Appunti di Reti di Calcolatori

Simone Ianniciello

A.A. 2020/2021

Contents

Chapter 1

Introduzione alle Reti

1.1 Introduzione

Rete Con rete si intende un'interconnessione di dispositivi in grado di scambiarsi informazioni. Le reti sono commposte da elementi quali:

- Sistemi terminali *(Host)*

  Macchine degli utenti finali

  Server *Fornitori di servizi*

- Switch: Dispositivi adibiti all'interconnessione locale di Host
- Router: Dispositivi di interconnessione di reti diverse
- Collegamenti: I mezzi tramite i quali vengono trasferite le informazioni

  Cavi in rame

  Fibra ottica

  Onde radio *(WiFi)*

1.2 Tipi di rete

LAN Con *LocalAreaNetwork* o *Rete Locale* si intende un insieme di Host appartenenti allo stesso ente *(Organizzazione, Casa, Scuola)* in grado di comunicare.
Le LAN possono essere:

- a cavo condiviso: Tutti gli host condividono lo stesso cavo per comunicare. Questo sistema non e' piu' usato perche' poco efficente.
- con switch: Tutti gli host sono collegati a uno switch che instrada le informazioni nella direzione desiderata. Questo sistema e' molto piu' efficiente perche' le macchine non hanno bisogno di monopolizzare la rete.

WAN Per *WideAreaNetwork* o *Rete Geografica* si intende una rete formata da piu' LAN e/o singoli host separati da grandi distanze. Essa viene gestita da da un operatore che fornisce il servizio di interconnessione ai clienti.
Le WAN si distinguono in:

- WAN punto-punto
- WAN a commutazione

Una applicazione tipica sono reti locali appartenenti ad un'azienda interconnesse tramite WAN p-p

1.3 Tecniche di commutazione

I due sistemi principali per determinare il percorso tra due host e dedicargli le risorse sono:

- Circuit-switched network *(Commutazione di circuto)*
- Packet-switched network *(Commutazione di pacchetto)*

Commutazione di circuito Per la commutazione di circuito si procede instaurando un cammino dedicato tra i due host: vengono assegnate le risorse necessarie alla comunicazione e sono garantite per l'intera durata della connessione. Cio' significa che una volta instaurata la connessione, essa non verra' disturbata in alcun modo.
I principali problemi di questa tecnica sono pero' il tempo di instaurazione della connessione *(risorse non disponibili)*, e il non sfruttamento delle risiorse disponibili durante i *silenzi* nella comunicazione.

Commutazione di pacchetto Nelle connessioni a commutazione di pacchetto il flusso di dati viene diviso in pacchetti ed essi vengono *spediti sulla rete* sul percorso prescelto. Le risorse vengono quindi utilizzate solo se necessarie e possono essere condivise da pacchetti provenienti da host differenti.

Ogni nodo della rete si occupa di ricevere e riservire i pacchetti che gli arrivano. Per fare cio' il commutatore, dopo aver ricevuto un pacchetto, lo mette in una coda di tipo FIFO; quando e' pronto a ritrasmettere preleva il primo pacchetto dalla coda. Cio' porta a dei ritardi (Il commutatore deve ricevere l'intero pacchetto per reinviarlo, i pacchetti potrebbero dover *aspettare* in coda) e a delle perdite di pacchetti (coda piena).
Questo metodo si chiama **Store and Forward**.

1.4 Internet

Con **i**nternet si intende un sistema formato da due o piu' reti comunicanti. L'**I**nternet e' l'insieme di reti piu' comune. Ogni rete che intende aggiungersi ad essa deve seguire L'Internet Protocol (IP) e rispettare certe convenzioni.

L'infrastruttura di Internet fornisce servizi di comunicazione alle applicazioni

- Senza connesione **(UDP)**
- Orientati alla connessione **(TCP)**

Sono stati definiti dei **protocolli** di comunicazione per le applicazioni piu' comuni di Internet *(TCP, IP, HTTP, FTP...)*

Ci sono delle organizzazioni adibite alla definizione degli standard di internet:

- **IETF** Internet Engineering Task Force
  - Studia e sviluppa i protocolli in uso su internet.
  - Pubblica i documenti ufficiali che li descrivono sotto forma di RFC/STD *(Request For Comments, STanDards)*
- **ICANN** Internet Corporation for Assigned Names and Numbers
  - Coordina i DNS
  - Assegna i gruppi di indirizzi di rete
  - Ha funzioni di controllo semplice dello sviluppo di Internet
- **W3C** World Wide Web Consortium
  - Sviluppa di standard aperti *(HTML, XML...)*

1.4.1 Strati della rete

Le reti degli host si collegano a Internet tramite gli ISPs *Internet Service Provider*. I livelli della rete sono:

Livello 3 ISP di accesso: Sono quelli a cui si connettono comunemente le reti locali.

Livello 2 ISP regionali: Sono dei collegamenti intermedi che uniscono tutti gli ISP di livello 2 in una zona geografica

Livello 1 Dorsali: Esse sono la parte piu' *alta* di Internet, tutti gli altri ISP si connettono ad esse. *(ne esistono circa 11)*

1.4.2 Peering point

Sono accordi tra due ISP che gli permettono di ricevere e riinoltrare il traffico da uno all'altro: per fare cio' esistono gli IXP *(Internet eXchange Point)* ovvero sistemi, anche gestiti da aziende di terzi, che effettuano il peering.

1.4.3 Reti di acesso

Il collegamento tra l'utente e Internet e' detto **rete di accesso**.

- Accesso via rete telefonica
  - dial-up
  - Digital Subscriber Line **(DSL)**
  - Fibra ottica
- Accesso tramite reti wireless
  - 3G, 4G, 5G
- Collegamento diretto
  - Collegamenti WAN dedicati per aziende, universita'...

1.5 Metriche di riferimento

Larghezza di banda (Bandwidth) Larghezza in Hertz dell'intervallo di frequenze utilizzato per la trasmissione.

Velocita' di trasmissione (bitrate) Quantita' di dati trasmissibili nell'unita' di tempo (bps).

Troughput Quantita' di dati trasmissibili da un nodo A ad un nodo B in una unita' di tempo. Tiene di conto anche di perdite sulla rete, protocolli, ecc...

Latenza Tempo che intercorre tra l'invio e la ricezione del primo bit di un messaggio.

$$\text{Latenza} = \text{elaborazione} + \text{accodamento} + \text{trasmissione} + \text{propagazione}$$

1.5.1 Ritardi

Ritardo di elaborazione Causato dal sistema di controllo degli errori e dal sistema che determina il canale di uscita.

Ritardo di accodamento Tempo tra l'inserimento nella coda di trasmissione e la ritrasmissione del pacchetto.

Ritardo di trasmissione Tempo impiegato per trasmettere un pacchetto sul mezzo di trasmissione.

Misurato in PacketLenght / BitRate

Ritardo di propagazione Tempo che inpiega un bit ad essere propagato da un nodo all'altro.

Misurato in LinkLenght / PropSpeed $(3 - 2 * 10^{-}8)$

Ritardo end-to-end Ritardo cumulato tra tutti i nodi in una trasmissione. E' pari alla sommatoria dei ritardi tra i vari nodi del collegamento.

1.5.2 Prodotto rate-ritardo

Numero massimo di bit che possono *essere contenuti* nel mezzo trasmissivo.

1.6 Modelli stratificati

Il modello statificato permette di scomporre un sistema complesso in piu' sistemi piu' facili da implementare e comprendere. La modularizzazione dei livelli permette di dividere l'interfaccia dall'implementazione di un servizio. Percio' dall'esterno ogni modulo e' visto come un'interfaccia che: accetta determinati parametri in ingresso, esegue le sue mansioni, e ritorna una risposta. Quindi l'implementazione puo' essere cambiata senza che il resto del sistema *ne venga a conoscenza*

1.7 OSI RM **(Open System Interconnection Reference Model)**

Nel 1976 sono iniziati i lavori per definire uno standard aperto per i protocolli di Internet. La ISO ha da prima pubblicato questi standard sotto forma del OSI-RM, che poi e' diventato uno standar internazionale nel 1983 (ISO 7498).

Il modello ISO/OSI prevede la statificazione del protocollo di telecomunicazione.

Gli strati OSI sono:

7 Applicazione - elaborazione dati

6 Presentazione - unificazione dati

5 Sessione - controllo del dialogo

4 Trasporto - trasferimento dati tra hosts

3 Rete - instradamento del traffico

2 Datalink - consegne trame sul link

1 Fisico - trasmette un flusso di bit

Le informazioni si propagano dal livello piu' alto (applicazione) fino al piu' basso per poi passare sul mezzo trasmissivo e risalire i livelli fino alla destinazione.

Ogni livello aggiunge all'informazione del livello superione una propria sezione informativa (header / trailer) Questo processo di incapsulamento delle informazioni e' reversibile percio' ogni livello e' in grado di estrarre i dati degli strati superiori.

