DISCLAIMER

Il materiale contenuto nel drive è stato raccolto e richiesto tramite autorizzazione ai ragazzi frequentanti il corso di studi di Informatica dell'Università degli Studi di Salerno. Gli appunti e gli esercizi nascono da un uso e consumo degli autori che li hanno creati e risistemati per tanto non ci assumiamo la responsabilità di eventuali mancanze o difetti all'interno del materiale pubblicato.

Il materiale sarà modificato aggiungendo il logo dell'associazione, in tal caso questo possa recare problemi ad alcuni autori di materiale pubblicato, tale persona può contattarci in privato ed elimineremo o modificheremo il materiale in base alle sue preferenze.

Ringraziamo eventuali segnalazioni di errori così da poter modificare e fornire il miglior materiale possibile a supporto degli studenti.

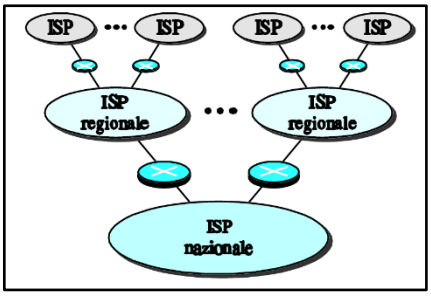
CoScienze
Associazione

Internet: la storia e la sua struttura

Una rete di calcolatori è un insieme di dispositivi o nodi indipendenti ed interconnessi tra loro attraverso un canale di comunicazione, questi nodi sono in grado di scambiarsi informazioni e cooperare al fine di ridurre i costi e avere un'affidabilità maggiore.

I tipi di Rete sono **WAN** (Wide Area Network) caratterizzata da un'area estesa e nodi distribuiti su medio-lunga distanza (nazione, continente), **LAN** (Local Area Network) e **MAN** (Metropolitan Area Network) spesso interconnesse tra loro. Oppure possiamo avere un'interrete, cioè reti differenti collegate tra loro mediante elementi di interfaccia che possono avere un'estensione mondiale.

Internet: ISP



I sistemi terminali accedono a Internet tramite gli **Internet Service Provider** che possono fornire diversi tipi di accesso alla rete (Dial-up, DSL, wireless, ecc.). Utenti connessi con **ISP** sono inoltre in grado di comunicare fra di loro, grazie alla struttura gerarchica tramite la quale i vari ISP sono connessi.

Internet è quindi una **struttura gerarchica**, composta **13 ISP** di reti di primo livello mondiali (tra cui troviamo anche Telecom Italia Sparkle). Passiamo poi a quelle di secondo livello (nazionali o distrettuali) fino ad arrivare a quelli di terzo livello, che gestiscono le connessioni locali.

I vari host all'interno di una rete fanno uso di protocolli per regolamentare l'invio e la ricezione dei dati. Gli elementi chiave di un protocollo sono:

- **Sintassi:** formato dei messaggi, ovvero l'ordine in cui i vari elementi della comunicazione devono essere presentati.
- **Semantica:** significato della sequenza di bit inviata.
- **Sincronizzazione:** uniformare le diverse velocità alla quale operano mittente e destinatario.

Generalmente i protocolli sono organizzati come livelli sovrapposti, dove un livello superiore esegue richieste al livello sottostante e i livelli uguali conversano tramite lo stesso protocollo. Questo permette di suddividere il problema e generare un sistema modulare.

Internet: Standard:

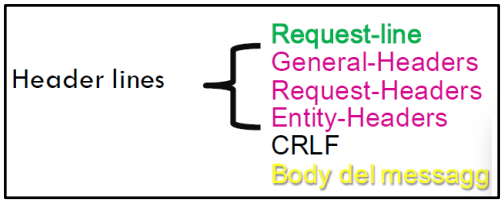
Gli standard sono di fondamentale importanza per avere un mercato libero in cui i vari produttori di dispositivi possono operare per garantire l'interoperabilità di diversi apparati. Forniscono inoltre delle linee guida a chiunque sia coinvolto nello sviluppo di una interrete pubblica. Esistono due categorie di standard:

- **De Facto:** non sono stati approvati da nessuno organizzazione ma si sono imposti nella pratica nel corso degli anni.
- **De Jure:** proposti e approvati da organizzazione riconosciute ufficialmente come la ISO o la IEEE.

Lo **Standard Internet** è lo standard adottato per il funzionamento di internet. La produzione di uno di essi segue un processo ben preciso: - si propone un "Internet draft" – Si dà la possibilità di commentarlo per un periodo di tempo – se approvato diventa un RFC (Request for Comment) che può essere utilizzato e migliorato per raggiungere il livello di maturità necessario per far diventare la propria proposta uno vero standard internet.

Il protocollo HTTP e l'interazione tra HTTP e TCP

Hypertext Transfer Protocol è un protocollo del livello di applicazione per la trasmissione di informazioni in formati e linguaggi multipli. La sua sintassi è basata su **MIME** (Multipurpose Internet Mail Extensions) il cui scopo era quello di poter definire messaggi con diversi tipi di dati.

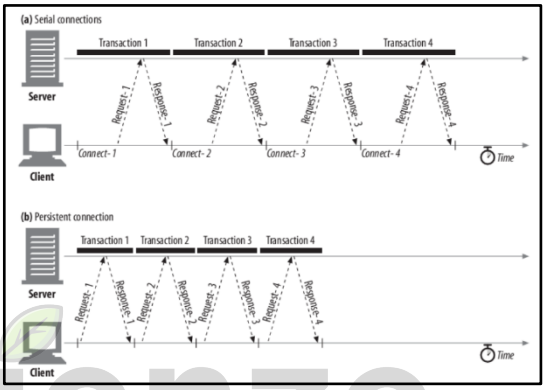


Proposto nel 1990 da Tim Berners Lee ha avuto una lenta ma significativa evoluzione, basti pensare che la versione HTTP2 è arrivata nel 2015 a 14 anni di distanza dal HTTP1.1.

Una richiesta http prevede la seguente struttura: **Request Line**, **Header Line** (general-headers, request-headers, entity-headers), **CRLF** e il **body del messaggio**. Mentre la risposta è caratterizzata da 5 classi ben definite:

- 1xx: informative.
- 2xx: successo.
- 3xx: redirezione.
- 4xx: errore client.
- 5xx: errore server.

Per la gestione delle connessioni, siccome **HTTP** è stratificato su **TCP**, le performance di quest'ultimo sono dipendenti da quelle del **TCP**. Per quanto riguarda i ritardi, questi possono dipendere dal carico della rete, taglia di messaggi di richiesta e risposta, distanza tra client e server e dettagli puramente intrinsechi del protocollo **TCP**. Il problema, infatti, del **HTTP** è che usa esclusivamente **TCP** come protocollo di trasporto che non è ottimizzato per lo scambio frequente di piccoli pacchetti **HTTP** di breve durata, ad esempio il **three-way handshake** di **TCP** e i **4 pacchetti** (syn, syn+ack, ack) di chiusura portano una richiesta http (tipicamente in 10 pacchetti) a diventare di 17 pacchetti.



Per ridurre la latenza e i costi delle connessioni TCP si potrebbe utilizzare il concetto di connessione persistente andando a trasferire coppie multiple di richiesta e risposta in una stessa connessione TCP; in più si potrebbero utilizzare delle **pipe** per far sì che avvenga la trasmissione di più richieste senza attendere l'arrivo delle risposte (ovviamente le risposte devono essere nello stesso ordine delle richieste). Questo però potrebbe portare a una maggiore congestione della rete. Il server, infine, può chiudere la connessione allo scadere di un timeout o di un numero massimo di richieste da servire sulla stessa connessione.

TCP timers

Il protocollo **TCP** usa alcuni **timers** per attivare alcune operazioni tra cui la ritrasmissione di pacchetti persi, ripetizione dello stato di avvio lento e il recupero dello stato da una connessione terminata. Questi timers però vanno ad influenzare in maniera negativa le performance del web. Questo perché il protocollo IP, non fornisce alcun dettaglio sullo stato di congestione della rete, (il sender TCP non può inviare più dati di quanti non ne possano essere ricevuti) costringendo i sender TCP ad inviare dati relativi alla **receiver windows** per non causare overflow del receiver buffer e **congestion window** per non congestionare la rete. Per tener conto sia dello stato del destinatario sia dello stato di congestione della rete, il mittente utilizza una dimensione della finestra data dal minimo tra i due valori precedenti.

I meccanismi per la gestione della finestra da parte del TCP sono 3:

- **Avvio lento (aumento moltiplicativo):** tecnica che consiste nell'aumentare in modo esponenziale la finestra di congestione fino a raggiungere la saturazione della rete. (aumento si lento, ma non così tanto).
- **Prevenzione della congestione (aumento additivo):** tecnica molto simile alla precedente ma che utilizza un aumento di tipo additivo, con lo scopo di ridurre i rischi di creare congestione. Viene utilizzato di solito per effettuare un aumento a grana più fine una volta terminato quello di tipo moltiplicativo.
- **Rilevamento della congestione (diminuzione moltiplicativa):** tecnica utilizzata quando si verificano situazioni di congestione e quindi la dimensione della finestra di congestione deve essere decrementata. Viene chiamata diminuzione moltiplicativa in quanto ad ogni iterazione la dimensione della finestra di congestione viene dimezzata.

La creazione di una connessione TCP può essere ritardata di diversi secondi a causa di un **retrasmission timeout (RTO)** troppo grande, infatti i Web downloads richiedono tempo per essere completati per dare tempo al TCP sender di rilevare che un pacchetto IP è andato perso.

Ogni richiesta http richiede di stabilire una connessione TCP, l'invio della richiesta, la ricezione della risposta e il rendering della pagina richiesta. Un ritardo in una di queste fasi è subito visibile trattandosi di un'applicazione interattiva. Per fare in modo di stabilire una connessione TCP si utilizza il concetto di three-way handshake. In caso di successo il client trasmette la richiesta http dopo un RTT, in caso di pacchetto perso (Syn o syn-ack) il protocollo TCP può rivelarlo tramite ack duplicati o retrasmission timeout (RTO).

Ma un RTO grande = latenza troppo alta || RTO piccolo = ritrasmissioni inutili

La soluzione è far scegliere al mittente l'RTO sulla base dell'RTT misurato verso il destinatario. Supponendo di avere un RTT di 3 secondi, alla perdita di un pacchetto viene raddoppiato per evitare di sovraccaricare una rete già congestionata. L'utente medio opta per il "interrompi-ricarica" appena percepisce la latenza del server, ma in questo modo aumenta solo il carico sulla rete e sul server in quanto permette di inviare pacchetti multipli in brevi periodi di tempo aumentando il traffico (già congestionato).

Perdita pacchetto -> I timeout di ritrasmissione lunghi nel mezzo di un trasferimento Web sono meno frequenti per due ragioni:

Valori minori dell'RTO: il sender TCP ridefinisce il valore dell'RTO osservando il ritardo subito dai data packets. A regime il valore dell'RTO stabilito dinamicamente dal mittente TCP si avvicina al valore dell'RTT tra mittente e destinatario.

Ack duplicati: quando un pacchetto arriva a destinazione, il TCP receiver invia un ACK che riflette il numero di byte contigui arrivati fino a quel momento. In caso di mancata ricezione di un pacchetto, il destinatario non incrementa il numero di ack. Alla ricezione di 3 ACK duplicati il sender invia il pacchetto mancante senza aspettare lo scadere dell'RTO.

