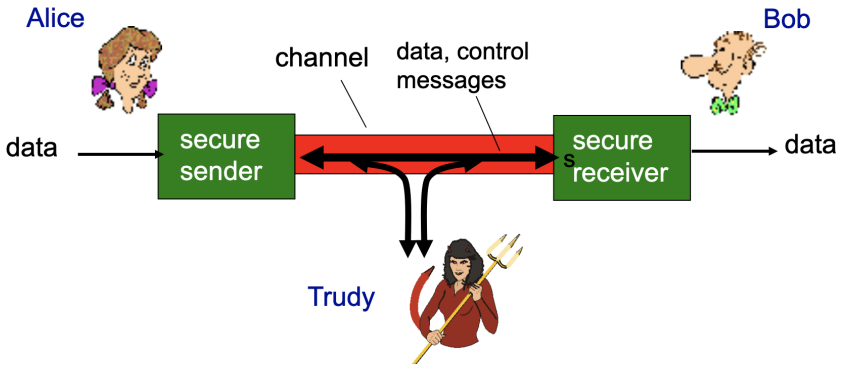


Figure 16: Schema (molto brutto) della comunicazione fra Alice e Bob

Alcune note "linguistiche"

Quando si tratta di crittografia bisogna avere presenti alcune abbreviazioni e nomenclature usate per semplificare la trattazione.

- m indica il messaggio **in chiaro** (plaintext message);
- $K_A(m)$ indica il messaggio cifrato con la chiave K_A (la chiave di Alice);
- $m = K_B(K_A(m))$, ovvero la chiave di Bob è l'inversa di quella di Alice.

Crittografia simmetrica

La crittografia simmetrica è il modo più semplice per proteggere una comunicazione: Alice e Bob si accordano su una chiave comune K_S da usare per cifrare e decifrare i messaggi.

Un banale esempio di scambio di messaggi cifrato con crittografia simmetrica è il Cifrario di Cesare

Il problema di questa tecnica è che Alice e Bob devono accordarsi sulla chiave da usare. Se si incontrano dal vivo e si scambiano la chiave certamente la connessione sarà cifrata, ma questa modalità non può avvenire sempre: i server di due banche non possono "incontrarsi dal vivo" per concordare una chiave.

> La crittografia simmetrica non sarà utile per scambiare informazioni su internet, ma è utilissima per proteggere ad esempio i nostri hard disk e le nostre chiavi USB, dato che in questo caso solo a noi serve sapere la chiave.

Comunicare la chiave in chiaro all'inizio della comunicazione è inutile e dannoso in quanto all'attaccante basta intercettare la chiave e poi tutti i messaggi saranno come fossero plain-text.

> Nel momento in cui la chiave è comunicata in plain-text non ha più neanche senso tentare di comunicare informazioni private, è come se fossero mandate tutte in plain-text.

Data Encryption Standard

Il DES è uno dei primi algoritmi di crittografia simmetrica ideati. Esso fa uso di chiavi da 56 bit.

Ad oggi può essere bypassato con un attacco brute-force in meno di un giorno.

Advances Encryption Standard

AES è il successore di DES, che ha sostituito a partire dal 2001. È sempre un algoritmo per la crittografia simmetrica; rispetto a DES è molto più sicuro: là dove per bypassare una cifratura in DES con bruteforce basti 1 secondo, per bypassare una cifratura AES che usi la stessa chiave servirebbero 149 bilioni di anni.

Crittografia a chiave pubblica

Gli algoritmi RSA (Rivest–Shamir–Adleman, dal nome dei ricercatori che lo hanno elaborato) è un algoritmo a chiave pubblica/privata che risolve il problema della crittografia simmetrica: non c'è più bisogno per i comunicanti di scambiarsi la chiave segreta, il mittente userà la chiave pubblica del destinatario per crittare il messaggio e il ricevente a sua volta lo decritterà con la propria chiave privata.

Il meccanismo si basa su complessi algoritmi di aritmetica modulare, il risultato però permette ad Alice di usare la chiave pubblica di Bob (che sarà disponibile in un server o database convidiso con tutti) per cifrare il messaggio che poi solo Bob, con la sua chiave privata, sarà in grado di decifrare.

Per maggiori dettagli sul funzionamento matematico dell'algoritmo rimando alla pagina wikipedia.

Ciò che rende RSA sicuro è che, anche conoscendo la chiave pubblica di Bob, abbiamo bisogno di fattorizzarla per trovare la chiave privata d e fattorizzare un numero grande non è un compito semplice.