1.8 Stack Protocollare TCP-IP

E' la famiglia di protocolli attualmente utilizzata in Internet. E' definita attualmente da cinque livelli:

Applicazione Applicazioni di rete, collegamento logico end-to-end, scambio di messaggi tra processi. *ftp, smtp, http*

Trasporto Trasferimento dati end-to-end. *tcp, udp*

rete Instradamento dei datagrammi. *IP, ICMP*

Link Trasferimento dei dati in frame tra elementi vicini. *ppp, ethernet, ...*

Fisico Trasferimento di bit di un frame sul mezzo trasmissivo.

Chapter 2

Lo Strato Applicativo: URI-HTTP

2.1 Applicazioni di rete

Sono applicazioni formate da processi distibuiti comunicanti; Cio' significa che i processi possono essere eseguiti su host diversi sulla rete. Sullo stesso host piu' processi possono comunicare anche attraverso la comunicazione inter-processo definita dal sistema operativo (quindi esterna alla rete). Due processi su macchine diverse comunicano tramite la rete con dei **messaggi**. Grazie alla statificazione dei livelli, i livelli applicazione comunicano tra loro come se esistesse un collegamento diretto tra i due.

2.2 Protocollo dello Strato di Applicazione

Esso definisce:

- i tipi di messaggi scambiati
- la sintassi dei messaggi *(i campi)*
- la semantica dei campi *(il significato)*
- le regole per l'invio dei messaggi *(quando e come)*

2.3 Paradigmi

Client-Server I server *(pochi ma potenti)* offrono un servizio e sono sempre in attesa di richieste.

Peer-tp-Peer (p2p) I peer offrono e richiedono servizi contemporaneamente.

Misto Programmi che utilizzano entrambi i paradigmi per utilizzi diversi *(Skype, . . .)*

2.4 Application Programming Interface (API)

E' un insieme di regole da rispettare per utilizzare le risorse di un servizio. Per esempio, se un processo vuole inviare un messaggio sulla rete, utilizza delle chiamate al sistema operativo che comunica con gli strati inferiori dello stack TCP-IP attraverso l'Interfaccia Socket.

L'Interfaccia Socket E' un'API che collega gli strati applicazione e trasporto. Grazie ad essa un programmatore che intende sviluppare un'applicazione con accesso a Internet non deve preoccuparsi di tutte le *formalita'* per la connessione. Per identificare un processo sulla rete si utilizza una coppia:

<Indirizzo IP (32b) + numero di porta (16b)>

2.5 Uso dei servizi di trasporto

Nel livello di trasporto sono previsti dule protocolli principali:

TCP connection-oriented: client e server devono *salutarsi* prima di iniziare a comunicare; c'e' controllo degli errori sui messaggi.

UDP connection-less: **CHAOS!** Non c'e' nessun controllo, il mittente tira i messaggi sulla rete e suppone che il destinatario lo riceva.

UDP e' utile se si ha bisogno di un alto troughput e/o un basso ritardo, ma solo se non ci interessa un trasferimento affidabile al 100% dei dati *(streaming, videogiochi, . . .).*

2.6 Web e HTTP

2.6.1 Il Web

Una pagina Web consiste di:

- Un file HTML
- Diversi oggetti referenziati indirizzabili tramite una URL (Uniform Reference Locator)

2.6.2 Uniform Resource Identifier

E' una forma generale per identificare una risorsa sulla rete. Ci sono due tipi di URI:

Uniform Resource Locator (URL) Risorse identificate tramite il meccanismo di accesso *(HTTP, FTP, ...)*

Uniform Resource Name (URN) Identificatore globalmente unico, anche se la risorsa diventa non disponibile.

2.6.3 Sintassi URL

scheme://host:port/path

- scheme: protocollo di accesso
- host: nome di dominio o indirizzo IP
- port: numero di porta del servizio richiesto
- path: identifica la risorsa nel contesto sopra-descritto

Nelle URL HTTP si aggiunge anche un parametro *?query* dopo *path*. Questa stringa viene interpretata dal server e serve a passare informazioni aggiuntive nella richiesta.

2.6.4 URI Assolute e Relative

URI assoluta identifica una risorsa indipendendentemente dal contesto.

URI relativa identifica una risorsa in relazione ad un'altra URL e viene interpretata dal client prima di mandare la richiesta.

2.7 HyperText Transfer Protocol (HTTP)

Protocollo stateless pubblicato nel 1990 e utilizzato fino ad ora nel World Wide Web. Una caratteristica fondamentale dell'HTTP e' la tipizzazione e negoziazione della rappresentazione dei dati percio' non e' limitato all'ipertesto.

Il client stabilisce una connessione e invia richieste (request) tramite HTTP al server, il quale invia messaggi di risposta (response) Usa TCP dato che il trasferimento deve essere senza perdite e che i ritardi non sono *un problema*

Dalla versione 1.1 la connessione client-server, a meno che diversamente espresso dal client, e' persistente; percio' il client puo' assumere che il server manterra' la connessione aperta. Cio' porta a minor utilizzo di CPU e ritardi minori dato che non c'e' bisogno di riaprire una nuova connessione per ogni richiesta. Inoltre si ha un mignor controllo di congestione.

La connessione viene chiusa dal server solo quando esplicitamente richiesto dal client nell'header del messaggio, o se non riceve piu' richieste (time out)

2.7.1 HTTP Request

La richesta e' formata da:

Request-Line
*(general-header
| request-header
| entity-header)
CRLF
[message-body]

2.7.2 HTTP Response

La risposta e' formata da:

Status-Line
*(general-header
| response-header
| entity-header)
CRLF
[message-body]

Request-Line

Method SP
Request-URI SP
HTTP-Version CRLF

Status-Line

HTTP-Version SP
Status-Code SP
Reason-Phrase CRLF

General-header Relativi alla connessione.

Entity-header Relativi all'entita' trasmessa.

Request-header Relativi alla richiesta.

Response-header Nel messaggio di risposta.

2.7.3 Content Negotiation

Le risorse possono essere disponibili in piu' rappresentazioni; Il client puo' richiederne una in particolare tramite richieste su Request e Entity headers.

2.7.4 Request methods

OPTIONS Il server ritorna solo le opzioni di comunicazione associate ad una URL (capacita', metodi esposti, ...).

GET Richiede il trasferimento della risorsa indicata. Sono possibili i **conditional-get** (If-Mofidied-Since, If-Match, ...).

HEAD Simile alla GET ma il server non trasferisce il message-body. Utile per controllare lo stato dei documenti (validita', modifiche, ...).

POST Serve per inviare al server informazioni inserite nel body del messaggio.

PUT Chiede al server di creare/modificare una risorsa (Recuperabile poi tramite GET).

DELETE Il client chiede di eliminare una risorsa identidficata dalla Request-URI.

2.7.5 Caching

E' possibile memorizzare copie temporanee di risorse Web e ri-servirle al client senza contattare il server. Si possono avere:

User-Agent Cache Risorsa mantenuta dal browser.

Proxy Cache Il proxy mantiene una copia delle risorse richieste. Quando un browser richiede una risorsa cached essa gli viene riservita dal proxy senza interrogare nuovamente il server.

2.7.6 Cookies

Essendo stateless, non e' mantenere una memora su un'applicazione web (come ad esempio l'identita' di un client). Una soluzione e' utilizzare i cookies, cioe' id unici che il client presenta al server ogni volta che esso effettua una richiesta.

Alla prima connessione, il server invia la normale risposta HTTP + una line **Set-cookie: id**. Il client si salva il cookie e lo associa al server. Ogni successiva richiesta a quel server conterra' quel cookie.

2.8 Versioni HTTP

HTTP/1.1 Permette di instaurare piu' connessioni contemporaneamente perciò il client puo' inviare piu' richieste al server contemporaneamente (Pipelining).

Il server deve pero' rispondere nello stesso ordine in cui sono arrivate le richieste perciò, se una richiesta richiede piu' tempo per essere elaborata, tutte le risposte successive non possono essere inviate (HeadOfLineBlocking).

HTTP/2 Aggiunge caratteristiche allo standard HTTP come:

- Multiplexing delle richieste su un'unica connessione TCP attraverso la definizione di:

  Frame Un messaggio HTTP viene mappato da una sequenza di Frame (come Data Frame, Headers Frame, ...).

  Stream E' un flusso bidirezionale di frame all'interno di una connessione TCP; mediante l'astrazione di essi, e' possibile effettuare il multiplexing delle richieste (piu' stream su un'unica connessione). E' possibile anche associargli un peso (priorità), e una dipendenza verso altri stream.

- Compressione delle intestazioni.

- Server Push (permette al server di inviare risorse aggiuntive oltre a quella richiesta dal client).

HTTP/3 E' un'evoluzione di HTTP/2 che usa i servizi di QUIC, un protocollo di trasporto basato su UDP. QUIC aggiunge controllo del flusso e della congestione, e rilevazione dell'errore e delle perdite. Grazie all'astrazione dello stream a livello di trasporto, il rallentamento o la perdita di uno di essi non influisce sugli altri.

Chapter 3

Lo Strato Applicativo: Telnet

TErminaL NETwork e' un protocollo che permette l'uso di shell su macchine remote. Permette al client di effettuare una sessione di login al server, dopo la quale esso puo' accedere a comandi e programmi disponibili sulla macchina remota.

Dopo il login, vengono passate le battute dei tasti al server come standard-input e l'output viene riportato direttamente al client.