Tuttavia, le risposte Web consistono di una quantità di dati molto piccola (risposte web 8-12 kb), questi piccoli trasferimenti spendono la maggior parte del tempo nella fase di slow-start per il controllo della congestione. Con una congestion window piccola, la probabilità di instradare con successo pacchetti multipli dopo la perdita di un pacchetto è molto bassa e preclude al server la possibilità di mandare ulteriori pacchetti finché non riceverà l'ACK per il primo pacchetto. A questo punto bisognerà attendere lo scadere dell'RTO che comporterà uno stallo nel trasferimento dei dati dal server. Ecco la risposta alla domanda. Risultato: RTO comportano lunghi ritardi durante i trasferimenti Web, quindi, è preferibile Stop e Restart per una ritrasmissione più veloce.

Componenti software del WWW

Componenti software: Proxy o Edge server

Sono localizzati sulla comunicazione **user-agent origin-server** con lo scopo di ridurre il carico sugli origin server e per ridurre la latenza percepita dagli utenti finali.

Proxy/Edge server: intermediario che agisce come server per il client e come client per il server, ponendosi nel mezzo di una comunicazione. Le informazioni note ai proxy sono svariate: risorsa a cui accedere, frequenza degli accessi, header e body delle richieste/risposte. L'uso di SSL obbliga, inoltre, il proxy ad agire come un tunnel. Alcuni esempi di server proxy sono: apache, Nginx, Squid, ecc.

Gateway: server che agisce da intermediario per server non HTTP (Mail, GTP server). Il gateway agisce come un origin server, lo user-agent può non essere consapevole che un intermediario è presente tra sé e l'origin server.

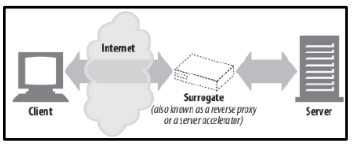
Tunnel: agisce da relay a livello sintattico non semantico. Non altera il flusso http e il suo ciclo di vita si limita alla connessione, non oltre.

Caching: strumento software che permette di mantenere copie locali dei documenti più popolari per ridurre il carico sui web server e la latenza percepita dagli utenti finali.

Filtering: componente software che permette di effettuare censure all'interno di internet oscurando siti e certi tipi di contenuti.

Document access controller: componente software utilizzato per l'implementazione di strategie di accesso, con lo scopo di aumentare la sicurezza dei dati. Rientrano in questa categoria anche i **firewall.**

Surrogate o reverse proxies: proxies che si mascherano da web server. Possono iniziare una comunicazione con altri server. Usati per migliorare le performance di slow Web server.



Content router: rete distribuita di contenuti replicati

Content transcoding: componente software che permette funzionalità di Language Translation e conversione dei contenuti per dispositivi mobili.

Anonymizer: strumenti software che permettono la rimozione di identificativi all'interno di messaggi HTTP con lo scopo di proteggere la privacy degli utilizzatori. Eseguono anche operazioni di filtering di informazioni critiche, blocco delle richieste verso siti di terze parti. Alcune tecniche utilizzate sono: eliminazione dei cookies, rimozione web bugs (immagini 1x1 per il tracking degli utenti), blocco dell'esecuzione di codice dinamico all'interno delle pagine HTML, blocco delle richieste destinate a siti terzi, blocco di banner, advertisement, meta tag, ecc.

Web Content Accessibility Guideline: sono un insieme di linee guida, consigli e regole per la creazione e progettazione di pagine web che siano accessibili da qualsiasi tipologia di utente.

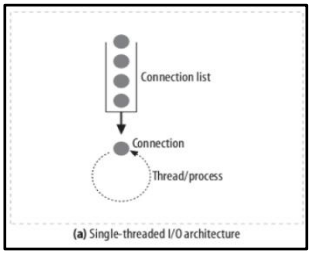
Tor network anonymization: un sistema in grado di garantire protezione della privacy su internet deviando delle comunicazioni attraverso una rete distribuita di relay tramite un sistema basato su web proxies.

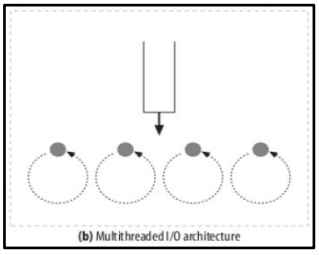
Web Server

Un Web server (o HTTP server) è un server che risponde alle richieste di risorse inviate attraverso il protocollo HTTP. Un client richiede una risorsa inviando un comando HTTP e un URL che identifica univocamente la risorsa nel WWW. Il web server risponde inviando un documento memorizzato staticamente sul file system oppure generato dinamicamente, usando la richiesta come input per un programma e inviandone l'output al client. Mentre le operazioni che un web server è in grado di gestire sono:

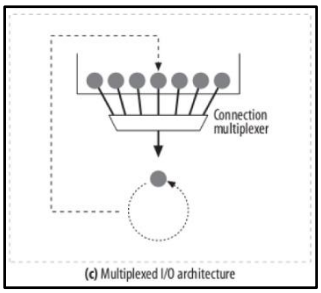
- **Gestione di una richiesta client:** processo che si divide in diverse fasi: parsing della richiesta, Check delle autorizzazioni, associazione URL con un filename, costruzione della risorsa, trasmissione della risposta.
- **Controllo dell'accesso:** processo per il quale si effettua l'identificazione dell'utente che origina la richiesta. Di solito tale autorizzazione è basata su una coppia nome/password. Inoltre, anche se il protocollo HTTP non offre alcun supporto alle sessioni l'utente ha comunque un'illusione di quest'ultima, perché il client continua ad inviare la coppia username/password ad ogni richiesta. Tale processo di autenticazione è detto Challenge/response. Per determinare le politiche di accesso ad una determinata risorsa per un utente, si utilizzano delle **access control list**.
- **Risposte dinamiche e gestione dello stato:** un web server oltre a fornire risorse statiche, ovvero già presenti all'interno del file system è anche in grado di generare risorse dinamiche, tramite **script** o **server-side-includes.** Inoltre, i web server si occupano anche della gestione dello stato, tramite l'utilizzo di **cookies** che vengono settati dal server ma mantenuti in locale sul client, il quale li invia ad ogni nuova richiesta.
- **Caching:** tecnica che consiste nel mantenere certe risorse, particolarmente popolari, che vengono caricate in memoria per velocizzarne l'accesso. È inoltre necessario implementare dei meccanismi di check della data, per evitare di fornire delle informazioni all'interno delle risorse ormai obsolete.

Dal punto di vista puramente architetturale invece un web server può essere **single-threaded** o **multi-threaded.** La differenza principale fra le due architetture sta nel numero di richieste che vengono gestite contemporaneamente da parte del server.



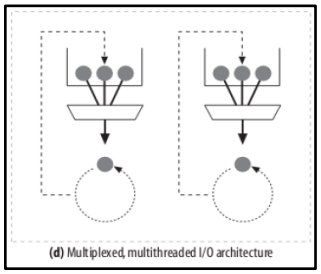


Architettura Mutliplexed multi-threaded: un tipo di architettura all'interno della quale tutte le connessioni sono pronte ad essere eseguite. Quando una connessione cambia stato una parte del processing viene eseguita sulla connessione, quanto il processing è terminato la connessione viene restituita alla open connection list per il prossimo cambiamento di stato. Questo permette di evitare che threads e processi restino in attesa su una connessione in idle.



Architettura Multiplexed multi-threaded: alcuni sistemi combinano multithreading e multiplexing per avvantaggiarsi della presenza di CPU multiple su un computer. Threads multipli (uno per ogni processore fisico) possono vedere le connessioni aperte (o un sottoinsieme delle connessioni aperte) ed eseguire una piccola quantità di lavoro su ogni connessione.

Alcuni esempi di server popolari sono: Apache, Nginx e Tomcat.



Benchmarking: definizione ed esempi

Con il termine **benchmarking** si intende quel processo continuo di misurazione di prodotti, servizi e prassi aziendali, mediante il confronto con i concorrenti più forti di uno specifico settore. Più nello specifico tale processo serve a misurare e incrementare le performance di un'impresa, di una pubblica amministrazione o di un prodotto sia hardware che software. La comparazione è effettuata proprio sulla base dei **benchmark**, ossia di prestazioni di riferimento rispetto alle quali effettuare la comparazione e verso cui tendere nell'attuare le azioni di cambiamento. Per compiere del buon **benchmarking** sono necessari i seguenti passi:

- Individuare le migliori aziende/concorrenti nel settore in cui si opera.
- Confrontare i propri risultati con quelli delle migliori aziende/concorrenti.
- Imparare dal questo confronto.
- Implementare ciò che si è imparato.
- Ripetere il benchmarking.

Le dimensioni misurate sono di solito relativa a: **qualità**, **tempi** e **costi**. Esistono inoltre diverse tipologie di benchmarking che variano a seconda del settore economico o industriale a cui si riferiscono.

Performance benchmarking:

Questo tipo di benchmarking ha lo scopo di analizzare le prestazioni di un sistema con l'obiettivo di acquisire i punti di forza delle migliori aziende in termini di componenti software o scelte tecnologiche. Il confronto fra diversi sistemi richiede di applicare un workload comune e riproducibile a ciascun prodotto (i famosi benchmark delle schede video). Il workload deve essere scelto in modo tale che i risultati siano rappresentativi di come il prodotto si comporta nella pratica. Per l'analisi delle prestazioni di sistemi web, per esempio, le principali metriche di confronto sono:

- **Page load time;**
- **Application availability;**
- **Web Page size and content;**
- **Third-part analysis;**
- **User engagement/transactions;**

Benchmarking Tools: all'interno di un contesto web i tool per il benchmarking si dividono in due categorie principali:

- **Benchmark lato client:** LoadNinja, JMeter, Webload, Loadview.
- **Benchmark lato server:** Httperf, ApacheBench, Curl-loader, FunkLoad, Siege.

Web server benchmarking:

Tale analisi si svolge principalmente misurando la capacità di un web server di servire adeguatamente carichi di lavoro adeguati i parametri vengono pertanto misurati in termini di: **numero di richieste concorrenti servite** (al secondo), **tempo di risposta per ogni nuova connessione o richiesta** (millisecondi) e **throughput** (byte per secondo). Le misurazioni, inoltre, devono considerare un numero variabile di client e di richieste per client. Per quanto riguarda invece le procedure da seguire è buona norma:

- Usare la stessa configurazione hardware e kernel per tutti i test.
- Usare la stessa configurazione di rete per tutti i test.
- Eseguire il test.
- Eseguire il reboot del server dopo ogni test.

Infine, prima di avviare la fase di benchmarking vera e propria è necessario, definire un **workload**, generalmente costituito da: **file HTML** di piccole e grandi dimensioni e **script** CGI e Perl.

PERFORMANCE TUNING DI APACHE DA VEDERE SULLE SLIDE (ma abbastanza inutile).

HTTPERF

È un tool di benchmark sviluppato dl HP Research Labs per misurare le performa di Web server e proxy. Fornisce funzionalità per la generazione di HTTP workload e si pone come obbiettivo quello di fornire un tool robusto ed altamente performante per facilitare la realizzazione di micro/macro-level benchmark. Le tre caratteristiche salienti di HTTPERF sono: **robustezza**, **supporto al HTTP/1.1** ed **estendibilità** (utile per generare nuovi workload ed ottenere ulteriori misure di performance). Il tool è composto da tre componenti:

- **Core HTTP Engine**: che gestisce le comunicazioni con il server occupandosi delle connessioni, delle richieste HTTP o della gestione delle risposte.
- **Workload Generator:** responsabile della generazione di stream di richieste HTTP a tempi prestabiliti in modo da caricare il server con particolari workload.
- **Statistics Collector:** che misura vari parametri e produce statistiche sulle performance.