Il compromesso di questo algoritmo è che è 100 volte più costoso (computazionalmente) di DES, per questo motivo solitamente si usa RSA solo all'inizio di una comunicazione per accordarsi su una chiave simmetrica, in modo da proseguire poi con algoritmi meno esosi di risorse ma (adesso) altrettanto sicuri.

Autenticazione

Risolto il problema di cifrare la comunicazione se ne pone un altro: l'autenticazione, cioè come fa Bob ad essere sicuro di star parlando con Alice.

Attacco di tipo Playback

Supponiamo che Bob chieda ad Alice di dimostrare di essere veramente lei inviando un messaggio all'inizio della comunicazione contenente una parola d'ordine che poi alice userà in seguito ogni volta che comunicherà con Bob.

In uno scenario del genere a Trudy basta osservare la comunicazione fra Alice e Bob e copiare la parte iniziale, in cui Alice comunica la parola d'ordine, e inviarla a Bob ogni volta che vuole aprire una comunicazione con lui spacciandosi per Alice.

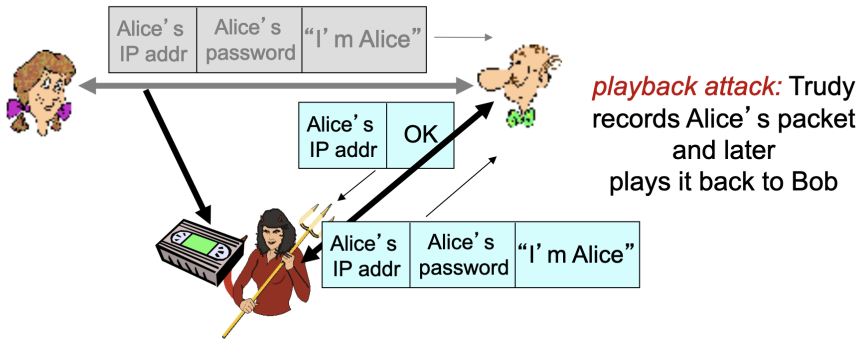


Figure 17: Playback attack

La soluzione per mitigare questo tipo di attacco Bob può mandare ad Alice un numero R noto come numero *once-in-a-lifetime*, che può essere usato per l'appunto una volta soltanto, che Alice deve restituire cifrato a Bob. Se in futuro Trudy dovesse tentare un playback attack Bob riceverebbe di nuovo la chiave mandata in passato e si renderebbe conto di non star parlando con Alice.

Digital Signature

Le firme digitali sono tecniche crittografiche analoghe alle firme "analogiche". Bob usa la sua firma digitale per certificare di essere il creatore/proprietario di un documento che manda ad Alice. Questa firma è verificabile e non-forgiabile: Alice può dimostrare che la firma appartiene a Bob e nessun altro.

Una volta che Alice riceve il messaggio firmato da Bob con la sua (di Bob) chiave privata, Alice può usare la chaive pubblica di Bob per decifrare il messaggio (un po' l'opposto della RSA concettualmente) e verificare che provenga effettivamente da Bob (o da qualcuno con la sua chiave privata).

Attacco di tipo Man in the Middle

Trudy si è accorta che non riesce più ad origliare le conversazioni di Alice e Bob e decide di passare ad un attacco di tipo **man-in-the-middle**.

In un attacco di tipo man-in-the-middle, Trudy intercetta le comunicazioni di Alice e Bob e consegna ad ognuno di loro delle chiavi pubbliche che in realtà sono di Trudy, in questo modo potrà intercettare tutti i messaggi da una parte e dall'altra e farne quello che vuole: per restare trasparente e non rendere nota l'infrazione a Trudy basterà inoltrare i messaggi da un capo all'altro della comunicazione.

Il rimedio ai tipi di attacco man-in-the-middle è l'uso di chiavi pubbliche *certificate*.

Public-key certification

Per evitare attacchi di tipo man-in-the-middle si fa ricorso ad enti terzi che certifichino l'autenticità delle chiavi pubbliche che Alice e Bob si scambiano.

Anziché scambiarsi direttamente le chiavi, sia Alice che Bob registrano la propria chiave presso una **Certificate Authority** (CA) e, nel momento in cui vorranno comunicare fra loro, richiederanno alle rispettive CA la chiave dell'altro.

A questo punto Trudy non potrà più spacciarsi per Alice o Bob in quanto la CA garantisce solo le vere chiavi di Alice e Bob.