Il protocollo TELNET [RFC854] detta che:

- Il client stabilisce una connessione di tipo TCP (tipicamente sulla porta 23) con il server. la connessione persiste per tutta la durata della sessione di login.
- Vengono *"collegate"* le STDIN e STDOUT dei tue terminali.

Il server Telnet utilizza uno Pseudo Terminal Driver per eseguire i comandi.

3.1 Network Virtual Terminal: NVT

Telnet deve poter operare con il numero massimo di sistemi quindi deve poter lavorare con client su sistemi operativi diversi.

Per fare questo il client e il server passano attraverso un terminale virtuale in modo da avere una sola codifica ed entrambi effettuano una conversione dalla propria a quella del NVT. Sulla rete vengono quindi trasferiti i comandi con l'**NVT Character Set**.

3.2 NVT Character Set

I dati vengono trasferiti tramite 7-bit US-ASCII.

- Ogni carattere e' inviato come un byte con il primo bit settato a 0.
- Per le sequenze di comandi si imposta il bit piu' significativo a 1.
- I comandi come EOL iniziano con 0xFF (Interpret As Command **IET**)

Chapter 4

Lo Strato Applicativo: E-Mail

Permette il trasferimento di messaggi tra un mittente e un destinatario. Dato che il destinatario potrebbe non avere il client mail aperto in quel momento, il servizio di posta elettronica deve basarsi su componenti intermediari per mantenere il messaggio finché il destinatario non e' in grado di riceverlo.

4.1 Principio di funzionamento

- L'utente invia un messaggio.
- Il sistema ne mantiene una copia nell'area di accodamento *(spool)*, insieme a id mittente, id destinatario, id destinazione e tempo di deposito.
- Il client avvia il trasferimento alla macchina remota stabilendo una connessione TCP.
- Se la connessione viene aperta inizia il trasferimento del messaggio e, a trasferimento completato, cancella la copia locale.
- Altrimenti il processo viene ripetuto periodicamente scandendo i messaggi nella spool.
- Oltre un certo intervallo di tempo, se il messaggio non e' ancora stato consegnato, viene inviata una notifica al mittente.

4.2 Indirizzi email

Formato da

local-part @ domain-name

local-part Specifica la cassetta a cui consegnare il messaggio.

domain-name Specifica il mail server.

4.3 Simple Mail Transfer Protocol (SMTP)

L'obbiettivo di SMTP e' il trasferimento affidabile ed efficiente di mail. Il protocollo e' indipendente dal sistema di trasmissione usato. Una caratteristica e' la capacita' di trasportare mail attraverso piu' reti; Un messaggio puo' quindi passare da server intermedi prima di arrivare al destinatario.

Modello: Un client apre un canale bidirezionale con il server SMTP. Inizia poi il trasferimento del messaggio (se non va a buon fine, comunica l'insuccesso). Per trovare il server SMTP, il client risolve il dominio attraverso un DNS.

Protocollo: Solitamente SMTP si basa sulla porta 25 TCP. Le fasi del trasferimento sono:

- handshaking
- trasferimento del messaggio
- chiusura della connessione

L'interazione e' di tipo **comando/risposta**

Handshaking Il client stabilisce la connessione e aspetta la risposta 220 READY FOR MAIL. Il client risponde con HELO ed il server risponde identificandosi.

Comandi SMTP:

- HELO *client-id*
- MAIL FROM: *reverse-path* [SP *mail-parameters*] <CRLF>
- RCPT TO: *forward-path* [SP *rcpt-parameters*] <CRLF>
- DATA
- QUIT

<CRLF>.<CRLF> determina la fine di un messaggio.

Formato dei messaggi

- Linee di intestazione (header)
  - To:
  - From:
  - Subject:
  - ...
- body
  - Soli caratteri ASCII a 7 bit.

Per ovviare al body solo testuale e' stato creato lo standard MIME (Multipurpose Internet Mail Extension). Questo standard *vive sopra* alla RFC 822 originale ma da nuove regole di interpretazione del body. Cio' ha permesso di mantenere i mail-server esistenti e cambiare solo gli user-agents.

Linee di intestazione aggiuntive per il MIME:

- MIME-Version:
- Content-Transfer-Encoding:
- Content-Type:

Content-Type: type/subtype;

- text
  - plain
  - html
- image
  - jpeg
  - gif
- audio
  - basics
  - 32kadpcm
- video
  - mpeg
- application
  - msword
  - ocet-stream

Esiste anche il tipo multipart: esso permette di inviare messaggi con piu' tipi separati da un boundary=

4.4 Sistemi pull per le mail POP3, IMAP e HTML

4.4.1 Post Office Protocol

Lo user-agent apre una connessione TCP (porta 110) verso il server di posta. Comandi permessi:

- Autorizzazione
  - user:
  - pass:
- Transazione
  - list:
  - retr:
  - dele:
  - quit:
- Aggiornamento
  - Dopo `quit` il server cancella i messaggi marcati per la rimozione.

4.4.2 Internet Mail Access Protocol

E' piu' complesso di POP3 ma permette la manipolazione dei messaggi memorizzati e da la possibilita' di estrarre solo alcuni componenti dei messaggi.

4.4.3 HTTP

Hotmail, Gmail, ...

Chapter 5

Lo Strato Applicativo: DNS

5.1 Introduzione

Ogni rete e' identificata sulla rete tramite un indirizzo IP. Il problema di essi e' che non sono facili da ricordare *e sono brutti*. E' quindi nato il bisogno di associare gli IP a dei nomi logici (e spesso mnemonici). Per associare gli IP ai nomi e' stato creato il DNS.

Il DNS adotta il paradigma client-server. Si affida al protocollo di trasporto sottostante per trasferire i messaggi. E' costituito da:

- Uno schema di assegnazione dei nomi
- Un database distribuito contenente le associazioni

$$nomedominio \implies IP$$

- Un protocollo per la distribuzione delle informazioni sui nomi tra i name server
  - Utilizza UDP (porta 53) [oppure TCP]

5.2 Servizi DNS

Risoluzione Traduzione $hostname \implies indirizzoIP$

Host aliasing Traduzione $nomi \implies nomecanonico/IP$

Mail server aliasing Stessa cosa degli host, permette tra le altre cose di usare nomi identici per mail e web server.

Distribuzione di carico Ad un hostname possono corrispondere piu' indirizzi IP; il DNS restituisce la lista di IP variandone l'ordinamento ogni volta.

5.3 Spazio dei nomi

Struttura "flat" Sequenza di caratteri senza alcuna ulteriore struttura (deprecato)

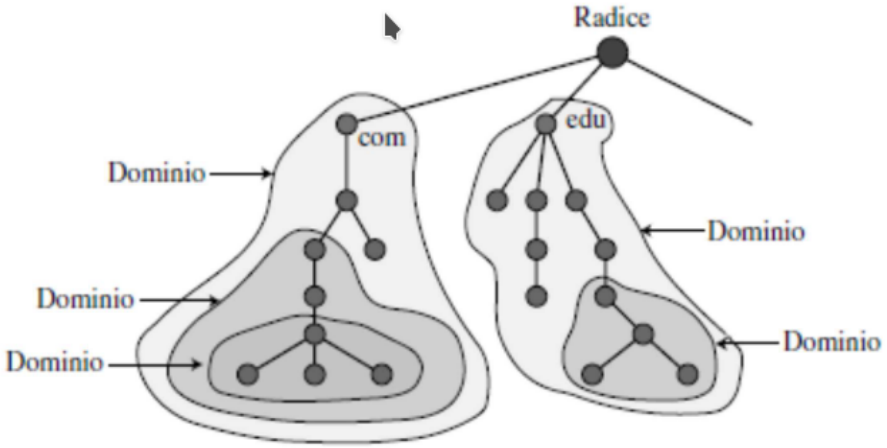
Struttura gerarchica Il nome e' costituito da diverse parti

(p.e. studenti.di.unipi.it)

L'assegnazione dei nomi e' delegabile al sistema decentralizzato.

5.3.1 Nomi di dominio

I nomi hanno una struttura ad albero. Ogni nodo e' individuato da un'etichetta.(la radice ha etichetta vuota)



La struttura gerarchica permette autonomia nella scelta dei nomi all'interno di un dominio.

5.3.2 Esempi di Top Level Domain (TLD)

com Organizzazioni commerciali

edu Istituti di istruzione

mil Gruppi militari

gov Istituzioni governative

net Centri di supporto alla rete

org Organizzazioni diverse dalle precedenti

it, uk, fr, us Schema geografico per nazioni

Mantenuti da IANA (Internet Assigned Numbers Authority)

5.4 Gerarchia dei Name Server

DNS Database distribuito implementato come gerarchia di piu' Name Server.

Name Server Gestisce le richieste. Ogni server immagazzina le informazioni relative alla propria zona inclusi i riferimenti ai server dei domini di livello inferiore.

5.4.1 Root Name Server

Restituisce le informazioni sui name server dei TLD. Ce ne sono centinaia in tutto il mondo.

5.4.2 TLD Server

Mantiene le informazioni dei nomi di dominio che appartengono a un TLD; Restituisce le informazioni sui Name Server di competenza dei sotto domini.