Le informazioni che HTTPERF può dare riguardo a un server sono molteplici: **carico massimo di richieste sostenibile**, **comportamento del server con differenti workload** e **tempo di risposta per determinati workload.** Riporta inoltre i seguenti risultati: Total, Connection, Request, Reply, CPU Usage, Errors, Session.

Internet Measurement & Internet Architecture

Internet è una rete di comunicazione versatile che connette un incredibile varietà di dispositivi client e viene utilizzata per un range di attività molto variegate fra di loro. Come si è visto i building block che compongono internet sono ben noti e ampiamente analizzati e studiati, mentre Internet come sistema globale invece non è mai stato analizzato, questo perché Internet è un **sistema decentralizzato**, ed è stata costruita da un insieme di grandi organizzazioni con differenti scopi. Inoltre, la sua natura è **dinamica** ovvero cambia con estrema rapidità in ogni sua caratteristica, per esempio di dispositivi che si connettono o disconnettono ogni secondo sono milioni. È pertanto impossibile avere un quadro generale dello stato di Internet, l'unica cosa che si può ottenere è una stima in un determinato istante della rete. Tra i principali ostacoli a misurare le proprietà di internet troviamo:

- Proprietà che restano **nascoste** a causa dell'architettura di internet stessa.
- I dati raccolti sono difficili da **memorizza**, **trasferire**, **processare** e **analizzare**.
- I **service provider** spesso non forniscono i dettagli delle reti interne.
- Alcune forme di Internet Measurement possono violare la **privacy**

Ma perché si dovrebbe voler misurare Internet? Principalmente per ragioni legate all'**ottimizzazione**, alla **rilevazione di problemi** e di **attacchi.** Più nello specifico queste ragioni possono essere:

- **Commerciali:** vendere un prodotto o fornire informazioni relative ad un prodotto richiede una varietà di misurazioni.

- **Sociali:** con lo scopo di avere info circa la caratterizzazione della popolarità e di contenuti o banalmente del numero di utenti connessi in una determinata aurea.
- **Tecniche:** se non si conoscono le condizioni in cui componenti hardware e software andranno ad operare una volta connessi o integrati all'interno di internet è impossibile progettarli.

Per misurare Internet correttamente è necessario conoscerne l'architettura.

Architettura di Internet:

L'organizzazione di Internet si basa principalmente sua una gerarchia di ISP (Internet Service Provider), più nello specifico distinguiamo tra:

- **ISP di primo livello:** Nazionali o internazionali (Sprint, Telecom, AT&T, UUNet, ecc.) noti anche come reti dorsali di Internet. (da 10 Gbps a 100 Gbps, le più recenti anche Tbps).
- **ISP di secondo livello:** hanno copertura nazionale o regionale e si connettono solo ad alcuni ISP di primo livello per instradare il traffico verso altre reti.
- **ISP di terzo livello o locali:** istituzionali, aziendali.

I **Tier-1 ISP** di solito hanno un piccolo numero di peer. I **Tier-2 ISP** sono connessi ad uno o più **Tier-1 ISP**, possibilmente anche ad altri **Tier-2 ISP** e pagano i **Tier-1 ISP** per la connettività verso il resto di internet. Con il passare degli anni però l'evoluzione di Internet verso scopi commerciali ha decentralizzato la rete, eliminando il singolo top-level ISP facendo in modo da non poter più raccogliere statistiche sul traffico. Infine, per quanto riguarda lo spostamento delle informazioni all'interno della rete I dati si muovono tramite pacchetti (Packet switching) e ognuno di essi contiene le info necessarie per raggiungere la propria destinazione. Più nello specifico questi pacchetti attraversano un'intricata rete di routers e links e che utilizzano protocolli per la loro gestione come per esempio il TCP/IP.

Principi base:

Internet è un'interconnessione di rati appartenenti a diversi proprietari. Non può infatti essere definita un'unica rete, in quanto non esiste un singolo proprietario o organizzazione responsabile della sua costruzione e configurazione, per aggiungere un nuovo host alla rete infatti è estremamente semplice, basta chiedere l'autorizzazione a un ISP locale, questo ha premesso di accelerare molto la sua crescita.

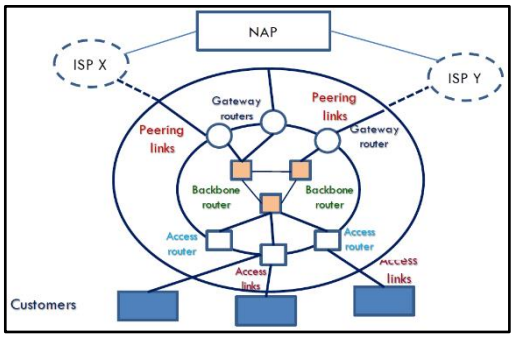
I protocolli utilizzati sono diversi per ogni livello del modello ISO/OSI. Uno solo però è il protocollo che viene usato al livello network quello **IP**, si tratta di un protocollo **stateless switching**, ovvero nessuna informazione viene mantenuta dai router circa le connessioni che passano attraverso essi (il router analizza solo l'header del pacchetto IP per determinare dove instradarlo).

Dettagli sulle operazioni Internet:

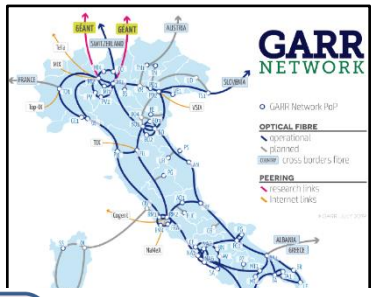
Le componenti su cui Internet si basa sono principalmente tre:

- **Endsystem:** che possono svolgere sia il ruolo di client, server o peers.
- **Router:** intermediari che si occupano di istradare i pacchetti attraverso la rete. I dati vengono trasmessi attraverso un link logico chiamato "**hop**", che viene selezionato tramite le informazioni contenuto nel header del pacchetto.
- **Links:** connessioni fisiche o wireless su cui viaggiano effettivamente i pacchetti.

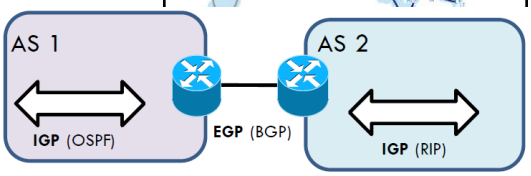
Una delle principali sfide organizzative di Internet è quella di permettere ad un insieme di operatori (organizzazioni no-profit e pro-profit) di rete eterogenei di cooperare per assicurare la connettività su larga scala (consentendo allo stesso tempo un'organizzazione indipendente). Queste organizzazioni indipendenti prendono il nome di **autonomous systems**, che non sono nient'altro che collezioni di elementi di rete sotto il controllo di un'unica organizzazione, ogni AS è identificato tramite un **AS Number** e vengono identificati in base alla scala in: **AS-Grandi** (decine di migliaia di Endsystem e router) e **AS Piccoli** (uno o due routers). Lo scambio di informazioni tra i vari punti della rete viene assicurato tramite i cosiddetti **peering points**, connessioni formate connettendo routers di diversi AS, che vengono chiamati **gateway routers**.



Ogni **AS** ha inoltre una singola e ben definita politica di **routing**. Un esempio è il **GARR** la rete della comunità scientifica ed accademica italiana. La dorsale della rete è costituita da circuiti basati su tecnologie ottiche di trasporto che permettono di raggiungere velocità di **10Gbit/s**. Interconnettendo oltre 40 **Pop** (Point of Presente) e collabora con le principali organizzazioni che operano nel campo del networking.



Routing:

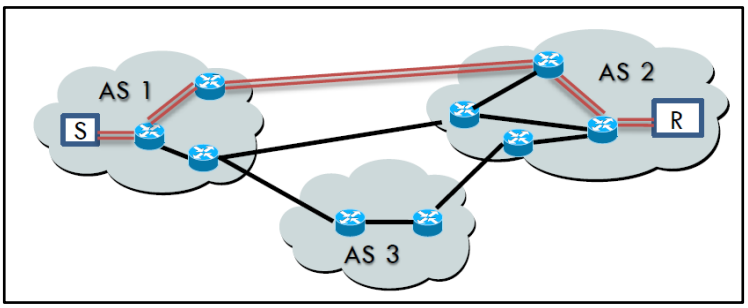


Un pacchetto inserito nella rete da un endsystem, viene inoltrato alla destinazione come specificato dall'indirizzo IP di destinazione del pacchetto tramite un processo che prende il nome di routing, più nello specifico questo processo viene svolto da dei dispositivi chiamati per l'appunto **router** e si divide in due fasi: una prima fase chiamata **routing** in cui il router sceglie su quale interfaccia di rete instradare il pacchetto appena ricevuto, per poi effettuare il **forwarding** e inviarlo effettivamente. L'intero processo all'interno di Internet è distribuito, ovvero non esiste un singolo router che si occupa di gestire tutti i pacchetti che viaggiano per la rete, questo perché la rete è così grande che sarebbe impossibile mantenere un database singolo e aggiornato (aggiunta di nodi o fallimenti) riguardo tutti i suoi nodi. Inoltre, tale sistema segue una gerarchia ben precisa, basata su due livelli: **intra-domain routing** e **inter-domain routing**. Differenti **AS** possono usare diversi **routing protocols**. L'essere gerarchico è dovuto al fatto che Internet è una rete troppo grande per un singolo database di routing, in questo modo un cambio di stato nella rete coinvolge solo un numero limitato di router, inoltre la composizione gerarchica permette ai vari AS di gestire le proprie reti con una certa libertà. Più nello specifico il **routing** all'interno di **intra-domain** avviene tramite l'uso di due protocolli (per entrambi l'obbiettivo è quello di ottenere lo Shortest Path):

- **IS-IS.**
- **OSPF (Open Shortest Path First):** ogni router mantiene una mappa della rete, che viene aggiornata ogni volta che un link viene aggiunto o rimosso (crash di un router), poiché ogni router mantiene una copia della **network map**, ogni router è in grado di calcolare il "best path" e quindi dove è meglio instradare i pacchetti che riceve, per assicurare che ogni router abbia una mappa aggiornata, ognuno monitora continuamente i link a cui è connesso. Quando viene rilevato un cambiamento viene avviato un protocollo di **flooding** per avvisare tutti gli altri router. È inoltre possibile da parte degli amministratori di rete definire dei pesi per determinati path. Mentre se la destinazione non si trova all'interno del Intra-domain si ricorre all'uso di gateway per instradarlo all'esterno tramite il protocollo **BGP**.

Protocolli per il routing Inter-domain:

- **BGP (Border Gateway Protocol):** ha il compito di fornire informazioni sulla raggiungibilità delle diverse reti tra più AS anche detto Advertising di una rete. Tutto questo viene realizzato attraverso lo scambio di **messaggi BGP** tra i gateway esterni ai vari AS, questo con lo scopo di specificare politiche di instradamento in base a diversi parametri (economici, sicurezza e gestionali). Più nello specifico il protocollo si basa sullo scambio di **path vector** ovvero un **network ID** oppure una sequenza di **network ID** che costituiscono un path verso la rete. In sostanza quando un **AS 1** riceve un path da un altro **AS 2**, aggiunge il proprio AS number all'inizi del path quindi comunica il nuovo path vector ad alcuni o tutti i suoi vicini. Se l'AS 1 trova già il suo AS number all'interno del path non deve aggiungere niente. Se più Path Vector vengono ricevuti si sceglie quello con lo Shortest Path.



