



Reti di Calcolatori-Appunti Dominick Ferraro
Anno: 2021-22

Definizioni:

Network = Rete.

Killer application: applicazioni che spopolano.

Computer Network: agglomerati di computer autonomi. Sono computer, dispositivi, un esempio è la rete Internet.

I sistemi telematici sono un'efficiente soluzione per la:

- Condivisione di risorse.
- Comunicazione.
- Cooperazione tra sistemi che collaborano per la risoluzione di un dato problema.

Uso delle reti: accesso alle info, comunicazione, commercio, intrattenimento, IOT(internet of things).

- Non bisogna confondere web con Internet. Il web nasce dopo internet col protocollo http.

Accesso alle info: web browser, social media platform, librerie digitali, web applications, modello client-server, comunicazione peer-to-peer.

In un client-server, un cliente chiede esplicitamente una info al server che la ospita.

C'è un processo client che fa una richiesta, il server la elabora e invia la risposta al client richiedente. Il client deve aspettare la risposta al messaggio.

Nel sistema peer-to-peer tutti possono comunicare con tutti e non c'è un ordine gerarchico, come nel caso di client-server.

Comunicazione Persona A Persona: messaggistica istantanea, social network, collaborazione tra utenti.

Commercio Elettronico: shopping online, aste.

Ci sono diversi tipi di shop online come:

- B2C: business to consumer
- B2B: business to business
- G2C: government to consumer
- C2C: consumer to consumer

- P2P: peer to peer
-

Intrattenimento:

- IPTV: Tv show basati su tecnologia IP.
 - Media Streaming Applications: Tv, film, radio.
 - Gaming.
-

THE INTERNET OF THINGS (IOT):

- Reti power line: mandano info con la corrente elettrica.
 - IoT: sensori e comunicazione. Collega ogni dispositivo elettronico a internet.
 - Ubiquitous computing: sistemi domestici, di sicurezza, frigoriferi smart, ecc...
-

TIPI DI RETE

- Rete mobile e banda larga: reti usate per l'accesso a internet.
 - Reti Data-Center: reti che ospitano dati e applicazioni.
 - Transit Network (reti di transito): reti che connettono altre reti ai data center.
 - Enterprise network: reti usati nei campus, negli uffici ecc...
-

RETI DI ACCESSO A BANDA LARGA:

- Reti di uso domestico: ascoltare musica, vedere video, foto ecc...
 - Legge di Metcalfe: il valore di una rete è proporzionale al quadrato del nr degli utenti.
 - Reti a banda larga: spedita a casa usando rame, fibra ottica ecc...
 - o Velocità per banda larga: gigabit per secondo per ogni casa.
-

Mobile E wireless Access Network

- Hotspot basati sullo standard 802.11 IEEE
- Reti wireless e mobile
- Gli smartphone combinano aspetti mobile e computer.
- GPS (global positioning system)
- Geo-tagging.

GPS e Geo-tagging non è sicura.

!! Reti wireless e mobile non sono la stessa cosa:

Wireless	Mobilità	Usi
No	No	Computer da tavolo negli uffici
No	Sì	Computer portatile usato in una stanza di albergo
Sì	No	Reti negli edifici non cablati
Sì	Sì	Palmare per l'inventario di magazzino

!!!!!!

M-commerce: mobile commerce

- NFC (Near Field Communication): permette i dispositivi mobili di interagire con le smartcard per i pagamenti e altro. Per interagire deve esserci un reader nelle vicinanze.
 - Reti di sensori: dispositivi che prelevano ed elaborano info circa il mondo che ci circonda (sensore di fumo, di fuoco, telepass...).
-

CONTENT PROVIDER NETWORKS

- Reti data-center: servizi internet cloud.
 - CDN: larga collezione di server, distribuiti geograficamente in modo che il contenuto è più vicino all'utente richiedente. Usano host per la distribuzione di contenuti. Es quando WhatsApp è offline ci sarà un problema di CDN in una parte del globo.
-

ENTERPRISE NETWORK:

- Condivisione di risorse.
 - VPN: rete virtuale privata che crea una rete su altre reti. Ampiamente usata tra impiegati.
 - IP o VoIP (Voice over IP): reti tecnologiche per chiamate telefoniche.
 - Permette lo sharing del desktop.
-

- Personal Area Networks (PAN): permette i device di comunicare su un range di una persona.