Trudy però non vuole lasciare e decide di raddoppiare la posta: mette sù un man-in-the-middle attack in cui Trudy questa volta impersona anche le CA di Alice e Bob.

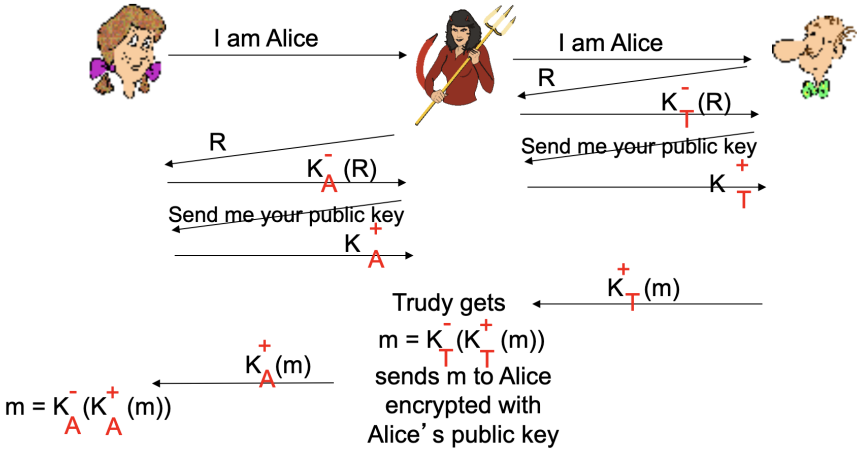


Figure 18: Man in the middle attack

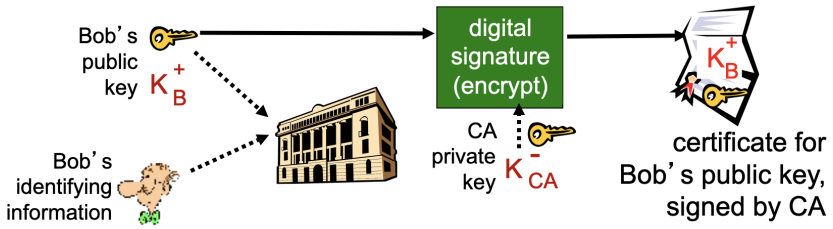


Figure 19: Certificate Authority

Per mitigare questa casistica le CA sono a loro volta certificate da altre CA, creando un network che di fatto impone a Trudy di perpetrare molti man-in-the-middle attacks per riuscire ad entrare nella conversazione fra Alice e Bob. Non è impossibile ma è certamente molto molto difficile.

E-mail sicure

Supponiamo che Alice voglia mandare una mail confidenziale a Bob. Come abbiamo visto, i passaggi che Alice dovrebbe seguire sono:

- Generare delle chiavi simmetriche casuali K_S;
- Cifrare i messaggi con le chiavi K_S;
- Cifrare K_S con la chiave pubblica di Bob K_B;
- Inviare a Bob sia il messaggio cifrato $K_S(m)$ che la chiave cifrate $K_B(K_S)$.

A questo punto Bob con la sua chiave privata può decifrare la chiave simmetrica e usarla a sua volta per decifrare il messaggio mandato da Alice.

Come abbiamo visto la cifratura non basta, Alice quindi vuole anche firmare il messaggio in modo che Bob sappia che proviene da lei e nessun altro. In sostanza Alice vuole che la comunicazione abbia le proprietà di: security, sender authentication, message integrity.

Per ottenere ciò Alice userà quindi 3 chiavi: la chiave simmetrica K_S per cifrare il messaggio, la chiave pubblica di Bob K_B per cifrare la chiave simmetrica e infine la propria chiave privata K_A per firmare il messaggio, usando quindi tutte le tecniche che abbiamo visto prima.

Rendere sicure le connessioni TCP: SSL

Per garantire integrità, sicurezza e autenticazione agli applicativi web che fanno uso del protocollo TCP la tecnologia più usata è **SSL (Secure Socket Layer)**.

SSL è disponibile per tutte le applicazioni che fanno uso di TCP: si pone fra l'applicazione ed il protocollo TCP e fornisce le API per rendere sicura una connessione.

SSL utilizza 4 chiavi per realizzare una comunicazione: la chiave di cifratura del client, la chiave **MAC (message authentication code)** del client, la chiave di cifratura del server e la chiave MAC del server. la chiave MAC viene usata per l'appunto per autenticare i messaggi scambiati.

La tecnologia SSL prevede un **Handshake** per iniziare la conversazioen.