ESEMPIO DI BGP SULLE SLIDE

Protocollo ICMP:

Primo protocollo per il network monitoring e management mai realizzato. Prevede 11 tipi di messaggi ognuno dei quali inviato tramite un singolo pacchetto IP. Questo protocollo viene utilizzato all'interno dell'implementazione del comando **Ping** per verificare la connettività a livello di rete tra due host, o all'interno del comando **Traceroute** che viene utilizzato per identificare il percorso seguito da un pacchetto per raggiungere una certa destinazione.

Protocollo SNMP:

Utilizzato principalmente per il performance measurement, debugging e capacity planning. Utilizza il protocollo UDP come protocollo di trasporto. Più nello specifico un device, per essere gestito da SNMP, deve eseguire un **SNMP management process** chiamato **SNMP agent**. Le informazioni raccolte vengono immagazzinate in un formato chiamato **Management Information Base (MIB)** che includono informazioni relative a stato di un device, attività del device e workload.

Analytic Background

Funzione di ripartizione F di una variabile aleatoria X: probabilità che la variabile aleatoria X assuma un valore minore o uguale di x.

Variabile casuale discreta: una variabile random che può assumere un numero finito o al più numerabile di possibili valori.

Variabile casuale continua: una variabile random che può assumere un numero non finito di valori è detta continua. Definita come un intervallo di valori, rappresentato come un'area sotto una curva (integrale).

Varianza di una variabile aleatoria X: funzione che fornisce una misura della variabilità dei valori assunti dalla variabile stessa; misura di quanto essi si discostino quadraticamente rispettivamente dalla media aritmetica

Deviazione standard: Se la deviazione standard (σ) è grande, i valori della distribuzione sono dispersi. Viceversa, se la deviazione standard è piccola, i valori sono concentrati vicino alla media.

Covarianza: di due variabili random X e Y, è un numero che fornisce una misura di quanto le due varino assieme, ovvero della loro dipendenza.

Special Issues In The Internet

Processi stocastici: Un processo stocastico è pertanto una **collezione** o **sequenza** di variabili random indicizzate su un insieme (tale indice in generale denota il tempo). Da un *punto di vista pratico*, un processo stocastico è una forma di rappresentazione di una grandezza che *varia nel tempo* in modo **casuale** e con certe caratteristiche ad esempio un segnale elettrico, il numero di autovetture che transitano su un ponte, ecc.

Arrivals: eventi che occorrono in un determinato istante di tempo.

Arrival process: un processo stocastico in cui variabili random successive corrispondono agli istanti di tempo degli arrivals.

Time series of counts: fissato un intervallo di tempo si contano il numero di arrivals in questo intervallo. L'intervallo di tempo T prende il nome di **timescale**, esso rappresenta la forma più comune in cui viene riportato il traffico Internet.

Risulta importante studiare una parte specifica delle distribuzioni:
- La coda (**tail**) di una distribuzione
- Il modo in cui la coda arriva a 0

Le distribuzioni più comuni hanno code che **diminuiscono esponenzialmente** (molto velocemente). Al contrario, una distribuzione la cui coda diminuisce più lentamente è detta **subexponential** distribution, i campioni da questa distribuzione mostrano osservazioni grandi con una frequenza non trascurabile, questo tipo di distribuzione è detta avere una long **tail**. Mentre un caso speciale è la **heavy-tailed distribution**, una variabile random che segue questa distribuzione mostrerà una alta variabilità. Confronto tra una **short-tailed** (linea continua) e una **long-tailed distribution** (linea tratteggiata).

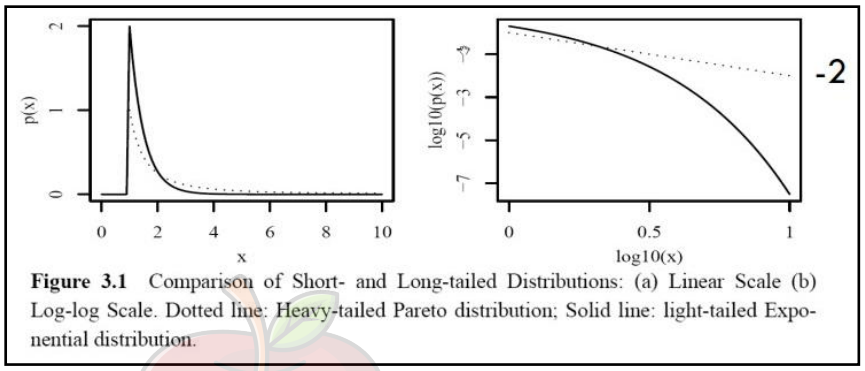


Figure 3.1 Comparison of Short- and Long-tailed Distributions: (a) Linear Scale (b) Log-log Scale. Dotted line: Heavy-tailed Pareto distribution; Solid line: light-tailed Exponential distribution.

I dati da analizzare possono variare di diversi ordini di grandezza:

- Alcune osservazioni assumono valori bassi
- Molte osservazioni assumono valori grandi
- Altri valori molto grandi.

Quando il range dei dati varia su tre ordini di grandezza la distribuzione è detta **highly skewed**. Diversi piccoli valori mischiati a pochi valori grandi anche se le quest'ultimi sono pochi, essi sono in grado di dominare le statistiche empiriche del dataset, cioè media e varianza.

Dunque, media e varianza **non** adatte in caso di alta variabilità nei dati.

Ad esempio, determinare la proporzione di richieste che usano il metodo GET e la proporzione di richieste il cui response code è 200 OK si traduce nel contare il numero totale di trasferimenti effettuati. Quando i parametri variano molto, la media può ingannare dal momento che può essere deviata da un piccolo numero di valori abbastanza grandi. Calcolare sia la media che la mediana può dare una migliore rappresentazione dei dati, la mediana più piccola della media suggerisce la presenza di un piccolo numero di risposte con taglia grande, le **distribuzioni di probabilità** sono in grado di mostrare **meglio** come i parametri possono variare in un range molto alto di valori.

Power law: Il power-law implica che i piccoli eventi sono estremamente comuni, mentre i casi di grandi dimensioni sono estremamente rari. La distribuzione power law può essere utilizzata per descrivere un fenomeno in cui un piccolo numero di elementi è raggruppato nella parte superiore di una distribuzione (o nella parte inferiore), occupando il 95% delle risorse. *In altre parole, implica una piccola quantità di eventi è comune, mentre eventi più grandi sono rari*.

Ad esempio, dove è in gioco la distribuzione del reddito, ci sono pochissimi miliardari, ogni quattro individui con reddito annuo pari a diecimila euro, ne esiste uno con reddito pari a ventimila (distribuzione della ricchezza). Altri

esempi di fenomeni che seguono questa legge: frequenza delle parole in un linguaggio, frequenza dei nomi, taglia dei terremoti ecc.

Una funzione power law è una relazione polinomiale che esibisce la proprietà di **scale invariance,** caratteristica di oggetti o leggi che non cambiano se la dimensione delle scale di osservazione sono moltiplicate di un fattore costante.

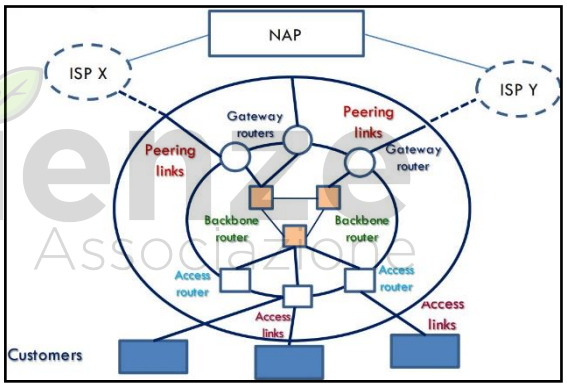
Pereto distribution: Una distribuzione power-law in grado di modellare fenomeni sociali, scientifici, geografici, ecc. Dovuta all'economista italiano Vilfredo Pareto. Il principio: "regola 80-20 - il 20% della popolazione possiede l'80% della ricchezza". La "probabilità" o frazione della popolazione f(x) che possiede una piccola quantità di ricchezza per persona (x) è piuttosto alta, e stabilmente decresce con l'aumentare della ricchezza. Pareto è applicabile per calcolare la dimensione degli insediamenti umani (poche grandi città, molti agglomerati), distribuzione della taglia dei files nel campo dell'Internet Traffic (molti files piccoli, pochi di grosse dimensioni), distribuzione dei job assegnati ai supercomputers, dimensione delle particelle di sabbia, ecc.

Pratical Issues in Internet Measurement

Dove possono essere fatte le misurazioni

Le misurazioni possono essere effettuate in un qualunque punto dello spettro di internet:

- vicino ai routers
- sui link che connettono i routers
- sulle LAN,
- sugli entry points della rete,
- all'interno di network backbones,
- WAN.



In blu le **locazioni** in cui possono essere effettuate le misurazioni internamente ed esternamente agli ISPs. I **backbones routers**, **access routers** e **gateway routers** comunicano usando un intra-AS protocol. I **gateway** sono responsabili per il routing del traffico tra un ISP e altri ASes.

I **gateway routers** scambiano routing tables molto grandi e comunicano usando un inter-AS routing protocol (BGP), sono anche detti peering routers.

Gli **access routers** forniscono connettività ad un insieme di customer networks su access links.

Gli **access links** variano in capacità sulla base del tipo di ISP e di customer needs.

Un **Network Access Point (NAP)** rappresenta un punto di scambio di ISP multipli che possono scambiare traffico.

LAN: eseguite solo per testbed locali e non hanno nessun interesse significativo nel contesto delle misurazioni internet. Performance measurement degli strati più alti sono normalmente eseguite su LAN, qui possono essere effettuate misurazioni precise.

INSIDE A BACKBONE: le misurazioni sono di norma effettuate all'interno di backbones. Gli ISPs monitorano la rete per i seguenti motivi:

- assicurare disponibilità,
- scanning per eventuali attacchi,
- cambio della topologia,
- trend nel traffico.

Un obiettivo chiave delle misurazioni sui backbone è il **capacity planning,** cioè verificare se ulteriori **PoP** [Punto di acceso alla rete fornito da un ISP] sono necessari o se deve essere aumentata la capacità di quelli esistenti.

La tecnica più semplice per raccogliere dati del traffico è usare il meccanismo di **polling di SNMP** per raccogliere le caratteristiche del traffico: dati SNMP forniscono info per misurare **packet loss**, delay, thoughput e questi possono essere raccolti in ogni punto della rete.

Un'altra tecnica è il **packet tracing** secondo cui vengono forniti **timestamps** ad alta precisione e viene fatto un monitoring periodico del **backbone traffic** su piccoli **timescale** per identificare potenziali burst di traffico. Può essere realizzata con piccoli packet traces raccolti su high speed links all'interno di backbones.

Un'attività essenziale eseguita dagli ISP è l'allocazione della banda ai vari link all'interno degli ISP. Il traffico all'interno di una rete deve essere misurato per un corretto "provisioning della banda".

Le misurazioni all'interno di backbone forniscono una vista precisa di tutto il traffico associato ad un insieme di customers. Inoltre, è importante assicurare che i **SLAs**, service level agrements firmati e contrattati dall'ISP con i vari customers, vengano rispettati. Un altro vantaggio riguarda il monitoring della rete per eventuali attacchi. Una tecnica consiste nel monitorare periodicamente l'utilizzo dei link e verificare la presenza di incrementi sostanziali.

ENTRY POINTS INTO A NETWORK: rappresentano il luogo migliore per filtrare unwanted traffic. Esportano info circa flussi di traffico (attraverso il tool **netflow**). I dati raccolti sono sufficienti per ottenere info sul traffico destinato ai customer all'interno della rete e info sulle porzioni di traffico che transita attraverso la rete.

