Riassunto Reti 1

Introduzione

Cos'è Internet

Internet è una rete di calcolatori pubblica in cui miliardi di dispositivi sono interconnessi tra loro, ogni dispositivi, detto host, è connesso a reti di collegamento e a commutatori di pacchetto e invia /riceve dati da ulteriori host utilizzando i pacchetti. Ogni pacchetto passa da un host all'altro attraverso le reti di collegamento, esse possono avere velocità di trasmissione differenti (misurabili in bit/s), passando da un commutatore all'altro, questi ultimi possono essere di due categorie:

- router: commutatori a livello di rete che instradano i pacchetti verso il nucleo della rete;
- i commutatori a livello collegamento instradano i pacchetti verso una rete di accesso oppure all'interno di una sottorete.

Gli host accedono a interno attraverso gli Internet Service Provider (ISP), essi sono formati da insiemi di reti di collegamento e commutatori e forniscono l'accesso alla rete a banda larga. Ogni rete ISP è gestita in modo indipendente e fa uso del protocollo IP (Internet Protocol) per quanto riguarda nomi e indirizzi, gli host e i commutatori utilizzano invece protocolli che favoriscono l'invio/ricezione di dati all'interno della rete, essi sono TCP (Transmission Control Protocol) e IP. I servizi di Internet vanno dalla posta elettronica alla navigazione web, esse sono tutte applicazioni distribuite dato che più host si scambiano dati in modo reciproco, esse non vengono eseguite sugli host stessi e non sui commutatori, il loro unico compito è solamente quello di portare i pacchetti a destinazione. Gli host forniscono un'interfaccia socket, un'insieme di regole che specifica il modo in cui un programma possa inviare/ricevere dati da ulteriori programmi eseguiti su altri host.

Cos'è un protocollo

Un protocollo è un'insieme di regole che definisce il formato e l'ordine dei messaggi scambiati tra due host, essi vengono utilizzati in modo estensivo in Internet e qualsiasi tipo di comunicazione tra due o più entità è governata da essi.

Confini della rete

Tutti gli host fanno parte del bordo o confine della rete, essi possono essere suddivisi in:

- Client: host che richiedono servizi in Internet;
- Server: host che forniscono dei servizi, al giorno d'oggi è più giusto parlare di data center, delle strutture in cui risisedono milioni di server.

Reti di accesso

Accesso residenziale

Gli accessi residenziali più diffusi sono l'accesso DSL (Digital Subscriber Line) e la connessione via cavo, la prima è solitamente fornita dalla compagnia telefonica, rendendolo in questo modo anche un ISP. Il DSL utilizza un modem che converte i dati in input e li alla centrale locale tramite il cavo telefonico, all'interno di una centrale vi è una DSLAM (DSL Access Multiplex), il cui compito è quello di dividere i dati internet da quelli telefonici e instradarli nella giusta rete. Tutto questo avviene in contemporanea alle chiamate telefoniche grazie all'utilizzo di 3 bande di frequenza:

- la prima è il canale telefonico che va da 0 a 4kHz;
- la seconda è il downstream, ovvero la parte verso l'abitazione e va da 50k a 1 MHz;
- la terza è l'upstream verso la DSLAM che va da 4 a 50 kHz.

La DSL può essere asimmetrica quando la velocità in download differisce da quella in upload, quest'ultima non è mai quanto quella indicata dal contratto perchè entrano in gioco limiti fisici quali la distanza dell'abitazione dalla centrale, il numero stesso di abitazioni, di utenti connessi, ecc.
La connessione a internet via cavo utilizza l'infrastruttura della televisione via cavo, funzionando in un modo simile a DSL, tutte le abitazioni di un quartiere sono connesse alla dorsale tramite un cavo coassiale, il quale porterà i dati verso una giunzione e da lì vanno verso la stazione attraverso un canale in fibra ottica (questo sistema è infatti detto HFC, ovvero Hybrid Fiber Coax). In ogni abitazione vi è un cable modem la cui funzione è simile a quella di un DSL modem, nella stazione vi è il CMTS (Cable Modem Termination System) la cui funzione è simile a quella della DSLAM nelle reti DSL. Una caratteristica importante del sistema HFC è il fatto che rappresenta un mezzo condiviso, infatti ogni pacchetto inviato dalla stazione viaggia verso tutti i collegamenti e abitazioni e viceversa, per questo motivo la velocità effettiva è significamente inferiore rispetto a quella data. La soluzione più recente è FTTH (Fiber To The Home) e permette l'accesso a internet attraverso un collegamento in fibra ottica, ogni abitazione ha un ONT (Optical Network Terminator) con una funzione simile a quella del modem DSL, esso è connesso a uno splitter di quartiere, quest'ultimo permette di connettere più abitazioni a un unico collegamento in fibra verso la centrale. All'interno della centrale vi è l'OLT (Optical Line Terminator), il quale converte i segnali ottici in elettrici e li invia al router della compagnia telefonica. Questo tipo di architettura è detta PON (Passive Optical Networks) e in essa ogni pacchetto inviato dall'OLT allo splitter viene replicato su tutte le linee in modo simile a HFC. Nelle località in cui non è disponibile la rete DSL è possibile collegarsi a internet attraverso i satelliti, l'accesso in dial-up è basato sulle reti DSL e purtroppo sono molto lente.

Accesso aziendale

L'accesso di tipo aziendale a Internet è solitamente utilizzato nelle aziende e nelle università, in esse gli host sono connessi al router attraverso una Local Area Network (LAN) tramite apparati appositi. Dal momento che sempre più utenti si connettono tramite dispositivi senza fili, si è deciso di aggiungere gli access point alla LAN, creando in questo

modo una Wireless LAN (WLAN). Oltre all'utilizzo aziendale/universitario, le reti LAN e WLAN stanno prendendo piede anche nelle normali abitazioni.

Accesso su scala geografica

Gli smartphone utilizzano l'infrastruttura wireless della telefonia cellulare per l'invio/ricezione di pacchetti tramite la stazione telefonica, esse a differenza del normale wi-fi possono scambiare dati anche per decine di chilometri anziché le poche decine di metri. Da queste infrastrutture sono in seguito nate le reti wireless di terza generazione (3G), consentendo l'accesso a internet con velocità superiore a 1 Mbps, e le reti di quarta generazione (4G) che sfrutta la basi della precedente per raggiungere velocità superiori a 10 Mbps.

Mezzi Trasmissivi

Tutti i dati che partono/arrivano da ogni host passano in un mezzo trasmissivo, esso propa i bit più volta passando da un dispositivo a un altro fino alla destinazione. Un mezzo trasmissivo deve essere per necessità dello stesso tipo per ogni coppia trasmettitore - ricevitore e può essere:

- vincolato: le onde sono contenute nel mezzo fisico stesso;
- non vincolato: le onde si propagano nell'atmosfera o nello spazio esterno.

I mezzi trasmissivi rappresentano di per sè la parte meno costosa all'interno di una rete, il costo della loro installazione fisica però lo può superare di vari ordini di grandezza.

Doppino intrecciato

Il doppino intrecciato è il mezzo trasmissivo meno costoso e più utilizzato, è presente infatti nella rete telefonica e negli ambienti casalinghi/lavorativi, è formato da due fili di rame intrecciati per ridurre l'interferenza elettrica, più doppini possono essere raggruppati per formre un cavo UTP, utilizzato solitamente nelle reti LAN. La velocità di trasmissione dipende dallo spessore del filo e dalla distanza tra trasmettitore e ricevitore.

Cavo coassiale

Il cavo coassiale è tipicamente utilizzato per la rete televisiva ed è formato da un filo interno, rappresentante il polo caldo, e da una maglia di rame esterna che funge sia da secondo filo che da schermatura per le interferenze elettromagnetiche. Questo mezzo permette la comunicazione bidirezionale e l'utilizzo di canali multipli su di esso (la cosiddetta broadband).

Fibra ottica

La fibra ottica è un mezzo che sfrutta la rifrazione della luce per l'invio/ricezione dei pacchetti attraverso segnali luminosi, ciò lo rende immune da ogni tipo di interferenza elettromagnetica e i suoi ripetitori possono trovare anche molto distanti tra loro, favorendolo in questo modo per quanto riguarda i collegamenti intercontinentali e le dorsali Internet (nonostante l'alto costo d'installazione ne abbia impedito l'utilizzo a corto raggio).

Canali Radio

I canali radio permettono di trasportare segnali all'interno dello spettro elettromagnetico, permettendo l'invio/ricezione dei dati senza l'ausilio di ulteriori cavi. I segnali però possono essere soggetti a riflessioni, interferenze o possono essere ostruiti da oggetti. I canali radio possono essere:

- terrestri: possono essere utilizzati a corte, medie e lunghe distanze a seconda dell'utilizzo;
- satellitari: vi è un ritardo di propagazione di 280 ms e possono essere geostazionari o a bassa quota.

Il nucleo della rete

Il nucleo della rete è formato dai router e dai relativi collegamenti che permettono ai router di arrivare a destinazione.

Reti a commutazione di pacchetto

Le reti a commutazione di circuito sono reti in cui i pacchetti vengono distribuiti tramiti dispositivi detti appunto commutatori di pacchetto (ad esempio i router), essi inviano i pacchetti da un commutatore all'altro attraverso a una velocità pari o superiore alla velocità totale di trasmissione del collegamento,il tempo di trasmissione si calcola attraverso il rapporto tra la dimensione del pacchetto e la velocità della linea:
Ttx=DimPacchetto/Vtx
Tecnica Store-and-Forward
la tecnica store-and-forward è una tecnica in cui il commutatore di pacchetto deve ricevere tutto il pacchetto prima di trasmettere il primo bit agli altri, supponendo che tutte le linee degli n-1 router tra sorgente e destinazione abbiano la stessa velocità, il tempo di trasmissione totale sarà il tempo di trasmissione per il numero totale di linee:
Tendtoend=n*Ttx

Ritardi e perdite di pacchetti

Ogni pacchetto che arriva a un commutatore viene messo in una coda per poi essere inviato nuovamente verso la destinazione, esso subisce il cosiddetto ritardo d'accodamento, ovvero il tempo di permanenza del pacchetto all'interno della coda, esso è variabile e dipende dal livello di traffico della rete. Dato che i commutatori non hanno dimensione infinita, vi possono essere casi in cui non è possibile memorizzare ulteriori pacchetti e di conseguenza si verificano delle perdite.

Tabelle e protocolli di instradamento

Ogni host viene identificato in una rete attraverso un indirizzo IP, ogni pacchetto infatti contiene un'intestazione in cui è presente l'indirizzo di destinazione, il quale presenta una struttura gerarchica: quando il pacchetto giunge al router della rete, esso esamina una parte dell'indirizzo e lo inoltra a quello adiacente, ogni router possiede infatti una tabella di instradamento in cui gli indirizzi IP sono messi in relazioni coi collegamenti in uscita. Per

quando riguarda l'instradamento end-to-end, il pacchetto rimbalza tra i router del nucleo fino a quando non arriva al router della rete in cui è presente la destinazione, lì viene controllato il suo indirizzo IP e viene inoltrato al relativo host. La compilazione della tabella di instradamento avviene attraverso i cosiddetti protocolli di instradamento, essi possono determinare il percorso più corto e utilizzarlo per compilare le tabelle.

Reti a commutazione di circuito

Le reti a commutazione di circuito son reti in cui i pacchetti vengono inoltrati in un percorso predeterminato, riservando le risorse richieste per tutta la durata della connessione, esse infatti non sono riservate nelle reti a commutazione di pacchetto, di conseguenza vengono utilizzate solamente quando necessario (in questo modo però i pacchetti potrebbero restare in attesa per accedere a un collegamento). Prima della trasmissione di dati, la rete deve instaurare una connessione col destinatario, riservando una frazione della velocità di trasmissione (equivalente a Vtx/numCollegamenti) in maniera costante, permettendo così l'invio/ricezione dei dati a una velocità garantita.

Multiplexing nella commutazione di circuito

Il circuito è implementato tramite il multiplexing, esso può essere in due modi:

- FDM: lo spettro di frequenza viene suddiviso tra le connessione stabilite sul collegamento stesso, dedicando una banda di frequenza a ognuna per tutta la durata della connessione;
- TDM: Il tempo viene suddiviso in frame di durata fissa, i quali vengono ripartiti in un numero fisso di slot, ogni connessione stabilita avrà uno slot temporale unicamente dedicato in ogni frame;

Un problema delle reti a commutazioni di circuito è il fatto che nei "periodi di silenzio", le risorse sono sottoutilizzate o non utilizzate nei casi peggiori, di conseguenza non possono essere sfruttate da altre connessione a causa della riservatezza. Al contrario, le reti a comunicazione di pacchetto, nonostante offra una miglior condivisione della banda e un utilizzo più semplice ed efficiente, non è adatto ai servizi in tempo reale. La commutazione di pacchetto rispetto a quello di circuito può avere anche prestazioni migliori, la differenza sta nella condivisione di risorse: se la prima garantisce una velocità minima riservando le richieste per tutta la connessione, la seconda utilizza in modo più efficiente le risorse senza però garantire una velocità minima.

Una rete di reti

Gli host si connettono a internet tramite l'ISP, la connessione host-ISP è solo una piccola parte delle cose che costituiscono internet. Per connettere tutti gli host, infatti, occorre connettere gli ISP tra loro, rendendo così internet una rete di reti. Un primo approccio è connette ogni ISP a tutti gli altri attraverso uno o più collegamenti, questa soluzione è però costosa. Un secondo approccio è connettere tutti gli ISP a un unico ISP di transito, questa soluzione però non è conveniente in quanto l'ISP di transito dovrebbe farsi pagare per ogni ISP connesso. L'ultimo approccio è simile al primo con la differenza che vi sono più ISP di transito e ogni ISP può connettersi a quello che vuole, permettendo così un'offerta più varia di costi e servizi. Essa è una gerarchia in cui gli ISP di transito sono interconnessi tra loro e

si trovano nella parte alta, in quella bassa invece vi sono i normali ISP. Gli ISP si connettono a ISP regionali, a loro volta connessi a ISP di primo livello, Questi ultimi tipi di ISP sono in competizione tra loro per ottenere ulteriori ISP. Un normale ISP può anche connettersi direttamente a un ISP di primo livello, pagando in modo diretto i servizi offerti. I PoP (point of presence) sono gruppi di router vicini tra loro in cui gli ISP possono connettersi ai fornitori contenuti. Per ridurre i costi causati da ISP connessi a più ISP fornitori (multi-homing), viene effettuato il peering, ovvero si connettono direttamente due ISP vicini in modo che il traffico passi direttamente da un ISP all'altro senza un intermediario. Le aziende possono creare i cosiddetti IXP (Internet exchange point), ovvero punti di incontro dove più ISP possono fare peering tra loro. Oltre agli ISP, in internet sono sempre più frequenti i fornitori di contenuti, essi fanno peering direttamente con altri ISP oppure attraverso un IXP.

Ritardi e perdite nella commutazione di pacchetto

Purtroppo i servizi internet non permettono la spostamento di quantità di dati in modo istantaneo e senza perdite tra due host, le reti infatti limitano il throughput, ovvero la quantità di dati al secondo trasferibile tra due host, introducendo ritardi e a volte causando delle perdite. Un pacchetto è soggetto ai seguenti tipi di ritardo:

- ritardo di elaborazione: tempo richiesto per esaminare l'intestazione del pacchetto per determinare in quale linea inoltrarlo, è spesso trascurabile ma può influenzare anche pesantemente il throughput;
- ritardo di accodamento: tempo di attesa del pacchetto all'interno del buffer del router, esso dipende dalla quantità di traffico nel collegamento stesso e dal numero di pacchetti arrivati in precedenza.
- ritardo di trasmissione: tempo richiesto per inviare un pacchetto da un host/router a un altro, esso dipende dalla dimensione del pacchetto e dalla velocità di trasmissione, infatti si calcola attraverso il loro rapporto (dimPacchetto/Vtx). NON DIPENDE DALLA DISTANZA.
- Ritardo di propagazione: tempo richiesto dal primo bit di un pacchetto di propagarsi in un collegamento, esso si calcola rapportando la distanza tra gli host/router con la velocità di propagazione del mezzo, di solito equivalente alla velocità della luce (d/Vp =d/c). NON DIPENDE DALLA GRANDEZZA DEL PACCHETTO O DALLA VELOCITà DI TRASMISSIONE.

Il ritardo di accodamento, a differenza degli altri, non dipende da variabili note ma varia da pacchetto a pacchetto: un buffer vuoto infatti permette di ottenere un ritardo praticamente nullo sul primo pacchetto, esso crescerà al'aumentare del numero di pacchetti che arrivano contemporaneamente. Per misurare il ritardo di propagazione si fa quindi utilizzo di misure statistiche. Il rapporto dimPacchetto*Vmedia/bitRate è l'intensità di traffico e spesso assume un ruolo importante nella stima del ritardo d'accodamento, infatti:

- se è maggiore di 1, la velocità media d'arrivo dei bit supera quella di ritrasmissione in uscita, quindi la coda tenderà a crescere all'infinito.
- se è minore o uguale a 1, la natura del traffico influisce sulla coda: se i pacchetti arrivano periodicamente, la coda sarà sempre vuota e quindi non ci saranno ritardi d'accodamento, altrimenti se arrivano a raffiche periodiche, si possono verificare ritardi di lunghezza media.

Il processo d'arrivo in coda tuttavia non segue uno schema predefinito e quindi i pacchetti arrivano a intervalli casuali. Dal momento che la capacità della coda non è infinita, l'arrivo di un pacchetto quando essa è piena ne comporta la perdita, essa verrà considerata come un pacchetto che non è mai arrivato a destinazione (anche se effettivamente c'é arrivato), di conseguenza verrà richiesta la ritrasmissione. Quindi le prestazioni non vengono solo misurate in termini di ritardo ma anche in numero di pacchetti persi. Il ritardo end-to-end tra una sorgente e una destinazione è dato dalla somma dei ritardi di propagazione, trasmissione e elaborazione moltiplicati per il numero di linee tra essi.
Ttot=(Tel + Ttx+ Tpr)*numLinee
Una misura efficiente dei ritardi all'interno delle reti è possibile attraverso il traceroute, esso funziona inviando dei pacchetti speciali verso una destinazione determinata dall'utente, essi passano attraverso una serie di router i quali, dove aver capito che il pacchetto è speciale, inviano un messaggio contenente nome e indirizzo IP alla sorgente. In questo modo è possibile ricostruire il percorso intrapreso dai pacchetti determinandone anche i ritardi di andata e ritorno per ogni tratto, questo esperimento viene effettuato tre volte da Traceroute stesso. Ulteriori ritardi riguardano gli host , essi infatti possono ritardare volontariamente la trasmissione dei pacchetti per poter condividere il mezzo con altri host, poi c'è il ritardo di pacchettizzazione, ovvero il tempo di trasposizione dei pacchetti di un flusso multimediale: per poter essere trasmesso, il pacchetto deve essere completamente riempito, codificato e in seguito inviato in internet. Un'altra misura che determina le prestazioni di una rete è il throughput end-to-end, esso può essere:

- istantaneo: la velocità in bps alla quale un router/host sta ricevendo un file in ogni istante;
- medio: Dato un file, il throughput medio è la velocità in bps del trasferimento del file, si calcola rapportando la dimensione del file al tempo di trasferimento (dimFile/Ttx).

