Appunti di

Reti (F. Granelli)

Davide Parpinello

Aprile 2020

Indice

Capitolo 1

Roadmap

1.1 Internet

Internet è costituito da milioni di dispositivi, chiamati **sistemi terminali**, collegati tra loro da collegamenti in rame, fibre ottiche, oppure via radio come onde elettromagnetiche o satelliti.

La frequenza di trasmissione in internet è data dall'ampiezza di banda disponibile.

Sulla rete internet inoltre sono presenti particolari host, denominati **router**, che si occupano di instradare i pacchetti verso la loro destinazione finale.

Nello scambio di messaggi tra host vengono implementati dei **protocolli**, che ne definiscono formato e ordine d'invio, così come le azioni intraprese in fase di trasmissione e/o ricezione di un messaggio o un altro evento.

Internet è un'infrastruttura di comunicazione per applicazione distribuita, viene anche chiamato "rete delle reti" ed è organizzato in modo gerarchico.

1.2 Ai confini della rete

Sul bordo della rete internet troviamo applicazioni e sistemi terminali, raggruppati tra loro e connessi tra di loro mediante collegamenti cablati e wireless.

I sistemi terminali (o host) fanno girare diversi programmi applicativi e possono essere organizzati con architettura client/server oppure P2P.

L'accesso a internet può avvenire mediante diversi modi:

- **Accesso residenziale:** viene utilizzato un modem dial-up o DSL.
- **Accesso aziendale:** una LAN collega i sistemi terminali di aziende e università all'**edge router**, i sistemi terminali sono collegati tra loro mediante uno switch ethernet.
- **Accesso wireless:** i terminali vengono collegati mediante **access point**.
- **Reti domestiche:** sono costituite da un modem DSL o via cavo, un router/firewall/NAT, switch ethernet e accesso wireless. Spesso queste funzioni vengono raggruppate in un unico dispositivo (modem/router).

1.3 Il nucleo della rete

Al centro della rete si trovano invece router interconnessi tra loro, che creano quindi una rete delle reti.

I dati nel nucleo della rete vengono trasferiti con due modalità differenti:

- **Commutazione di circuito:** è presente un circuito dedicato per l'intera durata della sessione
- **Commutazione di pacchetto:** i messaggi di una sessione utilizzano le risorse su richiesta, di conseguenza potrebbero dover attendere per accedere a un collegamento.

1.3.1 Esempio di commutazione di circuito

Consideriamo un file L di 640.000 bit, un bitrate totale C da 1.536 Mbps, TDM con 24 slot/s, 500ms per stabilire la connessione.

Trovo inizialmente la capacità di un singolo slot:

$$C_{1slot} = \frac{C}{24} = 0.064Mbps = 64Kbps$$

Successivamente calcolo il tempo necessario alla trasmissione:

$$T_{tx} = \frac{L}{C_{1slot}} = 10s$$

Infine, calcolo il tempo totale:

$$T_{tot} = 500ms + 10s = 10,5s$$

1.3.2 Esempio di commutazione di circuito

I secondi necessari per trasmettere un pacchetto in uscita su un collegamento da R bps sono dati da L/R, mentre il ritardo 3L/R

1.3.3 Confronto fra commutazione di pacchetto e di circuito

La commutazione di pacchetto consente un utilizzo della rete da parte di maggiori utenti, ed è ottima per i dati a raffica.

Dal lato negativo, presenta un'eccessiva congestione, causando ritardi e perdite di pacchetti. Sono quindi necessari protocolli per il trasferimento affidabile e per il controllo della congestione.

1.3.4 Struttura gerarchica

La rete internet ha una struttura fondamentalmente gerarchica:

- Al centro sono presenti **ISP di livello 1**, che offrono una copertura nazionale e/o internazionale
  - Comunicano tra loro come fossero pari
- **ISP di livello 2:** ISP più piccoli, copertura nazionale/distrettuale.
  - Si può connettere solo ad alcuni ISP di livello 1 e ad altri di livello 2
  - Paga l'ISP di livello 1 che gli fornisce la connettività per il resto della rete
- **ISP di livello 3 e ISP locali (di accesso):**
  - sono le reti "ultimo salto", le più vicine agli host
  - Sono clienti degli ISP di livello superiore che li collegano all'intera internet.