I router che utilizzano il protocollo BGP per scambiare info di routing possono essere pubblici e privati. Lo scopo delle misurazioni a questo livello può essere da un punto di vista economico, per assicurare uno scambio.

Access routers: connettono i backbones ad un insieme di customers. Le misurazioni sono realizzate per assicurare che il tasso di fallimento sia da basso o non esistente. Anche qui ci sono degli SLA da rispettare. Alcuni customers possono richiedere che il packet filtering sia eseguito per assicurare che solo pacchetti da una data sorgente e pacchetti per un dato insieme di destinazioni siano instradati.

WAN: Multi-site measurement. Alcune applicazioni richiedono che le misurazioni vengano effettuate **simultaneamente** da locazioni multiple attraverso la rete. Il multi-site measurement eseguito in modo coordinato richiede una Clock synchronization, una Execution serialization e diversi meccanismi per controllare accessi e risorse. Diverse piattaforme eseguono misurazioni simultanee da differenti locazioni (NIMI,PlanetLab).

Ruolo del tempo

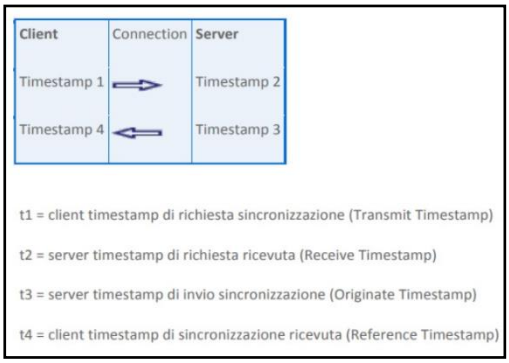
Diversi measurement tasks richiedono accurate misurazioni del tempo. Ad esempio:

- Misurare il RTT di pacchetti
- Misurare il ritardo subito dai pacchetti che attraversano routers e links
- Produrre una vista di misurazioni raccolte in differenti locazioni della rete
- Misurare le performance delle componenti internet (response time e throughput)

Internet è un sistema distribuito dove le varie componenti possono trovarsi su grandi distanze, ottenere informazioni di tempo accurate è un compito complesso, anche in presenza di **clock precisi**, le distanze tra le componenti induce latenza e quindi rende "vecchie" le informazioni relative al tempo quando esse arrivano. Esiste un certo numero di classi di sorgenti di tempo tipicamente usate nel campo delle misurazioni Internet:

- Sorgenti di tempo **esterne**: radio clocks per disseminare informazioni sul tempo, forniscono una precisione nell'ordine di millisecondi;
- Global Positioning System (**GPS**): catturando informazioni da più satelliti è capace di fornire una precisione più alta (range da poche centinaia di nanosecondi a microsecondi) svantaggio sono le grosse antenne;
- PC-based **clock** (Hardware clock, Software clock).

Importanza del tempo nel **One-way packet delay measurement**, si richiede che venga misurato il tempo in cui il pacchetto parte in una locazione e l'arrivo del pacchetto in un altro punto (la destinazione), per far ciò bisogna usare clock sincronizzati. Un server si **sincronizza** confrontando il suo orologio con quello di diversi altri server di strato superiore o dello stesso strato, questo permette di aumentare la precisione e di eliminare eventuali server scorretti.
A specifici intervalli un client invia una richiesta ad un server e ottiene una risposta, la sincronizzazione tra Client e Server funziona in questo modo:



- Il client manda un messaggio **NTP** al server e marca il momento dell'invio (**Transmit Timestamp**);
- Il server riceve il pacchetto e marca il momento della ricezione (**Receive Timestamp**);
- Il server analizza il pacchetto e rimanda indietro il messaggio **NTP** al client e marca il momento dell'invio (**Originate Timestamp**);
- Il client riceve il pacchetto e marca il momento della ricezione (**Reference Timestamp**) poi analizza i 4 timestamps con il client **NTP** per determinare la correzione da apportare al proprio orologio interno.

Da questi quattro timestamps il client può calcolare due valori: il ritardo (**Delay**), il tempo che è stato necessario per trasferire i pacchetti sulla rete, e la deviazione (**Offset**), la differenza di tempo tra gli orologi interni dei due computers.

Ruolo di Internet directories e database

La maggior parte delle misurazioni su internet richiede la traduzione di nomi in indirizzi e viceversa. Il problema dell'accesso ai database riguarda il fatto che le info potrebbero essere vecchie (le info possono trovarsi in cache in varie locazioni e tali info potrebbero essere vecchie). Le info, inoltre, potrebbero essere incomplete o accessibili solo ad utenti autorizzati presenti in access-control lists.

Misurazioni attraverso i protocolli di rete

Il meccanismo di data gathering varia a seconda del layer del protocol stack. Tre categorie:

- **Lower- level protocol data** (router e link-level data**)**
- **packet trace e flow-level data**
- **application-level data** (raccolti a livello applicativo attraverso una serie di tools).

Lower-level: La problematica principale al primo level **(router level)** riguarda la quantità di dati che devono essere esaminati. Una possibile tecnica è quella del **polling** e raccolta periodica dei dati (**SNMP** si potrebbe utilizzare). L'aggregazione realizzata raggruppando insiemi di comunicazioni in flussi e pubblicando periodicamente un record contenente le informazioni chiave sul traffico scambiato (indirizzi IP coinvolti, numero pacchetti, ecc).

Flow-level: il volume dei dati è ancora grande ed il processing in real time sempre complicato. Oltre ai **packet traces** i **router** possono raccogliere informazioni circa flussi di pacchetti tra due **endpoints**. Un flusso unidirezionale è definito da:

- Indirizzo IP sorgente e numero porta
- Indirizzo IP destinazione e numero porta
- Protocollo
- Interfaccia sorgente del router
- Tipo di servizio

Il router raccoglie meta-informazioni in termini di: numero pacchetti e bytes e periodicamente esporta record di flussi. Mantenere traccia di questo volume di dati può sovraccaricare un router. Una soluzione è il **sampling** (anziché catturare ogni pacchetto, si tiene traccia di un pacchetto ogni n pacchetti).

Il tool **CISCO netflow** definisce record di 30 campi. I campi più usati sono: indirizzo IP sorgente e destinazione e numero di porta, protocollo, tempo di inizio e fine del flusso, ASes. I **record CISCO netflow** sono scartati quando: i due endpoints hanno un periodo di inattività di 30 secondi (inactive timeout configurable), l'endpoint remoto ha chiuso la

connessione, il flusso ha ecceduto un lungo periodo di attività (30 minuti), il router ha necessità di pulire la memory cache.

Application-level: La raccolta di dati a livello applicativo è più semplice che a basso livello (tecnica più utilizzata è il **logging**). I dati possono essere catturati lato client, server o proxy.

Infrastructure

Proprietà dei dispositivi fisici di internet:

Links: struttura fisica sulla quale un pacchetto viene instradato da un hop ad un altro. Ogni link ha due proprietà che lo caratterizzano: **ritardo di propagazione** e **capacità**. Mentre le proprietà legate alle performance di un link sono:

- **Packet delay;**
- **Packet loss;**
- **Jitter;**

Routers: struttura fisica che si occupa di analizzare ogni pacchetto entrante e selezionare un'interfaccia uscente sulla base dell'indirizzo di destinazione del pacchetto e della forwarding table. Siccome i pacchetti possono arrivare ad una velocità maggiore rispetto a quella con cui i pacchetti vengono smistati in uscite, si ricorre all'utilizzo di **buffer** interni per mantenere i pacchetti all'interno di code **FIFO**. Se il buffer si riempie, ulteriori pacchetti verranno scartati (**Drop-tail service**). Alcuni routers possono essere configurati per eseguire un **active queue management** con lo scopo di definire quali pacchetti scartare. Le proprietà di interesse relativi ai router possono essere divise in due categorie:

- **Proprietà statistiche:**
  - Insieme di indirizzi IP usati;
  - Locazione geografica dei router;
  - Tipo di router o protocol supportate;
- **Proprietà dinamiche:**
  - Tempo che il router impiega per rispondere ad un messaggio ICMP e quello che impiega per instradare un pacchetto.

Wireless: il canale wireless più comunemente utilizzato è quello basato su frequenze radio. Lo scopo principale è quello di collegare gli utenti a infrastrutture wired (in cui le onde radio viaggiano da **wireless adapter** ad un **wireless AP** connesso ad Internet usando cavi). La scelta della tecnologia wireless detta le proprietà di quest'ultima:

- **Distanza a cui è possibile ricevere segnali;**
- **Data rate;**
- **Affidabilità;**
- **Potenziali Interferenze;**
- **Numero di utenti concorrenti;**

Esistono diverse tecnologie per il wireless networking ma la scelta principale ricade nella famiglia **802.x** standard istituiti dall'IEEE. **802.11a, 802.11b** e **802.11g** sono le 3 versioni dello standard che variano in **throughput**, **data rate**, **numero di canali**.

Un'altra possibile scelta è quella del **Bluetooth** che offre connettività a distanze più brevi rientrando nella categoria del **Personal Area Networking PAN** e mette in comunicazioni differenti device attraverso frequenze radio nel raggio di una decina di metri.

Un'altra tecnologia ancora è il **WiMAX**, tecnologia wireless che rientra nello standard **802.16 Wireless Metropolitan Area Network**. Infine, i **5G cellular systems**.

Le misurazioni riguardanti tecnologie wireless devono considerare diversi aspetti: **potenza del segnale**, **quantità di potenza consumata**, **data bit rate**, **grado copertura**, **informazioni sulle sessioni**, **error rates**. Altre metriche includono

la **capacità dei link**, **banda effettiva e disponibile**, **identificazione dei colli di bottiglia**. Inoltre, le misurazioni risultano complicate a causa della combinazione di wired e wireless network con cross-traffic.

Topologia:

AS graph: Internet è un insieme di reti indipendenti organizzate in ASes. Un AS consiste di una singola rete o più reti. Una singola organizzazione può operare su uno o più ASes. Il routing all'interno di un AS è gestito da un domain administrator ed usa il protocollo **BGP** tra ASes. L'insieme delle interconnessioni fra gli ASes formano un grafo (**AS graph**).

Router graph: all'interno di un AS i router sono organizzati in un insieme di locazioni fisiche chiamate **Point of Presence** (**PoPs**). Una vista più dettagliata della topologia si ha con il router graph (i router rappresentano i vertici, gli archi **one-hop** connections tra routers).