Per una rete formata da due soli collegamenti, il throughput è il bitrate minimo tra i due, di conseguenza il throughput medio di un file è il rapporto tra la dimensione del file e il minimo tra le due linee, questo vuol dire che la velocità di trasmissione di uno dei due collegamenti fa da collo di bottiglia. Nel caso in cui N client e N server condividono un collegamento, esso deve spartire la velocità di trasferimento tra tutti gli N client, questo vuol dire che il collo di bottiglia non è più nella rete di accesso ma nel collegamento condiviso nel nucleo. Quindi il throughput dipende dalla velocità di trasmissione dei collegamenti sui quali passano i dati, in caso di assenza di traffico, si può approssimare alla velocità di trasmissione minima lungo il percorso.

Livelli di protocolli e modelli di servizio

Tutti i componenti che hanno un ruolo nella rete e in Internet sono caratterizzati da un'architettura a livelli, essa consente di definire sistemi articolati e complessi e grazie alla modularità permette una semplificazione degli stessi. In questo modo è possibile cambiare l'implementazione di un livello senza dover fare lo stesso anche con gli altri. Ogni livello o strato utilizza dei servi appartenenti al precedente per fornirne altrettanti al successivo. I protocolli che utilizzano questo modello possono essere implementati via software, hardware o un misto tra le prime due. La stratificazione dei protocolli presenta vantaggi sia concettuali sia strutturali, infatti fornisce un modo strutturato per trattare i componenti e la modularità

rende più semplici gli aggiornamenti. L'unico svantaggio è la possibile ridondanza che si va a creare quando un livello duplica le funzioni di quello inferiore.

L'incapsulamento

L'incapsulamento permette ai singoli livelli di introdurre ulteriori informazioni ai messaggi che arrivano dai livelli più alti, esse saranno utilizzaterì dai livelli riceventi per capire a chi si riferisce il messaggio e quindi consegnarglielo.

Reti sotto attacco

Dal momento che miliardi di persone e cose sono connesse a Internet, sono necessarie delle impostazioni per evitare venga violata la privacy o il malfunzionamento dei servizi da cui dipendono. Tra i possibili attacchi informatici vi sono i malware, dei software malevoli che possono penetrare all'interno dei dispositivi e infettarli cancellando file o raccogliendo informazioni private per darle a malintenzionati. I dispositivi infettati possono essere inseriti in una botnet, una rete utilizzata dai malintenzionati per effettuare spam o attacchi DoS. I malware odierni sono autoreplicanti, ovvero che cercano altri host da infettare ogni volta che un host è stato infettato. Un malware può essere di differenti tipi, i principali sono:

- virus: malware che richiedono l'interazione dell'utente per infettare un dispositivo;
- worm: malware che possono entrare nei dispositivi senza interazioni con l'utente, per esempio passando attraverso la rete locale.

Attacchi alle infrastrutture

La maggior parte degli attacchi è di tipo Dos (Denial of Service), ovvero rendere inutilizzabile un servizio da parte degli utenti, essi possono essere di tre categorie:

- Attacchi allla vulnerabiltià degli host: consiste nell'invio di pochi messaggi ben costruiti a un'applicazione vulnerabile dell'host bersaglio;
- Inondazione di banda: si inviano grandi quantità di messaggio che intasano i collegamenti di accesso e quindi impedendo di raggiungere il server;
- Inondazione di connessione: vengono stabilite numerose connessioni TCP completamente o parzialmente aperte all'host bersaglio, non permettendo l'utilizzo delle risorse dato che sono "occupate".

Per rendere più efficiente un DoS, viene utilizzata una botnet, in questo modo risulta difficile difendersi e individuare l'attacco.
Oltre agli attacchi DoS, vi sono anche i packet sniiffer, dei ricevitori passivo che possono ottenere una copia di ogni pacchetto trasmesso, essi possono anche essere utilizzati in ambienti cablati e studiati successivamenti per ricavare informazioni sensibili. Dal momento che i packet sniffer sono passivi, essi sono difficili da individuare e quindi possono ottenere copie di pacchetti senza essere scoperti, un modo per difendersi da essi è la crittografia.

Mascheramento

é possibile creare un pacchetto con un indirizzo sorgente falso e inviarlo nella rete, i ricevitori di questo pacchetto non si accorgeranno di nulla, questa tecnica è detta IP spoofing e permette agli utenti di spacciarsi per altri. Per risolvere questo problema, bisogna

utilizzare un meccanismo di autenticazione in cui bisogna avere la certezza che i messaggi inviati provengano effettivamente da dove si suppone.

Livello Applicazione

Il livello applicazione si trova in cima a tutti gli altri nel modello TCP/IP e in esso si trovano i protocolli e le applicazioni di rete. Il cuore delle applicazioni di rete è costituito dai programmi eseguiti sugli host, per essi non occorre predisporre programmi per i dispositivi del nucleo della rete, essa ha un'architettura fissata che fornisce uno specifico uso dei servizi e stabilendo l'organizzazione dei sistemi, essa può essere:

- Client-Server: vi sono host sempre attivi, detti server, che rispondono alle esigenze di ulteriori host detti client, questi ultimi non comunicano direttamente tra loro e possono contattare il server in qualsiasi momento.
- P2P (Peer to peer): l'infrastruttura dedicata ai server è minima o addirittura assente e gli host possono essere sia client che server, rispondendo alle esigenze e chiedendone a ulteriori host. La comunicazione avviene in maniera diretta tra coppie di host detti peer, essi sono controllati dagli utenti e non dai fornitori di servizi. Un vantaggio è la sua scalabilità: ogni peer infatti può richiedere ad altri un file e quindi aggiunge capacità di servizio al sistema, rispondendo così alle richieste di altri peer, in più non ha bisogno di server e infrastrutture ad-hoc. Gli unici svantaggi riguardano la sicurezza, le prestazione e l'affidabilità.

Un'applicazione di rete è formata da processi comunicanti appartenenti a differenti host, essi comunicano scambiandosi messaggi attraverso la rete, essi possono essere client o server, il primo è quello che avvia la comunicazione mentre il secondo è quello che attende. I processi si interfacciano alla rete attraverso astrazioni software dette socket, ogni messaggio inviato passa attraverso questa interfaccia la quale lo esporterà nella rete. Allo stesso modo, i socket dell'host destinatario ricevono il messaggio dalla rete e lo recapitano al processo corretto, esso è quindi un punto d'incontro tra il livello applicazione e il livello trasporto. I progettisti esercitano un totale controllo del livello applicazione ma ben poco sul livello trasporto, esso infatti può solamente scegliere il protocollo e alcuni parametri relativi a quest'ultimo livello.

Indirizzamento

Affinchè la consegna avvenga nel modo corretto, il destinatario deve avere un indirizzo mentre il processo è identificato, oltre che dall'indirizzo stesso, anche da un identificativo. Per identificare un processo, o meglio il socket che riceve i dati (necessario perché più applicazioni di rete potrebbero essere in esecuzione), viene utilizzato un numero, per le applicazioni più note sono stati utilizzati numeri di porta specifici.

Servizi di trasporto

Le reti mettono a disposizione vari protocolli di trasporto, tra essi vi sono quelli di trasferimento dati affidabile: dato che la perdita di pacchetti può portare a gravi conseguenze, bisogna garantire che i dati inviati vengono consegnati correttamente e completamente. Un servizio importante è il trasferimento affidabile di processi, il processo mittente può passare i propri dati al socket e sapere con certezza che essi arriveranno

correttamente al destinatario. Quando un protocollo non fornisce il trasferimento affidabile, vuol dire che i dati potrebbero non arrivare mai a destinazione, ciò è ammissibile per le applicazioni che tollerano le perdite.

Throughput

Un'applicazione di rete può richiedere un throughput garantito e i protocolli assicureranno che ciò avvenga, se non è possibile, l'applicazione dovrà codificare i dati a un livello inferiore oppure rinuncia. Questo tipo di applicazione è detto sensibile alla banda, nonostante alcune di esse utilizzano tecniche adattative per la codifica. Le applicazione elastiche invece possono far utilizzo di tanto o poco throughput a seconda della disponibilità.

Temporizzazione

Un protocollo può anche fornire garanzie di temporizzazione, questo tipo di servizio può interessare le applicazione in tempo reale le quali, per essere efficaci, richiedono stretti vincoli temporali per la consegna dei dati. Per gli altri tipi di applicazione, invece, sono preferibili ritardi inferiori rispetto a quelli più consistenti, ma non sono presenti stretti vincoli per i ritardi end-to-end.

Sicurezza

I protocolli possono anche fornire uno o più servizi di sicurezza, essi possono crittografare i dati del mittente e decifrarli prima di darli al ricevente.

Servizi di trasporto di Internet

I protocolli di trasporto messi a disposizione da Internet sono TCP e UDP. TCP prevede un servizio orientato alla connessione e il trasporto affidabile dei dati, esso fa in modo che client e server si scambiano informazioni di controllo prima dei messaggi veri e propri, il cosiddetto handshaking. La connessione è full-duplex, ovvero che il client e il server possono inviare e ricevere messaggi in contemporanea. Grazie a questo protocollo lo scambio di dati avviene correttamente e senza errori, vi è un meccanismo di controllo delle congestione "strozzando" il processo mittente quando il traffico è eccessivo. UDP invece è un servizio minimalista, è senza connessione (non necessitando quindi di handshaking) e fornisce un servizio di trasporto non affidabile, non garantendo che i messaggi arrivino a destinazione, non vi è nessun controllo della congestione e quindi i dati possono essere inviati al livello di rete a qualsiasi velocità.

Protocolli a livello applicazione

I protocolli applicativi definiscono come i processi si scambiano i messaggio, definendone il tipo, la sintassi, la semantica dei campi e le regole per l'invio/ricezione. Alcuni protocolli sono di dominio pubblico, definendo regole che devono essere rispettate per garantire un corretto funzionamento, altri invece sono privati e volutamente non disponibili al pubblico. Un protocollo applicativo è solo una parte di un'applicazione di rete, essa consiste anche in standard per i formati di documento, browser e web server, il protocollo definisce infatti il formato e la sequenza di messaggi scambiati tra browser e web server.

Web e HTTP

Il web è stata la prima applicazione internet ad aver catturato l'attenzione del pubblico cambiando il modo di interagire con gli ambienti di lavoro e non solo. Il web funziona utilizzando il protocollo HTTP, un protocollo che permette di visualizzare pagine web. Una pagina web è costituita da oggetti: un file HTML principali e i vari contenuti multimediali, essa può essere visualizzata attraverso un browser che implementa appunto HTTP. Quando l'utente richiede una pagina web, il browser invia richieste HTTP al server interessato, quest'ultimo risponde con una risposta HTTP contenente gli oggetti. HTTP utilizza TCP come protocollo di trasporto, questo vuol dire che a ogni richiesta viene instaurata una connessione. Il server invia file ai client senza memorizzare informazioni, infatti invierà nuovamente il file anche quando è stato richiesto in precedenza, questo permette a HTTP di essere classificato come protocollo stateless.

Connessioni persistenti e non

Le richieste effettuate da un applicazione possono essere effettuate in sequenza, a intervalli regolari oppure a intermittenza. Quando le richieste/risposte vengono inviate separatamente, viene utilizzata la cosiddetta connessione non persistente, ovvero che per ogni richiesta/risposta viene instaurata una connessione, al contrario di quella persistente che invia tutto utilizzando un'unica connessione. Per quanto riguarda le connessioni persistenti, il processo client inizializza la connessione TCP con la porta 80 (quella di HTTP), poi tramite i socket invia al server una richiesta HTTP. Quest'ultimo, una volta ricevuta la richiesta, incapsula l'oggetto in un messaggio e lo invia al client, comunicando inoltre di chiudere la connessione TCP (non termina fino a quando non è certo che abbia ricevuto la risposta). Tutti questi passaggi vengono ripetuti per ogni oggetto appartenente al file HTML.Di default i browser aprono dalle 5 alle 10 connessioni TCP parallele, gestendole in contemporanea e quindi risparmiando tempo. Il tempo che impiega un pacchetto ad andare e tornare è detto Round Trip Time (RTT), esso include tutti i ritardi di elaborazione, propagazione e accodamento e permette un calcolo approssimato del tempo impiegato al download di una pagina web. Dato che ogni connessione effettua l'handshaking prima dell'invio effettivo dei dati più una conferma della chiusura di una connessione, il tempo di download approssimato equivale a due RTT più il tempo di trasmissione dell'oggetto.

RTT= Tel + Tpr + Tac

Tdown=2*RTT + Ttx

Le connessioni non persistenti, a causa della loro "natura", devono allocare continuamente nuovi buffer e variabili, in più il ritardo di 2 RTT pone dei limiti per quanto riguarda il tempo.

Nel caso in cui la connessione è persistente, vi è un unico handshake e il file HTML e gli oggetti utilizzando essa, senza aprire ulteriori connessioni,La connessione viene chiusa quando non viene utilizzata per un certo intervallo di tempo. Il tempo totale di download sarà quindi data dal tempo dell'handshake, dal tempo di trasferimento del file HTML e da quello di tutti gli oggetti.

Tdown=3/2 RTT + Thtml + somma(Tobj)

Formato dei messaggi HTTP

Richieste

I messaggi HTTP vengono scritti in testo ASCII in modo da essere leggibili dagli utenti, i messaggi di richiesta possono essere costituiti da un numero indefinito di righe:

- la riga di richiesta comprende tre campi, il metodo, l'URL e la versione di HTTP. Il campo metodi può assumere diversi valori, essi corrispondono alle azioni da eseguire su URL;
- La riga Host specifica l'host in cui risiede l'oggetto;
- La riga Connection: close indica che il browser sta comunicando al server che non deve occuparsi delle connessioni persistenti, vuole però che la chiuda una volta inviato l'oggetto richiesto;
- User-agent: specifica il browser utilizzato, utile perchè il browser potrebbe avere diverse versioni dell'oggetto per diversi tipi di browser.
- Accept-language: indica la lingua dell'utente.

Dopo l'intestazione vi è il corpo, esso sarà vuoto nel caso di GET ma viene utilizzato da POST quando l'utente riempie un form, alcuni form HTML utilizzano il metodo GET includendo i dati nell'URL. Il metodo HEAD è simile a GET, con la differenza che, quando il server riceve una richiesta HEAD, risponde al messaggio tralasciando gli oggetti, esso serve per verificare la correttezza del codice. Il metodo PUT permette l'invio di un oggetto a un percorso specifico su uno specifico web server, esso viene utilizzato dalle applicazioni che inviano oggetti ai web server. Il metodo DELETE invece cancella oggetti da un server.

Risposte

Le risposte HTTP sono formate da tre sezioni: la riga di stato iniziale, l'intestazione e il corpo (il fulcro del messaggio). La riga di strato è composta da tre campi: la versione del protocollo, il codice e il messaggio di stato, le altre 6 righe sono:

- Connection:close comunica al client che chiude la connessione dopo l'invio del messaggio;
- Date indica data e ora della creazione;
- Server indica che il messaggio è stato da un web server di un certo tipo;
- Last-Modified indica la data dell'ultima modifica (importante per la gestione in cache);
- Content-Length contiene il numero di byte dell'oggetto inviato.
- Content-type indica il tipo dell'oggetto.

Il codice di stato indica il risultato di una richiesta:

- 200 OK : la richiesta ha avuto successo;
- 301 Moved Permenently: l'oggetto è stato trasferito permanentemente, il nuovo URL è specificato in Location;
- 400 Bad Request: la richiesta non è stata compresa dal server;
- 404 Not Found: il documento richiesto non è presente nel server;
- 505 HTTP Version Not Supported: il server non dispone della versione HTTP richiesta.

I cookie

I cookie sono file di testo che memorizzano le informazioni degli utenti, consentendo ai server di tenere traccia di questi ultimi, essi presentano 4 componenti: una riga di intestazione nella risposta HTTP, una riga di intestazione nella richiesta HTTP un file mantenuto dall'utente e gestito dal browser e un database del sito. Quando l'utente visita un sito per la prima volta, il cookie crea un nuovo record nel database indicizzato da un id, il server risponde includendo l'id nel campo get-cookie. Il browser, ricevuta la risposta, aggiunge una riga al file dei cookie che gestisce, includendo nome del server e id. Ogni volta che viene richiesta una pagina web, il browser consulta il file ed estrae l'id (se presente), ponendo nella richiesta HTTP la riga di intestazione del cookie. In questo modo è possibile monitorare l'utente sapendo esattamente a quali siti ha avuto accesso e in quali orari. Infatti se si torna in un sito già visitato, il browser continuerà a inserire la riga d'intestazione corrispondente a esso nella richiesta HTTP e, nel caso ci dovessimo registrare, il sito può memorizzare tutte le informazioni del caso. Nonostante semplificano molte azioni via Internet, i cookie sono spesso fonte di controversie in quanto violano la privacy dell'utente.

Web caching

Una web cache è un'entità di rete che soddisfa le richieste HTTP al posto del web server effettivo, essa ha una memoria su disco in cui conserva copie di oggetti richiesti di recente. Ogni volta che viene inviata una richiesta HTTP, si controlla la web cache per vedere se è presente, in tal caso viene presa da lì altrimenti si chiede direttamente al server e si salva una copia nella web cache. Una web cache può fare sia da client sia da server e solitamente è installata da un ISP, essa si è sviluppata per due ragioni:

- riduce sostanzialmente i tempi di risposta alle richieste dei client;
- possono ridurre il traffico sul collegamento di accesso a Internet, col vantaggio di non dover aumentare la banda e quindi riducendo i costi.