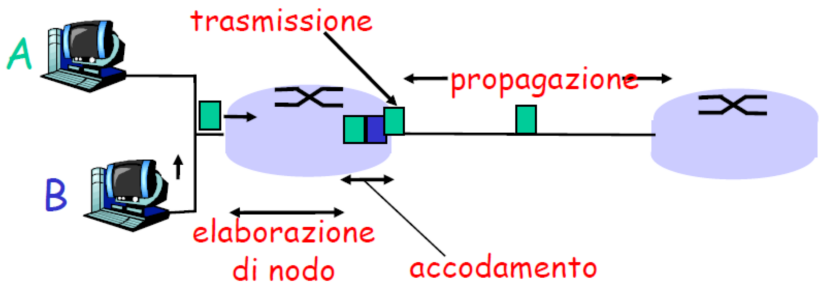
Un pacchetto attraversa un sacco di reti, dal livello più basso fino al principale e poi nuovamente a scendere.

1.4 Ritardi, perdite e throughput nelle reti a comunicazione di pacchetto

Nella rete si verificano dei ritardi quando il tasso di arrivo dei pacchetti eccede la capacità di evaderli, con la conseguenza che vengono accodati nei buffer del router in attesa del proprio turno. Se non ci sono buffer liberi i pacchetti vengono scartati e ritrasmetti dal nodo precedente o, in alcuni casi, non venire proprio ritrasmessi.

Quattro cause di ritardo dei pacchetti sono le seguenti:

1. Ritardo di elaborazione sul nodo
   - Controllo errori sui bit
   - Determinazione del canale di uscita
2. Ritardo di accodamento
   - Attesa di trasmissione
   - Livello di congestione del router
3. Ritardo di trasmissione (L/R)
   - R = frequenza di trasmissione del collegamento
   - L = lunghezza del pacchetto
4. Ritardo di propagazione (d/s)
   - d = lunghezza dl collegamento fisico
   - s = velocità di propagazione del collegamento ($\tilde{}2 \cdot 10^8$ m/s)



1.4.1 Ritardo di un nodo

Il ritardo di un nodo è dato dalla seguente formula:

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop} \tag{1.1}$$

dove:

- d_{proc} = ritardo di elaborazione (processing delay)
  - in genere di pochi microsecondi, o anche meno
- d_{queue} = ritardo di accodamento
  - dipende dalla congestione

- d_{trans} = ritardo di trasmissione (transmission delay)
  - significativo sui collegamenti a bassa velocità
- d_{prop} = ritardo di propagazione (propagation delay)
  - da pochi microsecondi a centinaia di millisecondi

Ritardo di accodamento

- R = frequenza di trasmissione (bps)
- L = lunghezza del pacchetto (bit)
- a = tasso medio di arrivo dei pacchetti

Se calcoliamo La/R otteniamo l'intensità di traffico:

- Vicino a 0: poco ritardo
- Minore o uguale a 1: traffico consistente
- Maggiore di 1: più lavoro in arrivo di quanto possa essere effettivamente svolto, ritardo medio infinito.

Il **throughput** viene calcolato come la frequenza (bit/unità di tempo) alla quale i bit sono trasferiti tra mittente e ricevente. Può essere istantaneo o medio (in un periodo più lungo).

In internet si considera anche il **collo di bottiglia**, ovvero un collegamento su un percorso punto-punto che vincola un throughput end to end.

1.5 Livelli di protocollo e i loro modelli di servizio

Si considerano principalmente 5 livelli di protocollo:

1. **Applicazione:** di supporto alle applicazioni di rete
2. **Trasporto:** trasferimento dei messaggi del livello applicazione tra modulo client e server, connessione tra processi applicativi
3. **Rete:** instradamento dei datagram dall'origine al destinatario
4. **Link *(collegamento)*:** instradamento dei datagram attraverso una serie di commutatori di pacchetto
5. **Fisico:** trasferimento dei singoli bit

1.6 Reti sotto attacco: la sicurezza

I malintenzionati installano malware negli host attraverso internet. Il malware può raggiungere gli host attraverso virus, worm o cavalli di Troia. Un malware di spionaggio può registrare quanto viene digitato, i siti visitati e informazioni di upload. Gli host infettati possono diventare botnet e essere usati per lo spamming e attacchi DDoS. Un malware è spesso auto-replicante e da un host attaccato può passare ad altri host.

