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 Pipelining e HeadOfLineBlocking

Per migliorare le le prestazioni il client puo' inviare contemporaneamente piu' richieste al server, il quale **DEVE** rispondere nello stesso ordine in cui sono state ricevute.

Se pero' una richiesta richiede piu' tempo al server per essere processata blocca tutte le risposte successive e porta all'HOLB. In questi casi non si ha un miglioramento delle performance.

2.7.2 HTTP Request

La richesta e' formata da:

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

2.7.3 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.

2.7.4 Content Negotiation

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

2.7.5 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.

2.7.6 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.

2.7.7 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.