Con l'aumento della distribuzione di contenuti, i proxy server stanno assumendo un ruolo sempre più importante in Internet.

GET Condizionale

L'unico grosso svantaggio della web cache è il fatto che le pagine web in memoria possono scadere, per risolvere questo problema si utilizza il GET condizionale, una versione alternativa del GET che permette di capire se la pagina è scaduta o meno e quindi richiederla alla giusta destinazione. Una richiesta HTTP riferita al GET condizionale è simile a GET, la differenza è che la prima presenta un'ulteriore riga, ovvero If-modified-since, che indica la data di ultima modifica. Per prima cosa il proxy invia una richiesta al web server per conto del browser, ricevuta la risposta, inoltra l'oggetto a quest'ultimo il quale effettuerà i controlli. Se il valore di If-modified-since è diverso da quello di Last-Modified, vuol dire che l'oggetto in cache è scaduto e quindi il browser lo richiede al web server.

Posta Elettronica

La posta elettronica via Internet è formata da tre componenti:

- gli user agent consentono agli utenti di leggere, rispondere, salvare e comporre messaggi;

- I server di posta sono la parte centrale dell'infrastruttura , esso contiene le caselle di posta relative a ogni utente che a loro volta contengono i messaggi stessi. Un tipico messaggio parte dallo user agent, viene inviato alla casella del mittente e da lì a quella del destinatario. Nel caso in cui non sia possibile consegnare la posta, essa verrà trattenuta in una coda e trasferita in un secondo momento.
- Il protocollo SMTP permette il trasferimento affidabile della posta tra il server mittente e quello destinatario, esso presenta un lato client e un lato server, entrambi possono essere eseguiti su tutti i server di posta.

SMTP

Questo protocollo costituisce il cuore della posta elettronica via Internet, trasferendo i messaggi dal server di posta mittente a quello destinatario. Il protocollo tratta il corpo di questi messaggi, allegati inclusi, come ASCII 7 bit. Quando si vuole inviare un messaggio, si invoca lo user agent per fornire l'indirizzo del destinatario, comporre il messaggio e inviarlo al proprio mail server, Il lato client SMTP apre una connessione TCP tra il server mittente e quello destinatario e in essa viene inviato il messaggio, alla ricezione il messaggio viene messo nella casella. SMTP non fa mai passare la posta da server intermedi, anche se si trovano agli angoli opposti del pianeta. Il trasferimento SMTP è simile a quello umano: il client e il server creare una connessione, si presentano e vengono inviati i messaggi, chiudendo infine la connessione. I comandi di SMTP sono HELO, MAIL FROM, RECEIVE TO, DATA e QUIT, per ogni messaggio il client inizia il processo con MAIl FROM e indica la fine con un punto isolato, QUIT viene inviato solo dopo avere spedito tutti i messaggi.

Confronto con HTTP

Le principali differenze tra HTTP e SMTP sono:

- HTTP trasferisce file tra web server e web client, SMTP trasferisce messaggi da un mail server a un altro;
- HTTP è un protocollo pull, se qualcuno carica un oggetto su un web server, gli altri se lo prendono utilizzando HTTP, SMTP è invece un protocollo push, il mail server spinge i messaggi in ricezione;
- SMTP compone il messaggio con ASCII a 7 bit mentre HTTP non impone tale vincolo;
- HTTP incapsula ogni oggetto in una risposta mentre SMTP incorpora tutti gli oggetti in un unico messaggio.

Formato dei messaggi

I messaggi di posta elettronica sono formati da un'intestazione, comprendente l'indirizzo del mittente, del destinatario, la data e il soggetto del messaggio, e il corpo vero e proprio. Ogni intestazione deve contenere una riga From e una To, esse sono differenti dai comandi SMTP in quanto, se le prime fanno parte del messaggio, le seconde fanno parte del protocollo stesso. Alla fine dell'intestazione vi è una riga vuota e subito dopo il corpo del messaggio.

Protocolli di accesso alla posta

L'accesso alla posta elettronica utilizza un'architettura client-server, tuttavia lo user agent non dialoga direttamente col serve destinatario, esso infatti spinge la posta utilizzando SMTP perchè, senza il proprio mail server come punto intermedio, non saprebbe come gestire un destinatario non raggiungibile. Per fare ciò si fa utilizzo di appositi protocolli: IMAP, POP3 e HTTP (quest'ultimo si utilizza solamente quanto si controlla la posta dal browser), essi entrano in gioco quando si deve trasferire la posta dal mail server allo user agent.

POP3 (Post Office Protocol 3)

POP3 è un protocollo di accesso alla post semplice ma con funzionalità limitate, esso scarica la posta dal mail server utilizzando una connessione TCP tra quest'ultimo e lo user agent, procedendo in tre fasi:

- Autorizzazione: invia user e password per l'autenticazione dell'utente utilizzando i comandi user e pass;
- Transazione: vengono recuperati i messaggi, è anche possibile marcarli per la cancellazione e ottenere statistiche sulla posta. In questa fase POP3 può rispondere ai comandi in due modi: +OK se il comando scelto va bene e -ERR indica che qualcosa non ha funzionato;
- Aggiornamento: quando il client invia il comando quit, il server di posta rimuove tutti i messaggi segnati come da cancellare.

La sequenza di comandi inviati da uno user agent dipende dalla modalità, in "scarica e cancella" viene chiesto al server di elencare la dimensione dei messaggi memorizzati, recuperarli e cancellarli dopo l'invio, un problema di questa modalità è che il destinatario potrebbe voler accedere ai propri messaggi da più dispositivi, quindi si è passati alla modalità "scarica e mantieni" la quale non cancella i messaggi dopo l'invio allo user. Durante la sessione, il server mantien alcune informazioni di stato, in particolare tiene traccia dei messaggi marcati come cancellati, tuttavia non è in grado di trasportarle tra le differenti sessioni.

IMAP (Internet Message Access Protocol)

Dato che POP3 permette la creazione di cartelle dopo avere scaricato i messaggi, è possibile avere una cosa simile ma nel server? Sì, si utilizza IMAP, un protocollo la cui funzione è simile a quella di POP3 con la differenza che esso gestisce la posta direttamente nel server e non scaricandola, rendendolo in questo modo più complesso. Un server IMAP associa a ogni messaggio una cartella, quelli in arrivo sono associati a INBOX, è comunque possibile creare ulteriori cartelle in cui si possono inserire messaggi.IMAP a differenza di POP3 permette la conservazione delle informazioni di stato tra una sessione e un'altra, una caratteristica principale è la presenza di operazioni che permette di ottenere le singole parti di un messaggio, utile quando la connessione è limitata.

Posta basata sul Web

La posta sul browser è tutt'oggi in costante crescita, grazio a questo servizio, lo user agent è il browser e l'utente comunica con la propria casella attraverso richieste HTTP al posto di

IMAP/POP3, la comunicazione tra il server mittente e quello destinatario rimane quella di sempre.

DNS (Domain Name System)

Per riconoscere gli host in Internet vengono utilizzati i cosiddetti hostname, essi sono comprensibili dagli utenti ma forniscono poche informazioni nella rete, per questo motivo vengono utilizzati anche gli indirizzi IP, formato da 4 byte in una struttura gerarchica che ne definisce la collocazione. Per capire qual è l'indirizzo di un hostname e viceversa si utilizza il DNS, un database distribuito implementato in una gerarchia di DNS server e, allo stesso tempo, un protocollo applicativo che permette agli host di interrogare il database, esso utilizza UDP e la porta 53 e viene comunemente utilizzato da altri protocolli per tradurre gli hostname in indirizzi IP. Quando viene inviata una richiesta HTTP su un determinato hostname, per ottenere il suo indirizzo IP:

- il browser estrae l'hostname e passarlo al client DNS (la macchina utente);
- il client DNS interroga il DNS server utilizzando l'hostname;
- ricevuto l'indirizzo IP, esso viene inoltrato al browser il quale potrà iniziare a mandare richieste.

Il DNS introduce un ulteriore ritardo per le applicazioni che lo utilizzano, tuttavia l'indirizzo IP richiesto è quasi sempre nella web cache e quindi il problema si può dire risolto. Il DNS offre anche i seguenti servizi:

- Host aliasing: permette di dare a un host più sinonimi, quello originale verrà detto hostname canonico. Quindi il DNS può anche essere chiamato per richiedere un hostname canonico dato un sinonimo;
- Mail server aliasing: permette di fare ottenere a un mail server il nome canonico di un sinonimo fornito attraverso i record MX, essi permettono ai server di posta di una società di avere un hostname identico a quello del web server.
- Load distribution: serve per distribuire il carico tra server replicati, soprattutto per i siti con molto traffico. Ogni sito è su un host diverso con un indirizzo IP differente, essi sono tutti contenuti nella stessa voce del database DNS, infatti quando il client richiede quel sito, il DNS risponde con l'intero insieme, variando l'ordine ogni volta. Lo stesso procedimento viene effettuato anche coi server di posta.

Funzionamento del DNS

Il DNS è paragonabile a una scatola nera che fornisce un servizio di trasporto semplice e diretto, è costituito da un gran numero di server sparsi per il mondo, se così non fosse i client dirigerebbero le richieste su un singolo server, cosa inappropriata per l'Internet odierno perchè:

- In caso di questo dell'unico server, tutto Internet ne soffre;
- Un singolo server dovrebbe gestire tutte le richieste di tutti i client, creando traffico;
- Un singolo database non può essere vicino a tutti i client;
- Il database DNS sarebbe vasto e difficile da aggiornare.

Il DNS è quindi un database gerarchico e distribuito nel mondo su tre tipi di classi:

- Root Server: server DNS in cima alla gerarchia che fornisce gli indirizzi dei server TLD;
- Top Level Domain Server (TLD): server che si occupano dei domini di primo livello;

- server autoritativi: server contenenti parte del database DNS in cui per ogni host in internet vi è un record DNS pubblicamente accessibile, un'organizzazione può scegliere se implementare un proprio server autoritativo oppure se affittarne uno da un fornitore.

Un altro importante tipo di DNS è il DNS server locale, esso non appartiene strettamente alla gerarchia ma è centrale nell'architettura, esso è presente in ogni ISP e, quando un host si connette a esso, quest'ultimo fornisce un indirizzo IP tratto da uno o più server locali.
Quando un client ha bisogno di convertire un hostname, contatta il server locale il quale interroga i root server i quali restituiscono una lista di TLD server che esso andrà a contattare, una volta ottenuto l'indirizzo IP dell'hostname da parte del server autoritativo, esso verrà inoltrato al client. Vi è anche un implementazione ricorsiva in cui il server locale contatta solo il root server, tutti gli altri server della gerarchia verranno contattati "in profondità" fino a quando non si ottiene l'indirizzo IP richiesto.

DNS caching

Il DNS sfrutta una cache in modo estensivo per diminuire i ritardi e il numero di messaggi DNS, i DNS possono mettre le risposte che ricevono in cache in modo da riutilizzarle quando vengono richieste. Il DNS server può quindi fornire un indirizzo IP contenuto in cache anche se non è autoritativo, questo ha permesso di velocizzare le operazione senza ulteriori interrogazioni. I server locali possono memorizzare gli indirizzi dei TLD server, consentendogli di aggirare i root server.

Record e messaggi DNS

I server che implementano il database DNS memorizzano i cosiddetti record di risorsa oltre a quelli di nomi e indirizzi, ogni messaggio DNS trasporta uno o più di essi, essi vengono identificati da 4 valori:

- TTL: Il time to live è il tempo di vita di un record e determina quando deve essere rimossa dalla cache;
- Type: Indica il tipo di record, se è A allora Name è l'hostname e Value l'indirizzo IP, se è NS allora sono rispettivamente il dominio e l'hostname del server autoritativo, se è CNAME, allora Value e il nome canonico dell'host per il sinonimo Name e se è MX allora Value è il nome canonico di un mail server che ha il sinonimo Name.

Un DNS server autoritativo contiene un record A per l'hostname, se non lo è allora contiene un record NS per il dominio e un record A che fornisce l'indirizzo IP del DNS server.
Un messaggio DNS ha la seguente struttura:

- I primi 12 byte sono la sezione di intestazione, formata da un numero di 16 bit che lo identifica, esso consente al client di far corrispondere le risposte ricevute con le query inviate, un campo flag e i bit di richiesta/risposta (rispettivamente 0 e 1). Un'ulteriore bit è la richiesta di ricorsione e viene impostato quando il client desidera che effettui una ricerca in ricorsione quando non dispone del record. Vi sono poi altri 4 campi che indicano il numero di occorrenze di ogni sezione;
- La sezione delle domande contiene informazioni riguardanti le richieste che stanno per essere effettuate, include un nome e un tipo;
- La sezione delle risposte contiene i recordi di risorse relativo al nome richiesto in origine, contiene il campo Type, Value e TTL;

- La sezione autoritativa contiene i record di altri server autoritativi;
- La sezione aggiuntiva contiene ulteriori record utili;

Quando si deve inserire un record nel database DNS, per prima cosa si deve registrare il nome di dominio presso un ente di registrazione, un'azienda che erifica l'unicità del nome e lo inserisce nel database DNS. Alla registrazione di un nome di dominio, bisogna fornire nomi e indirizzi IP dei server autoritativi primario e secondario, per ognuno di essi si accerterà dell'inserimento di un record NS e di un record A nei TLD server relativi al suffisso. In più, bisogna accertarsi che il record A e il record MX vengano immessi nei server autoritativi,

Distribuzione di file P2P

Se in una rete client-server il server deve inviare una copia del file a ciascun peer, consumando però un'enorme quantità di banda, una distribuzione P2P ogni peer può ridistribuire tutti i pezzi che ha a disposizione, aiutando in questo modo il server a condividere il file. Server e peer sono connessi a Internet attraverso collegamenti di accesso il cui tempo di distribuzione equivale al tempo richiesto affinchè tutti i peer ricevano il file.
Il calcolo del tempo di distribuzione per una rete client-server deve tenere conto del fatto che tutto il carico è sul server ed esso deve trasmettere il file a ogni peer. Quindi se vi sono N peer, un file da distribuire grande F e la banda del server s, si ha che il server ci mette un tempo NF/s per distribuire il file a tutti i peer. Bisogna però tenere conto anche della banda dei singoli peer (prendiamo una banda p come la banda minima tra tutti i peer), un peer infatti ci mette infatti un tempo F/p per scaricare il file, possiamo quindi dire in conclusione che il tempo di distribuzione è il massimo tra quello impiegato dal server e quello dei peer: Tcs >= max{ NF/s , F/p }.
Il tempo di distribuzione di una rete client-server aumenta linearmente in base al numero di peer connessi. Per quanto riguarda le reti P2P, quando un peer riceve una parte di un file, esso può ricondividerlo con altri peer che non ce l'hanno. Inizialmente il file è presente solo nel server, per trasmetterlo deve inviare ogni bit in ogni collegamento, quindi il tempo di distribuzione, dato un file grande F e una banda s, è F/s. Come nelle reti client-server, si tiene conto della banda più bassa tra i peer p, quindi i per ci mettono un tempo F/p per ricevere il file. Complessivamente la capacità di upload equivale alla velocità di upload del server più quella di ogni peer, quindi il sistema de consegnare NF bit nella banda totale (NF/(s + somma(p)). Possiamo quindi concludere che il tempo di distribuzione è il valore massimo calcolato dalle tre formule viste in precedenza:
Tp2p >= max{ F/s , F/p , NF/(s + somma(p) }
Il tempo di distribuzione minimo in una rete P2P non è solo più piccolo rispetto a un'architettura client-server ma è anche minore per ogni numero di peer, questo conferma la scalabilità dell'architettura e una conseguenza del fatto che i peer possono distribuire le loro parti di file.

BitTorrent

BitTorrent è uno dei più diffusi protocolli P2P in cui tutti i peer permettono la distribuzione di particolari file detti torrent. I peer scaricano parti di file di uguale dimensione, quando entra un nuovo peer nella condivisione, esso scarica alcune parti di file il quale metterà a disposizione quando entreranno nuovi peer, scaricando nel frattempo altre parti. Una volta

acquisito l'intero file, il peer può lasciare la rete o continuare a distribuire. Ogni volta che un peer si connette a una rete P2P, si registra a un nodo detto tracker, informandolo periodicamente se si trova ancora nel torrent o meno, e seleziona un sottoinsieme di peer, con essi stabilirà delle connessioni TCP gestite in contemporanea. Ogni peer ha una lista delle parti a sua disposizione, esse verranno messe a disposizione di tutti i vicini per verificare le parti mancanti. Con queste informazioni, si possono prendere due decisioni: quali parti prendere per prime e quali inviare ai vicini, nel frattenpo si applica la tecnica del rarely first, ovvero si determinano quali sono le parti più rare e si distribuiscono quando richiesti. Per determinare a quali richieste bisogna rispondere, si utilizza un algoritmo di trading in cui si danno priorità ai vicini che inviano dati alla velocità più alta. Ogni 0 secondi vengon determinati 4 peer aventi la velocità più elevata, essi vengono detti unchoked e ogni 30 secondi ne viene scelto uno casualmente tra tutti per garantire la distribuzione totale del file, esso potrebbe diventare un unchoked se invia i dati con una velocità abbastanza elevata. Tutti gli altri peer sono choked, ovvero che non ricevono parti di file, questo meccanismo è detto tit-for-tat. Nelle applicazioni P2P sono anche prensenti delle tabelle hash di distribuzione, dei database semplici in cui i record sono distribuiti tra i peer.

Streaming video

Lo streaming video rappresenta l'attuale maggior parte del traffico nelle reti con ISP residenziali, i contenuti vengono memorizzati su server a disposizione degli utenti e utilizzabili su richiesta. I video sono sequenze di immagini visualizzati con un rate che varia dai 24 alle 30 immagini al secondo, esso consiste in un array di pixel codificato con un numero di bit che rappresenta la luminanza e la crominanza, esso possono essere compressi per raggiungere un compromesso tra qualità e bit rate, più è alto, più la qualità è alta. Una misura importante per lo streaming è il throughput medio, esso deve essere almeno pari al bit rate per garantire una riproduzione continua.