Durante l'handshake si autentica il server, viene negoziato un accordo sull'algoritmo di cifratura da usare, si generano le chiavi e (opzionale) si autentica il client.

Nello specifico il client fornisce al server una lista di algoritmi di cifratura che può usare e il server restituirà al client l'algoritmo scelto e il certificato. Il client quindi verifica il certificato del server, prende la chiave pubblica del server e cifra il pre_master_secret che poi invia al server. A questo punto sia client che server useranno il pre_master_secret per generare le chiavi di cifratura e MAC ed entrambi aggiungeranno il MAC per tutti i successivi messaggi dela fare handshake.

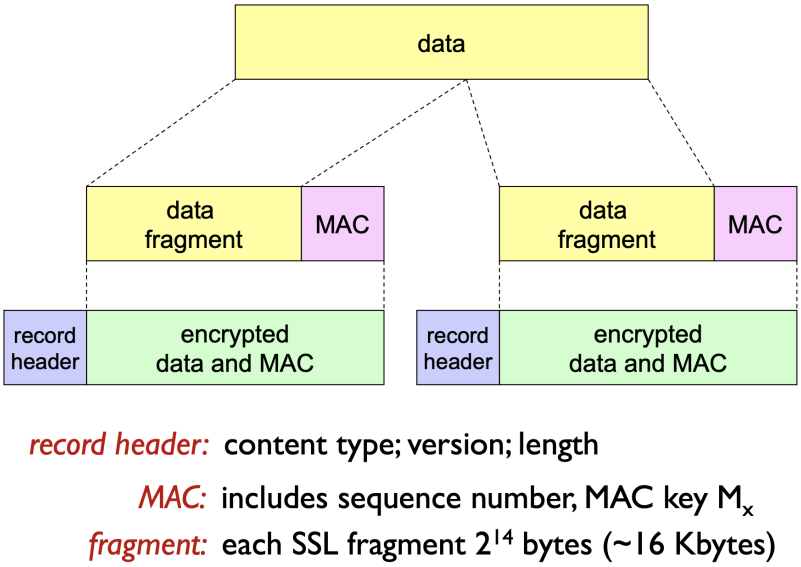


Figure 20: Schema SSL

Rendere sicuro il livello rete: IPsec

Rendere sicuro il livello network ha senso per proteggere tutta una serie di dati: dalle pagine web alle email, ai messaggi ICMP.

Fra le varie tecniche per proteggere la comunicazione a livello rete ci sono: VPN e IPsec

Le VPN

Sono usate (ad esempio) dalle imprese e dalle istituzioni per fornire una rete protetta ai dipendenti. Sono costose da realizzare in quanto bisogna predisporre dei router appositi e una infrastruttura DNS.

In pratica le VPN permettono di gestire il traffico 'inner-office' usando la rete internet pubblica, in quanto tutte le comunicazioni sono cifrate prima di accedere

ad internet e sono logicamente separate dall'altro traffico.

IPsec

I servizi IPsec forniscono le solite proprietà di integrità dei dati, autenticazione, protezione dai playback attack e confidenzialità della conversazione.

IPsec è fornita da due protocolli: AH e ESP.

- **Authentication Header (AH)** fornisce autenticazione della provenienza e integrità dei dati ma non la confidenzialità
- **Encapsulation Security Protocol (ESP)** fornisce anche la confidenzialità ed infatti è più usato di AH.

Security Associations (SA)

Vengono realizzate all'inizio della comunicazione fra il mittente ed il destinatario, entrambi conservano informazioni sullo stato della SA, in questo modo si passa da un protocollo connectionless, IP, ad uno connection-oriented, IPsec.

Le SA sono usate assieme ad esempio alle VPN per stabilire una connessione sicura fra due entità (ad esempio un computer di un azienda che acceda dall'esterno della rete aziendale), in quanto fra le altre cose permettono di conservare le informazioni sul tipo di algoritmo di cifratura accordato, sugli identificatori usati e sulle chiavi di autenticazione e cifratura.

Solitamente gli endpoint conservano le informazioni in dei database chiamati SAD.

Internet Key Exchange (IKE)

IKE è un servizio che si usa per lo scambio di chiavi di cifratura/autenticazione quando non è possibile eseguire il processo a mano (ad esempio per gestire VPN con centinaia di endpoints).

IKE gestisce l'autenticazione tramite o chiavi condivise a monte (PSK, pre shared keys) o tramite il meccanismo di chiave pubblica e certificazione.