LAN: local area network: sia Wi-Fi che ethernet. Lo standard per le reti LAN è il Wi-Fi. Adottate per connettere sistemi (PC, periferiche) presso organizzazione private o pubbliche, si distendono nell'ambito di un singolo edificio o campus (dimensione: meno di 1 km).

- Reti Domestiche: facili da installare, sicure, poco costose.
- MAN (Reti di aree metropolitan): i segnali di internet e di tv sono cablati insieme per la distribuzione ai cittadini.
- WAN (wild area network): reti con linee dedicate per la comunicazione tra aree. Col tempo si è capito che conviene usare internet piuttosto che linee dedicate.
- WAN BASATE SU SATELLITE: Ogni router invia e riceve info dal satellite broadcast *downlink* (*dal satellite alla terra*) e *broadcast uplink*.
- WAN BASATE SU RADIO AL SUOLO: entro una certa distanza massima, ogni router comunica via onde radio con i propri vicini.

- Una WAN può essere anche realizzata in maniera mista: parte cablata e parte basata su radio o satellite.
 - In caso di *Internetwork* (*collegamento di LAN, MAN e WAN*) i *gateway* (*router multiprotocollo*), effettuano instradamento e conversioni di protocollo.
-

Differenza tra PAN e LAN con WAN: la velocità.

INTERNET: collezione di reti interconnesse. Le reti combinano sottoreti e host. Le sottoreti possono essere descritte con ISP o WAN.

Le reti sono operate in maniera indipendente. Posso connettere una LAN a una WAN o più LAN, i gateway fanno queste connessioni tra due o più reti. Non ha senso collegare una rete a sé stessa, causerebbe un loop infinito con errori.

Lez 2

Internet ha tante entità amministrate in maniera indipendente ma interconnesse tra loro.

L'ARPANET (era divisa in due parti, sottorete ed host) è stato il primo tentativo di internet, abbiamo poi la NSFNET e l'architettura Internet.

Ci sono poi le reti mobili e wireless (Wi-Fi).

Altri esempi di reti:

- Tv via cavo
- HFC
- DOCSIS

Convenzionalmente, l'architettura di internet è vista come una gerarchia, ci sono gli acquirenti, i provider locali, quelli continentali e i nazionali, col tempo la gerarchia si è più compattata con i national, regional e customer IP (scritte dall'alto verso il basso).

RETI MOBILI

L'architettura delle reti mobili comprende diverse parti.

Quando un utente si sposta dal range della stazione base e si trova in un altro, c'è l' "hopping" cioè il salto di cella tra una stazione e l'altra. Tutto il flusso

di dati deve essere “re-routed” dalla vecchia alla nuova stazione.

Il packet switching deriva dalla comunità internet: ci sono pacchetti che vengono inviati e ogni pacchetto può essere ruotato indifferentemente, se alcuni router vanno off durante la sessione, il sistema si può riconfigurare autonomamente.

Il circuit switching deriva dalle compagnie telefoniche, c’era un numero da chiamare per ricevere una connessione, era meno efficiente ma supportava una qualità di servizio ottima.

- First-generation → 1G
- 2G → la parte voce è trasmessa in maniera digitale e offre messaggi
- 3G → si utilizzano servizi dati digitali ma aveva scarsità di banda
- 4G → offre maggiore velocità
- 5G → arrivano fino a 10 GigaBit

Per evitare le interferenze tra utenti, le aree di copertura sono divise in **celle**.

RETI WIRELESS (Wi-Fi)

- IEEE creò una LAN soprannominata 802.11. Lavora su bande non licenziate. L'utilizzo è industriale, scientifico, medico...
 - Funzionano sia in modalità Ad Hoc (tra due dispositivi) o di Access Point.
 - Si basano sullo schema CSMA.
 - L'aspetto mobilità delle reti Wi-Fi è limitato, si preferiscono infatti le reti mobili.
 - Per quanto riguarda l'aspetto di sicurezza c'è stato WEP, WPA, WPA2, WPA3.
 - Le frequenze usate dall'802.11 possono essere disturbate e possono arrivare delle onde indirette, ciò significa che posso avere segnali Wi-Fi anche se è riflesso da pareti o oggetti.
 - Il range di un'entità radio potrebbe non coprire un intero sistema.
-

PROTOCOLLI DI RETE

Scopi della progettazione di rete:

1. Affidabilità
2. Sicurezza
3. Evoluzione
4. Allocazione di risorse

Il design dei protocolli è stratificato.