Streaming HTTP e DASH

Nello streaming HTTP i video sono memorizzati in un server HTTP come un file ordinario con rispettivo URL, i client per visualizzarli effettuano connessioni TCP con il server e inviano dei GET, questi ultimi inviano il video il più velocemente possibile e vengono messi in un buffer contenuto nell'applicazione client, quando il buffer supera una certa soglia, può iniziare la riproduzione. Vi è però uno svantaggio: i client ricevono una sola versione del file indipendentemente dalla banda di ognuno, questo problema viene risolto con lo streaming adattativo, il DASH (Dynamic Adaptive Streaming over HTTP). Nel DASH vi sono più versioni di un video codificate con un bit rate differente e con un URL diverso in modo da richiedere il file giusto in base alla banda disponibile in un certo intervallo di tempo. Il server HTTp che contiene i video ha anche un file, detto manifesto, in cui vi è l'URL di ogni versione del video e il relativo bit rate, i client infatti prima di scaricare il video richiedono questo file per scegliere la versione più adatta al caso: a ogni istante sceglie un blocco e, mentre lo scarica, misura la banda, essa servirà a un algoritmo per scegliere il blocco successivo.

Reti di distribuzione

L'approccio più diretto per la distribuzione di contenuti in streaming è il datacenter unico, esso però oltre a essere un possible Single Point Of Failure, non è vicino a tutti ISP del mondo, in più il throughput di uno dei collegamenti è minore, lo sarà anche quello totale, oltre al fatto che l'azienda è costretta a pagare più volte l'ISP per inviare gli stessi dati. Per risolvolre questo problema si utilizzano le CDN (Content Distribution Network), esse gestiscono server sparsi per il mondo memorizzando copie di contenuti di altri server e cercando di dirigere le richieste degli utenti nella CDN col servizio migliore. Una CDN può essere privata, ovvero quando proprietaria del fornitore, oppure di terze parti, quando vengono affittate dalle aziende per la distribuzione, ognuna adotta una delle due sequenti politiche di dislocazione:

- Enter deep: si entra nel profondo delle reti di accesso degli ISP installando dei gruppi di server detti cluster all'interno, l'obiettivo è quello di essere vicini agli utenti finali in modo da diminuire il ritardo e migliorare il throughput, diminuendo il numero di collegamenti e router tra utenti e CDN. Questo approccio però è molto dificile da gestire e mantenere;
- Bring home: si porta a casa l'ISP costruendo grandi cluster in pochi punti chiave e interconnetterli con una rete privata ad alta velocità. Queste CDN pongon ogni cluster vicino ai point of presence e presentano una manuzione e una gestione più facile, tuttavia potrebbe esserci una minore qualità del servizio.

Molte CDN utilizzano una strategia in cui, se un client chiede a un cluster un contenuto che quest'ultimo non possiede, esso ne prende una copia dall'archivio centrale o da un altro cluster e lo memorizza una copia, i video meno richiesti vengono poi rimossi quando c'è carenza di spazio.

Funzionamento di una CDN

Quando un browser richiede un video, la CDN deve poter intercettare la richiesta per:

- determinare il cluster più approppriato;
- dirigere la richiesta a uno dei server;

Molte CDN sfruttano il DNS per intercettare e ridirigere le richeste, esso funziona nel seguente modo:

- Quando si seleziona un link di una pagina web, viene inviata un'interrogazione al DNS;
- Il server locale invia l'interrogazione a un server autoritativo che, osservando la stringa del nome, restituisce il nome dell'host del dominio;
- il sever locale effettua quindi una seconda interrogazione per prendere l'indirizzo IP del server e lo inoltra al browser;
- Il browser stabilisce una connessione TCP col server e, nel caso sia DASH, riceverà il manifesto contenente gli URL;

Strategie di selezione dei cluster

La selezione dei cluster è un meccanismo che consente la direzione dinamica dei client verso cluster o data center della CDN, infatti la CDN deve selezionare un cluster appropriato dopo aver appreso l'indirizzo IP. La strategia più semplice è scegliere il cluster

geograficamente più vicino utilizzando la geolocalizzazione, essa funziona abbastanza bene nella maggior parte dei casi ma, anche se in linea d'aria potrebbe essere più vicino, potrebbe non esserlo dal punto di vista della rete, in più alcuni utenti potrebbero essere configurati per l'utilizzo di server locali remoti, oltre a non tenere conto dei ritardi e della banda della rete. Si utilizza quindi un approccio in cui vengono effettuate misure in tempo reale di ritardo e perdita basandosi sul traffico corrente, un problema però è che molti server locali sono configurati in modo da non rispondere a tali richieste.

Livello trasporto

Il livello trasporto permette il trasferimento dei messaggio del livello applicazione tra due punti periferici, la cosiddetta comunicazione logica. In questo livello vi sono due protocolli:

- TCP: fornisce un servizio orientato alla connessione con consegna garantita dei messaggi e controllo di flusso. In più vi è il frazionamento di messaggi troppo lunghi e il controllo della congestione in modo da regolare la velocità di trasmissione;
- UDP: fornisce un servizio senza connessione, non è affidabile e non presenta alcun controllo.

Il livello trasporto divide i messaggi applicativo in segmenti e ci aggiunge un'intestazione, in ricezione estrae invece i segmenti dai pacchetti a livello di rete. I servizi di livello trasporto sono vincolati da quelli forniti a livello di rete, infatti se quest'ultimo non può fornire garanzie riguardanti di ritardo o banda per i segmenti, allora valore lo stesso anche per il primo, questo ragionamento però non vale per tutti i protocolli dato che ne esistono alcuni che danno garanzie anche quando non sono presenti nel livello sottostante.

Livello trasporto di Internet

Internet, come tutte le reti TCP/IP, mette a disposizione TCP e UDP al livello applicazione, esso utilizzano i servizi offerti da IP, un protocollo di rete che fornisce la comunicazione logica tra host, il suo modello di servizio è il cosiddetto best effort, in cui il protocollo cerca di consegnare i pacchetti non offrendo però garanzie riguardo l'ordine originario e l'integrità dei dati. I protocolli a livello trasporto estendono IP fornendo la comunicazione logica tra processi attraverso il multiplexing/demultiplexing a livello trasporto, implementando anche un controllo di integrità e il riconoscimento degli errori. TCP inoltre permette il trasferimento dati affidabile, assicurandosi che i messaggi arrivino a destinazione correttamente e in ordine, e il controllo della congestione per evitare che l'elevato traffico intasi router e collegamenti regolando la velocità di immissione dei pacchetti.

Multiplexing e demultiplexing

L'invio/ricezione dei segmenti avviene da e verso la porta di destinazione attraverso il multiplexing e il demultiplexing. Quando più processo devono inviare dei segmenti alla stessa socket, il livello trasporto li raccoglie e li indirizzi verso essa mentre, al contrario, se dei processi attendono tutti sulla stessa socket, il livello trasporto li preleva e li consegna. Il multiplexing richiede che le socket vengano identificate in modo univoco, per farlo si utilizzano i numeri interi di 16 bit, le cosiddette porte, alcune di esse sono riservate a dei protocolli applicativi e hanno un numero specifico. In questo modo quando arriva un

segmento il livello trasporto esamina il numero di porta e lo dirige verso il socket corrispondente.

Multiplexing/Demultiplexing non orientato alla connessione

Il livello trasporto crea i segmenti che includono i dati i numeri di porta di origine e due valori non importanti, passa il segmento al livello di rete il quale lo incapsula in un pacchetto e lo invia al destinatario. Arrivato a destinazione, il livello trasporto estrae il segmento dal pacchetto e lo consegna alla porta di destinazione appropriata.

Multiplexing/Demultiplexing orientato alla connessione

I socket orientati alla connessione, a differenza dei precedenti, sono identificati da quattro parametri: indirizzo IP e porta sia del mittente sia del destinatario. Quindi quando un segmento giunge a destinazione, vengono utilizzati questi quattro parametri per dirigerlo verso la socket appropriata. Quando bisogna inviare dei dati, il server TCP presenta un socket di benvenuto in attesa di richieste di connessione, il client TCP crea un socket e genera un segmento per stabilire una connessione col server. Una richiesta di connessione è un segmento TCP con porta 12000 e un bit speciale posto a 1 nell'intestazione, vi è anche la porta di origine scelta dal client. Il server, alla ricezione del segmento, localizza il processo e crea una nuova connessione. Il livello trasporto server prende nota della porta di origine, destinazione e i relativi indirizzi IP, tutti i segmenti successivi coincidono coi valori citati prima. Un server permette di ospitare più socket TCP in contemporanea connesse a differenti processi.

Web server e TCP

Quando si richiede una pagina web a un web server, la connessione viene eseguita sulla porta 80 dal client, ogni processo viene riconosciuto dal server attraverso l'indirizzo IP e ognuno ha una propria socket su cui vengono inviate le risposte HTTP. Se client e server usano HTTP persistente, i messaggi HTTP vengono scambiati attraverso la stessa socket per tutta la durata della connessione. In caso di HTTP non persistente, invece, per ogni coppia richiesta-risposta viene creata una nuova connessione, l'apertura e la chiusura frequente può avere un impatto sulle prestazioni.

Trasporto UDP

UDP è un protocollo senza connessione che prende i messaggi dai processi, aggiunge porte di origine e destinazione e lo manda al livello di rete che lo incapsula e lo invia. DNS, come molte altre applicazioni, utilizzano UDP perchè:

- vi è un controllo più fine su quali dati sono inviati e quanto a livello di applicazione, ovvero che non appena i messaggi giungono a UDP, essi vengono spediti immediatamente al contrario di TCP che esegue ulteriori controlli. Per questo motivo TCP non è adatto a molte applicazioni in quanto quelle in tempo reale richiedono una velocità minima.
- Non vi è stato di connessione, quindi non vengono allocati buffer e variabili, cosa che TCP fa;
- L'intestazione è meno spaziosa, è di 8 byte contro i 20 di TCP.

UDP non implementa il controllo della congestione, necessario al fine di evitare un'enorme traffico nella rete che non permetterebbe di raggiungere la destinazione. In più, l'alta frequenza di perdite indotte da UDP provoca una diminuzione del tasso di invio di TCP, quindi questa mancanza può avere come conseguenza un'alta percentuale di perdite tra mittente e destinatario. Un trasferimento dati affidabile è comunque possibile attraverso UDP, il protocollo QUIC, infatti utilizza UDP e implementa l'affidabilità a livello applicazione.

Struttura di un segmento UDP

Un segmento UDP è formato da un'intestazione e da un corpo, nella prima vi sono quattro capi di 2 byte ciascuno, essi sono la porta di origine e destinazione, la lunghezza del segmento, necessaria perché la lunghezza di un segmento è variabile, e il checksum, nella seconda parte vi sono i dati veri e propri.

Checksum UDP

Il checksum serve per il rilevamento degli errori, nel lato mittente viene effettuato il complemento a 1 della somma di tutti i 16 bit nel segmento e l'eventuale riporto viene sommato al primo bit, il risultato viene poi inserito nell'apposito campo. La presenza del checksum è data dal fatto che non vi è alcuna garanzia che tutti i collegamenti tra sorgente e destinazione controllino gli errori. In più potrebbero verificarsi errori anche quando il segmento si trova nella memoria del router e, dal momento che non sono garantite nè affidabilità dei collegamenti nè rilevamento degli errori in memoria, UDP deve metterlo a disposizione. Il livello trasporto fornisce il controllo degli errori come misura di sicurezza anche se si suppone che IP funzioni correttamente coi protocolli di secondo livello, nonostante questo UDP non permette di risolverli, alcune implementazioni scartano il segmento mentre altre lo inviano con una avvertimento.

Principii del trasferimento dati affidabile

Il trasferimento dati affidabile è un problema che riguarda il livello trasporto, applicazione e collegamento. L'astrazione offerta dai livelli superiori è un canale affidabile in cui si possono trasferire dati, con esso nessun bit è corrotto e l'ordine viene rispettato. Il compito di un protocollo che utilizza questo servizio è quello di implementarlo, questo diventa difficile quando i livelli inferiori non presentano utilizzano servizi di trasferimento inaffidabili. A livello di rete, il "canale di trasferimento" può consistere di un collegamento o di una rete, entrambi si possono considerare canali inaffidabili.

Costruzione di un protocollo di trasferimento affidabile

Utilizzando la macchina a stati finiti e considerando solamente il trasferimento half-duplex, il mittente verrà invocato per il trasferimento dati con send() mentre il destinatario sarà chiamato per la ricezione con receive(), infine i pacchetti verranno consegnati al livello superiore con deliver().

Modello 1

Il caso più semplice è dato da mittente e destinatario che attendono una chiamata da parte dell'utente. Quando ciò avviene, il mittente crea un pacchetto e lo invia mentre il

destinatario, alla ricezione del pacchetto, estrae i dati e li dirige al livello superiore, infine entrambi ritornano allo stato iniziale. Tutti i pacchetto fluiscono da mittente a destinatario attraverso un canale affidabile senza che il destinatario avvisi il mittente di eventuali errori.

Modello 2

Un modello più realistico è quello in cui i bit possono essere corrotti, di conseguenza bisogna gestire tali situazioni, per farlo si notifica il mittente utilizzando dei pacchetti appositi, rispettivamente un ACK quando tutto va bene e un NAK quando non è così, essi permettono al mittente di sapere se un pacchetto è stato ricevuto correttamente o meno e, nel caso, ritrasmetterlo. Il mittente, dopo aver inviato il pacchetto, rimane in attesa del feedback del destinatario: se riceve un ACK, ritorna allo stato iniziale, altrimenti se riceve un NAK, ritrasmette il pacchetto. Il destinatario, appena riceve il pacchetto, controlla se è corrotto o meno e, a seconda della situazione, invia un ACK o un NAK e ritorna allo stato iniziale. Questo modello presenta però un difetto: non può gestire le possibilità in cui ACK o NAK vengano persi durante il trasferimento, infatti se un ACK/NAK è corrotto o perso, il mittente non ha modo di sapere se il pacchetto è giunto correttamente a destinazione, vi possono essere quindi tre possibilità:

- nessuno capirebbe il pacchetto dell'altro e viceversa;
- si potrebbe aggiungere un checksum per trovare e correggere gli errori, tuttavia non risolve il problema delle perdite;
- il mittente invia nuovamente il pacchetto, questo però introduce dei duplicati nel canale e soprattutto il destinatario non sa se l'ultimo ACK/NAK inviato sia stato ricevuto correttamente dall'utente.

Modello 3

Un soluzione al problema precedente è raddoppiare il numero di stati nel mittente/destinatario e numerare i pacchetti, il destinatario dovrà quindi controllare se il pacchetto deve essere ritrasmesso o meno. Nel primo caso, il numero del pacchetto è lo stesso di quello appena ricevuto mentre nel secondo sarà diverso. Il mittente sa che l'ACK/NAK ricevuto si riferisce al pacchetto trasmesso più recentemente. Il destinatario invia un ACK quando il pacchetto ricevuto non è corrotto però manda i dati all'applicazione solamente quando è in sequenza, altrimenti manda un NAK.

Modello 4

Il modello 4 è molto simile al precedente con la differenza che non vengono più utilizzati i NAK, al loro posto vengono utilizzati degli ACK numerati: infatti se il destinatario riceve il pacchetto 0 anzichè 1 o viceversa, chiede al mittente la ritrasmissione mandando un ACK il cui numero è diverso da quello che attende. Lo stesso discorso vale anche quando il pacchetto è corrotto.

Modello 5

Assumendo ora la possibile perdita dei pacchetti, bisogna ora gestire le situazioni in cui ciò accade. Il mittente si preoccuperà di rilevare e risolvere eventuali perdite, per farlo si può utilizzare un timer il cui tempo di durata è dato dal ritardo minimo tra mittente e destinatario più il tempo di elaborazione del pacchetto, se non si riceve un ACK entro questo periodo, viene effettuata una ritrasmissione. Tuttavia vi possono essere trasmissione anche quando

vi è un ritardo troppo lungo, questo può causare pacchetti duplicati all'interno della rete. Implementare un meccanismo basato sul tempo richiede un contatore in grado di segnalare le scadenze. Il mittente dovrà quindi inizializzare il contatore a ogni invio del pacchetto, rispondere agli interrupt del timer con l'azione appropriata e fermare il contatore.

Trasferimento affidabile con pipeline

Nonostante il modello 5 sia corretto, le prestazioni però non lo sono: considerando due host il cui RTT è 30 ms in un canale R da 1 Gbps e pacchetti L da 1 Mb, il tempo richiesto è 8L/R=8 us. L'ultimo bit entra nel canale dopo 8 us, quindi giunge a destinazione all'istante RTT/2 + 8u= 15 ms. L'ACK arriverà quindi all'istante RTT=30ms, possiamo quindi definire l'utilizzo del mittente come il tempo in cui è effettivamente occupato, quindi: Um=(L/R)/(RTT+L/R)=0,00027=270 us
Il mittente è quindi occupato solamente per l'1% del tempo, per risolvere ciò si utilizza un'altra modalità al posto dello stop-and-wait che consente di inviare più pacchetti senza attendere gli ACK. Questa tecnica è detta pipelining e permette di aumentare l'utilizzo del mittente, tuttavia:

- l'intervallo dei sequence number deve essere incrementato dato che ogni pacchetto deve averne uno univoco e ci possono essere ancora ACK in transito;
- I lati di invio e ricezione dei protocolli devono poter memorizzare in un buffer i pacchetti, soprattutto quelli che non hanno ricevuto un ACK;
- La quantità di sequence numero e i requisiti del buffer dipendono dalla reazione del protocollo agli smarrimenti.

Go-Back-N (GBN)

Con la tecnica GBN, il mittente può trasmettere fino a N pacchetti senza attendere alcun ACK, esso utilizza la cosiddetta finestra: la variabile base indica il numero di sequenza del pacchetto più vecchio non confermato da un ACK e indica l'inizio della finestra, tutti quelli precedenti sono già stati confermati. La variabile nextseqnum è il più piccolo sequence number inutilizzato, ovvero il numero del prossimo pacchetto da inviare, i pacchetti tra base e nextsegnum-1 sono quindi quelli inviati ma non ancora confermati con un ACK mentre quelli superiori a quest'ultima variabile non sono stati ancora considerati. Come risultato il destinatario invia il cosiddetto ACK cumulativo il cui numero all'interno indica che tutti i pacchetti fino a esso sono stati ricevuti correttamente. Il GBN risponde a tre tipi di evento:

- Invocazione dall'alto: il mittente controlla se vi sono N pacchetti in sospeso senza ACK, se è così, invia un pacchetto e aggiorna le variabili, altrimenti restituisce i dati all'applicazione e controlerà più tardi;
- Ricezione di un ACK: il destinatario restituisce l'ACK cumulativo citato prima;
- Evento di timeout: Al verificarsi del timeout, il mittente invia tutti i pacchetti spediti e in attesa di un ACK. Se il mittente riceve un ACK ma ci sono ancora pacchetti in attesa, il timer viene fatto ripartire, verrà azzerato quando nessuno è in attesa.