IKE opera in due fasi:

1. Nella prima fase viene stabilita la SA bi-direzionale, conosciuta come ISAKMP security association;
2. Nella seconda fase ISAKMP viene usata per negoziare in modo sicuro le coppie di SA usate per IPsec.

Riassumendo

IPsec usa essenzialmente due tecnologie per gestire la comunicazione:

- IKE per stabilire gli algoritmi di cifratura, scambiare le chiavi e i numeri SPI;
- AH o ESP per fornire integrità, autenticazione e confidenzialità alla conversazione.

Gli Alice e Bob di una conversazione IPsec possono essere due end systems, due router/firewall oppure un end system ed un router/firewall.

Mettere al sicuro le reti locali (LAN)

Per evitare che una persona esterna manometta o osservi le comunicazioni all'interno di una rete locale sono stati nel tempo ideati diversi algoritmi.

WEP

L'algoritmo WEP è pensato per utilizzare un sistema di cifratura simmetrica, per essere self-synchronizing (ovvero ogni pacchetto è cifrato autonomamente senza per questo rischiare di non accorgersi di aver perso un pacchetto) e per essere efficiente ed implementabile sia tramite hardware che software.

Il protocollo prevede che il mittente:

- Calcoli l'ICV (Integrity Check Value), un hash di 4 byte, sui dati inviati;
- I due comunicanti hanno una chiave condivisa di 104 bit;
- Crei un vettore di inizializzazione (IV) di 24 bit e lo aggiunge alla chiave, ottenendo così una chiave di 128 bit;
- Aggiunga anche una chiave identificativa di 8 bit;
- La precedente chiave di 128 bit viene data in input ad un generatore di numeri random per ottenere il keystream;
- I dati del frame e la ICV vengono cifrati con l'algoritmo RC4 e poi il payload viene mandato al destinatario.

Il destinatario dal canto suo:

- Estrae il vettore di inizializzazione;
- Usa l'IV e la chiave condivisa come input per un generatore pseudo casuale per ottenere il keystream;
- Con uno XOR fra il keystream e i dati cifrati si decifrano i dati e il ICV;
- L'integrità dei dati viene verificata con l'ICV e la comunicazione può dirsi conclusa.

Perché non si usa più WEP

WEP non è più uno standard sicuro a causa di un security hole che è parte del protocollo stesso: dal momento che l'IV è composto di soli 24 bit e dato che ne viene utilizzato uno per ogni frame scambiato, prima o poi si riutilizzeranno gli stessi IV e, dato che l'IV viene inviato in chiaro, è facile per Trudy capire quando un IV viene riutilizzato.

Non appena Trudy identifica un IV riutilizzato può usare il dato per calcolare la chiave condivisa. Facendo questa operazione per tutti gli IV riutilizzati che trova Trudy è in grado di risalire a tutte le chiavi condivise e, la volta successiva, intercettare e decifrare i pacchetti.

Ormai è sconsigliato utilizzare WEP, lo standard di riferimento attuale è WPA2.

I Firewall

I firewall isolano una rete da internet permettendo l'ingresso solo di pacchetti selezionati. Sono molto utili per prevenire gli attacchi DoS, per impedire l'accesso e la modifica illegali di dati interni, permettere solo alle entità autorizzate di accedere al network.

I firewall possono essere di tre tipi:

- **Stateless packet filtering**

  I dispositivi interni sono connessi ad internet mediante il firewall del router, il quale filtra ogni pacchetto decidendo se farlo passare o meno (la decisione viene presa in base agli indirizzi IP, alle porte TCP o UDP e ai messaggi ICMP contenuti nei pacchetti);

  Esempio: il router blocca tutti i segmenti TCP interni contenenti ACK=0, con il risultato che nessun computer esterno possa connettersi come client ad uno interno, ma tutti i computer interni possono connettersi a server esterni;

- **Stateful packet filtering**

  A differenza dell'approccio stateless, in questo tipo di firewall viene tracciato lo status di ogni connessione TCP per capire quali pacchetti in transito abbiano senso (se una connessione è stata buttata formalmente giù non ha senso far transitare altri pacchetti in seguito);

- **Application gateways**

  Questo tipo di firewall filtra i pacchetti basandosi sui dati delle applicazioni e sui campi IP/TCP/UDP.

I firewall però non sono infallibili: sono suscettibili ad attacchi di IP spoofing (non hanno modo di verificare che l'IP che proclama il pacchetto sia effettivamente il suo). Inoltre non sono strumenti senza compromessi: per godere del livello di protezione garantito dai firewall bisogna rinunciare ad un certo grado di comunicazione con il mondo esterno.