5.4.3 Authoritative Name Server

E' l'autorità per una certa zona. Puo' effettuare traduzioni nome $\Longrightarrow$ indirizzo o inoltrare la richiesta a altri Name Servers. Sono di due tipi:

Server primari Mantengono il file di zona.

Server secondari Ricevono il file di zona e offrono il servizio di traduzione.

5.4.4 Local Name Server

E' un Name Server mantenuto dagli ISP. Le query DNS vengono prima rivolte al server locale il quale, in caso non riesca a risolverle, le inoltra alla gerarchia DNS.

5.4.5 Query DNS

La query puo' essere:

Ricorsiva La richiesta fa il *giro completo* attraversando ogni name server fino a trovare il dominio che sta' cercando.

Iterativa Le risposte sono restituite direttamente al client insieme al riferimento al server da contattare.

Se un server supporta le query ricorsive, imposta il bit RA *(Recursion Available)* a true. Se anche il client imposta il bit BD *(Recursion Desired)* a true, viene effettuata la query in modalità ricorsiva.

5.4.6 DNS Caching

I name server mantengono una memoria delle associazioni gia' richieste, le quali vengono cancellate dopo un tempo di timeout.

5.5 Resource Records

RR format: **(name, value, type, ttl)**

TTL Tempo di vita della risorsa della cache.

Type A

Name hostname

Value IP

Type CNAME

Name hostname

Value nome canonico

Type NS

Name Nome di dominio

Value hostname dell'authoritative name server per quel dominio

Type MX

Name Nome di dominio

Value Nome canonico del server di posta elettronica associato

V. messaggi DNS

Chapter 6

Lo Strato Applicativo: File Transfer Protocol

6.1 Introduzione

Il File Transfer Protocol FTP e' un protocollo che vive sullo strato applicativo di reti TCP/IP per il trasferimento di file da/a un host remoto: Utilizza il protocollo client/server

Client Il lato che richiede il trasferimento.

Server Host remoto che fornisce il servizio.

Lo standard non prevede la modifica *"live"* dei file, ma soltanto la richiesta di copie e l'eventuale aggiornamento del file successivamente. Fornisce inoltre l'accesso interattivo alle directory nel filesystem remoto. Permette anche l'autenticazione degli utenti.

6.2 Modello

Dettato dalla RFC 959. Permette due tipi di connessione:

Control connection Scambio di comandi e risposte, segue il protocollo Telnet.

Data connection Scambio di dati che possono essere parte, l'intero, o un set di file.

Viene usato il protocollo TCP per il trasporto.

FTP e' un protocollo stateful: il server deve tener traccia dello stato dell'utente *(connessione di controllo associata ad un account, current directory, ...)*

6.2.1 Control Connection

E' una connessione persistente.

Il client contatta il server FTP (sulla porta 21 di default). Dopo che il server gli fornisce l'autorizzazione, il client puo' iniziare ad inviare comandi *(change dir, invio file, list files, ...)* sotto forma di caratteri codificati con standard NVT ASCII.

6.2.2 Data Connection

Quando viene richiesto un trasferimento di file tramite la connessione di controllo, il server apre una nuova connessione TCP (porta 20 di default) con il client, vi trasferisce i dati, e la richiude subito dopo.

6.2.3 Active / Passive Mode

6.2.4 Standard di comunicazione

Come per Telnet, anche per il trasferimento dati bisogna definire uno standard per la struttura dei file e delle directory. Il trasferimento viene preparato attraverso uno scambio di informazioni tramite la connessione di controllo.

6.2.5 Modalità di trasmissione

Stream mode I dati vengono inviati cone flusso continuo di bit allo strato TCP.

Block mode I dati vengono inviati a TCP suddivisi in blocchi. Ogni blocco e' preceduto da un header.

Compressed mode Al posto di inviare direttamente i file, vengono inviate delle versioni compresse.

6.3 Comandi di controllo

6.3.1 Client-to-Server

USER Username

PASS Password

LIST Elenca i file della directory corrente

NLST Elenca file e dirs nella directory corrente

RETR Recupera un file dalla current directory

STOR filename: Memorizza un file nell'host remoto

ABOR Interrompe tutto

PORT Indirizzo e numero di porta del client

SYST Restituisce il tipo di sistema

QUIT Chiude la connessione

6.3.2 Server-to-Client

6.4 Altre caratteristiche

6.4.1 Anonymous FTP

E' possibile attivare il supporto a connessioni senza autenticazione. Queste connessioni solitamente hanno accesso a una piccola parte del filesystem e permettono meno operazioni (es. no PUT)

6.4.2 Secure FTP with TLS

Descritto dalla RFC 2246, permette di implementare sicurezza e autenticazione attraverso il protocollo TLS

Chapter 7

Lo Strato di Trasporto: Introduzione

7.1 Obbiettivo

Realizza una comunicazione logica tra processi su terminali diversi. Cio' significa che essi comunicano come se ci fosse un collegamento diretto tra i due. Lo strato applicazione trasmette dati allo strato di trasporto, il quale poi comunica con gli strati inferiori per trasmetterli all'host di destinazione.

7.2 Servizi offerti

Lo strato di trasporto si occupa principalmente di multiplexing/demultiplexing e del controllo degli errori.

7.2.1 Protocolli

In base al protocollo utilizzato per la connessione, lo strato di trasporto si occupa anche di altri fattori. I due protocolli (principali?) sono:

TCP (RFC 793) Si occupa della gestione della connessione, dei controlli di flusso e congestione, e assicura una consegna affidabile.

UDP (RFC 768) Protocollo connection-less, non affidabile ma piu' veloce.

7.2.2 Multiplexing/Demultiplexing

Multiplexing Lo strato di trasporto si occupa di *accorpare* i flussi di dati e spedirli verso la loro destinazione insieme ad una *busta* di trasporto.

Demultiplexing Lo strato di trasporto si occupa di ricevere i dati dalla rete e instradarli verso i processi desiderati.

Entrambi si basano sul socket addres (IP + porta) per identificare i processi. In base al protocollo usato, lo strato di trasporto si occupa anche del

Demultiplexing connectionless - UDP La *busta* contiene solamente la socket address del destinatario. I datagrammi provenienti da host differenti vengono consegnati tutti alla stessa socket di destinazione.

Demultiplexing orientato alla connessione - TCP La *busta* contiene le socket address di mittente e destinatario. Un host puo' supportare piu' socket contemporaneamente (p.e. server Web). Lo strato di trasporto puo' quindi inviare i dati alla socket appropriata in base al mittente.

7.2.3 Le porte

Ogni processo che intende comunicare con la rete (punto di demultiplexing) e' identificato tramite un socket address. La porta e' un valore u-int16(0-65535). I range standard sono:

System ports (0-1023) Assegnate da IANA, identificano i processi server.

User ports (1024-49151) Assegnate da IANA, non e' possibile la duplicazione.

Dynamic ports (49152-65535), Non assegnate da IANA, muoiono al momento della chiusura della connessione.

7.2.4 Well known ports

20/tcp	ftp-data
21/tcp	ftp
22/tcp	ssh
23/tcp	telnet
25/tcp	smtp
53/udp	dns
53/tcp	dns
69/udp	tftp
80/tcp	www-http
110/tcp	pop3
119/tcp	nntp
161/udp	sntp
220/tcp	imap3
510/udp	RIP
3306/tcp	mysql

Chapter 8

Lo Strato di Trasporto: TCP

8.1 Proprietà del servizio

I processi effettuano un handshake COVID!1!. Lo strato della connessione risiede sui puni terminali e non sui nodi della rete.

Permette di trasferire un flusso continuo di dati in formato bidirezionale (full-duplex). Inoltre consente di assegnare una connessione diretta processo - processo. Offre anche controllo di connessione tramite meccanismi di inizio e fine trasmissione. Tramite alcuni bit di controllo, permette di correggere alcuni tipi di errore. Con il controllo di flusso si evita di spedire piu' dati di quanti il destinatario sia in grado di trattare. Nel caso di sovraccarico della rete, tramite il controllo di congestione, si hanno sistemi di recupero della connessione.

Trasferimento bufferizzato Il protocollo TCP puo' suddividere il flusso di byte in segmenti. Per farlo e' necessario un buffer dove immagazzinare i dati finché non ce n'e' un numero sufficiente da essere spediti. Cio' consente un minor numero di messaggi scambiati per trasferire una sequenza di byte.

8.2 Formato dei segmenti

8.2.1 Numeri di sequenza e di riscontro

Numero di sequenza e' il numero del primo byte di un segmento. Generalmente si parte da un Initial Sequence Number (ISN) casuale.

Numero di riscontro e' il +1 del numero dell'ultimo byte ricevuto correttamente.

$ACK{=}x$ = significa: ho ricevuto tutti i byte fino a *x-1*, aspetto *x*.