Quando un pacchetto viene ricevuto correttamente e in ordine, il destinatario manda un ACK per quel pacchetto e consegna i dati all'applicazione, altrimenti lo scarta e manda l'ACK più recente. Il vantaggio di GBN è la semplicità, esso infatti non deve memorizzare i pacchetti che giungono fuori sequenza, deve solo memorizzare il sequence number del pacchetto successivo nell'ordine. Lo svantaggio è che, anche se ha ricevuto correttamente un

pacchetto fuori sequenza, la trasmissione può andare persa o alterata, richiedendo così ulteriori ritrasmissioni.

Ripetizione selettiva

La ripetizione selettiva è una tecnica simile a GBN con la differenza che essa è capace di mantenere i pacchetti fuori ordine. Essa utilizza una finestra grande N come GBN, spostandosi al prossimo sequence number non validato quando arriva l'ACK riferito al pacchetto base, infatti la finestra non può avanzare senza di esso. Tuttavia le finestre del mittente e del destinatario non sempre coincidono, la mancanza di sincronizzazione infatti ha conseguenze importanti quando vi è un intervallo finito di sequence number, dato che alcuni pacchetti confermati da un parte potrebbero non esserlo dall'altra a causa delle perdite. I pacchetti non possono essere riordinati in un canale tra mittente e destinatario, tuttavia se il canale è una rete, c'è il rischio che essi vengano mischiati. Una manifestazione di questo fenomeno è la possibile comparsa di vecchie copie anche non contenute nelle finestre del mittente e del destinatario. In pratica ci si assicura che il numero di sequenza non venga utilizzato di nuovo finché il mittente non è sicuro che ogni pacchetto inviato in precedenza sia scomparso dalla rete, assumendo che essi non possano sopravvivere all'interno della rete per un certo periodo di tempo.

Trasporto TCP

Il trasporto TCP è detto con connessione in quanto i processi devo fare handshake prima di inviare i dati, sia mittente che destinatario devono inizializzare delle variabili di stato associate alla connessione stessa, essa non è un circuito end-to-end in quanto lo stato della connessione si trova nei sistemi periferici. TCP va in esecuzione solo sugli host e non negli elementi intermedi e offre un servizio full-duplex: mittente e destinatario possono sia inviare che ricevere dati. TCP è anche punto-punto, ovvero che vi sono solo un mittente e un destinatario, ciò non lo rende utilizzabile per servizi come il multicast. Per stabilire una connessione, il processo client informa il livello trasporto di voler fare verso un processo nel server, specificando il nome e la porta di destinazione. Il client invia così al server un segmento TCP speciale e quest'ultimo risponde con un ulteriore segmento TCP speciale e Infine il client invia un'altro segmento TCP, anch'esso speciale, in cui vi sono anche dati utili, questo approccio è detto three-way handshake. Una volta instaurata la connessione i processi possono iniziare a scambiarsi i dati, il mittente manda un flusso di dati attraverso la socket, essi verranno raccolti da TCP in un buffer di invio da cui verranno raccolti e diretti verso il livello di rete. La massima dimensione del segmento (MSS) indica la quantità di dati più grande che TCP può prelevare, esso è un valore variabile ma solitamente fissato 1460 byte, dato che i protocolli Ethernet e PPP hanno la massima dimensione del pacchetto (MTU) grande 1500 byte, ovvero il segmento più 40 byte di intestazione. TCP forma i segmenti accoppiando i blocchi di dati a delle intestazioni, essi verranno poi inviati al livello di rete che li impacchetta e li invia. Il destinatario, ricevuto il pacchetto, estrae il segmento e memorizza i dati nel buffer di ricezione della connessione per poi essere letti dall'applicazione.

Struttura dei segmenti TCP

Un segmento TCP è limitato da MSS ed è formato da un'intestazione e dalla parte dati, la prima è grande 20 byte e contiene i numeri di porta di origine e destinazione (utilizzati per il multiplexing/demultiplexing)un campo checksum, il sequence number, il numero di ACK, la finestra di ricezione (utilizzata per il controllo di flusso), la lunghezza dell'intestazione, il campo opzioni (di lunghezza variabile, viene utilizzata per negoziare la MSS) e il campo flag, la seconda invece è lunga di solito un byte. il campo flag è composta da 6 bit ognuno con un ruolo specifico:

- ACK indica che il valore all'interno del campo ACK è valido, il segmento contiene un ACK per un segmento ricevuto con successo;
- RST, SYN e FIN vengono utilizzati per impostare e chiudere la connessione;
- CWR e ECE sono utilizzati per il controllo della congestione;
- PSH quando ha valore 1 indica che i dati devono essere inviati immediatamente all'applicazione;
- URG indica la presenza di dati marcati come urgenti, la posizione dell'ultimo byte dei dati urgenti viene denota da un apposito campo puntatore di 16 bit, TCP inoltre deve informare l'applicazione e passarle il suddetto puntatore.

Sequence number e numeri di ACK

I sequence number e i numeri di ACK rappresentano una parte critica del trasferimento affidabile di TCP, dato che quest'ultimo vede i dati come flussi di byte ordinati e non strutturati. Il sequence number si applica quindi al flusso di byte trasmessi e non alla serie di segmenti trasmessa, quindi lo si può identificare come il numero del primo byte nel flusso. Il numero di ACK scritto da un host A è il sequence number successivo che esso attende da un host B, questo vuol dire che se A ha ricevuto da B i byte numerati fino a 535, esso è in attesa di 536 e superiori. TCP, dato che effettua acknowledgment dei byte nel flusso fino al primo mancante, si dice che esso offra ACK cumulativi. Quando TCP riceve segmenti fuori sequenza, il destinatario può scartarlo oppure mantiene i byte non ordinati e attende quelli mancanti per colmare i vuoti. I partecipanti della connessione all'inizio scelgono un sequence number a caso per minimizzare la possibilità che un segmento si trovi ancora nella rete.

Caso di studio: Telnet

Telnet è un protocollo applicativo utilizzato per il login remoto, esso utilizza TCP ed è progettato per funzionare con ogni coppia di host. Supponiamo che un host A (client) inizi un Telnet con un host B (server), ogni carattere verrà spedito al server e quest'ultimo invierà una copia dello stesso al client per assicurarsi che i caratteri siano ricevuti ed elaborati in remoto. Per esempio, il segmento inviato da A verso B ha il sequence number 42 e numero di ACK 79, B risponderà quindi con un segmento con sequence number 79 e numero di ACK 43 e così via. E' possibile notare che il riscontro dei dati da A a B viene trasportato da un segmento che a sua volta trasporta dati dal segmento che va da B ad A, questo tipo di acknowledgement è detto piggybacked.

Timeout e stima del tempo di andata e ritorno

Per individuare i pacchetti persi, TCP utilizza un meccanismo di timeout, esso dovrebbe essere più grande di un RTT (tempo di un pacchetto per andare verso un host e tornare) per evitare ritrasmissioni inutili. Per misurare il tempo RTT di un segmento, si prendono dei campioni di tempo e si inseriscono nella variabile SampleRTT, essi variano da segmento a segmento in base alle congestione dei router e del carico degli host. Per ottenere una stima reale, si fa una media pesata di i valori, essa viene aggiornata a ogni campione in quanto quest'ultimo indica le condizioni della rete più recenti:
EstimatedRTT= (1-a)*EstimatedRTT + a*SampleRTT
con a=0,125
Oltre a una stima su RTT, è anche importante avere la cosiddetta variabilità di RTT, DevRTT, ovvero la stima di quanto SampleRTT si discosta da EstimatedRTT:
DevRTT= (1-b)*DevRTT + b*|SampleRTT - EstimatedRTT|
con b=0,25
EstimatedRTT e DevRTT vengono entrambi utilizzati per il calcolo del timeout, esso non deve essere molto maggiore di EstimatedRTT onde evitare lentezza nelle ritrasmissioni, perciò si utilizza DevRTT per indicare la possibile variazione, facendo così da "margine":
Timeout = EstimatedRTT + 4*DevRTT
Il valore iniziale di Timeout viene raccomandato a 1 secondo, quando si verifica Timeout viene raddoppiato per evitare futuri timeout prematuri, tuttavia quando si riceve un segmento, EstimatedRTT viene aggiornato e Timeout viene ricalcolato.

Trasferimento dati affidabile

Il trasferimento dati è possibile attraverso tre eventi principali:

- dati dall'applicazione: TCP incapsula i dati in un segmento, li passa a IP e avvia il timer se non è in funzione;
- timeout: il segmento viene ritrasmesso e il timer viene riavviato;
- ricezione di un ACK: viene confrontata la base col sequence number dell'ACK, se quest'ultimo è maggiore, allora il mittente aggiorna la sendBase fino al pacchetto non ancora confermato dall'ACK. Se non c'è nessun pacchetto in attesa, il time viene riavviato.

Per esempio, se bisogna trasferire un segmento con 8 byte di dati e sequence number 92, il client deve attendere l'ACK 100. Se scatta il timeout, il segmento viene ritrasmesso. Nel caso in cui il segmento abbia 20 byte di dati e sequence number 100, l'ACK che il client deve attendere è 120.

Raddoppio del tempo di timeout

Quando si verifica un timeout, TCP ritrasmette il segmento non confermato col sequence number più basso. ogni volta che succede, il tempo di timeout viene raddoppiato rispetto al precedente valore, offrendo una forma limitata di controllo della congestione.

Trasmissione rapida

Il mittente può rilevare una possibile perdita di un segmento ben prima dell'evento di timeout, questo è possibile grazie alla ricezione di ACK duplicati relativi a un segmento già

ricevuto dall'utente. Quando il destinatario riceve un segmento con un sequence number maggiore rispetto a quello atteso e in ordine, rivela un buco nel flusso di dati causati da possibili segmenti persi o riordinati all'interno della rete. Per risolvere il proble il destinatario invia nuovamente l'ACK relativo all'ultimo segmento ricevuto in ordine. Se il mittente riceve 3 ACK relativo allo stesso dato, considera il segmento come perduto e quindi effettua la ritrasmissione del segmento mancante prima della scadenza del timer.

GBN o ripetizione selettiva?

TCP è un protocollo in cui gli ACK sono cumulativi e i segmenti vengono ricevuti correttamente anche se in disordine, quindi lo si può considerare un'istanza di GBN, tuttavia alcune implementazione memorizzano i segmenti fuori sequenza in un buffer e, in più, ritrasmette al massimo il segmento perso (in alcuni casi nemmeno quello se il mittente riceve l'ACK relativo al segmento successivo). Una modifica di TCP è il cosiddetto riscontro selettivo, in cui il destinatario manda ACK in modo selettivo per i segmenti fuori sequenza anzichè in modo cumulativo.

Controllo di flusso

TCP permette di eseguire il controllo di flusso tra mittente e destinatario per evitare che il primo saturi il buffer di ricezione del secondo, esso è possibile confrontando le velocità d'invio del mittente e quella di lettura del ricevente. Per eseguire questo controllo, TCP mantine una variabile chiamata finestra di ricezione, la quale fornisce al mittente indicazioni riguardanti lo spazio libero nel buffer del destinatario. Se il mittente vuole inviare un grosso file al destinatario,quest'ultimo alloca un buffer la cui dimensione è data da ReceiveBuffer e per il controllo vengono definite le seguenti variabili:

- LastByteRead: numero dell'ultimo byte del flusso letto dal ricevente;
- LastByteReceived: numero dell'ultimo byte del flusso copiato nel buffer di B.

Dal momento che TCP non può mandare in overflow il buffer, si deve rispettare la seguente condizione:
LastByteReceived-LastByteRead<=ReceiveBuffer
la finestra di ricezione rwin verrà impostata nel seguente modo:
rwin= ReceiveBuffer - [LastByteReceived - LastByteRead]
Il ricevente comunica al mittente lo spazio disponibile nel buffer scrivendo il valore di rwin nell'apposito campo del segmento e lo manda. Rwin viene inizializzato con ReceiveBuffer, sarà compito del ricevente aggiornarlo e tenere traccia delle specifiche variabili. Il mittente tiene traccia di due variabili, LastByteSent (l'ultimo byte inviato del flusso) e LastByteAcked (l'ultimo byte confermato con ACK). Mantenendo la quantità di dati senza ACK sotto rwin, si garantisce che il mittente non mandi in overflow il buffer del destinatario, assicurandosi che la seguente uguaglianza perduri per tutta la connessione:
LastByteSent - LastByteAcked <= rwin
Vi è però un problema tecnico: quando rwin è uguale a 0 e il mittente non ha più nulla da inviare, il ricevente svuota il buffer ma TCP non invia più nuovi segmenti con rwin aggiornato, il mittente così non viene informato e rimane bloccato. Per risolvere questo problema, le specifiche TCP richiedono che il mittente continui a inviare dati anche quando rwin è 0, il destinatario risponderà con degli ACK.

Gestione della connessione TCP

Quando un processo vuole inizializzare una connessione con un altro processo, innanzitutto informa il lato client di TCP di volerlo fare e dopo vengono eseguiti i seguenti passi:

- Il lato client TCP invia al server un segmento con SYN a 1, prima però sceglie a caso un sequence number e lo inserisce in esso;
- quando il segmento arriva al server, alloca i buffer e le variabili per la connessione e invia un segmento al client con SYN a 1, il campo ACK prende il valore successivo al sequence number del segmento ricevuto e viene scelto un sequence number casuale;
- Il client, dopo aver ricevuto il segmento, alloca i buffer e le variabili e invia un'altro segmento con SYN a 0 e con il sequence number successivo del segmento ricevuto nel campo ACK. Nel campo dati possono essere presenti dati utili.

Ogni processo che partecipa alla connessione può terminarla inviando un segmento con FIN a 1 e deallocando le risorse. Alla ricezione del segmento, il client/server spedisce il proprio segmento di shutdown e attende la ricezione dell'ACK. I protocolli TCP in esecuzione attraversano vari stati:

- inizialmente si parte dallo stato CLOSED;
- quando si vuole inizializzare una connessione, si invia il segmento e si passa a SYN_SENT in cui si attende l'ACK;
- Una volta ricevuto, si passa a ESTABLISHED in cui si possono inviare/ricevere i segmenti con dati utili;
- Quando si vuole terminare la connessione, si invia l'apposito segmento e si passa a FIN_WAIT_1 in cui si attende l'ACK;
- Ricevuto il segmento, si passa a FIN_WAIT_2 in cui si attende un segmento con FIN a 1;
- Ricevuto il segmento, si manda un'ACK e si passa a TIME_WAIT, esso consentendo di ritrasmettere un ACK nel caso in cui il primo vada perduto, la sua durata dipende dall'implementazione.

Per analizzare una specifica porta TCP su un host bersagli, nmmap manderà a esso un segmento SYN verso la porta di destinazione scelta, vi possono essere tre possibili risultati:

- il mittente riceve un ACK dal destinatario, quindi un'applicazione è in esecuzione sulla porta scelta, nmap restituirà "open";
- il mittente riceve un segmento RST, quindi non vi è alcuna applicazione su questa porta però essa non è bloccata da firewall;
- il mittente non riceve nulla, probabilmente il segmento SYN è stato bloccato da un firewall;

Nmap può anche raccogliere informazioni riguardanti porta UDP aperte, firewall e relative configurazioni.

Principii di controllo della congestione

La congestione all'interno di una rete succede quando i router non riescono a smaltire in tempo tutti i pacchetti nei loro buffer, causando ritardi o perdite. Per evitare ciò, TCP utilizza appositi meccanismi che permettono di rilevarla, suppondento che un client stia inviando richieste a un server a una frequenza di f bps senza eseguire nessun controllo, il tasso

d'invio del client è quindi f. I pacchetti che vanno dal client al server, passano per un collegamento condiviso con rate R e un router, quest'ultimo memorizza i pacchetti quando la velocità di arrivo supera quella del collegamento. Qui è possibile misurare il throughput per connessione in funzione del tasso di invio: finchè non supera R/2, equivale alla velocità di invio del client, altrimenti è R/2. Questo limite superiore è causato dalla condivisione della capacità del collegamento tra le due connessioni. Quando la velocità di invio si avvicina a R/2, il ritardo medio cresce sempre di più, tendendo a infinito quando si supera la soglia.
Assumendo ora che il buffer del router sia limitato, vi è una possibilità che i pacchetti vengano scartati, bisogna quindi prestare attenzione al tasso di invio. Considerando f' come il tasso con cui il livello trasporto invia i segmenti, le prestazioni dipendono da come viene effettuata la ritrasmissione: se il buffer ha spazio disponibile, non vi è alcun smarrimento e quindi f'=f e il throughput è f. Se però il mittente ritrasmette il pacchetto quando è certo che è stato perso, ammettendo un valore di f' equivalente a R/2, il tasso di consegna dei pacchetti è R/3, in cui il mittente deve compensare le perdite con delle ritrasmissioni. Considerando ora un timeout prematura da parte del mittente, si ha una situazione in cui si ritrasmette un pacchetto che ha subito ritardi in coda ma non è andato perso, quindi sia la copia che quello originale possono raggiungere il destinatario. Il lavoro effettuato dal mittente è quindi sprecato dato che il destinatario scarterà una delle due copie quando ha ricevuto l'altra. Queste ritrasmissioni non necessarie costringono il mittente a utilizzare la banda e per questo si ha un throughput equivalente a R/$ quando f' tende a R/2.
Considerando ora uno scenario con più collegamenti con rate R, buffer limitati e 4 host in cui vi sono meccanismi di timeout e ritrasmissione per il trasferimento affidabile e un carico f, se A manda segmenti a C, deve passare per R1 e R2 (condivisi con la connessione B-D), si ha che con valori piccoli di f, il throughput è direttamente proporzionale al traffico in rete mentre, con valori di f grandi, il traffico inoltrato da R2 a R1 non può essere più grande di R. Con valori f' estremamente grandi per la connessione B-D, il tasso di arrivo è maggiore rispetto a quello di A-C, quindi entrano in competizione per ottenere spazio nel buffer. Il tasso A-C diventa sempre più picco all'aumentare di quello di B-D, se quest'ultimo tende a infinito, esso è in grado di riempire un buffer vuoto di un router annullando il tasso del primo. La diminuzione del throughput è dal fatto che il lavoro effettuato dal primo router verso il secondo viene sprecato a causa dei pacchetti scartati da quest'ultimo.