- **AFFIDABILITA'**: una rete deve continuare ad operare anche se è stato compromesso un insieme di componenti, ad esempio si genera un errore sul cavo. Le parti di routing devono consentire alla rete di prendere decisioni automatiche, trovando percorsi efficienti.
 - **ALLOCAZIONE DELLE RISORSE**: bisogna fare design di soluzioni scalabili, al crescere del numero di nodi il tutto deve funzionare. Questo problema può capitare in qualsiasi rete. Si implementa il servizio di qualità: se abbiamo una rete condivisa ogni dispositivo deve ricevere una determinata “porzione” di rete.
 - Evoluzione: ogni problema deve essere risolto. Emergono sempre nuove problematiche che devono essere risolte e aggiornate.
 - Sicurezza: garantire la confidenzialità del dato, che non può essere intercettato o modificato, garantendo l'integrità.
-

PROTOCOLLI DI STRATIFICAZIONE

Le reti sono organizzate come uno stack di livelli, ogni livello è costruito su quello al di sotto. Le comunicazioni attraverso strati corrispondenti sono

dette “protocollo di livello n”. un’architettura di rete è un insieme di strati e protocolli.

CONNESSIONI E AFFIDABILITA’

- Connessioni oriented-service: sono modellate con i sistemi telefonici. L’utente fa una connessione, la usa e poi la rilascia.
- Servizi Connectionless: modellate con sistemi postali. Le info (spezzettate) possono seguire percorsi diversi e arrivare in ordine diverso da quello di partenza o addirittura non arrivare mai.

PRIMITIVE DEI SERVIZI

Le primitive dicono al servizio di effettuare un’azione o un report su un’azione intrapresa da un peer.

Ci sono 6 primitive di base:

- Listen: blocca in attesa di una connessione.
 - Connect: stabilisce una connessione.
 - Accept: accetta una connessione.
 - Receive: blocca in attesa di un messaggio.
 - Send: invia un messaggio al peer.
 - Disconnect: termina una connessione.
-

DIFFERENZA TRA SERVIZIO E PROTOCOLLO

- Un *servizio* è un insieme di primitive che un livello offre al livello superiore.
 - Un *protocollo* è un insieme di regole che governano il formato e il significato delle info che gli host si scambiano.
-

COME FUNZIONA UN'ARCHITETTURA DI RETE

Un'architettura di rete è un software organizzato a livelli, ciascuno costruito sopra il precedente.

Il livello n offre servizi al livello $n+1$, nascondendo dettagli implementativi.

Al variare della rete possono essere differenti:

- Numero e nomi dei livelli.
- Contenuto e funzioni dei livelli.

Il livello n su un host conversa, secondo apposite regole, con il livello n di un altro host.

Il dialogo tra due *peer entity* di livello n viene realizzato tramite i servizi offerti dai sottostanti $n-1$ livelli.

Sistemi con caratteristiche diverse possono comunicare solo se implementano gli stessi protocolli.

DIALOGO NEL MODELLO A LIVELLI

Le info prodotte dal livello i del mittente, giunte a destinazione, saranno elaborate dal livello i del destinatario.

LATO MITTENTE:

il livello i del mittente passa i dati ricevuti dal livello $i+1$, più le proprie info di controllo al livello $i-1$.

Giunti al livello più basso, cioè quello **fisico**, i dati vengono inviati al destinatario.

LATO DESTINATARIO:

i dati giunti al destinatario, vengono passati dal livello i al livello $i+1$ fino a raggiungere il livello n .

il livello i , ricevuto i dati del livello sottostante, gli sottrae le info di controllo e le passa al livello superiore $i+1$.

LEZ 3

MODELLI DI RIFERIMENTO

- ISO/OSI
 - TCP/IP
-

MODELLO ISO/OSI

Prevede 7 livelli, ognuno con delle funzioni.

È uno standard per la connessione di sistemi aperti.

È un modello rispetto a cui confrontare le varie architetture di rete.