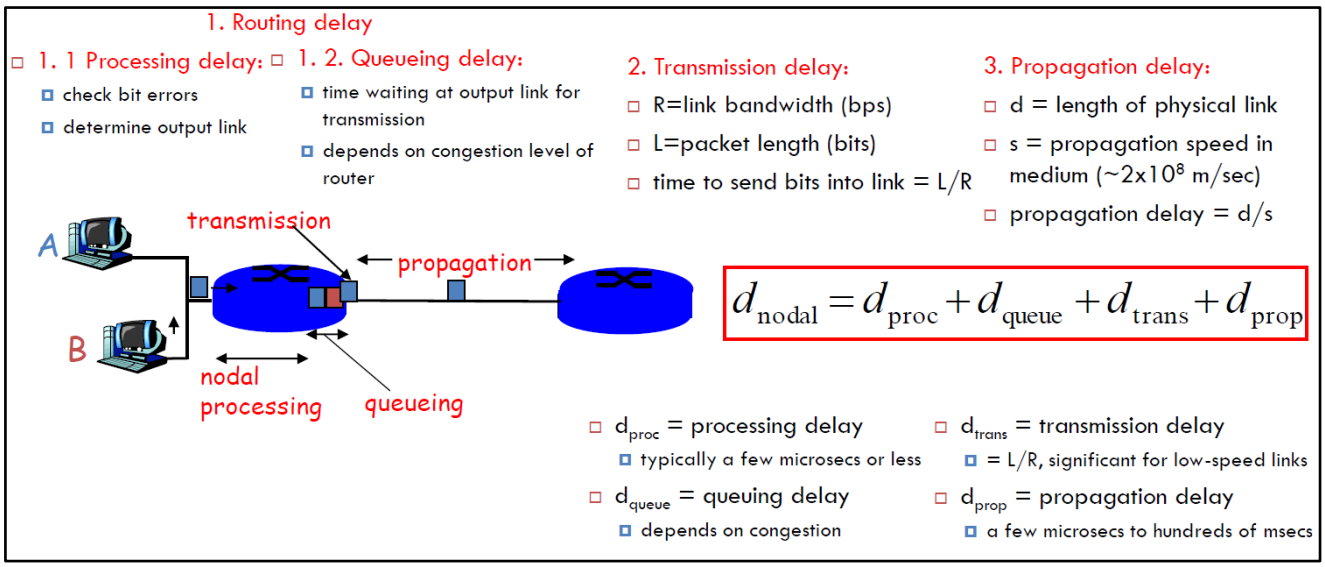
Interface graph: Un'ultima vista è data dall'**interface graph** dove i vertici sono le interfacce dei router, gli archi **one-hop connection**. Questo grafo viene usato dal tool **traceroute**.

Traffico:

Le principali proprietà legate al traffico sono:

Packet delay: che rappresenta il ritardo a cui un pacchetto può essere soggetto. Più nello specifico questo valore è dato dalla somma di varie informazioni:

- **Routing Delay:** che rappresenta il tempo speso all'interno del router. Più nel dettaglio questo tempo che il pacchetto passa all'interno del router si divide in tre parti:
  - **Packet processing delay:** tempo necessario al **lookup** dell'indirizzo di destinazione del pacchetto in arrivo e del **forwarding table** per determinare la porta di output appropriata.
  - **Queuing delay:** tempo speso in coda all'interno del router.
  - **Additional delay:** possibili comportamenti sbagliati (come il fallimento di un instradamento). Altri ritardi non dovuti né al processing né all'accodamento.
- **Transmission Delay:** tempo necessario per posizionare un pacchetto su un link. Se un pacchetto ha taglia **S** ed il link ha capacità **T** allora il transmission delay è **S/T**.
- **Propagation Delay:** tempo richiesto da un pacchetto per passare da un'end di un link **all'end** opposto (misurato in secondi). Per un link di lunghezza fisica **D** in cui i segnali vengono propagati con velocità **V**, il propagation delay può essere modellato come **D/V**.



Packet loss: si verifica quando un pacchetto viene scartato, corrotto o rifiutato a causa di un **error-checking** (Tramite **checksum** fallito). La causa più frequente di perdita di pacchetti è la congestione. L'informazione precisa circa il

momento in cui un pacchetto viene scartato è difficile da ottenere, in generale il packet loss può essere caratterizzato come un **time series of counts**. Una misura comunemente utilizzata è il **packet loss rate** dove:

- **Cn:** rappresenta il numero di pacchetti che entrano nella rete al tempo n.
- **Ln:** rappresenta il numero di pacchetti persi durante questo periodo.
- Il **packet loss rate** può essere definito come **Ln / Cn**.

Throughput: limiti sulla capacità degli elementi di rete e traffico indotto dalla congestione pongono seri limiti al **throughput** (rate con cui il traffico può fluire attraverso la rete). Se l'intervallo di tempo **T** ha un valore abbastanza grande confrontato con il tempo richiesto per attraversare il network path, il throughput del path durante il periodo di tempo **N** può essere stimato come **Cn/T** dove **Cn** rappresenta il numero di pacchetti che attraversano il path senza nessuna perdita. Inoltre, Il throughput su una sequenza di hop è determinata dagli elementi con minima capacità disponibile (il collo di bottiglia può essere un **endsystem** o un **network-internal element**).

Jitter: Misura la variabilità dei tempi di inter-arrivo di pacchetti individuali (variazione del ritardo tra pacchetti inviati in sequenza da un nodo ad un altro). Si verifica a causa della variabilità dei **time queuing delay**s nei router lungo un path. **Packet arrivals** con un valore **jitter** minimo sono più predicibili e comportano performance maggiori.

Alcune problematiche

The "**stupid network**" è un approccio stateless nel progetto delle componenti Internet dove i routers non mantengono informazioni di stato e quindi nessuna traccia delle connessioni. Il vantaggio è associato al fatto che ha contribuito alla crescita esponenziale di Internet, lo svantaggio riguarda la perdita di osservabilità in diversi punti della rete.

Il **modello IP hourglass** nasconde i dettagli degli strati inferiori (fornisce un'astrazione per consentire interoperabilità, ma impedisce una visione dettagliata degli strati inferiori). Misurazioni dettagliate, come packet capture, non sono in grado di rilevare la differenza tra due tipi differenti di link.

Hidden Pieces-Middleboxes: l'end-to-end argument definisce come alcune funzioni possono essere eseguite correttamente solo dagli endsystem questo implica che gli elementi intermedi nel data path devono eseguire solo il forward dei pacchetti IP senza tener conto della semantica della connessione e che le connessioni TCP devono terminare sugli endsystem e non sugli elementi intermedi. Le ragioni dell'uso dei middleboxes sono moltreplici: **sicurezza**, **management**, **performance** e **address translation**. Alcuni middleboxes sono:

- **Firewalls:** forniscono protezione, bloccando il traffico non voluto analizzando la semantica delle connessioni (numeri di porta) per verificare se un pacchetto deve essere scartato.
- **Traffic shapers:** assistono nel **traffic management** scartando pacchetti in modo selettivo. Un traffic shaper può essere utilizzato per assicurare che non più del 10% del traffico della rete possa essere utilizzato per applicazioni **P2P**.
- **Proxies:** terminano le connessioni all'interno della rete (retrieving di una pagina Web dalla cache del proxy senza che venga interrogato l'origin server).
- **NAT:** è una tecnica che consiste nel modificare gli indirizzi IP dei pacchetti in transito sulla rete e consente un efficiente utilizzo dello spazio degli indirizzi IP (indirizzi all'interno della rete mappati su port numbers all'esterno della rete).

Ognuna di queste componenti impedisce la visibilità delle componenti di rete, i firewalls possono bloccare probing requests (pacchetti UDP o ICMP usati da traceroute), il NAT nasconde il numero di hosts e la struttura della rete (quando eseguito un ping).

Administrative Barriers: gli ISP spesso preferiscono nascondere i dettagli della loro rete interna, le configurazioni dei router, pattern di interconnessione, le qualità del traffico che attraversa i link, queste sono tutte informazioni che ritengono competition-sensitive. Per questo motivo diversi ISP bloccano il traffico che può essere usato per misurare le infrastrutture (ICMP ECHO packets (usato da ping) possono essere bloccati dai router in ingresso). Le informazioni che essi spesso forniscono sono semplificate, infatti, invece di pubblicare router-level topologies, gli ISP spesso pubblicano PoP- level topologies.

Tools: Il processo di misurazione del traffico sulla rete richiede facilities sia di tipo hardware che di tipo software direttamente connesse alla rete, utili a osservare il traffico (di pacchetti) sulla rete e filtrare e collezionare solo i dati di interesse. Assumono tecnologie di tipo broadcast-based e permessi superuser. Questi tools possono essere classificati in **hardware** e **software measurement tools**. Mentre Il tipo di misurazione può essere **attiva** o **passiva**, mentre il sito della misurazione può essere **single**-**point** o **multi-point**. Più nello specifico:

- **Tecniche di misurazioni attive o intrusive:** aggiungono traffico alla rete (pacchetti measurement probes) allo scopo di effettuare misurazioni.
- **Tecniche di misurazioni passive o non intrusive:** analizzano il traffico esistente attraverso tecniche di sniffing.

La classificazione vede la presenza di: **Active Measurement**, **Passive Measurement**, **Fused/Combined Measurement**, **Bandwidth Measurement**, **Latency Measurement**, **Geolocation** e **altri**.

Active Measurement: metodi che aggiungono traffico alla rete allo scopo di effettuare misurazioni (Ping è Traceroute).

Ping: invia un pacchetto **ICMP ECHO** e cattura un pacchetto **ECHO REPLY**. Utile per misurare **RTT**. Solo il sender ha necessità di essere sotto controllo sperimentale.

Traceroute: è utile per determinare il path da una sorgente ad una destinazione. Usa il campo **TTL** (Time To Live) dell'header IP. Inizializzato dal host sorgente e decrementato ogni volta che il pacchetto passa attraverso un router. Quando raggiunge il valore 0, il protocollo **IP** scarta il pacchetto ed un messaggio di errore viene restituito alla sorgente (**ICMP TIME EXCEEDED PACKET**). L'indirizzo sorgente del **ICMP TIME EXCEEDED PACKET** è l'interfaccia del router che ha scartato il pacchetto originale. Se un pacchetto ha **TTL** pari ad **N** e viene inviato verso una particolare destinazione, il router che è l'n-esimo hop lungo il path, può generare il pacchetto ICMP TI… Un sistema per le misurazioni su larga scala che usa traceroute per scoprire la topologia della rete è **Archipelago** (**Ark**) che è un'evoluzione di **skitter**. Il traceroute come metodo per il discovering di path presenta però un insieme di problematiche:

- **Asimmetria dei path:** Il **forward path** scoperto da traceroute può essere diverso dal **reverse path** (path dalla destinazione verso la sorgente) perché potrebbe includere altri nodi. L'output di traceroute deve essere interpretato solo in termini di un path diretto dalla sorgente alla destinazione.

- **Path non stabili e archi falsi:** se i path non sono stabili in un certo periodo di tempo, probes successivi possono seguire differenti path. Una possibile soluzione potrebbe essere non considerare output di traceroute che mostrano instabilità nei path (path multiple osservate in piccoli periodi di tempo).

- **Risoluzione di Alias:** traceroute scopre interfacce e non router che sono soliti avere interfacce multiple, ognuna con il proprio indirizzo IP. L'indirizzo IP sorgente nel messaggio di risposta **ICMP TIME EXCEEDED** è l'indirizzo dell'interfaccia che il router usa quando invia pacchetti alla sorgente. Un metodo per risolvere il problema **dell'alias resolution** è inviare un pacchetto **ICMP ECHO** ad entrambe le interfacce della stessa sorgente. Se le interfacce appartengono allo stesso router allora la risposta verrà inviata dalla stessa interfaccia. Eseguendo il matching dei messaggi **ECHO REPLY** che hanno la stessa source interface, si può dedurre che il pacchetto ECHO originale era stato inviato allo stesso router.

- **Carico delle misurazioni Asimmetria dei path:** traceroute inserisce un notevole carico sulla rete, specialmente se deve effettuare il discovery su larga scala della topologia della rete. Esistono due varianti di questo problema: eseguire traceroute da una singola sorgente verso destinazioni multiple provocherà carichi elevati sui link vicino alla sorgente, eseguire traceroute da sorgenti multiple verso una singola destinazione provocherà carichi elevati sui link vicino alla destinazione.