Approcci al controllo della congestione

TCP deve necessariamente utilizzare il controllo end-to-end dato che IP non offre feedback sulla congestione della rete. La perdita di segmenti viene considerata come congestione e quindi TCP dimunuisce l'ampiezza della finestra. Un'altra soluzione è il controllo assistito, in cui i router forniscono dei feedback al mittente relativi all congestione, essi possono essere bit oppure cose più sofisticate. Il controllo della congestione ATM ABR permette di informare il mittente esplicitamente sulla frequenza trasmissiva che i router possono supportare su ogni collegamento uscente, questa informazione può essere fornita in due modi: un avviso diretto tramite chokepacket (un pacchetto di strozzatura) oppure un pacchetto impostato appositamente dal router per indicare la congestione.

Controllo della congestione

L'approccio scelto da TCP per controllare la congestione è quello di imporre a ogni mittente un limite alla velocità di invio sulla propria connessione, se uno di essi si accorge di condizioni di scarso traffico, aumenta il suo tasso trasmissivo, lo diminuisce in caso contrario. TCP riduce la velocità di invio utilizzando le variabili LastByteRead e rwin più una terza variabile: la finestra di congestione (cwin) con cui impone un vincolo alla velocità di immissione di traffico. La quantità di dati non confermata da ACK non può eccedere oltre il minimo tra i valori delle due finestre:
LastByteSent - LastByteAcked <= min{cwin, rwin}
La quantità di dati ancora da confermare è quindi limitata solamente da cwin, il vincolo però limita la velocità trasmissiva in modo indiretto, infatti considerando una connessione con ritardi e perdite trascurabili, all'inizio di ogni RTT, il vincolo permette la trasmissione di cwin byte di dati su di essa e al termine riceve i relativi ACK, la velocità d'invio è quindi approssimabile a cwin/RTT bps. Considerando ora un mittente che percepisce la congestione con la ricezione di tre ACK duplicati o un timeout, si ha un caso di congestione eccessiva quando uno o più router vanno in overflow e, di conseguenza, causando un'evento di perdita. Un caso ottimista è quello in cui non si verificano perdite, gli ACK vengono ricevuti dal mittente, TCP considera quindi tali ricezioni come una segnalazione che tutto va bene, di conseguenza aumenta la finestra di congestione in base alla frequenza di ricezione degli ACK (per questo motivo TCP è detto auto-temporizzato). Se i mittenti trasmettono a una frequenza troppo alta, c'è il rischio di incorrere nella congestione, al contrario "una cautela maggiore" causerebbe un sottoutilizzo della banda, quindi si trova una via di mezzo. Per stabilire con quale velocità trasmettere i segmenti, si seguono i seguenti punti:

- La perdita di un segmento indica congestione e quindi il tasso di trasmissione diminuisce a ogni perdita;
- L'ACK indica che va tutto bene e quindi si aumenta il tasso di trasmissione fino a quando non arrivano ACK duplicati;
- Rilevamento della larghezza di banda: il mittente aumenta il proprio tasso fino a quando non iniziano le congestione, rallenta e quindi inizia una nuova fase di rilevamento dato che il punto di congestione potrebbe essere cambiato.

Un'algoritmo per rilevare le congestione è costituito da tre parti: slow start, congestion avoidance e fast recovery.

Slow start

All'inizio di questa fase, il valore di cwin è inizializzato a 1 MSS, comportando quindi una velocità iniziale approssimabile a MSS/RTT. Dal momento che la banda disponibile è più grande, il mittente aumenta cwin di 1 MSS per ogni ACK ricevuto e invia un segmento in più. La velocità cresce quindi in modo esponenziale. Quando avviene una perdita, TCP riporta cwin a 1 e ricomincia il processo, in più pone la variabile di stato ssthresh (soglia di slow start) a cwin/2. Questa fase termina quando il valore di cwin è maggiore o uguale a ssthresh oppure alla ricezione di tre ACK duplicati.

Congestion avoidance

In questa fase, cwin è circa la metà del suo valore precedente, ovvero quando è stata rilevata la congestione. Esso, invece di raddoppiare, viene incrementato di 1 MSS ad ogni RTT, questo metodo è ottenibile in diversi modi, il più comune è incrementarlo per MSS*(MSS/cwin) ogni volta che viene ricevuto un ACK. Al verificarsi di un timeout, si comporta allo stesso modo di slow start, tuttavia in caso di ricezione di tre ACK duplicati, cwin viene dimezzato (aggiungendo 3 MSS per contare gli ACK duplicati) e ssthresh viene impostato come la metà di cwin al momento della ricezione.

Fast Recovery

Questa fase è l'unica non obbligatoria ma raccomandata e consiste nell'incrementare cwin per ogni ACK duplicato ricevuto relativo al segmento perso che ha casuato l'entrata in questa fase. TCP ritorna in congestion avoidance quando arriva un'ACK di un segmento perso dopo aver ridotto cwin. Se si verifica un timeout, invece, si passa a slow start e si porta cwin a 1. Il tipo di TCP che adotta questa fase è detto Reno che, a differenza della versione precedente (Tahoe) è che la prima pone la soglia a metà cwin e cresce linearmente mentre la seconda ricomincia il giro.

Retrospettiva sul controllo della congestione

Assumendo che le perdite vengano segnalate solamente con la ricezione di tre ACK duplicati, il controllo della congestione consiste quindi in un incremento lineare di cwin pari a 1 MSS per RTT e a un decremento moltiplicativo in caso di perdita. Questo permette un comportamento a dente di sega in cui cwin aumenta linearmente per poi scendere di un fattore 2 quando vi sono delle perdite e così via. L'algoritmo Vegas permette di evitare le congestioni mantenendo un buon throughput, esso rileva le congestioni prima che si verifichino e abbassa la velocità di perdita in modo lineare quando vi è una perdita imminente, quest'ultima viene predetta osservando il valore di RTT: più è grande, più è probabile.

Descrizione del throughput

Durante un RTT, la velocità di invio è in funzione di cwin e dell'RTT corrente, quindi la frequenza trasmissiva è approssimabile a cwin/RTT. TCP cerca anche banda aggiuntiva aumentando di 1 MSS cwin a ogni RTT fino al verificarsi di una perdita. Segnando come W il valore di cwin con cui si verifica tale evento, la velocità trasmissiva si trova tra W/2RTT e W/RTT. La rete elimina il pacchetto quando la velocità sale a W/RTT, di conseguenza viene dimezzata e si incrementa cwin di MSS/RTT fino a quando non raggiunge W/RTT. Per calcolare il throughput medio si una la seguente formula:

thrM=0,75*W/RTT

La formula qua sopra non funziona però in tutti i casi, infatti supponendo di voler inviare 10 Gbps con segmenti da 1500 byte e RTT di 100 ms, con essa si ottengono 83.333 segmento per avere un throughput di 10 Gbps. Per risolvere questo problema si utilizza un'ulteriore formula in funzione del tasso di perdita L, RTT e MSS:

thrM=1,22*MSS/(RTT*radice(L))

In questo modo gli odierni algoritmo possono tollerare una perdita di di segmenti di 2*pow(10,10) quindi si vuole un throughput di 10 Gbps.

Fairness

Considerando K connessioni TCP, ognuna con un differente percorso ma con un collegamento in comune con rate R che costituisce un collo di bottiglia mentre tutti gli altri non sono congestionati e hanno alto rate, si deve trasferire un file di grandi dimensioni attraverso il collo di bottiglia. Un meccanismo di controllo della congestione, per essere equo, deve avere una velocità trasmissiva media in ogni connessione approssimabile a R/K, ognuna di essa avrà la stessa porzione di banda. Consideriamo adesso due connessioni TCP con un router e un collegamento in comune, inoltre assumiamo che MSS e RTT siano gli stessi per entrambe e che bisogna trasmettere una grande quantità di dati. Se TCP sta suddividendo equamente la banda del collegamento, il throughput dovrebbe essere R/2, non è però piacevole dividere la banda in porzioni uguali, quindi l'obiettivo è quello di ottenere un throughput compreso tra R/2 e R, ovvero il massimo utilizzo.Supponiamo di avere un finestra in cui le due connessioni in cui il primo throughput non è molto maggiore di R/2 mentre il secondo è minore, dal momento che l'utilizzo congiunto di banda è minore di R, non si verificheranno perdite e connessione incrementeranno la loro finestra fino a quando non vi è una perdita. L'utilizzo congiunto della banda è quindi superiore a R, il che vuol dire che possibili perdite causeranno la diminuzione delle finestre. I thoughput continueranno a salire e scendere avvicinandosi sempre più a R/2. Questo scenario serve a far intendere come TCP distribuisca equamente la banda tra le connessioni, nella pratica invece queste condizioni non si verificheranno mai e quindi le applicazioni avranno porzioni di banda differenti. Inoltre è stato dimostrato che quando più connessioni condivido un collo di bottiglia, quelle con RTT inferiore riescono ad acquisire banda più velocemente non appena si libera.

Fairness e UDP

Alcune applicazioni preferiscono UDP a TCP perchè hanno bisogno di una velocità minima e quindi non possono essere utilizzate con quest'ultimo a causa dei rallentamenti dovuti alla congestione. Le applicazioni UDP, oltre a non avere nessun controllo di flusso o congestione, non cooperano e non adeguano la loro velocità trasmissiva in modo appropriato tanto da essere in grado di soffocare il traffico TCP. Nonostante questo, anche se potessimo forzare il traffico UDP a essere equo, il problema non sarebbe risolto dato che nulla vieta a un'applicazione TCP di utilizzare più connessioni in parallelo.

Notifica esplicita di congestione (ECN)

Le ECN sono forme di controllo della congestione assistito in cui è possibile la segnalazione esplicita di congestione attraverso varie implementazioni di IP e TCP. A livello di rete vengono utilizzati due bit in un apposito campo del protocollo IP, nel caso un router sia congestionato, essi vengono impostati e consegnati al destinatario il quale informa il mittente. Lo standard RFC non definisce quando un router è congestionato o meno (anche se lo raccomando quando la congestione è persistente), questa decisione è una scelta di configurazione dell'operatore di rete. A livello trasporto, quando il destinatario riceve la

segnalazione, informa il mittente utilizzando un segmento col bit ECE a 1, quest'ultimo dimezza la finestra di congestione e invia al ricevente un'ulteriore segmento col bit CWR a 1.

Livello Rete

Il livello di rete permette il trasferimento dei pacchetti in Internet da un host all'altro, il protocollo principale è IP il quale, data la sua unicità, deve essere supportato da tutti i dispositivi che lavorano in tale livello o in uno superiore. Altri protocolli a livello di rete permettono il calcolo dei percorsi che i pacchetti devono seguire per arrivare a destinazione. Il ruolo principale di questo livello è trasferire i pacchetti da un router all'altro fino a quando non arriva a destinazione, questo è possibile attraverso due funzioni:

- Inoltro: All'arrivo di un pacchetto, bisogna trasferirlo nel collegamento in uscita appropriato;
- Instradamento: si calcola il percorso che i pacchetti devono seguire attraverso gli algoritmi di instradamento, è solitamente implementato via software dato che avviene in grandi scale temporali.

L'inoltro dei pacchetti è effettuato dai router attraverso l'ausilio di tabelle di instradamento, il risultato del confronto tra le righe della tabella e i valori del pacchetto determina il percorso da seguire.

Approcci di controllo

Un approccio più tradizionale indica che gli operatori dovrebbero interagire tra loro per assicurarsi che le tabelle siano compilate in modo da portare i pacchetti a destinazione.
Un'altro approccio prevede che un controller remoto compili e distribuisca a tutti i router, esso può essere implementato in un data center o essere gestito da un ISP/terza parte. Questo tipo di approccio è detto software-defined networking in quando il controller compila le tabelle e interagisce coi router che implementano il software.

Modelli di servizio

I modelli di servizio della rete definisce le caratteristiche del trasporto dei pacchetti tra origine e destinazione, essi potrebbero offrire:

- Consegna garantita: assicura la consegna del pacchetto alla destinazione;
- Consegna garantita con ritardo limitato: assicura la consegna del pacchetto a destinazione e il rispetto di un ritardo specificato;
- Consegna ordinata: assicura che i pacchetti giungano a destinazione nell'ordine in cui sono stati inviati;
- Banda minima garantita: emula un comportamento con un bit rate specificato, non vi è garanzia sulla consegna dei pacchetti, però finchè l'host trasmette i bit a una velocità inferiore a quella del rate dato, non si verificheranno perdite;
- Servizi di sicurezza: il livello di rete può cifrare tutti i pacchetti inviati, la riservatezza viene fornita a tutti i segmenti di livello trasporto.

Il livello di rete mette a disposizione solamente un servizio: best-effort, ovvero non vi è garanzia sulla consegna del pacchetto, sul ritardo e sulla banda minima ma si farà del proprio meglio. Nonostante lo sviluppo di possibili alternativa, il modello best-effort rimane abbastanza buono da supportare quando combinato con una banda di larghezza adeguata.

Struttura di un router

Un router è generalmente strutturato nel seguente modo:

- Porte di ingresso: svolgono funzioni a livello fisico e collegamento in ingresso al router, necessarie per inter-operare con l'altro capo del collegamento di ingresso. Esse svolgono la funzione di ricerca in modo che il pacchetto inoltrato nella struttura di commutazione esca nell'uscita corretta;
- Struttura di commutazione: connette fisicamente le porte di ingresso con quelle di uscita;
- Porte di uscita: Memorizzano i pacchetti che provengono dalla struttura di commutazione e li trasmettono nel collegamento in uscita corretto, solitamente è accoppiata alla medesima porta di ingresso della stessa scheda.
- Processore di instradamento: esegue le funzioni del piano di controllo e i protocolli di instradamento, gestisce le tabelle di instradamento e le informazioni sui collegamenti ed elabora la tabella di inoltro del router. In più è responsabile della comunicazione con il controller remoto ricevendo le occorrenze della tabella e installandole alle porte di ingresso.

L'implementazione delle porte di ingresso, di uscita e della struttura di commutazione, sono quasi sempre in hardware, esso si occupa dell'inoltro e può essere proprietario oppure assemblato da terze parti. Le funzioni di controllo sono invece implementare via software ed eseguite dal processore. L'inoltro può essere di due tipi:

- Basato sulla destinazione: il router legge la destinazione del pacchetto e ne determina l'uscita adatta;
- generalizzato: il router determina un'uscita adatta al pacchetto in base in base a molti fattori come la sua origine.

Inoltro basato sulla destinazione

L'elaborazione è centrale per la funzionalità del router, infatti è qui che viene determinata la porta di uscita in cui dirigere il pacchetto attraverso la struttura di commutazione. Una copia della tabella di inoltro è solitamente memorizzata su ogni porta di ingresso, considerando il caso in cui l'inoltro è basato sulla destinazione, una possibile implementazione della tabella stessa è quella di avere una riga per ogni indirizzo di destinazione, questa opzione non viene però presa in considerazione data la mole degli indirizzi. Come alternativa, il router utilizza un prefisso dell'indirizzo di destinazione per indicare la porta di destinazione, nel caso vi siano più corrispondenze, si adotta la regola della riga di corrispondenza al prefisso più lungo, ovvero si sceglie la porta di destinazione associata al prefisso più lungo tra le corrispondenze della tabella. Una volta determinata la porta di output, il pacchetto viene inviato alla struttura di commutazione, vi sono però architetture in cui il pacchetto potrebbe essere fermato temporaneamente nel caso in cui la commutazione venga utilizzata da altre porte di ingresso, quindi verrà accodato alla relativa porta di input e schedulato per il passaggio a essa. Nonostante l ricerca sia l'azione più importante dell'elaborazione alle porte di input, ve ne sono altre:

- elaborazione a livello fisico/collegamento;
- il numero di versione del pacchetto;
- il checksum;

- il TTL;
- i contatori della gestione di rete;

L'azione di cercare la corrispondenza tra IP di destinazione e l'invio del pacchetto alla porta di uscita specificata, è un caso specifico di match-action.

Struttura di commutazione

La commutazione può essere ottenuta in vari modi:

- Commutazione in memoria: dato che i primi router erano calcolatori e la commutazione tra ingresso e uscita veniva effettuata sotto il controllo diretto della CPU. Le porte di ingresso e uscita funzionavano come dispositivi I/O: Quando sopraggiungeva un pacchetto, ne veniva segnalato l'arrivo tramite un interrupt in cui il processore, per gestirlo, doveva estrarre da esso l'IP di destinazione. L'ampiezza della memoria è tale da poter leggere/scrivere P pacchetti al secondo, il throughput complessivo è quindi inferiore a P/2 per necessità. i router attuali, a differenza dei precedenti, effettuano la ricerca dell' IP di destinazione e la memorizzazione del pacchetto direttamente sulla scheda e non dal processore.
- Commutazione tramite bus: le porte di ingresso trasferiscono il pacchetto direttamente a quelle di uscita tramite un bus condiviso e senza l'intervento del processore. Viene solitamente utilizzato aggiungendo un'etichetta al pacchetto da trasferire, essa verrà rimossa dalla porta di output in quanto è utilizzata solamente all'interno della commutazione. Quando vi sono più pacchetti provenienti da più ingressi diretti verso un'uscita, essi dovranno aspettare in quando è possibile trasferire solamente un pacchetto alla volta;
- Commutazione attraverso una rete di interconnessione: per superare la limitazione di banda del singolo bus condiviso, è possibile utilizzare una rete di interconnessione, una matrice formata da 2n bus che connettono n porte di input con n di output. Ogni bus verticale interseca tutti gli orizzontali a un punto di incrocio che può essere aperto o chiuso in qualsiasi momento dal controller della commutazione. Al contrario degli altri due approcci, in questo tipo è possibile inoltrare più pacchetti in parallelo, la matrice è quindi detta non-blocking, ovvero che un pacchetto in via di inoltro verso un'uscita non viene bloccato tranne nel caso in cui non vi sia un'altro pacchetto in inoltro sulla stessa uscita. Tuttavia, un pacchetto dovrà accodarsi nel caso ve ne sia un altro diretto sulla stessa uscita. Alcune reti più sofisticate permettono, attraverso elementi di commutazione a più stadi, l'inoltro di pacchetti da più ingressi a un'unica uscita contemporaneamente.