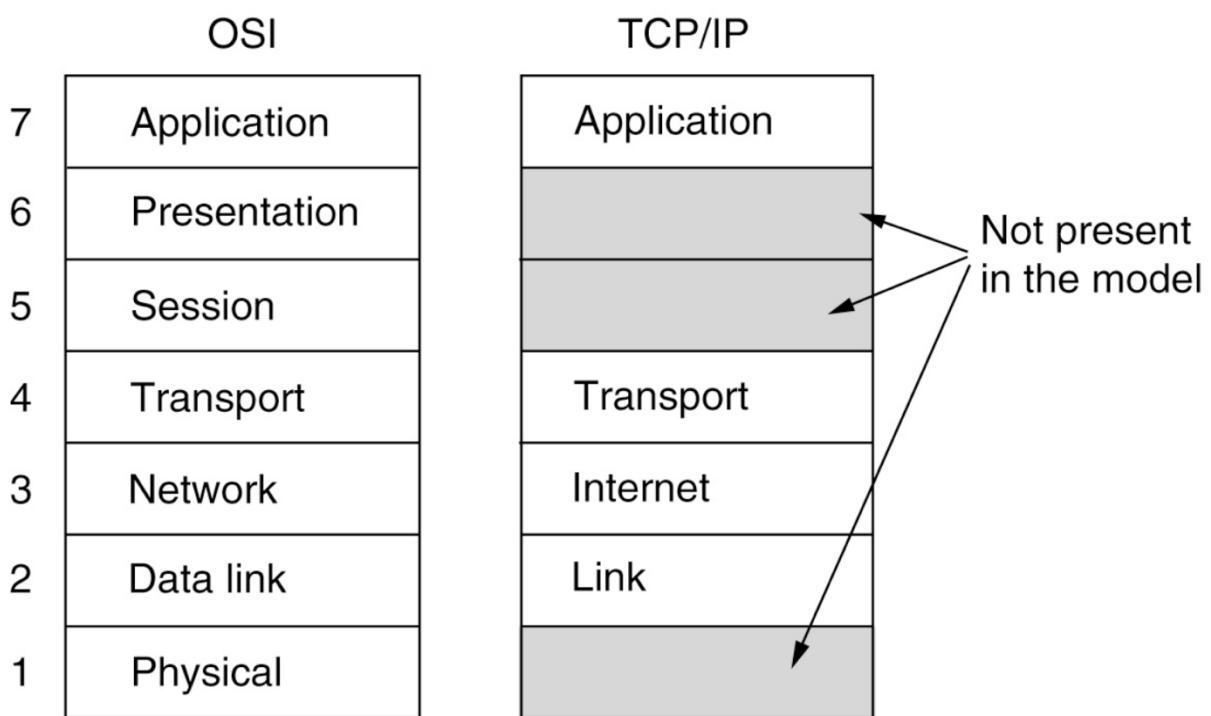
7 LIVELLI DEL PROTOCOLLO ISO/OSI

Partendo da quello più basso:

- Fisico: trasmissione dei dati lungo il supporto fisico di comunicazione.
- Data link: definizione del frame
- Rete: definizione dei pacchetti, indirizzamento e instradamento.
- Trasporto: invio e ricezione di dati con controllo degli errori.
- Sessione: instaura e chiude le comunicazioni.
- Presentazione: formatta e trasforma i dati.
- Applicazione: interfaccia tra il sistema di comunicazione e le applicazioni.

TCP/IP

- Il livello link descrive quali link devono considerare i bisogni di questa trasmissione senza connessione.
- Il livello internet consente agli host di inserire dei pacchetti in ciascuna rete e farli raggiungere la loro destinazione. Definisce un pacchetto e un protocollo chiamato IP.
- Il livello di trasporto si trova sopra il livello internet e usa due protocolli di trasporto “end-to-end”.
- Il livello Application contiene tutti i protocolli ad alto livello.