Analisi di pacchetti Chiamato anche packet sniffing, quando un'interfaccia di rete legge/registra tutti i pacchetti che la attraversano.

IP spoofing Invio di pacchetti con un indirizzo sorgente falso.

Record-and-playback Vengono "sniffati" dati sensibili per utilizzarli in un secondo momento.

Capitolo 2

Il livello Applicazione

2.1 Principi delle applicazioni di rete

Le applicazioni di rete sono costruite con diverse architetture

- Client-Server
  - L'host (o client) interagisce direttamente con il server
  - Più client ci sono più il servizio diminuisce
- Peer-to-Peer (P2P)
  - Non c'è un server sempre attivo
  - Coppie arbitrarie di host comunicano direttamente fra di loro
  - Facilmente scalabile ma difficile da gestire
- Architetture ibride
  - Skype, messaggistica istantanea
  - Connessione client-client, utilizzo del server per la ricerca dell'indirizzo della parte remota
- Cloud computing
  - Un insieme di tecnologie che permettono sia di archiviare dati che elaborarli tramite l'utilizzo di risorse distribuite e virtualizzate in rete
  - Creazione di copie di sicurezza preventive in modo automatico trasferendo tutta l'operatività online
  - Dati memorizzati in server farm
  - Sempre client-server ma basata sulla virtualizzazione

Un processo è un programma in esecuzione su un host. All'interno dell'host due processi comunicano utilizzando schemi di interprocesso, mentre su host differenti comunicano attraverso lo scambio di messaggi.

- **Processo client:** processo che dà inizio alla comunicazione
- **Processo server:** processo che attende di essere contattato

Un processo invia/riceve messaggi mediante la sua socket. La socket è analoga ad una porta mediante la quale un processo che vuole inviare un messaggio lo fa uscire. Questo presuppone l'esistenza di un'infrastruttura esterna che trasporterà il messaggio attraverso la rete fino alla porta del processo di destinazione.

Per il trasporto internet vengono utilizzati i servizi dei protocolli TCP e UDP.

Servizio TCP

- Orientato alla connessione: è richiesto un setup fra i processi client e server
- Trasporto affidabile fra i processi d'invio e ricezione
- Controllo di flusso: il mittente non vuole sovraccaricare il destinatario
- Controllo della congestione: "strozza" il processo d'invio quando la rete è sovraccaricata
- **Non offre** temporizzazione, garanzie sull'ampiezza di banda minima, sicurezza
- **Più affidabile ma si paga con ritardi di trasferimento**

Servizio UDP

- Trasferimento dati inaffidabile fra processi d'invio e ricezione
- **Non offre** setup della connessione, affidabilità, controllo di flusso, controllo della congestione, temporizzazione né ampiezza di banda minima e sicurezza
- Mandare i dati il più velocemente possibile ma senza completa affidabilità, usato per gaming online e VoIP

2.2 Web e HTTP

L'HTTP, o HyperText Transfer Protocol, è un protocollo a livello applicazione per il web, che considera un client che richiede, riceve e visualizza gli oggetti del Web e un server, che invia gli oggetti in risposta ad una richiesta.

Viene utilizzato TCP nel seguente modo:

- Il client inizializza la connessione TCP con il server (crea una socket) tipicamente sulla porta 80
- Il server accetta la connessione TCP dal client
- Avviene lo scambio di messaggi HTTP tra browser e server web
- Chiusura della connessione TCP

HTTP è un protocollo *stateless* (senza stato), cioè il server non mantiene informazioni sulle richieste fatte dal client.

Le connessioni HTTP possono essere non persistenti o persistenti.

RTT Round trip time, tempo impiegato da un piccolo pacchetto per andare dal client al server e ritornare al client.

2.2.1 Connessioni non persistenti

Viene trasferito un solo oggetto su una singola connessione TCP. Il tempo di risposta è dato da:

- Un RTT per inizializzare la connessione TCP
- Un RTT per inviare la richiesta HTTP e i primi byte
- Il tempo necessario alla trasmissione del file

Quindi totale = 2 RTT + tempo di trasmissione. Le connessioni non persistenti presentano alcuni svantaggi:

- Richiedono 2RTT per ogni oggetto
- Overhead dell'OS per ogni connessione TCP
- I browser aprono spesso connessioni TCP parallele per caricare gli oggetti referenziati

2.2.2 Connessioni persistenti

Il server lascia la connessione TCP aperta dopo l'invio di una risposta, i successivi messaggi tra gli stessi client/server vengono trasmessi sulla connessione aperta, il client invia le richieste non appena incontra un oggetto referenziato, un solo RTT per tutti gli oggetti referenziati.