8.2.2 Forma dei segmenti

L'header aggiunto al messaggio contiene:

Porta sorgente 16bit

Porta destinazione 16bit

n-SEQ Numero di sequenza, 32 bit

n-ACK Numero di riscontro, 32 bit

HLEN Lunghezza dell'header espressa in parole da 4Byte, 4bit

Reserved 6bit

URG Il campo puntatore contiene i dati da trasferire in via prioritaria, 1bit

ACK Considerare n-ACK, 1bit

PSH Trasferimento immediato da trasporto a applicazione

RST Reset della connessione

SYN Sincronizza n-SEQ

FIN Chiusura della connessione

Dimensione finestra Numero di byte a partire da n-ACK, elaborabili, 16bit

Checksum Considera l'intero pacchetto; serve a rilevare gli errori, 16bit

Puntatore urgente Punta al primo Byte non URG partendo da n-SEQ, 16bit

Opzioni e riempimento Negoziazione di vari parametri opzionali, 0-40Byte

8.3 Gestione della connessione

8.3.1 Handshake a tre vie

Richiesta di connessione Il client invia una richiesta al server con:

SYN 1
n-SEQ #rand $_{= x}$

Il server inizializza due buffer e le variabili di connessione per il controllo di flusso e congestione.

Autorizzazzione connessione (SYNACK) Il server risponde con:

SYN 1
ACK 1
n-SEQ #rand $_{= y}$
n-ACK x+1

Il client inizializza gli stessi buffer e variabili.

ACK Riscontro positivo dell'avvenuta connessione, contiene:

SYN 0
ACK 1
n-SEQ x+1
n-ACK y+1

Connessione stabilita.

8.3.2 Chiusura della connessione

C $\Longrightarrow$ S Chiusura da parte del client:

FIN 1
n-SEQ x

Il client non puo' piu' inviare dati.

S $\Longrightarrow$ C ACK di chiusura:

ACK 1
n-ACK x+1

Il server puo' ancora inviare dati, il client no.

S $\Longrightarrow$ C Chiusura da parte del server:

FIN 1
n-SEQ y

Il server aspetta la ricevuta di chiusura.

C $\Longrightarrow$ S ACK di chiusura.

ACK 1
n-ACK y+1

Connessione terminata.

Quando il client riceve FIN dal server entra nello stato TIME_WAIT e ci resta per un tempo dettato da MSL (diverso su macchine differenti) prima di poter chiudere totalmente la connessione. Questo serve nel caso l'ultimo ACK venga perso perché a un certo punto uno dei due host richiederà la chiusura passiva e l'altro dovrà rispondere con un ACK. Allo stesso modo dell'handshake di apertura, si puo' avere un three-way handshake di chiusura.

8.4 Stati TCP

8.4.1 Stati client

SYN-SENT Dopo aver inviato una richiesta di connessione, si aspetta la conferma.

ESTABLISHED La connessione e' pienamente stabilita, e' possibile trasferire dati.

FIN-WAIT-1 Il client aspetta una richiesta di chiusura o l'ack di una sua richiesta di terminazione.

FIN-WAIT-2 Aspetta la richiesta di terminazione di connessione da un host remoto.

TIME-WAIT Vedi sopra.

CLOSED Connessione chiusa. Non e' piu' possibile comunicare.

8.4.2 Stati server

LISTEN In attesa di una connessione TCP qualunque.

SYN-RECEIVED Si entra in questo stato appena si riceve una richiesta di connessione.

ESTABLISHED Vedi sopra.

CLOSE-WAIT Si aspetta la richiesta di chiusura dal client.

LAST-ACK Si aspetta l'ack di chiusura.

CLOSED Vedi sopra.

8.5 Trasferimento di dati affidabile

8.5.1 Esempio Telnet su TCP

C $\Longrightarrow$ **S** L'utente digita C

n-SEQ 42
n-ACK 79
data "C"

S $\Longrightarrow$ **C** ACK per ricevuta di C

n-SEQ 79
n-ACK 43
data "C"

S $\Longrightarrow$ **C** ACK dell'ACK

n-SEQ 43
n-ACK 80

Il mittente puo' anche inviare piu' segmenti senza attendere il riscontro (pipelining).

8.5.2 Eventi lato mittente

Lo strato di trasporto riceve i dati dall'applicazione, li incapsula, gia assegna un numero di sequenza, ed avvia un timer di ritrasmissione (RTO).

La ritrasmissione avviene in caso di:

Timeout Non si riceve un ACK prima della scadenza del RTO.

ACK duplicato Il mittente riceve tre ACK uguali significa che il segmento successivo a quello e' andato perso. Si ha quindi la fast retransmission.

8.5.3 Segmenti fuori sequenza

Se i dati arrivano fuori sequenza (ordinamento sparso), l'entità TCP destinataria puo' memorizzarli temporaneamente ma il protocollo non specifica che utilizzo farne.

Nelle versioni piu' recenti si implementa SACK, che manda l'ACK dei pacchetti fuori sequenza in OPTIONS.

8.5.4 Eventi lato destinatario

Tutti i segmenti inviati per trasmettere dati includono ACK.

Delayed ACK Se il destinatario riceve un segmento atteso, puo' ritardare l'invio di ACK di 500ms a meno che non riceva un altro segmento.

Fast retrasmission request Se il destinatario riceve un segmento fuori sequenza, un duplicato, o un capisce che c'e' un segmento mancante, risponde subito con un ACK indicando il segmento da cui ripartire.

8.5.5 Calcolo del timeout

Il RTO deve essere maggiore del Round Trip Time (RTT) (tempo tra l'invio di un messaggio e la ricevuta dell'ACK).

$$eRTT = (1 - \alpha) * eRTT + \alpha * sRTT$$

eRTT RTT stimato, cumulativo su piu' misure.

sRTT RTT di un segmento trasmesso con successo.

alpha 1/8 (RFC2988)

$$dRTT = (1 - \beta) * dRTT + \beta * (sRTT - eRTT)$$

dRTT Deviazione di eRTT da sRTT

beta 1/4 (RFC2988)

$$\boldsymbol{RTO} = eRTT + 4 * dRTT$$

In molte implementazioni dopo un errore si raddoppia RTO

8.5.6 Finestra di trasmissione

Il buffer di invio, si dividono i byte in finestre. La finestra di trasmissione e' la sottosezione del buffer di invio che deve essere spedita con il prossimo messaggio. Ha dimensione variabile (modificata in base alla condizione della rete) per permettere i controlli di flusso e congestione. Viene fatta avanzare non appena si riceve l'ACK dell'ultima trasmissione.

8.6 Controllo di flusso

Il destinatario di un messaggio mantiene un buffer di ricezione. il processo sullo strato applicazione legge i dati da una finestra di ricezione (non necessariamente quando arrivano). Si intende con controllo di flusso la capacità del mittente di evitare di saturare il buffer del destinatario. Il mittente mantiene una variabile detta finestra di ricezione (rwnd) che detta lo spazio disponibile nel buffer ricevente; essa e' passata dal destinatario al mittente tramite il campo window nell'header TCP.

$$rwnd = RcvBuffer - (LastByteReceived - LastByteRead)$$

Il mittente si assicura che:

$$LastByteSent - LastByteAcked < rwnd$$

Se $rwnd = 0$ il mittente lo *richiede* con un segmento sonda da un byte

8.7 Controllo di congestione

Se si provano a spedire troppi dati su una rete che non puo' supportarli avviene il fenomeno della congestione; cio' puo' provocare:

- Lunghi ritardi (accodamenti nei buffer)
- Perdita di pacchetti (overflow dei buffer)

TCP permette il controllo di congestione: se si percepisce uno scarso traffico, viene aumentata la frequenza di invio; altrimenti diminuisce.

8.7.1 Algoritmo di controllo

L'algoritmo utilizzato per regolare la frequenza di invio in funzione della congestione e' costituito da tre fasi:

- Slow start
- AIMD
- Fast recovery

8.7.2 Finestra di congestione (Congestion Window)

Misurata in Max Segment Size (MSS); E' il numero di segmenti che si possono inviare insieme, senza bisogno di aspettarne l'ACK.

8.7.3 Slow start

All'inizio la congestion window (CW) e' posta a 1MSS. Per ogni ACK, CW viene aumentata di 1MSS. Cio' significa che essa raddoppia per ogni RTT.

8.7.4 Additive Increase Multiplicative Decrease

La CW viene aumentata in modo da avere un incremento pari a 1MSS per ogni RTT; Cio' significa che per ogni ACK, cwnd = cwnd+1/cwnd.

Ad ogni evento di perdita di ACK la CWND viene dimezzata.

8.7.5 TCP RENO

Viene definita una variabile di threshold alla quale viene assegnato un valore alto; Finché $cwnd < threshold$, siamo in slow start (cwnd aumenta esponenzialmente).
Non appena $cwnd > theshold$, si entra in AI.

Se ricevo 3 ACK duplicati entro in fast recovery:

$$threshold = \frac{cwnd}{2}$$

$$cwnd = threshold + 3MSS$$

Finché continuano ad arrivare ACK duplicati

$$cwnd = cwnd + 1$$

Non appena arriva un ACK non duplicato

$$cwnd = threshold$$

Se perdo un ACK per timeout entro in slow start:

$$threshold = \frac{cwnd}{2}$$

$$cwnd = 1MSS$$

8.7.6 TCP Tahoe

Molto simile a RENO ma timeout e 3 ACK duplicati vengono gestiti allo stesso modo (si va in slow start)

8.7.7 Throughput

Indicando con W la dimensione massima raggiunta dalla finestra, il throughput medio e'

$$Throughput = \frac{0,75 \times W}{RTT}$$

8.7.8 Equità (fairness)

In condizioni perfette, K connessioni TCP che passano sullo stesso link di capacita' R bit/s, le connessioni hanno gli stessi valori di MSS e RTT; in questo caso ogni connessione trasmette R/K bit/s. In condizioni reali pero' le connessioni con RTT piu' piccolo (migliori) variano piu' velocemente *congwin* e raggiungono throughput superiori.