Passive Measurement: consiste nel catturare ed analizzare il traffico generato da altri utenti ed applicazioni:

- **BGP:** l'inter-domain routing, implementato attraverso BGP, determina come il traffico viene scambiato attraverso gli ASes. Una BGP routing table fornisce informazioni parziali circa l'AS-level topology. Per esempio il fatto che due ASes appaiono in sequenza in un AS path è indice del fatto che sono direttamente connessi, informazione che

permette di dedurre una AS-level topology dalle BGP tables. Per comprendere il traffico all'interno di ogni AS è necessario ottenere le BGP tables (o views) da tutti gli altri ASes. Questo ciò porta alla costruzione di un routeviews repository che colleziona BGP views da grandi ASes. Compito principale di un routeviews è aiutare gli operatori di rete a comprendere lo stato del sistema BGP, oltre ad effettuare monitoring passivo e l'analisi della topologia di Internet.

  - **Vantaggi:** grandi insiemi di AS-AS e router-router connections possono essere studiate semplicemente analizzando tabelle (BGP views).
  - **Svantaggi:** la struttura risultante dell'analisi del routeviews è tree-like, con root il target AS. Ogni cross-connection tra ASes potrebbe non essere scoperto, perché non usate nel routing al target prefix. Inoltre, **route aggregation** e **filtering** tendono a nascondere alcune connessioni AS-AS. Alcuni AS (Specialmente tier1-AS) hanno connessioni fisiche multiple (efficienza e ridondanza), solo un singolo arco fra due AS sarà visibile nel AS graph risultante.

- **OSPF:** è possibile ottenere misurazioni passive anche all'interno di un AS catturando il traffico generato da IS-IS o OSPF. Con OSPF, i dati relativi agli announcements (LSAs) all'interno di un routing domain possono essere raccolti passivamente. Gli LSA possono essere il risultato di cambi nella topologia della rete (link non più disponibile). L'analisi passiva del traffico OSPF può risultare utile per comprendere la topologia di un router graph all'interno di un AS. Una delle tecniche di misurazione passiva più utilizzata consiste nell'utilizzo di **SNMP**. Tutti i device di rete forniscono **MIB** (**Management Information Base**) data. Un MIB per uno switch Ethernet fornisce info sui dati in ingresso ed in uscita su una porta, istogrammi dei frame sizes, numero e tipo di error frames, ecc. Altri MIB danno info sulla configurazione dello switch, stato della coda, CPU status. Esistono inoltre i **Remote Monitoring MIB** (**RMON**) possono listare i 10 utenti più attivi, e la matrice delle comunicazioni. I MIBs vengono interrogati ad intervalli regolari memorizzando i risultati in un database centralizzati. I maggiori network management systems (HP, Nagios) permettono SNMP polling database facilities.

Fused/Combined Measurement: come combinare active & passive measurement. Difficoltà dell'active measurement è la grande quantità di dati richiesta per il probe anche se bisogna mappare un AS singolo. Una possibile strategia per ridurre il probe traffic è quella di eseguire passive meas. attraverso l'accesso alle BGP views del singolo AS. Usando queste views è possibile identificare un insieme limitato di indirizzi che sono parte del target AS. È possibile a questo punto limitare il traffico probes solo a questi indirizzi.

Traffic

Fare misurazione e modellazione del traffico è importanti per:

- Analisi delle prestazioni: analisi del traffico necessario per rispondere a domande relative a throughput, packet loss, packet delay introdotti dalle componenti di rete, il timescale per queste attività variano da microsecondi a decine di minuti;
- Network engineering: network configuration, capacity planning, demand forecasting, traffic engineering, i timescale variano da minuti ad anni.

Proprietà:

Le **analisi sul traffico** osservato vengono riportate attraverso *processi stocastici*. Possiamo considerare solo i punti in cui in un certo istante di tempo un pacchetto arriva in uno specifico punto di osservazione (**arrival process**). Inoltre, è possibile studiare anche le proprietà degli arrivi dei processi.

Spesso un *arrival process view* può contenere più informazioni di quelle richieste, in questo caso è possibile studiare i **time series of counts** su un certo timescale T (quanti arrivi ci sono in un certo intervallo di tempo). Sebbene i packet arrivals siano importanti una misura del traffico più comune è data dal **numero** di **bytes** che essi contengono. Il time series of counts può essere facilmente usato in questo caso, contando il numero di bytes contenuti nei pacchetti che arrivano in ogni intervallo {Bn, n=0,1,..}. Possiamo descrivere la struttura del traffico come il risultato di una collezione di processi **ON/OFF**. Un processo ON/OFF alterna uno stato "on" in cui viene generata attività di rete, ad uno stato

"off" in cui è silente. I processi ON/OFF creano **bursty workloads** che presentano problematiche per le performance. Esistono tre principali livelli di attività ON/OFF nel traffico di rete:

- Al livello più basso i pacchetti stessi sono processi ON/OFF
- Al livello superiore i pacchetti formano "trains" dalla sorgente alla destinazione
- Al livello superiore insiemi di trains formano una sessione (una singola esecuzione di un'applicazione)

Flusso: un flusso è un insieme di pacchetti che attraversano un punto di osservazione durante un certo periodo di tempo, tutti i pacchetti hanno delle proprietà comuni (contenuto dei campi dell'header IP, caratteristiche del pacchetto stesso, come il pacchetto è processato). Sono separati da periodi idle. Tutti i pacchetti dal SYN iniziale all'ACK finale.

IP flows: un insieme di pacchetti distinguibili per indirizzo sorgente o destinazione (o per altri valori header) è chiamato IP flow.

Network flow: insieme di pacchetti che entrano in una rete in un certo tempo ed escono da un altro punto (origin-destination o ingress-egress flow).

Challenges:

Misurare le proprietà del traffico presenta diverse problematiche:

- **Pratical issues:**
  - Osservabilità;
  - Volume dei dati;
  - Condivisione di dati;
- **Statistical issues:**
  - Long tails e alta variabilità;
  - Stationarity e stability;
  - Autocorrelation and memory in system behavior.

PRATICAL ISSUES:

Osservabilità:

- **Core semplicity**: semplicità dei router che contribuiscono ad una perdita di osservabilità in diversi punti della rete;
- **Flussi**: non si possono raccogliere dati visto che il per-flow state non viene mantenuto nei router;
- **Packets**: packet capture in generale implementato sugli endsystems;
- **Distributed Internetworking**: non esiste più un singolo backbone e non esiste nessuna rete che fornisce un punto conveniente per la misurazione della maggior parte del traffico Internet, ogni punto ha una visione locale del traffico e delle proprietà della della rete e non è rappresentativo di altri punti;
- **IP Hourglass**: pacchetti corrotti, persi e ritardi al livello fisico possono essere mascherati da ritrasmissioni al link-layer, queste proprietà del traffico sono generalmente non visibili quando le misurazioni sono fatte al livello IP o superiore.

Volume dei dati:

Quanto più la velocità dei link aumenta, tanto più complesso diventa la raccolta dei dati di traffico (full packet capture). Problematiche relative a processing, storage e management. Queste operazioni sono appropriate solo su **short timescale**, normalmente un minuto o meno, mentre attraverso tecniche quali **sampling** è possibile arrivare a considerare timescale di giorni. Per estendere il periodo di sampling è possibile decidere di memorizzare solo **packet headers**.

Condivisione di dati:

Il traffico di rete contiene una grande quantità di informazioni critiche, il full packet capture memorizza le attività di rete degli utenti, da queste tracce è possibile estrarre informazioni quali siti Web visitati, password, contenuto di mail. Le misurazioni sul traffico possono fornire informazioni sulla

configurazione e sul funzionamento della rete che si sta monitorando, identità del peer, customers, interconnection points, politiche per il routing. Per diversi ISP queste informazioni devono essere riservate, la necessità di proteggere la privacy degli utenti ed i "segreti" dei network providers sono in contrasto con il network measurement.

STATISTICAL ISSUES:

Long tails e alta variabilità:

Diverse proprietà sul traffico Internet mostrano una alta variabilità causate da instabilità delle metriche:

- Instabilità delle metriche: media e varianza;
- Diverse osservazioni molto piccole e poche molto grandi.

Semplici statistiche come media, mediana, o varianza sono non affidabili in presenza di dati altamente variabili per uno studio delle proprietà del dataset, siccome su un dataset altamente variabile non si stabilizzano mai su un particolare valore. Le distribuzioni di probabilità forniscono un approccio più generale per rappresentare come i parametri variano in un ampio range di valori. Però ci sono delle difficoltà nella fase di modellazione, non sempre si è in grado di distinguere la giusta distribuzione, la maggior parte dei dataset mostra una variabilità di 3 o 4 ordini di grandezza e il comportamento della tail può essere studiato solo su un range limitato. La differenza fra una distribuzione lognormal o di Pareto può essere molto piccola.

Stationarity e stability Stabilità:

Stabilità:

- Consistenza del traffico nel tempo,
- L'analisi differisce a seconda del timescale.

I dati possono risultare instabili su timescale di ore. Con picchi nelle ore diurne della giornata. I dati possono risultare stabili su un timescale di un'ora.

Autocorrelation and memory in system behavior:

Networks ed endsystems mostrano memoria perchè contengono buffer e algoritmi che mantengono la storia passata in un modo che influisce sul comportamento corrente. Quando un link ha un alto utilizzo un router può mantenere una coda abbastanza grande che richiederà però tempo per svuotarla. Comporta ritardo nel forwarding di un pacchetto dovuto ad eventi del passato, questo comporta autocorrelazione nel comportamento del sistema. Il comportamento corrente è molto simile a quello di un passato recente.

Tools

I tool più importanti per l'analisi del traffico sono quelli di packet capture. Nella categoria del packet capture rientrano:

- Packet capture su General Purpose Systems
- Packet Capture su Special Purpose Systems
- Control Plane Traffic.

PACK CAPTURE SU GENERAL PURPOSE: un endsystem può catturare e memorizzare tutti i pacchetti che passano attraverso le sue interfacce di rete. La API più comunemente utilizzata è **libpcap**. Questa libreria
contiene entry points per specificare le interfacce da monitorare i pacchetti da catturare. Fornisce raw data.
I software devono essere utilizzati per il **parsing di header fields** e l'analisi dei protocolli i principali sono **tcpdump** e **wireshark**.

Il **tcpdump** viene comunemente utilizzato per il debugging di applicazioni di rete e della configurazione della rete, determina se tutti gli instradamenti necessari avvengono correttamente o meno, consentendo all'utente di isolare successivamente la sorgente del problema, per intercettare e visualizzare le comunicazioni di un altro utente o computer.

Wireshark è un packet sniffer che permette all'utente di osservare tutto il traffico presente sulla rete, offre le stesse funzionalità di tcpdump ma con un'interfaccia grafica.

Software non commerciali per il packet capture sono **OC3MON** e **OC12MON**.
Per risolvere i problemi relativi all'accesso autorizzato (richiesto da libpcap e tcpdump) sono disponibili altri tool (pktd, scriptroute). Consentono ad utenti senza privilegi di catturare pacchetti dalla rete e inserire pacchetti nella rete

PACK CAPTURE SU SPECIAL PURPOSE SYSTEMS: Catturare pacchetti all'interno di una LAN è semplice ma su Internet il compito diventa più complesso: il monitoring di core network links richiede hardware specializzato. Al livello più basso sul link bisogna consentire un accesso fisico per la copia dei dati sia per link elettrici sia per link ottici. Software non commerciali per il packet capture sono:

- OC3MON
- OC12MON

CONTROL PLANE TRAFFIC: Obiettivo è il controllo del traffico durante lo scambio di messaggi di routing. L'idea è quella di registrarsi per ottenere **BGP updates** stabilendo una sessione con un BGP-speaking router. Stabilire questa connessione significa trovare un router BGP (un amministratore di sistema di un BGP router che ci consenta di stabilire una sessione BGP). Il BGP router a questo punto fornisce messaggi come se il listening device fosse un altro router. Il listening device è passivo e non genera BGP updates. GNU Zebra cattura dati in formato binario. I dati raccolti possono essere tradotti in formato leggibile usando tools dall'MRDT tollkit. La grande mole di dati che si può catturare, e le tabelle BGP associate sono archiviate e rese disponibili dal Routeviews project.