2.2.3 Messaggi HTTP

I messaggi HTTP possono essere di richiesta o di risposta.

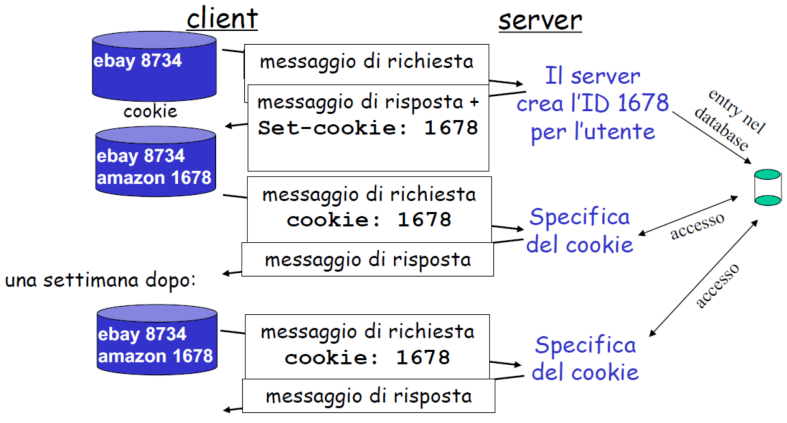
Richiesta HTTP

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close Accept-language:fr
```

Risposta HTTP