Chapter 9

Lo Strato di Trasporto: UDP

I datagrammi UDP possono essere perduti o consegnati fuori sequenza, perciò i processi devono inviare messaggi di dimensione limitate, e devono essere contenuti in un solo datagramma.

9.0.1 Differenze tra TCP e UDP

TCP	UDP
Connection-oriented	Connection-less
Mandatory checksum	Optional checksum
Introduce ritardi	Nessun ritardo aggiunto
Aggiunge informazioni	Datagramma minimale

9.0.2 Formato dei datagrammi

Porta sorgente 16bit

Porta destinazione 16bit

Lunghezza Lunghezza totale del segmento, 16bit

Checksum Controllo errore end-to-end, opzionale, 16bit

Data max 65535Byte - 8Byte intestazione

9.0.3 Utilizzi UDP

- Processi che richiedono scambio di dati con volume limitato e senza controlli di flusso/errori.
- Processi con meccanismi interni di controllo di flusso/errore.
- Trasmissioni multicast (piu' destinatari).
- Applicazioni interattive che non tollerano ritardi variabili.

Chapter 10

Lo Strato di Rete: IP

10.1 Concetti generali

Lo strato di rete si occupa dello scambio dei datagrammi tra gli hosts. Offre servizi allo strato superiore *(trasporto)* e utilizza i servizi di quello inferiore *(collegamento)*. Offre un'astrazione che consente a host e reti diverse di funzionare, dal punto di vista logico, come una singola rete.

10.1.1 Passaggi della comunicazione

Mittente L'entità a livello di rete riceve i segmenti e li incapsula in datagrammi prima di essere spediti verso il prossimo nodo della connessione (host o router).

Nodo intermedio Il router esamina i campi intestazione dei datagrammi che riceve da un collegamento in ingresso, e li inoltra al collegamento in uscita.

Destinatario I datagrammi arrivano allo strato di rete del destinatario, che li consegna al rispettivo strato di trasporto (demux TCP o UDP).

10.2 Internet Protocol (IP)

Descritto dalla RFC-791, e' un protocollo di tipo connection-less (Send-and-Pray, QoS non garantito, ...).

10.2.1 Protocol Data Unit (PDU)

<table>
<tr><td>Frame
header</td><td>IP
header</td><td>TCP
header</td><td>User data</td></tr>
</table>

10.2.2 Servizi offerti

Inoltro (forwarding) Permette di instradare verso il corretto collegamento di uscita. Per fare cio' il router legge l'indirizzo di destinazione dal campo di intestazione e lo confronta con una tabella di inoltro, la quale indica la giusta interfaccia di uscita per quell'indirizzo.

Instradamento (routing) E' il processo decisionale che sceglie il percorso migliore tra mittente e destinazione.

Indirizzamento Strumento per identificare gli host nell'internet.

Modello datagram Permette la segmentazione / riunificazione dei pacchetti, il controllo degli errori (nell'header), e la verifica TTL.

10.2.3 Formato dei datagrammi

Intestazione	Dati

Intestazione e' lunga 20-60Byte ed e' formata da:

ver Versione del protocollo IP (IPv4, IPv6).

head. len Lunghezza dell'header in parole di 4Byte.

type of service

lenght Lunghezza del datagramma in Byte.

16-bit identifier Per frammentazione.

flags Per frammentazione.

fragment offset Per frammentazione.

time to live Massimo numero di *hop* rimanenti (decrementato ad ogni router).

upper layer Protocollo di trasporto.

header checksum Calcolato ad ogni router.

32 bit source IP address Indirizzo sorgente.

32 bit destination IP address Indirizzo destinazione.

options Facoltativo.

data Lunghezza variabile, solitamente segmento TCP o UDP.

10.2.4 Frammentazione

Diversi protocolli di trasferimento possono piu' o meno dati in un solo frame. Il massimo e' determinato dal Maximum Transfer Unit (MTU). Cio' significa che un router potrebbe ricevere un frame piu' grande dell'MTU della rete verso la quale deve inoltrarlo; in questi casi il router deve dividere il datagramma in piu' datagrammi piu' piccoli detti frammenti. Il riassemblaggio dei frammenti viene effettuato dall'entità rete del destinatario. Se uno o piu' frammenti vengono persi l'intero messaggio viene scartato.

Per riordinare i frammenti vengono usati 3 campi intestazione:

Identificatore e' un valore associato alla coppia ¡mittente, destinatario¿ che identifica il datagramma.

Offset indica la posizione relativa come multiplo di 8Byte (Il primo frammento ha offset 0).

Flag Serve ad identificare l'ultimo frammento:

- **bit 0** *reserved*, sempre a 0 per il momento.
- **bit 1** *do not fragment*, vale 1 se il pacchetto non puo' essere frammentato.
- **bit 2** *more fragments*, vale 1 se il pacchetto non e' l'ultimo del frammento.

La frammentazione e' un processo critico perché richiede risorse ai nodi intermedi e destinazione e introduce ritardi perciò e' preferibile non usarla (impostare il *do not fragment* a 1 in IPv4, IPv6 non supporta la frammentazione).

10.3 Indirizzamento IPv4

Gli indirizzi IPv4 sono costituiti da 4Byte. Ogni host ha un indirizzo univoco diviso in 2 parti

Prefisso	Suffisso

Il Prefisso individua la rete metre il suffisso individua il singolo host.

10.3.1 Classful addressing

Per riconoscere un indirizzo c'e' bisogno di indicare la lunghezza del prefisso (e di conseguenza del suffisso).

Un sistema e' il classful addressing che definisce delle classi e le rispettive divisioni P/S. Per riconoscere le classi si imposta il primo byte dell'indirizzo in un range di valori prefissato.

Classe	Primo Byte	Pref. Len.
A	0-127	8bit
B	128-191	16bit
C	192-223	24bit
D	224-239	Non applicabile
E	240-255	Non applicabile

10.3.2 Classless addressing

Utilizza la notazione Classless Interdomain Routing (CIDR) cioe':

byte.byte.byte.byte/n

con n pari alla lunghezza del prefisso.

Dato un indirizzo a.b.c.d/n si definisce la Subnet mask come valore da 32bit con i primi n bit a sinistra impostati a 1. Effettuando una Bitwise-AND tra un indirizzo e la maschera si ottiene l'indirizzo della rete.

10.4 Assegnazione blocchi di indirizzi

Gli indirizzi IP *"globali"* sono gestiti da ICANN, che li assigna agli ISP in blocchi. Sono poi gli ISP ad assegnare sottoblocchi di essa ai suoi clienti.

10.4.1 Aggregazione di indirizzi

Per rendere piu' efficiente l'instradamento, gli ISP possono assegnare vari blocchi di indirizzi a piu' clienti per poi aggregarli in un singolo blocco da annunciare alla rete.

10.4.2 Indirizzi speciali

0.0.0.0 This host, usato come indirizzo sorgente quando un host non sa il proprio indirizzo.

255.255.255.255 Limited-broadcast, usato per inviare un datagramma a tutti i dispositivi all'interno della rete locale.

127.0.0.1 Loopback, il datagramma non lascia l'host locale.

10.0.0.0/8

127.16.0.0/12

192.168.0.0/16

169.254.0.0/16 Indirizzi privati, riservati per le reti locali.

224.0.0.0/4 Indirizzi multicast.

10.4.3 DHCP

Un indirizzo IP puo' essere attribuito a un host tramite configurazione manuale (l'host o chi lo gestisce configura l'indirizzo e le altre informazioni di servizio), oppure tramite DHCP (l'host ottiene tutte le informazioni automaticamente).

Il Dynamic Host Configuration Protocol e' un protocollo client-server che assegna indirizzi IP in modo dinamico. Appena un host si connette alla sottorete da inizio alla procedura di assegnazione comunicando con il server DHCP della sottorete su cui e' inserito.

10.5 Network Address Translation

L'accesso di una rete privata su Internet avviene tramite un router abilitato alla NAT. Tutto il traffico in uscita dal router di accesso ha come indirizzo IP quello pubblico del router. Tutto il traffico in ingresso alla subnet e' indirizzato verso l'IP pubblico del router.

Il router mantiene una tabella di traduzione NAT, le cui righe contengono associazioni del tipo:

(IP privato, porta) $\leftrightarrow$ (IP pubblico, porta)

10.6 Internet Control Message Protocol

L'ICMP e' usato da host e router per scambiarsi messaggi di errore o altre situazioni che richiedono intervento. Sono incapsulati all'interno di datagrammi IP ma il protocollo viene comunque considerato parte dello stato di rete. I pacchetti ICMP vengono instradati dai router prima dei pacchetti IP ordinari. Sono relativi solo al frammento 0 in caso di frammentazione del pacchetto.

10.6.1 Tipi di messaggio

I messaggi ICMP si dividono in

- Messaggi di segnalazione errore.
- Messaggi di richiesta/risposta.