La differenza tra ISO/OSI e TCP/IP è che il secondo possiede solo 4 livelli:

- *Host-to network*: il TCP/IP non specifica il livello host-to-network, prevedendo a riguardo l'uso di quelli disponibili per le varie piattaforme HW e conformi agli standard.
 - *Intenet*: permette ad un host di suddividere il messaggio in più pacchetti e di iniettarli in una qualunque rete e magari per diverse strade.
 - *Transport*: progettato per consentire la conversazione delle peer entity, prevede due distinti protocolli: TCP (protocollo connesso ed affidabile) e UDP (protocollo non connesso e non affidabile).
 - *TCP*: i pacchetti arrivano tutti e corretti.
 - TCP LATO MITTENTE: frammenta in pacchetti il flusso in arrivo (messaggi) dal livello superiore che poi passa al livello internet.
 - TCP LATO DESTINATARIO: i pacchetti in arrivo vengono riassemblati in un flusso di output (messaggio) per il livello superiore.
 - UDP: i pacchetti possono arrivare in ordine diverso o mai arrivare.
 - *Application*: i protocolli di questo livello sono:
 - *telnet*: terminale virtuale
 - *FTP*: file transfer
 - *SMTP e POP*: *e-mail*
 - *DNS*
 - *NNTP*
 - *HTTP*
-

STANDARDIDDAZIONE

Standard significa cosa è necessario per fare interoperabilità.

- Wi-Fi Alliance ha stabilito lo standard 802.11
- ONF ha stabilito l'interoperabilità dei protocolli.

2 categorie di standard:

- Quelli che nessuno ha deciso che sarebbero diventati standard (ed WhatsApp è diventato uno standard di messaggistica senza pianificazioni).
 - Standard legislativi introdotti con regole di enti internazionali.
-

Chi ha creato gli standard internazionali?

- ISO
- NIST
- IEEE

Chi ha creato gli standard di Internet?

- IAB
 - Internet Society
 - World Wide Web Consortium (W3C)
-

POLITICHE, LEGAL E PROBLEMI SOCIALI

Trasparenza, nessuna priorità a pagamento, nessuna censura.

I Provider System dovrebbero prevedere lo stesso servizio di qualità a qualsiasi applicazione.

SICUREZZA:

- Attacchi DDoS
- E-mail spam
- Phishing
- Botnets (reti SW malevoli installate sui dispositivi)

PRIVACY:

- Browser fingerprinting: il browser può essere identificato univocamente tra gli altri.
- Conservare cookies nei browser
- Tracciamento e profili social

DISINFORMAZIONE:

- Fake news

« manca la parte di cable»

SOMIGLIANZE TRA OSI E TCP/IP

Entrambi sono basati sul concetto di pila di protocolli indipendenti e hanno funzionalità simili per i vari livelli.

DIFFERENZE:

- OSI nasce come modello di riferimento
- TCP nasce coi protocolli, il modello viene a posteriori.

L'insuccesso di OSI ha diverse cause, sia perché la definizione dei protocolli è arrivata tardi, sia perché il modello a 7 livelli aveva molti difetti.

C'è da dire che nemmeno la TCP/IP è perfetta poiché:

- È poco utile come modello
- Non c'è una distinzione chiara tra protocolli, servizi e interfacce
- Alcune scelte di progetto cominciano a pesare, come gli IP a soli 32 bit.

LEZ 4

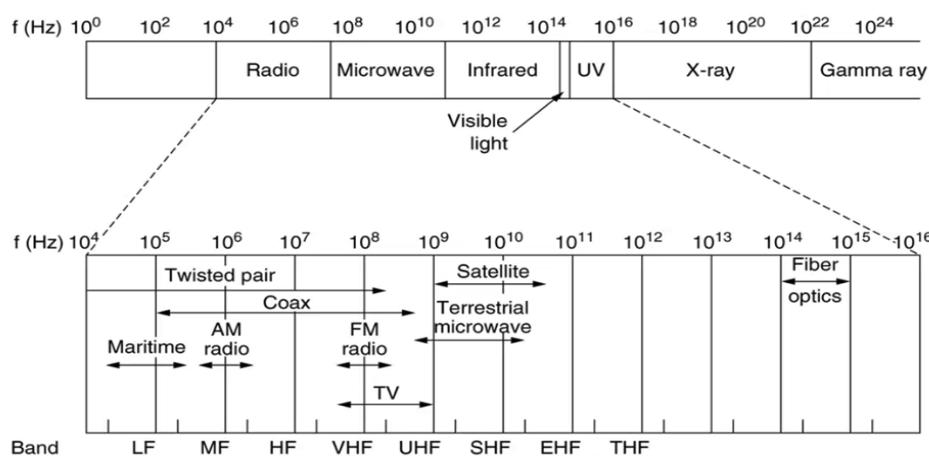
TRASMISSIONE WIRELESS

Anche il GPS è un meccanismo di trasmissione wireless. Parliamo di canali di comunicazione dati, è un meccanismo broadcast. Il GPS non veicola dati generici che possiamo far passare.