```
HTTP/1.1 200 OK
Connection close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 ...
Content-Length: 6821
Content-Type: text/html
```

2.2.4 Cookies



I cookies possono essere utilizzati per: autorizzazioni, dati carte per acquisti, raccomandazioni all'utente e possono mantenere lo stato del mittente e del ricevente per più transazioni; i messaggi HTTP trasportano lo stato. I cookies permettono ai siti di imparare molte cose sugli utenti.

2.2.5 Cache web (server proxy)

L'obiettivo di una web cache è quello di soddisfare la richiesta del client senza coinvolgere il server d'origine. La cache opera come client e come server, e consente di ridurre i tempi di risposta alle richieste dei client, riducendo il traffico sul collegamento a internet.

GET condizionale Ha l'obiettivo di non inviare un oggetto se la cache ne ha una copia aggiornata.
Per questo la cache specifica la data della copia dell'oggetto nella richiesta HTTP: `If-modified-since: <date>`
A lato server la risposta non contiene l'oggetto se la copia nella cache è aggiornata: `HTTP/1.0 304 not modified`

2.2.6 HTTP/2.0

È un'evoluzione di HTTP, che mantiene quindi i metodi HTTP, i codici di stato e la semantica migliorando però le prestazioni.

2.3 FTP

FTP, o File Transfer Protocol, viene utilizzato per trasferire file con un host remoto. Il server FTP opera solitamente sulla porta 21. Vengono aperte due connessioni, prima una di controllo e successivamente, se andata a buon fine la prima, una seconda per il trasferimento del file. La connessione di controllo è quindi "fuori banda" (out of band). Il server FTP mantiene lo stato

2.3.1 Comandi comuni

- `USER username`
- `PASS password`
- `LIST` elenca i file della directory corrente
- `RETR filename` recupera un file dalla directory corrente
- `STOR filename` memorizza un file nell'host remoto

2.3.2 Codici di ritorno comuni

- `331 Username OK, password required`
- `125 data connection already open; transfer starting`
- `425 Can't open data connection`
- `452 Error writing file`

2.4 Posta elettronica

La posta elettronica è costituita da tre componenti principali: agente utente, server di posta, SMTP (Simple Mail Transfer Protocol).

Agente utente Detto anche mail reader, si occupa della composizione, editing e lettura dei messaggi di posta elettronica (esempi: Outlook, Thunderbird). I messaggi in uscita sono memorizzati sul server.

Server di posta In esso è contenuta la casella di posta (mailbox) che contiene i messaggi in arrivo per l'utente e la coda dei messaggi da trasmettere.

Protocollo SMTP Utilizzato tra i server per inviare messaggi: il client è il server di posta trasmittente e il server il ricevente. Viene utilizzato TCP per trasferire in modo affidabile i messaggi di posta dal client al server sulla porta 25; il trasferimento è diretto. Ci sono tre fasi per il trasferimento: handshaking (saluto), trasferimento dei messaggi e chiusura. L'SMTP utilizza connessioni persistenti.

Esistono poi diversi protocolli di accesso alla posta:

- POP: Post Office Protocol. Utilizzato per autorizzazione e download
- IMAP: Internet Mail Access Protocol. Ha più funzioni e consente di manipolare i messaggi memorizzati sul server.
- HTTP: GMail, Hotmail, ecc...

POP3 Suddiviso in due fasi: autorizzazione e transazione, è un protocollo senza stato tra le varie sessioni

IMAP Mantiene tutti i messaggi sul server, consente all'utente di organizzare i messaggi in cartelle e conserva lo stato tra le varie sessioni (nomi delle cartelle, associazione tra identificatori dei messaggi e nomi delle cartelle).

2.5 DNS

Il DNS, Domain Name System è un protocollo a livello applicazione che consente a host, router e server DNS di comunicare per risolvere i nomi dei siti web. Il DNS traduce un hostname come *www.facebook.com* in un indirizzo IP.

Host aliasing In alcuni casi, un host può avere più nomi, come nel caso dell'aliasing dei server mail.