Elaborazione alle porte di uscita

L'elaborazione alle porte in uscita prende i pacchetti dalle porte di output e li trasmette sul collegamento uscente, può capitare però che si formino accodamenti sia in input che in output, la loro lunghezza dipende dalla quantità di traffico presente in rete, dalla velocità della commutazione e dalla memoria del router dato che, quando è piena, causa perdite. e la commutazione non è abbastanza veloce, possono verificarsi accodamenti in ingresso. Considerando una commutazione a rete di interconnessione e supponendo che i collegamenti abbiano lo stesso rate, i pacchetti possono essere trasferiti da ingresso a uscita con lo stesso intervallo con cui essi vengono ricevuti e che possano essere trasferiti in

modo FCFS, é possibile trasferire più pacchetti in parallelo quando le porte di uscita sono differenti, se non è così, tutti i pacchetti tranne uno dovranno attendere in coda. Quando la commutazione vuole trasferire un pacchetto da un ingresso a un'uscita perchè bloccato da un altro pacchetto, questo fenomeno è detto blocco in testa alla coda.
Per quanto riguarda l'accodamento in uscita, ne tempo richiesto per ricevere un singolo pacchetto, ne arriveranno altri dalle porte di input, essi dovranno mettersi in coda. La formazione di code in uscita è possibile anche quando la commutazione è molto più veloce della velocità complessiva delle porte. In assenza di memoria per inserire un pacchetto nel buffer, sono possibili due approcci: si scarta il pacchetto (drop-tail) oppure si eliminano alcuni pacchetti dal buffer per inserirlo. In alcuni casi conviene eseguire la seconda scelta in quanto si fornisce un segnale di congestione al mittente. Quando vi sono più pacchetti accodati sulle porte di output, l'ordine di trasmissione viene stabilito da uno scheduler apposito. La lunghezza della coda inizialmente era grande quanto un RTT per la capacità del collegamento C, in seguito a causa del gran numero di flussi TCP, la dimensione necessaria odierna è RTT*C/radice(N) in cui N è il numero di porte in entrata.

Schedulazione dei pacchetti

La schedulazione dei pacchetti permette di determinare l'ordine in cui essi vengono trasferiti sul collegamento in cui, essa può essere:

- FCFS: i pacchetti vengono trasmessi nello stesso ordine in cui arrivano.
- Coda di priorità: ai pacchetti vengono inizialmente inseriti in delle code che ne determinano la priorità, lo scheduler prenderà i pacchetti dalla coda con priorità più alta e li trasmetterà sul collegamento, passerà alla successiva nel caso quest'ultima sia vuota. Nell'accodamento a priorità non prelazionabile, la trasmissione non può essere fermata una volta iniziata;
- Round robin: i pacchetti vengono suddivisi in classi di priorità e lo scheduler scegliere un pacchetto dalle due code in base a essa.
- Accodamento equo: i pacchetti sono classificati e accodati in base alla classe e verranno schedulati come il round robin, a differenza di quest'ultimo però si garantisce a ogni classe una frazione peso(i)/sommapesi del servizio, quindi per un collegamento con rate R, la classe i avrà rendimento R*peso(i)/sommapesi.

Protocollo IP

Formato dei pacchetti IPv4

Un pacchetto IPv4 è formato dai seguenti campi:

- Numero di versione (4 bit): indica la versione del protocollo e consente ai router una corretta interpretazione del pacchetto;
- Lunghezza dell'intestazione (4 bit): indicano l'inizio effettivo dei dati del pacchetto, la maggior parte però non contiene opzioni e quindi la lunghezza è di solito 20 bit;
- Tipo di servizio (4 bit): indicano il servizio utilizzato per distinguere i pacchetti di un servizio da quelli di un altro;
- Lunghezza del pacchetto (16 bit): rappresenta la lunghezza totale del pacchetto, intestazione compresa, in byte (raramente superano i 1500 byte);
- Identificatore, flag e offset di frammentazione: servono per la frammentazione;

- Tempo di vita: indica il numero di hop possibili per quel pacchetto, serve per evitare che esso permanga troppo a lungo nella rete;
- Protocollo: indica il protocollo a livello trasporto a cui bisogna passare i dati, è l'anello di congiunzione tra i due livelli.
- Checksum: consente il rilevamento degli errori sui bit dei pacchetti ricevuti, per farlo si fa la somma in complemento 1 di ogni coppia di byte e si confronta con quello memorizzato, a differenza di quello TCP/UDP, esso riguarda solo l'intestazione.
- Indirizzi IP di sorgente e destinazione: indicano il proprio IP e quello di destinazione;
- Opzioni: permettono di estendere l'intestazione IP anche se sono utilizzate sporadicamente, la loro lunghezza variabile non permette di interpretare a priori l'inizio dei dati di un pacchetto e potrebbero richiedere l'elaborazione, aumentandone il tempo;
- Dati: parte il cui è presente il segmento o altri tipi di dati;

Frammentazione IPv4

I pacchetti, per essere trasportati da un router a un altro, vengono incapsulati in un frame a livello collegamento, esso permette di limitare la lunghezza del pacchetto stesso attraverso MTU, tuttavia l'unico problema è il fatto che tra mittente e destinatario potrebbero esserci diversi protocolli a livello collegamento che possono utilizzare MTU differenti. Dato che i pacchetti potrebbero essere più grandi della misura imposta da MTU, si divide il pacchetto in frammenti e si inviano singolarmente, il destinatario avrà quindi il compito di riassemblare il pacchetto prima di inviarlo al livello trasporto. Può capitare che, dato che IP non è affidabile, i frammenti non giungano a destinazione, per questo motivo si pone l'ultimo campo dell'ultimo pacchetto a 0, tutti gli altri ce l'hanno a 1 e si utilizza il campo offset per specificare l'ordine e quindi determinare i frammenti persi.

Indirizzamento IPv4

In genere gli host sono connessi alla rete utilizzando un unico collegamento, a differenza dei router che, per poter instradare i pacchetti, devono per necessità averne due. Il confine sottile tra host/router e collegamento è detto interfaccia, ognuna di essere deve avere un proprio indirizzo IP, essi sono scritti utilizzando la notazione decimale puntata, in poche parole un indirizzo è formato da 4 numeri divisi da un punto, ognuno è compreso tra 0 e 255 ed è univoco nella rete. Per determinare che più host appartengano a una stessa rete, si utilizzano gli stessi numeri (di solito i primi 3) per tutti gli indirizzi, i restanti verranno utilizzati per distinguere l'host all'interno della rete. Questi numeri vengono fissati ponendo alcuni bit come unici per quella rete, essa è la cosiddetta maschera di sottorete in quanto l'unico numero a variare è quello che identifica l'host. Una strategia di assegnazione degli indirizzi è detta classless interdomain (CIDR) e permette si selezionare un numero x di bit dell'indirizzo, essi formeranno, oltre alla maschera di sottorete, un prefisso mentre i rimanenti 32-x bit serviranno per la distinzione degli IP. Prima di CIDR vi erano le classi IP, ognuna prevedeva rispettivamente un prefisso di 8, 16 o 24 bit, di questi due indirizzi vengono riservati rispettivamente per la rete e il broadcast (rispettivamente il primo e l'ultimo). Questo metodo però sprecherebbe indirizzi inutilmente dato che non vi è una via di mezzo tra le classi.

Ottenere un blocco di indirizzi

Per ottenere un blocco di indirizzi IP da utilizzare in una sottorete, si contatta l'ISP, esso li richiederà a ICANN, un'autorità globale responsabile della gestione dello spazio di indirizzamento. Il ruolo di ICANN non è solo allocare indirizzi IP ma anche la gestione dei root DNS e l'assegnazione/risoluzione dei nomi di dominio, in più alloca gli indirizzi ai registri Internet regionali, questi ultimi si occuperanno dell'allocazione/gestione degli indirizzi nella rispettiva regione.

DHCP (Dynamic Host Configuration Protocol)

L'assegnazione di un indirizzo IP a un host/router avviene di solito manualmente, tuttavia viene utilizzato sempre più spesso il DHCP, un metodo che permette l'assegnazione automatica di un indirizzo persistente o temporaneo. DHCP è un protocollo client-server in cui il client è di solito un host appena connesso alla rete mentre il server è solitamente un router della sottorete. Nel caso più semplice, ogni sottorete dispone di un DHCP, altrimenti è necessario un'agente di relay che conosca l'indirizzo del DHCP server per essa. Il procedimento di assegnazione avviene nel seguente modo:

- l'host appena connesso invia in broadcast il pacchetto DHCP Discover, il cui compito è quello di trovare e attivare il DHCP server;
- Il DHCP server, ricevuto il pacchetto, invia in broadcast il pacchetto DHCP offer, esso verrà tenuto in considerazione solamente dall'host precedente e contiene l'ID del messaggio, l'IP proposto, la maschera di sottorete e la durata della connessione nel caso in cui l'indirizzo sia temporaneo;
- L'host, dopo aver scelto il DHCP server, manda a esso il pacchetto DHCP request, il quale riporta i parametri di configurazione;
- In conclusione, Il DHCP server invierà all'host il pacchetto DHCP ACK per confermare i parametri richiesti.

DHCP però non è privo di difetti: quando un host si connette a una sottorete, il DHCP gli rilascia un nuovo indirizzo, tuttavia questo non è possibile quando l'applicazione è remota.

NAT (Network Address Translation)

La NAT è un metodo in cui, per evitare l'allocazione di indirizzi su tutti gli host, viene utilizzato un unico indirizzo IP per fare comunicare un host verso l'esterno, essi infatti nascondono i dettagli della rete al mondo esterno. Questo metodo funzione utilizzando una tabella in cui si segnano l'indirizzo IP e la porta utilizzati da un host, tutti i pacchetti inviati dagli host di una sottorete verranno modificati cambiando l'IP di origine con la rispettiva corrispondenza della tabella NAT. Alla ricezione, il router confronta l'intestazione del pacchetto con le righe della tabella e lo invierà all'host corrispondente. NAT ha molti detrattori in quanto non rispetta il modello TCP violando il principio end-to-end: gli host infatti dovrebbero comunicare direttamente senza intromissioni di nodi né modifiche di indirizzi e porte.

IPv6

La versione 6 di IP è stata creata con la considerazione del fatto che quelli IPv4 potessero prima o poi finire, i progettisti però colsero l'opportunità di apportare delle migliorie, infatti rispetto alla versione precedente ha:

- Indirizzamento esteso: la dimensione dell'indirizzo aumenta da 32 a 128 bit;
- Intestazione di 40 byte ottimizzata: sono stati eliminati o resi opzionali dei campi IPv4, la finestra risulta così di 40 byte fissi permettendo una rapida elaborazione dei pacchetti, una nuova codifica della opzione ne rende più flessibile l'elaborazione;
- Etichettatura dei flussi: IPv6 presenta i flussi, quindi bisogna prevedere lo smistamento del traffico tra le diverse tipologie;
- Versione: un campo che indica la versione di IP;
- Classe di traffico: simile a TOP, può essere utile per dare priorità a determinati pacchetti del flusso;
- Etichetta di flusso: identifica il flusso di datagrammi;
- Lunghezza del payload: intero senza segno che indica il numero di byte del pacchetto che seguono l'intestazione di 40 byte;
- Intestazione successiva: identifica il protocollo a cui verrà consegnato il contenuto del pacchetto;
- Limite di hop: indica il numero di hop effettuabili dal pacchetto, viene decrementato ogni volta e cancellato se 0;
- IP sorgente e di destinazione;
- Dati;

Rispetto a IPv4, IPv6 non presenta la frammentazione/riassemblaggio nei router intermedi, questa procedura infatti avviene solamente all'interno degli host sorgente e di destinazione. Se un pacchetto IPv6 è troppo grande per essere inoltrato, il router lo scarta e invia un messaggio ICMP al mittente per fargli inviare nuovamente il pacchetto con dimensione ridotta. In più non è presente il checksum, secondo i progettisti il campo IPv4 risultava ridondante dato che questa operazione viene effettuata già in altri livelli, oltre che essere continuamente ricalcolato dai router. Le opzioni non fanno più parte dell'intestazione standard, esse però sono una delle possibili intestazioni successive di IPv6.

Passaggio da IPv4 a IPv6

Anche se i sistemi IPv6 sono retro compatibili, quelli IPv4 non sono in grado di gestire pacchetti IPv6. Per risolvere questo problema si utilizza il tunnelling, ovvero si incapsula un pacchetto IPv6 in uno IPv4 e si inoltra, se il router che riceve il pacchetto può leggere IPv6, allora "lo scarta" e lo inoltra normalmente.

Inoltro generalizzato

L'avvento delle middlebox, dispositivi a livello di rete simili a router ma che non instradano pacchetti, ha creato un po' di confusione nell'ambito di Internet, per risolvere ciò si è adottato un approccio unificato alla fornitura di funzioni dei livelli, l'inoltro generalizzato: esso utilizza una tabella match-action che generalizza il concetto di tabella di inoltro basato sulla destinazione. Nella pratica queste tabelle sono implementate da un controller remoto che le

calcola, le installa e le aggiorna. Ogni occorrenza della tabella match-action (o dei flussi) contiene i seguenti campi:

- Insieme di valori dei campi di intestazione coi quali verrà confrontato il pacchetto, il confronto è implementato a livello hardware ed effettuato dalle memorie TCAM. Un pacchetto senza riscontri può essere scartato oppure inviato al controller per ulteriori elaborazioni;
- Un'insieme di contatori che vengono aggiornati quando i pacchetti vengono associati alle occorrenze nella tabella match-action, essi possono contenere i numeri di pacchetto associati all'occorrenza e informazioni temporali riguardo l'ultimo aggiornamento;
- Un'insieme di azioni che verranno effettuate quando un pacchetto è associato a un'occorrenza della tabella dei flussi.

Match

L'astrazione Openflow del match permette che esso venga effettuato sui campi delle intestazioni di tre livelli di protocolli. Il campo Ethernet type corrisponde al protocollo al livello sovrastante, i campo VLAN riguardano le reti locali virtuali, la porta di ingresso si riferisce alla porta di entrata del packet switch sul quale il pacchetto viene ricevuto, IP sorgente e destinazione del pacchetto, il campo IP riferito al tipo di servizio e le porte di sorgente e destinazione. Le occorrenze della tabella dei flussi possono anche essere windcard. Non tutti i campi dell'intestazione IP possono essere confrontati, dato che bisogna raggiungere un compromesso tra funzionalità e complessità.

Action

Ogni occorrenza della tabella dei flussi ha una lista di azione che il pacchetto deve intraprendere quando vi è una corrispondenza. Le azioni possibili possono essere:

- Inoltro: il pacchetto viene inoltrato a una data porta di uscita, o in broadcast, o in multicast oppure essere inviato al controller;
- Scarto: Il pacchetto viene scartato;
- Modifica di campi: i valori dei dieci campi di intestazione del pacchetto possono essere riscritti prima dell'inoltro;

Piano di controllo

Il piano di controllo del livello di rete indica i modi in cui vengono calcolate le tabelle di inoltro e di flusso rispettivamente nei casi di inoltro basato sulla destinazione e generalizzato, esso può essere:

- Locale: ogni singolo router della rete esegue l'algoritmo di instradamento, per calcolarla scambiano informazioni tra le loro componenti di instradamento;
- Logicamente centralizzato: Un controller calcola e distribuisce le tabelle di inoltro a ogni router. Inoltre l'astrazione match-action permette l'aggiunta di ulteriori funzioni ai router.

Il controller interagisce con l'agente di controllo (CA) in ogni router configurando e gestendo la tabella di inoltro attraverso un protocollo. Tipicamente il CA ha funzionalità minime, esso infatti comunica col controller eseguendo quello che gli ordina, in più non comunicano direttamente tra loro e non interagiscono all'elaborazione della tabella in modo attivo. Per

controllo centralizzato si intende un servizio di controllo dell'instradamento a cui si accede come un singolo punto di servizio, anche se è implementato su più server per via della tolleranza alle perdita e scalabilità delle prestazioni.

Algoritmi di Instradamento

Gli algoritmi di instradamento sono algoritmi che permettono il calcolo dei percorsi tra un mittente e un destinatario, essi calcolano il cosiddetto cammino minimo, ovvero il percorso avente il costo minore o il numero più piccolo di hop. Una rete è rappresentabile come un grafo in cui i nodi rappresentano i router e gli archi i rispettivi collegamenti, ogni collegamento ha un costo associato, esso può definire la lunghezza fisica, la velocità o il prezzo. Gli algoritmi di instradamento possono essere:

- Centralizzato: il percorso viene calcolato avendo una conoscenza globale della rete, ciò richiede che ogni router invii informazioni al controller prima del calcolo vero e proprio, gli algoritmi di questo tipo sono detti Link-state;
- Decentralizzato: ogni router calcola la propria tabella di instradamento utilizzando le proprie informazioni più quelle dei suoi vicini, il risultato verrà in seguito in seguito scambiato attraverso dei processi iterativi. Gli algoritmi di questo tipo sono detti Distance-vector in quanto elaborano un vettore di distanze verso gli altri nodi nella rete.
- Statici: i percorsi calcolati con l'algoritmo cambiano raramente, solitamente tramite l'intervento umano;
- Dinamici: determinano il percorso migliore rispondendo ai cambiamenti della rete;
- Load-sensitive: i costi dei collegamenti variano in base al traffico e quindi anche i cammini minimi per evitare la congestione;
- Load-insensitive: i costi dei collegamenti non riflettono in modo esplicito la situazione della rete;

Instradamento Link-state

Nell'instradamento Link-state, tutti i costi dei collegamenti sono disponibili fin da subito, dato che il controller si fa inviare le informazioni di ogni router della rete. Questo tipo di instradamento utilizza l'algoritmo di Dijkstra, calcolando i cammini minimi a partire da una sorgente. Per evitare le oscillazioni sono possibili due soluzioni:

- La prima è stabilire che i costi dei collegamenti non dipendono dal traffico in rete, questo però non permette di evitare i collegamenti troppo congestionati;
- Una seconda soluzione è assicurarsi che i router non lancino l'esecuzione in contemporanea, tuttavia essi si possono auto-sincronizzare tra loro. Per evitare ciò è possibile determinare su ciascun router in modo casuale l'istante di emissione dell'avviso di collegamento;

Instradamento Distance-vector

L'instradamento distance-vector è:

- distribuito dato che ciascun nodo riceve parte dell'informazione dai suoi vicini a cui restituisce i risultati;
- asincrono nel senso che non richiede che i nodi operino al passo con gli altri;

- iterativo perchè ogni processo si ripete fino a quando ci sono vicini che inviano informazioni, bloccandosi in automatico quando finiscono.