Ci sono alcune tipologie di trasmissioni e agiscono sullo spettro radio:

- Spettro elettromagnetico: si può cambiare la frequenza e l'altezza dell'onda.
 - Frequency hopping spread spectrum: chi trasmette salta centinaia di volte da un'onda all'altra.
 - DSSS: la sequenza di dati viene diffusa su delle bande di frequenza più larghe.
 - UWC: inviano una serie di pulsazioni a bassa frequenza variando le loro frequenze di trasporto per comunicare informazioni.
-

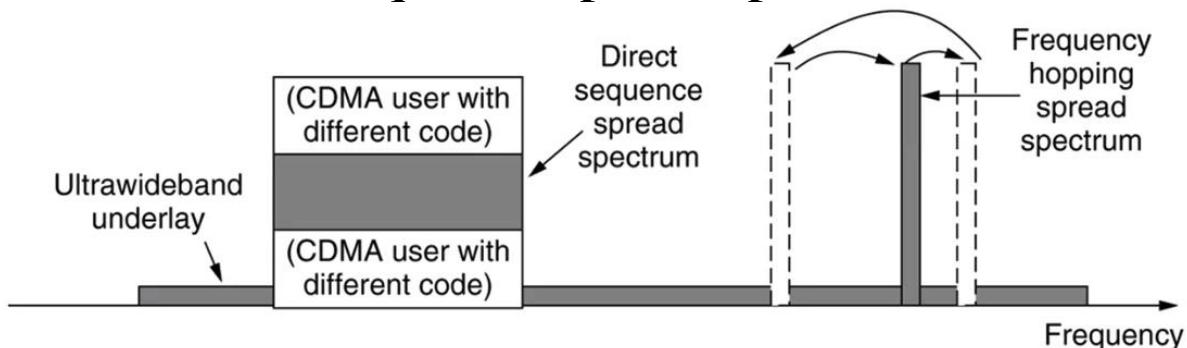
LO SPETTRO ELETTRONAGNETICO



- Raggi gamma
- Raggi x
- UV
- Infrarossi
- Microonde
- Radio

La trasmissione satellitare è quella con qualità migliore.

DSSS (direct sequence spread spectrum)



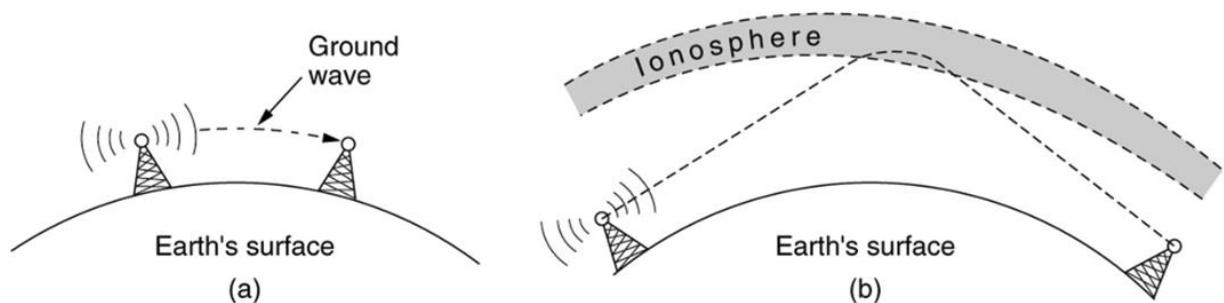
Usa una sequenza codificata per distribuire il segnale su bande con frequenze maggiori. A ogni segnale viene associato un codice diverso chiamato **CDMA**. Questo metodo è usato dal GPS e da connessioni 3G.

USO DELLO SPETTRO PER TRASMISSIONI

- Trasmissioni radio: onde omnidirezionali, facili da generare che lavorano a lunga distanza penetrando gli edifici. (sono le torri alte in strada)
- Trasmissioni microonde: onde direzionali che fanno da *uplink* e non penetrano edifici. Sono utili per portare il segnale da una parte all'altra.