Il DNS è un database distribuito implementato in una gerarchia di server DNS. Tipicamente, un server DNS radice viene contattato da un server DNS locale che non può tradurre un nome. A sua volta, il DNS radice contatta un server DNS autorizzato se non conosce la mappatura del nome, la ottiene e la restituisce al server DNS locale.

Server TLD (top-level domain Si occupano dei domini .com, .org, .net, .edu, ecc. e di tutti gli altri domini locali di alto livello, quali .uk, .fr, .ca e .jp.

Server di competenza (authoritative server Ogni organizzazione dotata di host internet pubblicamente accessibili (server web e di posta) deve fornire i record DNS di pubblico dominio che mappano i nomi di tali host in indirizzi IP; possono essere mantenuti dall'organizzazione o dal service provider.

Il server DNS locale non appartiene strettamente alla gerarchia dei server, ciascun ISP (università, società) ha un server DNS locale. Quando un host effettua una richiesta DNS, la query viene inviata al suo server DNS locale, che opera da proxy e inoltra la query in una gerarchia di server DNS.

Una volta che un server DNS impara la mappatura la mette nella cache, dove le informazioni vengono invalidate dopo un certo periodo di tempo. Tipicamente, un server DNS locale memorizza nella cache gli indirizzi IP dei server TLD, quindi i server radice non vengono visitati spesso.

Il DNS è un database distribuito che memorizza i record di risorsa (RR) che hanno il seguente formato: `(name, value, type, TTL)`

- Type = A
  - `name` è il nome dell'host (server)

  - `value` è l'indirizzo IP
- Type = NS
  - `name` è il dominio (`foo.com`)
  - `value` è il nome dell'host del server DNS di competenza di questo dominio
- Type = CNAME
  - `name` è il nome alias di qualche nome canonico (`www.ibm.com`)
  - `value` è il nome canonico (`servereast.backup2.ibm.com`)
- Type = MX
  - `value` è il nome del server di posta associato a `name`

2.5.1 Protocollo DNS

Il protocollo DNS è costituito da domande (query) e messaggi di risposta, entrambi con lo stesso formato.

Intestazione del messaggio (12 byte - 6 campi)

L'intestazione è costituita da un numero da 16 bit di identificazione della domanda, utilizzato uguale dalla risposta, e da un flag indicante:

- domanda o risposta
- richiesta di ricorsione
- ricorsione disponibile
- risposta di competenza

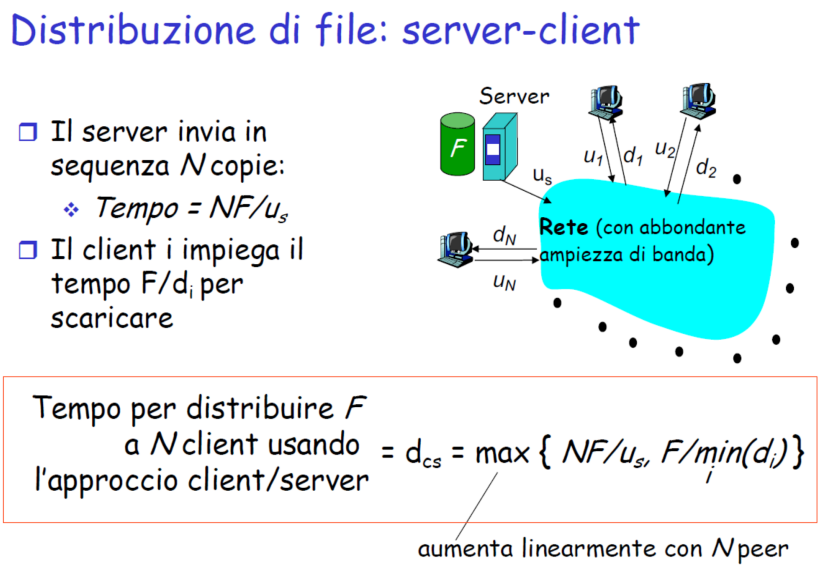
Resto del messaggio

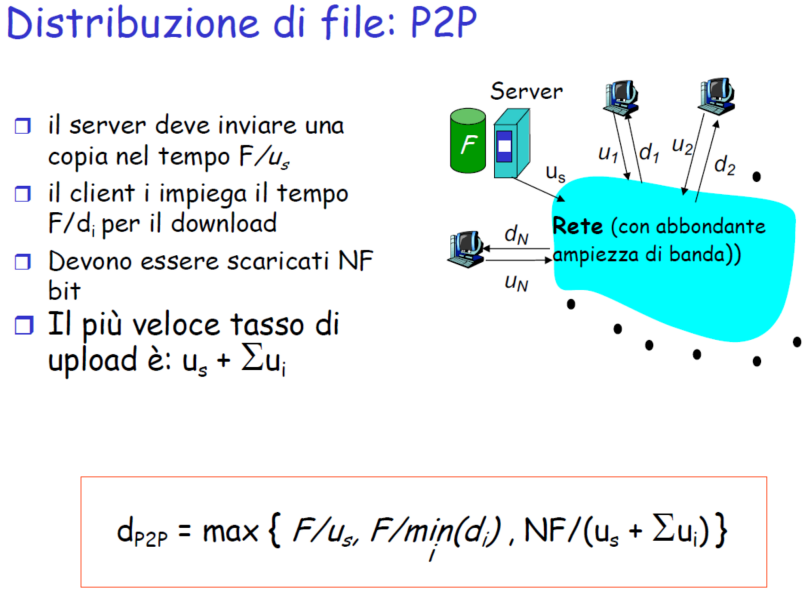
- Campi per il nome richiesto e il tipo di domanda
- RR nella risposta alla domanda
- Record per i server di competenza
- Informazioni extra che possono essere usate

2.6 Condivisione di file P2P

Nell'architettura P2P pura non c'è un server sempre attivo, infatti coppie arbitrarie di host (peer) comunicano direttamente tra loro. I peer non devono essere sempre attivi e possono cambiare indirizzo IP.

2.6.1 Confronto tra architettura server client e P2P





Tracker Il tracker tiene traccia dei peer che partecipano alla rete, dove un torrent è un gruppo di peer che si cambiano parti di un file.

Un file condiviso è diviso in più parti da 256 Kb, chiamati *chunk*. Il peer invia le sue parti a 4 vicini, quelli che gli stanno inviando alla frequenza più alta (aggiornata ogni 10 secondi). Successivamente, ogni 30 secondi viene scelto un peer a caso che riceve i chunk; oltre a questi 5 gli altri peer non riceveranno nulla.

Query flooding Ciascun peer indicizza i file che rende disponibili per la condivisione. Il messaggio di richiesta è trasmesso sulle connessioni TCP esistenti, il peer inoltra la richiesta e il messaggio di successo più trasmesso viene inviato sul percorso inverso.

2.7 Cloud Computing

L'architettura del cloud computing prevede uno o più server reali, generalmente in architettura ad alta affidabilità e fisicamente collocati presso i datacenter del fornitore del servizio.

2.7.1 CDN

Le CDN, o Content Delivery Networks, rappresentano una soluzione comune per la realizzazione di servizi su internet, la quale costruisce una rete "overlay" per la distribuzione di contenuti. Serve per memorizzare i dati il più vicino possibile ai consumatori in modo da ottimizzare le prestazioni di rete, ridurre la latenza ed evitare colli di bottiglia.