L'header ICMP e' formato da un campo Tipo (8bit) e un campo Codice (8bit).

Tipo	Codice	Descrizione
0	0	Echo replay. (ping)
3	0	Dest. network unreachable.
3	1	Dest. host unreachable.
3	2	Dest. protocol unreachable.
3	3	Dest. port unreachable.
3	6	Dest. network unknown.
3	7	Dest. host unknown.
4	0	Source quench. (congestion control)
8	0	Echo request.
9	0	Router advertisement.
10	0	Router discovery.
11	0	TTL expired.
12	0	Bad IP header

10.6.2 Ping

PING e' un'applicazione di ICMP utilizzata per verificare lo stato di funzionamento di un altro host. Viene inviata una richiesta eco (08,00) e si attende la risposta dell'host destinazione (00,00). Essa fornisce anche misure dell'RTT e permette in maniera molto grossolana di misurare l'affidabilità e la congestione dei router tra due host inviando una sequenza di messaggi richiesta-risposta.

10.6.3 Traceroute

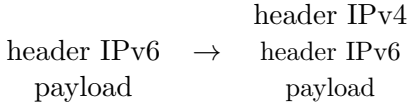
Traceroute si basa sui messaggi di errore (tempo scaduto e porta non raggiungibile). Invia datagrammi UDP con valori di TTL crescenti (partendo da zero); quando il TTL raggiunge 0, il router invia un messaggio di errore. L'host conta il tempo di ritorno dell'errore per trovare il RTT di ciascun router del percorso.

10.7 IPv6

Motivazione	Soluzione
Esaurimento indirizzi 32bit	Indirizzi a 128bit
Velocizzare elaborazione e forwarding	Lunghezza fissa header Eliminato checksum
Risparmiare costo frammentazione ai router	ICMPv6 msg 'packet too big'
Facilitare QoS	Introdotta 'flow label'

10.7.1 Dual stack

E' possibile mascherare un datagramma IPv6 (header + payload) come payload di un datagramma IPv4.



Chapter 11

Lo strato di rete: Forwarding e Routing

11.1 Architettura di un Router

11.1.1 Forwarding vs Routing

Forwarding Data plane

E' il processo che si occupa del trasferimento dei pacchetti sul collegamento in uscita appropriato.

Routing Control plane

E' il processo decisionale di scelta del percorso ottimale verso una destinazione.

Routing decentralizzato Ogni router si coordina con gli altri scambiandocisi messaggi ed esegue algoritmi di routing.

Software-Defined Networking (SDA) Un Remote Controller interagisce con tutti i Control Agents locali (router): I CA inviano informazioni sui collegamenti e sul traffico al RC; esso calcola i percorsi migliori ed invia ai CA i valori da inserire nelle tabella di inoltro.

11.1.2 Componenti di un router

Le quattro componenti principali di un router sono:

- Porte di input
- Porte di output
- Routing processor
- Switching fabric

11.1.3 Porte di input

Sono formate da:

Line termination La connessione fisica che riceve i bit.

Link layer protocol Il livello di collegamento, si occupa di interpretare i bit in ingresso dallo strato inferiore.

Lookup, Forwarding, Queueing Preleva dati dall'header e li confronta con una copia della tabella di inoltro per determinare la porta di output. Se il router non riesce ad inoltrare i datagrammi nella switching fabric abbastanza velocemente, essi entrano in una coda.

11.1.4 Porte di output

Formate da:

Datagram buffer, queueing Permette di mantenere una coda di datagrammi se essi arrivano ad una velocità maggiore della velocità di trasmissione sul collegamento in uscita. Permette anche di definire politiche di scheduling per dare piu' o meno priorità a determinati datagrammi.

Link layer protocol Vedi input.

Line termination Vedi input.

11.1.5 Routing processor

E' il *cervello* del router, il suo ruolo principale e' quello di definire le tabelle di inoltro.

11.1.6 Switching fabric

E' il sistema che si occupa di trasferire i datagrammi dal buffer di input al buffer di output. I sistemi di commutazione piu' standard si basano su:

Memory I dati passano dalla memoria del sistema di commutazione.

Bus Si ha un unico bus condiviso da tutte le porte.

Crossbar Anche detta matrice di commutazione, e' un collegamento *a griglia*, permette il trasferimento simultaneo su piu' collegamenti.

11.1.7 Approfondimenti

I buffer sono fonte di ritardi dovuti agli accodamenti o a overflow degli stessi.

11.2 IP Forwarding

E' il sistema che vive sui router che si occupa di inoltrare i pacchetti dal collegamento di ingresso al giusto collegamento di uscita. Puo' essere diretto o indiretto.

Inoltro diretto Avviene quando il pacchetto IP non esce dalla propria rete o subnet; L'indirizzo di destinazione a livello di link e' il suo MAC address.

Inoltro indiretto E' quando il pacchetto deve uscire dalla propria rete; l'indirizzo di destinazione a livello di link e' quello del router.

11.2.1 Fasi dell'inoltro

Diretto L'host sorgente confronta l'IP destinazione con il proprio IP e la subnet; se riconosce la destinazione come parte della sua rete procede con l'inoltro diretto. Cerca quindi l'indirizzo di livello collegamento (MAC address) del destinatario all'intero della propria ARP table (una tabella che contiene le associazioni $IP \leftrightarrow MAC$) e invia un frame con:

- src-MAC = *MAC Sorgente*
- dst-MAC = *MAC Destinazione*
- Pacchetto IP
  - src-IP = *IP Sorgente*
  - dst-IP = *IP Destinazione*
  - payload

Indiretto In caso di inoltro indiretto il frame conterrà il MAC Address del router, il quale dovrà poi reinoltrare il pacchetto verso il giusto host.

11.3 Routing

Lo scopo del livello di rete e' quello di consegnare un datagramma dalla sorgente alla/e destinazione/i.

Routing unicast significa che un datagramma e' destinato a una sola destinazione.

Routing multicasti significa che un datagramma e' destinato ad un gruppo di host.

Una rete puo' essere rappresentata come un grafo pesato che identifica i Router con i nodi e i collegamenti tra di loro con gli archi (il costo e' determinato da vari fattori come: distanza tra i nodi; numero di hop; banda disponibile...).

Il routing algorithm calcola il cammino di costo minimo tra due nodi.

11.3.1 Routing statico e dinamico

Statico Le entry delle tabelle di inoltro vengono configurate manualmente dall'operatore. Viene usato per reti molto piccole in cui e' possibile prevedere tutti i possibili percorsi di rete.

Dinamico Le tabelle di inoltro vengono compilate automaticamente da protocolli specifici dipendenti dallo stato attuale della rete. Viene utilizzato per grandi reti private e per Internet.

11.3.2 Algoritmi di routing globali e decentralizzati

Globali Si basano sulla conoscenza della topologia dell'intera rete; Vengono quindi generate e inviate le tabelle di inoltro a tutti i router che la compongono.

Decentralizzati Inizialmente ogni nodo conosce solamente i costi verso i nodi vicini, poi, scambiando informazioni con essi, elabora un vettore di stima dei costi verso tutti gli altri.

11.3.3 Link-State algorithm

E' un algoritmo globale. Ogni nodo invia e riceve i costi dei collegamenti attraverso il Link-State Broadcast. Ogni nodo dispone quindi dell'intera topologia di rete e puo' calcolare i percorsi migliori utilizzando l'algoritmo di Dijkstra.

Il Link-State Database e' una tabella $N \times N$ (Con N pari al numero di nodi) in cui

$$LSD(x,y) = \begin{cases} c(x,y) & \text{se } x,y \text{ sono adiacenti} \\ \infty & \text{altrimenti} \end{cases}$$

Viene creata utilizzando le informazioni che arrivano da tutti i nodi sotto forma di Link-State Packets, i quali contengono i costi dei percorsi di ogni nodo verso i suoi vicini.

11.3.4 Distance Vector algorithm

E' un algoritmo decentralizzato e asincrono: ogni nodo provvede al ricalcolo dei Distance Vectors se:

- Cambia il costo di uno dei collegamenti verso i nodi vicini.
- Si riceve un DV aggiornato da un nodo vicino.

Se il DV cambia, il nodo lo notifica ai vicini.

Algorithm 1: DistanceVectorRouting()

```
/* The node executing the code is identified as x      */
for y = 1 → N do
    if y.isAdj() then
        D[y] = c(x, y)
    else
        D[y] = ∞
D[x] = 0
while true do
    /* Wait for a modified distance vector from another
       node                                              */
    while !newVecFromAdj(D_v) do nothing;
    forall y = 1 to N do
        /* Compare the old route to y and the route
           passing through v                             */
        D[y] = min(c(x, v) + D_v, D[y])
```

11.3.5 Count to infinity

V. slide

11.4 Hierarchal Routing

Seppur semplice concettualmente, una rete costituita da un insieme di router omogenei interconnessi, questo approccio non e' applicabile alle dimensioni di Internet. In realtà i router sono divisi in gruppi detti sistemi autonomi (AS) gestiti da un unico amministratore.

11.4.1 AS Types

AS Stub Collegato ad un solo altro AS.

AS Multihomed Collegato a piu' AS ma trasporta solo traffico da o verso se stesso.