I cammini minimi vengono calcolati utilizzando l'algoritmo di Bellman-Ford, esso utilizza la seguente formula:
cammino(x,y)=min{costo(x,v)+cammino(v,y)}
in poche parole: un cammino minimo tra x e y equivale al cammino più piccolo tra un nodo v e y più il costo dell'arco (x,v). Questa formula è molto importante perchè essa fornisce le righe della tabella di instradamento del nodo x.

Modifica dei costi e guasti

Quando un nodo che esegue il distance-vector rivela un cambiamento del costo di un vicino, esso aggiorna la propria tabella e, se si verifica un cambiamento nel cammino minimo, la invia ai suoi vicini i quali faranno lo stesso. Tuttavia con questo metodo si rischia di ottenere un instradamento ciclico, il quale potrebbe aumentare di molto le distanze tra i nodi. Per risolvere questo problema, si aggiunge la cosiddetta inversione avvelenata,ovvero che se un nodo x deve andare da y passando per z, allora x avvertirà z dicendo che la sua distanza verso y è infinita, in questo modo z crede con x non abbia un percorso verso y e quindi non tenterà l'instradamento verso di esso. L'inversione avvelenata però non risolve totalmente il problema dei cicli, infatti non riguardano solamente i nodi adiacenti e quindi non possono essere rilevati con questa tecnica.

Confronto tra link-state (LS) e distance-vector (DV)

- in DV ciascun nodo dialoga solo coi vicini mentre in LS ogni nodo dialoga in broadcast con tutti gli altri ma comunica solamente coi vicini;
- in LS ogni nodo deve sapere il costo di ogni collegamento (cambiamenti inclusi) inviando dei messaggi mentre in DV il cambiamento di un costo viene propagato con l'instradamento se il cammino minimo viene cambiato;
- LS è un algoritmo quadratico mentre DV converge più lentamente e può presentare cicli;
- Con LS un router può inviare un costo sbagliato, tuttavia gli altri nodi si occupano di calcolare solamente le proprie tabelle di inoltro, garantendo così robustezza. In DV, invece, un nodo può comunicare percorsi errati a tutte le destinazioni, diffondendosi per l'intera rete.

Instradamento nei sistemi autonomi OSPF

Il modello di instradamento visto in precedenza risulta più semplice per questi due motivi:

- Scalabilità: al crescere del numero di router, il tempo richiesto per calcolare, memorizzare e comunicare le informazioni di instradamento diventa proibitivo;
- Autonomia amministrativa: Ogni ISP di Internet desidera gestire liberamente i propri router o nasconderli all'esterno;

Questi problemi sono risolvibili attraverso i sistemi autonomi (AS), gruppi di router sotto uno stesso controllo amministrativo. I router di un AS eseguono lo stesso algoritmo di instradamento e ognuno ha informazioni sugli altri.

OSPF (Open Shortst Path First)

OSPF è un protocollo link-state open source che utilizza il flooding di informazioni riguardo lo stato dei collegamenti, esso non fornisce una politica riguardo i costi dei collegamenti ma fornisce dei meccanismi in grado di determinare il cammino minimo per un dato insieme di costi. Ogni volta che si verificano cambiamenti nello stato dei collegamenti, OSPF manda in broadcast informazioni di instradamento a tutti i router, in più invia lo stato dei collegamenti periodicamente (anche se non sono cambiati). I vantaggi di OSPF sono:

- Sicurezza: gli scambi tra i router di un stesso AS possono essere autenticati: con l'autenticazione semplice i router possiedono tutti la stessa password (inclusa nel pacchetto) mentre in quella basata su MD5 si utilizza un sistema di chiavi segrete divise e configurate in ogni router, essi eseguono la funzione hash MD5 sul contenuto e lo inseriscono in un pacchetto. Alla ricezione si prende il contenuto del pacchetto, si esegue la funzione hash su esso e si confronta la chiave segreta per verificarne l'autenticità.
- Percorsi con lo stesso costo:Nel caso vi siano più cammini verso una destinazione avente lo stesso costo, OSPF ne consente l'utilizzo senza scegliene solamente uno;
- Supporto per il trasferimento unicast/multicast: MOSPF è un'estensione che aggiunge un nuovo tipo di annuncio sullo stato dei collegamenti al meccanismo di broadcast;
- supporto alles gerarchie di dominio di instradamento: possibilità di strutturare i sistemi autonomi in modo gerarchico, dividendolo in aree che eseguono algoritmi di instradamento differenti. Ogni area ha dei router di confine che si fanno carico dell'instradamento verso l'esterno. Un'area presente in ogni AS è l'area di dorsale, il cui compito principale è quello di tra tutte le altre aree. L'instradamento nell'area di un AS richiede che i pacchetti vengano prima instradati verso i router di confine e da essi all'area di destinazione.

Instradamento tra ISP attraverso il BGP

Per determinare i cammini minimi tra sorgente e destinazione che interessano più AS, si utilizza un algoritmo inter-AS: il border gateway protocol (BGP). In BGP i pacchetti vengono instradati non verso un indirizzo specifico ma verso un prefisso rappresentante una sottorete o una collezione delle stesse, le occorrenze della tabella di instradamento hanno quindi la forma (x,I) dove x è il prefisso di rete e I è il numero di interfaccia del router. BGP mette a disposizione a ciascuno router un modo per:

- Ottenere informazioni sulla ragiungibilità dei prefissi da parte dei sistemi confinanti, permettendo a ciascuna sottorete di dichiarare la propria esistenza in Internet;
- Determinare i cammini minimi verso le sottoreti: Dal momento che un router può venire a conoscenza di più cammini verso un prefisso, per determinare il migliore si esegue BGP localmente sulla base delle informazioni di raggiungibilità e delle politiche del sistema.

Distribuzione delle informazioni

Ogni router in ogni AS funge sia da gateway che da router interno. Per sapere come BGP distribuisce le informazioni, consideriamo tre AS connessi con dei collegamenti, la terza AS

invia un messaggio alla seconda per annunciare l'esistenza di un dato prefisso al suo interno. quest'ultima propagherà l'informazione alla prima AS. Nella pratica non sono gli AS a comunicare ma i router stessi utilizzando connessioni semi-permanenti, esse possono essere interne o esterne se coinvolgono o meno un router appartenente alla stessa AS.

Selezione delle rotte migliori

Quando un router annuncia un prefisso per una sessione BGP, include un certo numero di attributi insieme al prefisso, formando la cosiddetta rotta, i più importanti sono:

- AS-PATH: elenca gli AS attraversati dal prefisso, viene aggiornato ogni volta che quest'ultimo ne attraversa uno;
- NEXT-HOP: indica l'interfaccia del router inizia l'AS-PATH, in poche parole è il primo router dell'AS che viene attraversato.

Instradamento hot-potato

L'instradamento hot-potato indica il percorso avente i router NEXT-HOP col costo minore. L'idea è buttare fuori tutti i pacchetti dall'AS il prima possibile, senza preoccuparsi del costo delle restanti tratte.

Selezione delle rotte

BGP utilizza un algoritmo più complesso di hot-potato, esso ha come input l'insieme di tutte le rotte apprese e accettate verso lo stesso prefisso, invocando in seguito le seguenti regole:

- alle rotte viene assegnato un numero di preferenza, esso può farlo direttamente il router oppure viene appreso da un altro nella stessa AS, e si selezionano quelli con il valore più alto;
- Tra le rotte con lo stesso valore si seleziona quella con l'AS-PATH più corto;
- Nel caso due rotte abbiano anche la lunghezza di AS-PATH uguale, si selezione quella col NEXT-HOP più vicino;
- Nel caso rimanga ancora qualche rotta, il router si basa sugli identificatori BGP.

Anycast IP

BGP viene anche utilizzato per implementare l'anycast IP, utilizzato dal DNS. In molte applicazioni infatti si è interessati al replicare lo stesso contenuto su server di differenti aree geografiche e che l'utente acceda a un contenuto al server più vicino a esso. Durante la configurazione, le applicazioni assegnano lo stesso indirizzo IP a più server e utilizzano BGP per annunciare l'esistenza di ognuno.Quando un router riceve l'annuncio di più percorsi con lo stesso indirizzo, lo tratta come se portassero tutti alla stessa destinazione, quindi ogni router configura la propria tabella di inoltro e calcola il cammino minimo per quell'indirizzo IP.

Politiche di instradamento

Quando un router sceglie una rotta, le politiche di instradamento dell'AS possono intervenire facendo saltare ogni altra considerazione riguardante la brevità del cammino o l'hot-potato. Supponiamo di avere 3 reti stub connesse tra loro tramite tre reti di provider, ipotizziamo anche che queste ultime siano dei peer l'un l'altro. Tutto il traffico che entra in un ISP è destinato a tale rete mentre tutto quello che esce ha origine da essa. Se una rete è connessa a più ISP, essa è detta multi-homed in quanto può inviare/ricevere da tutti gli ISP

connessi. Tutte le reti sono responsabili del traffico in ingresso/uscita da quella multi-homed, quest'ultima, per impedire lo smaltimento del traffico, comunica ai suoi ISP di non avere altre destinazioni al di fuori di se stessa. Alcuni ISP possono apprendere dei percorsi da altri, quindi possono aggiungersi al cammino e presentarlo alle reti stub a cui sono connesse. Se i tre ISP sono tutti provider di dorsale, allora i provider possono pensare di non dover sopportare il peso del traffico nel caso vi sia un collegamento diretto tra gli altri due. Attualmente non esistono standard per gestire l'instradamento reciproco tra ISP dorsali, nella pratica però tutto il traffico che fluisce attraverso una dorsale ISP deve avere origine/destinazione in una rete stub.

Piano di controllo SDN

In un'architettura SDN possono essere identificate quattro caratteristiche fondamentali:

- Inoltro basato sui flussi: i pacchetti possono essere inoltrati utilizzando uno switch SDN basandosi sul valore dell'intestazione a livello trasporto, rete e collegamento. Openflow invece inoltra i pacchetti basandosi su 11 campi dell'intestazione, in contrasto con l'inoltro basato esclusivamente sulla destinazione;
- Separazione del piano dei dati da quello di controllo: il primo consiste di dispositivi semplici ma veloci, come gli switch, che effettuano il match-action nelle loro tabelle dei flussi mentre il secondo consiste di server e software che determinano e gestiscono queste tabelle;
- Controllo di rete esterno al piano dei dati: A differenza dei router tradizionali, il software non è eseguito dai router ma da server remoti rispetto agli switch nella rete. Il piano di controllo stesso è formato da due componenti: un controller SDN, che mantiene informazioni di stato sulla rete, e un insieme di applicazioni di controllo, che monitorano e controllano i dispositivi di rete;
- Una rete programmabile: La rete è programmabile grazie alle applicazioni di controllo della rete, esse utilizzano API fornite dal controller per specificare/controllare il piano dei dati.

Le SDN rappresentano quindi una separazione delle funzionalità di rete, esse infatti sono entità distinte che possono essere fornite da differenti organizzazioni. Questo è in contrasto col precedente modello SDN in cui router/switch, insieme al software di controllo, erano monolitici o integrati verticalmente. La separazione delle funzionalità ha quindi portato a un ecosistema più ricco e aperto.

Controller SDN e applicazioni di controllo

Un generico controller SDN ha funzionalità che possono essere divise in tre livelli:

- Livello di comunicazione: effettua le comunicazioni tra controller e i dispositivi controllati, esse ha bisogno di un protocollo (costituente la parte più bassa dell'architettura) che trasferisca le informazioni tra essi, in più i dispositivi devono essere in grado di comunicare eventi locali ai controller. Tali eventi forniscono una visione aggiornata dello stato della rete al controller. La comunicazione passa attraverso la cosiddetta interfaccia "southbound" del controller;
- Livello di gestione dello stato globale della rete: le decisioni finali prese dal piano di controllo richiedo che il controller abbia informazioni aggiornate sullo stato della rete. Le applicazioni di controllo possono quindi utilizzare i contatori della tabella. Lo

scopo finale è quello di determinare la tabella dei flussi dei dispositivi controllati, permettendo al controller di mantenere una copia;
- L'interfaccia a livello applicazione: il controller interagisce con le applicazioni di controllo utilizzando un'interfaccia northbound, essa permette la lettura/scrittura dello stato della rete e delle tabelle di flussi nel livello precedente. In più le applicazioni possono richiedere una notifica quando avviene un evento di cambiamento di stato in modo che possano rispondere adeguatamente a esso.

Tali servizi sono implementati nei server per ragioni di prestazioni, tolleranza agli errori e disponibilità.

Il protocollo OpenFlow

Il protocollo OpenFlow opera tra il controller e un dispositivo che implementa le relative API. Alcuni dei più importanti messaggi inviati dal controller sono i seguenti:

- Configuration:permette al controller di interrogare/impostare i parametri di configurazione;
- Modify-state: aggiunge, cancella o modifica le occorrenze della tabella dei flussi e imposta le proprietà delle porte del dispositivo;
- Read-state: raccoglie statistiche e i valori dei contatori nelle tabelle dei flussi e nelle porte;
- Send-packet: invia un pacchetto specifico fuori da una specifica porta;

Di seguito invece vi sono quelli i messaggi più importanti che il controller può ricevere:

- Flow-removed: informa il controller che un'occorrenza della tabella dei flussi è stata cancellata;
- Port-status: informa il controller di un cambiamento dello stato di una porta;
- Packet-in: invia al controller un pacchetto che non ha corrispondenze con la tabella dei flussi per ulteriori elaborazioni, è possibile farlo anche coi pacchetti avente almeno una corrispondenza a seguito di un'azione.

ICMP (Internet Control Message Protocol)

Il protocollo ICMP permette lo scambio di informazioni tra router e host a livello di rete, solitamente per la notifica degli errori. Questo protocollo è solitamente considerato parte di IP, dal punto di vista dell'architettura, però, si trova al di sopra di esso, dato che i suoi messaggi possono essere incapsulati in pacchetti. Infatti se un host riceve un pacchetto avente ICMP come protocollo di livello superiore, esso effettuerà il demultiplexing del pacchetto a ICMP. I messaggi ICMP hanno un campo tipo, un campo codice, l'intestazione e i primi 8 bit del pacchetto che ha causato la generazione del messaggio in modo che il mittente possa determinare la causa dell'errore. I messaggi ICMP permettono anche di ridurre il tasso trasmissivo al fine di evitare congestioni. Nel caso di traceroute, per evitare che il protocollo applicativo continui a inviare pacchetti, viene generato un messaggio ICMP con porta non raggiungibile e inviato alla sorgente, in questo modo essa sa che deve interrompere l'invio dei pacchetti. Una nuova versione di ICMP è stata introdotta con IPv6 introducendo nuovi tipi e codici e ridefinendo quelli vecchi.

Gestione della rete e SNMP

La gestione della rete comprende il funzionamento,l'integrazione e il coordinamento di hardware, software e personale tecnico per utilizzare le risorse della rete affinché soddisfino i requisiti di qualità del servizio e le funzionalità in tempo reale a un costo accettabile.

Infrastruttura di gestione:

- il server di gestione è un'applicazione controllata del responsabile il cui compito è quello di sovrintendere alla raccolta, elaborazione, analisi e visualizzazione delle informazioni di gestione. Qui vengono intraprese le azioni per verificare il comportamento della rete e interagisce coi dispositivi;
- Un dispositivo di rete gestito è un componente in cui si trovano oggetto quali componenti hardware o insiemi di parametri;
- A ogni oggetto da gestire sono associate informazioni in un database gestionale, un linguaggio di specifica dati assicura che la sintassi e la semantica dei dati sia ben definita per evitare ambiguità;
- L'agente di gestione è un processo che comunica col server ed esegue azione decise da quest'ultimo;
- il protocollo di gestione è un componente con cui il server può richiedere lo stato dei dispositivi e agire su di essi attraverso gli agenti, esso non gestisce direttamente la rete ma è uno strumento che l'amministratore opera su di essa.

SNMP (Simple Network Management Protocol)

SNMP è un protocollo applicativo che trasporta informazioni tra server di gestione e agenti, la modalità di funzionamento più comune è la richiesta-risposta: un'entità di gestione invia una richiesta all'agente, esso effettuerà le azione scritte in essa e invierà al mittente la risposta. Alcuni messaggi possono essere inviati anche quando non sono state mandate richieste, sono i cosiddetti messaggi trap il cui compito è segnalare situazioni eccezionali. SNMP definisce sette tipi di mssaggi detti PDU:

- GetRequest, GetNextRequest e GetBulkRequest permettono di gestire uno o più valori di oggetti MIB, gli identificatori di questi ultimi sono specificati nella porzione variabile obbligatoria della PDU. GetRequest può richiedere qualunque valore MIB, GetNetRequest richiede elenchi o tabelle di oggetti mentre GetBulkRequest restitutisce grandi quantità di dati evitando le ridondanze. In tutti e tre i casi viene restituita una Response contenente gli identificatori degli oggetti;
- SetRequest può impostare uno o più valori di oggetti MIB in un dispositivo, in caso di effettiva modifica viene restituito un Response contenente "noError";
- InformRequest trasmette informazioni MIB a un'altra entità di gestione, a confermare la ricezione viene restituito un Response con "noError";
- Response viene inviata dall'agente al server per indicare che l'azione è avvenuta con successo;
- I messaggi trap indicano situazioni eccezionali e viene emesso quando si verificano eventi cui è richiesta la notifica. Il server non è tenuto a fornire un riscontro ai messaggi trap ricevuti.

Anche se le PDU sono convogliate da qualunque protocollo di trasporto, di solito costituiscono il payload di un pacchetto UDP, quindi si può utilizzare il campo Request ID per rilevare la perdita di richieste/risposte. E' compito dell'entità di gestione decidere se ritrasmettere una richiesta entro un certo tempo e, dal momento che SNMP non prevede procedure di ritrasmissione, deve anche preoccuparsi della frequenza e durata delle ritrasmissioni.