Data management: Il full packet capture e storage presentano diverse problematiche per link ad alta velocità, bus bandwidth e velocità di accesso alla memoria su comuni PC, enorme quantità di dati da analizzare. La soluzione è l'uso di tools specializzati quali **Smacq** e **Windmill** per il packet filtere packet processing. La caratteristica dei dati di rete è la grande quantità di dati che arrivano incrementalmente nel tempo. Diversi tools sono stati sviluppati per interrogare database di dati: Tribeca, STREAMS system, TelegraphCQ (ultime due scritte in SQL). Il più importante è **Gigascope** sistema specializzato per l'analisi del traffico di rete (queries sono espresse in GSQL).

Data reduction: Problema: ridurre la quantità di dati che bisogna processare. I metodi più comuni che permettono di fare ciò sono:

- Uso di traffic counters e collezione di flussi record;
- Metodi basati su sampling;
- Modelli porbabilistici.

Counters: Aggregazione di dati usando series of counts di traffico (bytes o pacchetti per unità di tempo). Approccio supportato dai router attraverso il MIB-II Management Information Base, il cui accesso avviene attraverso SNMP. Il numero di pacchetti inviati e ricevuti su ogni interfaccia è dato da **ifInOctects** e **ifOutOctets**. I MIB counters mantengono dati totali dalla fase di inizializzazione dell'interfaccia (router reboot). Lo svantaggio è che alcuni pacchetti potrebbero andare persi perché l'SNMP si appoggia su UDP ed è difficile la sincronizzazione in quanto i dati sono collezionati via polling.

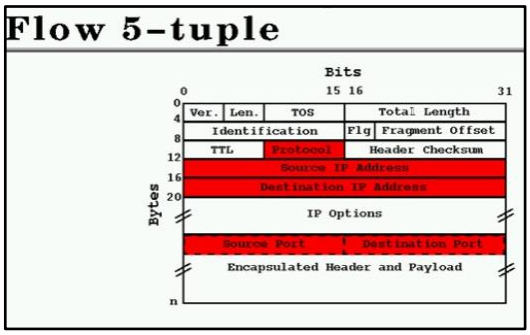
Flow capture: packet trains: I counter forniscono dettagli circa l'attività di rete, ma la semantica del traffico è completamente persa. Un'alternativa per preservare le informazioni del traffico per gli analisti di rete è quella di catturare e memorizzare packet trains or flows. Packet trains traces sono utili per il monitoring delle attività di base di una rete, monitoring di utenti ed applicazioni, network planning e security analysis. Un packet train record consiste delle seguenti informazioni:

- Source IP address;
- Destination IP address;
- Source port;
- Destination port;
- Protocol ID;
- Start time;
- End time;
- Numero di pacchetti;

- Numero di bytes contenuti nel packet train.

Una complicazione per quanto riguarda il packet train capture riguarda come definire la fine di un packet train. Analisi a livello applicativo devono determinare quando un high level data transfer è stato completato. Una possibile soluzione è osservazioni su un pacchetto FIN o RST.

Flow capture: packet flows: In alcune situazioni non è necessario sapere packet train start e packet train end ma solo la tupla di 5 valori in un determinato istante di tempo. **Flowtools** è un insieme di tools che collezionano, inviano, processano e generano reports da **Cisco NetFlow** o **cflowd**. Cisco NetFlowfornisce servizi per: network traffic accounting, network planning, security, denial of Service monitoring capabilities, network monitoring. NetFlow fornisce infomazioni su: utenti e applicazioni, tempi sui picchi di traffico e traffic routing.



Risultati riguardanti le proprietà del traffico Internet

- Long-Range Dependence (LRD)
- Self-Similarity
- Heavy-tails. LRD

Diverse issues nell'analisi del traffico dipendono dal particolare timescale di interesse. Su lunghi timescale (maggiori di un'ora) il traffico presenta variazioni predicibili, cioè **trends**, è possibile vedere se il traffico è stazionario, verificare la differenza tra le ore diurne e notturne e le differenze fra il traffico della settimana e quello del weekend. Per descrivere il traffico di rete su piccoli timescales si utilizzano modelli Stocastici (Insieme ordinato di variabili casuali indicizzate dal parametro t). Lo **Scaling behavior** del traffico di rete può essere descritto in termini di 2 fenomeni correlati:

- **Long-RangeDependence**
- **Self-similarity**

Long-RangeDependence: Per comprendere il concetto di correlazione partiamo dal concetto opposto: Indipendenza. Traffico modellato come un processo stocastico stazionario {Xn , n=0,1,…}. Se gli {Xn} sono indipendenti allora molte delle proprietà del processo sono semplici da calcolare e comprendere. Per descrivere la distribuzione dell'intero processo {Xn} sarà sufficiente caratterizzare la singola distribuzione P di x0. L'indipendenza influenza le proprietà di scaling:

- Se {X0} ha media μ e deviazione standard
- Per il teorema del Limite Centrale: Ogni componente dello scaled process Convergerà alla distribuzione Normale con media 0 e deviazione standard.

Teorema del Limite Centrale ci dice che se si ha una somma di variabili aleatorie Xi indipendenti e identicamente distribuite, allora indipendentemente dalla forma distributiva di partenza, al tendere della dimensione campionaria a infinito la somma tende a distribuirsi come una variabile casuale Normale.
Vantaggi del modello di Poisson per il performance modeling. È descritto da un solo parametro λ (Rate degli arrivi). Il numero di arrivi che si verificano su 2 non-overlapping intervals sono indipendenti e la distribuzione degli arrivi in un intervallo di lunghezza T segue la distribuzione di Poisson:

$$p_{X_0}(n) = e^{-\lambda T}\frac{(\lambda T)^n}{n!} \qquad n>=0$$

Il traffico Internet mostra correlazione e quindi NON può essere modellato con modelli indipendenti come i processi di Poisson. Un processo stazionario è Long-Range Dependence se la funzione di autocorrelazione è non sommabile (converge a 0 cosi lentamente che la loro somma non converge). Il traffico LRD mostra instabilità su un ampio range di timescales, non esiste pertanto nessun timescale nel quale anche approssimativamente le assunzioni viste per i modelli indipendenti continuino a valere. La presenza di fluttuazioni (picchi) su parecchie o su tutte le scale temporali

significa che il traffico non appare omogeneo in nessun timescale Presenza di memoria di tipo long range, i valori ad ogni istante sono fortemente correlati con valori, anche lontani, di istanti passati, questa proprietà è catturata dalla nozione di SelfSimilarity.

Self-Similarity: La presenza di fluttuazioni (picchi) su parecchie o su tutte le scale temporali significa che il traffico non appare omogeneo in nessun timescale (presenza di memoria di tipo long range). Questa proprietà è catturata dalla nozione di Self-Similarity. Intuizione: Qualcosa che "feels the same" indipendentemente dalla scala. Una proprietà tipicamente associata ai frattali, oggetti che appaiono essere gli stessi a dispetto della scala su cui sono visualizzati. Descrive il fenomeno in cui il comportamento di un processo viene preservato a dispetto di operazioni di scaling (di spazio o tempo). Variabilità in un ampio spettro di scale temporali. Non trova riscontro nei modelli adoperati per anni per la caratterizzazione del traffico nelle reti. Il modello di Poisson genera un traffico bursty se osservato su scale dell'ordine dei millisecondi. Al crescere della scala questa caratteristica si perde. **Curva di Koch snowflake:** una delle prime curve frattali di cui si conosca una descrizione. Apparsa in un documento del 1904 intitolato "Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire" del matematico svedese Helge von Koch. **Modello di Poisson**: il traffico osservato su piccoli timescale è bursty. Su grandi time scale il traffico appare omogeneo. **Self-Similar Model**: il traffico bursty ha caratteristiche simili su ogni scala. Il traffico ha proprietà statistiche simili su differenti timescale (ms, sec, mins, hrs,days). Il merging del traffico non comporta un traffico omogeneo.

Heavy Tails: Diversi fattori contribuiscono alla presenza di Long-Range Dependence e Self-Similarity nel network traffic, la causa principale è l'influenza dei **packet trains**. Un packet train che si estende su più osservazioni contribuisce per i pacchetti di ciascuna osservazione, e quando un packet train influisce sull'analisi di più di un range of counts i valori nei range diventano correlati. I fenomeni di **LRD** e **self-similarity** possono essere individuati per la presenza di heavy tails nella durata dei packet trains. Le lunghezze dei packet train sono **heavy-tailed** (lunghezze dei periodi ON) perchè la taglia dei dati stessi è heavy-tailed. Esempi di traffico **self-similar**: Ethernet Traffic, WWW traffic, TCP,FTP, TELNET traffic..

Principali osservazioni scaturite da ricerche recenti nel campo del Network Traffic Measurement:

- **Il traffico Internet continua a cambiare e crescere su piccoli timescales.** Ogni data set raccolto da una qualunque componente di rete rappresenta solo una istantanea di un singolo punto dell'evoluzione di Internet;
- **Il traffico aggregato è self-similar (o frattale).** Caratterizzarlo è difficile per diverse ragioni: natura eterogenea di Internet; diverso mix di applicazioni, eterogeneità e varietà per quanto riguarda velocità delle reti e tecnologie di accesso; cambiamento nel comportamento degli utenti;
- **Il traffico di rete esibisce la proprietà di "locality",** cioè la struttra del traffico di rete non è random ma è imposta implicitamente dai task iniziati dagli utenti a livello applicativo. I pacchetti non sono indipendenti, ma parte di un flusso riconoscibile;
- **Il traffico non è uniformemente distribuito attraverso gli host della rete.** Una osservazione comune è che il 10% degli hosts è presente nel 90 % del traffico. 28 Conseguenze importanti sulla struttura power-law di diversi aspetti del traffico Internet e della topologia di Internet;
- **La distribuzione della taglia dei pacchetti è bimodale**. La maggior parte dei pacchetti piccoli sono per traffico interattivo e acknowledgements (40% sono di piccola taglia), la maggior parte dei pacchetti di grosse dimensioni sono relativi alle applicazioni (50%), pochi pacchetti tra le due differenti taglie (il rimanente 10%);
- **Il session arrival process è di tipo Poisson.** Gli utenti Internet operano in maniera indipendente e random quando accedono ad alcune risorse sul Web.
- **Il packet arrival process non è di tipo Poisson**. I tempi di interarrivo non sono esponenzialmente distribuiti e non sono indipendenti (traffico Bursty);
- **La maggior parte delle conversazioni sono brevi**. Il 90% dei dati trasferiti è inferiore ai 10 kilobytes, il 50% delle connessioni interattive persiste per meno di 90 secondi, le distribuzioni sono ''heavy tailed'';
- **I flussi di traffico sono bidirezionali ma asimmetrici**. Natura download-intensive del Web;
- **Il protocollo TCP è il più usato su Internet**, nonostante applicazioni Internet quali video streaming, napster, IP telephony, e multicast usino UDP.

In conclusione, l’Internet traffic measurement è una metodologia che ha come obiettivo quello di comprendere il traffico su Internet. Dalle prime misurazioni su LAN a diverse intuizioni su proprietà, protocolli e utenti di Internet. Richiederà il costante monitoring delle assunzioni fatte fino ad oggi e la dimostrazione che i modelli formulati continuino a valere nella realtà.