AS di transito Implied by name.

Interior Gateway Protocol (IGP) Sono i protocolli di routing utilizzati all'interno di un AS.

Exterior Gateway Protocol (EGP) Sono i protocolli di routing fra AS.

11.4.2 INTRA-AS e INTER-AS Routing

Per trasferire pacchetti all'interno di un AS si usano protocolli detti INTRA-AS: I piu' comuni sono

- Routing Information Protocol (RIP) - Di tipo DV
- Open Shortest Path First (OSPF) - Di tipo LS

Nel momento in cui ci si deve *spostare* da un AS a un altro si utilizzano i protocolli INTER-AS: Il piu' usato e' il Border Gateway Protocol (BGP)

11.4.3 Routing Information Protocol (RIP)

E' un protocollo di routing INTRA-AS che implementa Distance Vector con Poisoned Reverse (inf = 16). Il costo dei collegamenti e' dato dal numero di sottoreti attraversate. Cio' porta a una limitazione al numero massimo di sottoreti pari a 15. La tabella di inoltro conterrà quindi:

- Rete di destinazione
- Next hop (prossimo router)
- Costo (in hop)

I messaggi contenenti le tabelle di routing vengono scambiati, tramite la porta UDP 520, ogni 30sec o se la tabella cambia. Quando un router (R) riceve una tabella di inoltro da un suo vicino (V), procede a modificare la sua se:

- $D_R[y]$ non esiste.
- $D_V[y]+1 < D_{R_{old}}[y]$
- $D_V[y]$ e' cambiato e V e' nextHop di y

Inoltre si aggiunge 1 a ogni valore prelevato da D_V dato che si deve effettuare un hop in piu' $R \rightarrow V$.

11.4.4 Open Shortest Path First (OSPF)

Protocollo di routing INTRA-AS che implementa LS su IP. In OSPF il costo dei collegamenti e' tarato su variabili determinate dall'amministratore della rete (latenza, banda, n.hop ...) Inoltre l'AS puo' essere partizionato in aree per ridurre il numero di pacchetti LS che attraversano la rete.

11.4.5 Border Gateway Protocol (BGP)

BGP4 e' l'unico protocollo INTER-AS usato in Internet. I router si scambiano informazioni su connessioni TCP. Si considerano le sessioni BGP esterne (Tra AS diversi) e interne (All'interno dell'AS). Un gateway riceve informazioni da gateway di altri AS su eBGP e ri-distribuisce queste informazioni all'interno del suo AS tramite iBGP. I pacchetti di Advertisement (ADV) BGP contengono:

prefisso + attributi

Gli attributi principali sono:

AS_PATH La sequenza di AS attraversati per raggiungere il prefisso.

NEXT_HOP IP del primo router lungo il percorso annunciato (al di fuori dell'AS che riceve l'ADV)

Dato che si possono ricevere piu' rotte per lo stesso prefisso bisogna avere una sequenza di regole per selezionare quella migliore.

1. Attributo LOCAL-PREF, scelta dall'amministratore o impostato dei router dell'AS.
2. Shortest AS_Path.
3. Closest NEXT_HOP interface ("hot potato routing").

Chapter 12

Lo Strato di Collegamento

I collegamenti possono essere:

Punto-punto Collegamento dedicato a due soli dispositivi.

Broadcast Collegamento condiviso tra piu' dispositivi.

12.1 Servizi offerti

Framing I datagrammi del livello di rete vengono incapsulati all'interno di un frame. I frame sono formati da:

- Campo dati
- Intestazione
- Trailer (opzionale)

Per identificare origine e destinazione si utilizzano i MAC addresses

Consegna affidabile Non necessaria per collegamenti *affidabili* come quelli cablati. Si utilizza spesso nei collegamenti wireless.

Controllo di flusso Evita che il nodo trasmittente saturi il ricevente.

Rilevazione degli errori Il trasmittente puo' inserire dei bit di controllo degli errori all'interno del frame permettendo al ricevente di individuarli.

Correzione degli errori Grazie ad algoritmi di correzione, il ricevente puo' correggere gli errori.

12.2 Indirizzi a livello collegamento

E' associato alla scheda di rete e non al nodo, tipicamente e' permanente. Per le LAN Ethernet (IEEE 802) e' lungo 6Byte. L'univocità di tali indirizzi e' garantita perché in parte gestita da IEEE (I primi 24 bit sono assegnati da IEEE mentre i restanti vengono gestiti dalle aziende).

12.2.1 Indirizzamento

Quando il nodo mittente spedisce un frame, vi inserisce l'indirizzo MAC della scheda di destinazione. Tutti i nodi collegati su quel collegamento ricevono il frame e lo scartano se non e' indirizzato a loro. Se si vuole spedire un frame a tutte le schede di rete su un collegamento , immette nel campo destinazione l'indirizzo broadcast ovvero FF-FF-FF-FF-FF-FF.

12.3 Address Resolution Protocol

Nel momento in cui viene inserito un nodo in una nuova rete esso conosce:

- Il proprio indirizzo fisico (MAC)
- Il proprio indirizzo IP
- Il proprio indirizzo alfanumerico

Per comunicare con i nodi vicini pero', come visto sopra, ha bisogno degli indirizzi fisici di essi. Per trovarli e memorizzarli utilizza il Protocollo di Risoluzione degli Indirizzi. (Protocollo a livello di rete)

ARP Table Nella tabella ARP vengono memorizzate le associazioni IP → MAC, insieme ad un TTL (quanto tempo deve valere l'associazione, solitamente 20min)

ARP Packet Tramite richieste ARP (fatte in broadcast) un nodo puo' richiedere l'indirizzo fisico di un altro nodo nella rete. Il pacchetto contiene l'indirizzo del nodo che fa la richiesta cosicché il destinatario sa a chi rispondere. Il pacchetto ARP viene incapsulato in un frame e spedito sulla rete; ogni nodo riceve ed elabora la richiesta; il nodo che riconosce il proprio indirizzo IP restituisce una risposta contenente il proprio IP e MAC, esso viene inviato in modalità unicast al nodo richiedente.

12.3.1 Esempio forwarding diretto

A vuole inviare un messaggio a B, entrambi sono sulla stessa rete.

- A richiede l'indirizzo MAC di B tramite un messaggio ARP in broadcast.
- B riceve il pacchetto ARP e risponde ad A (in unicast) comunicandogli il proprio indirizzo MAC.
- A riceve il MAC Address di B, aggiorna la ARP Table, e invia a B il frame.

12.3.2 Esempio forwarding indiretto

A vuole inviare un messaggio a B, i due host si trovano su reti differenti ma sono collegati da un router R.

- A, grazie a DHCP o configurazione manuale, conosce l'indirizzo IP di R.
- Tramite ARP ricava anche l'indirizzo MAC di R.
- Invia il frame destinato a B, con indirizzo IP di destinazione di B e indirizzo MAC di destinazione di R.
- R riceve il frame, che viene decapsulato passando al livello IP.
- R vede che il datagramma e' destinato a B quindi lo incapsula in un frame con indirizzo MAC destinazione di B (eventualmente ricavato con ARP)

12.4 Ethernet

E' il sistema di collegamento cablato piu' diffuso.

Nella configurazione a BUS (il primo tipo di connessione ethernet) si avevano piu' host collegati ad un unico *cavo di rete*. Questo sistema non era particolarmente efficiente per colpa della facilita con cui si riscontravano collisioni tra pacchetti.

Nella configurazione a stella invece, tutti ghi host si collegano ad un nodo centrale detto hub (o switch).

12.4.1 Struttura dei pacchetti Ethernet

Preamble	Dst.addr	Src.addr	Type	Data	CRC

Preamble I pacchetti Ethernet iniziano tutti con 7Byte (10101010) e 1Byte (10101011) che segnano l'inizio di un frame e servono a sincronizzare i clock dei riceventi con il trasmittente.

Dst.addr 6Byte, La scheda di rete che riconosce il proprio MAC Address in questo campo (o vede l'indirizzo di broadcast), ne trasferisce il contenuto al livello di rete.

Src.addr 6Byte, Indirizzo della scheda di rete che trasmette il frame.

Type 2Byte, Permette di supportare e riconoscere i vari protocolli di rete (ARP, IPvn ...)

Data I dati da trasmettere. Se il campo dati e' troppo piccolo, viene riempito (stuffed).

CRC Cyclic Redundancy Check, contiene i bit di controllo utilizzati dal ricevente per rilevare eventuali errori.

12.4.2 Dispositivi di interconnessione

Repeater Opera a livello fisico, rigenera il segnale che riceve sull'ingresso e lo inoltra sull'uscita.

Hub Esattamente come un repeater ma ha piu' porte: tutto cio' che riceve su una lo inoltra sulle altre.

Switch Operano anche a livello Link leggendo il MAC nel frame; tramite una tabella di commutazione inoltra il pacchetto verso la porta dove si trova il destinatario. La tabella di commutazione si genera automaticamente ogni volta che si riceve un frame da una porta.

Router Opera fino a livello network per verificare l'indirizzo IP di destinazione e inoltrare il pacchetto verso l'interfaccia giusta. Opera solo sui frame che hanno come Des.addr l'indirizzo del Router.