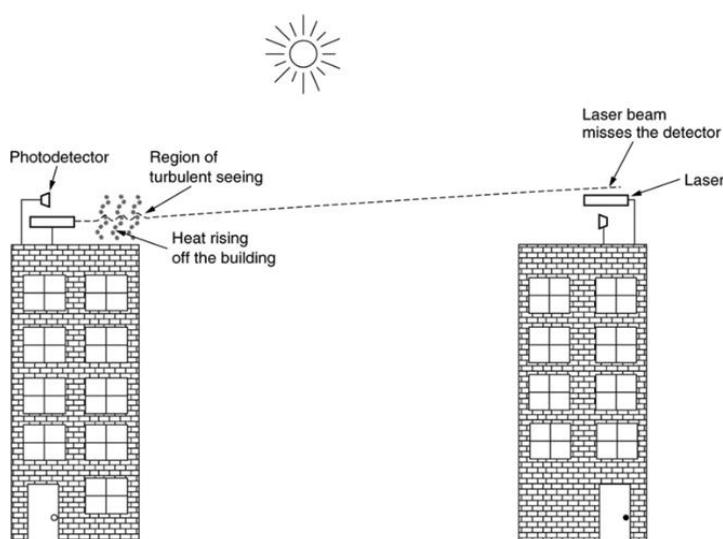
- Infrarossi: un esempio sono i telecomandi. Sono onde non guidate e si usano per comunicazioni a corto raggio, costano poco e sono facili da implementare, non passano mura solide.
 - Trasmissioni luce: sono comunicazioni ottiche non guidate.
-

TRASMISSIONI RADIO



Le onde medie seguono la curvatura della terra mentre le alte anno verso la ionosfera.

TRASMISSIONI LUCE



Correnti convenzionali possono interferire con le comunicazioni laser.

Il calore che sale dalla palazzina interferisce con la trasmissione luce.

La parte atmosferica influenza molto la larghezza di banda, anche per le trasmissioni radio.

Anche le fasi lunari influenzano le comunicazioni.

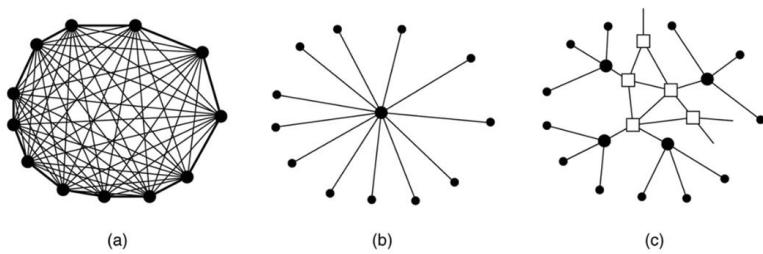
Questo è uno dei motivi per il quale si preferisce la fibra per le connessioni a lungo raggio.

A volte il digitale terrestre se piove non si vede bene.

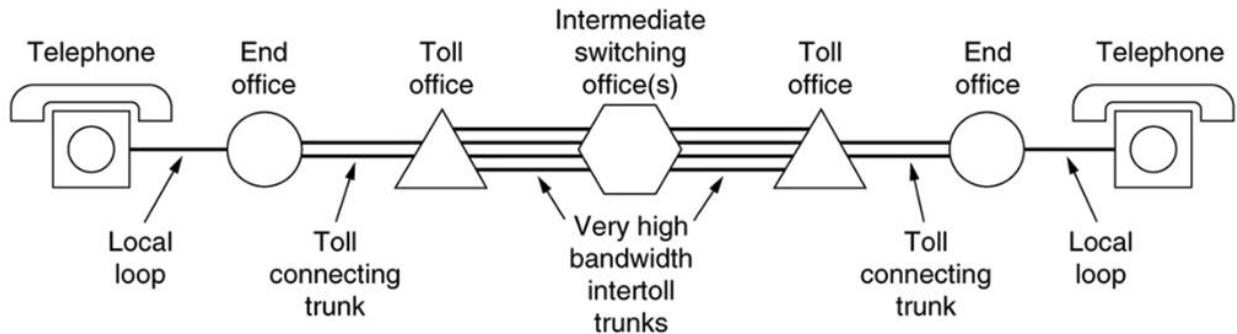
RETI FISSE

PSTN per gli americani. Hanno una componente locale detta local loop che include modem, ADSL e fibra.

STRUTTURA DELLE RETI FISSE



(a) Fully interconnected network. (b) Centralized switch. (c) Two-level hierarchy.



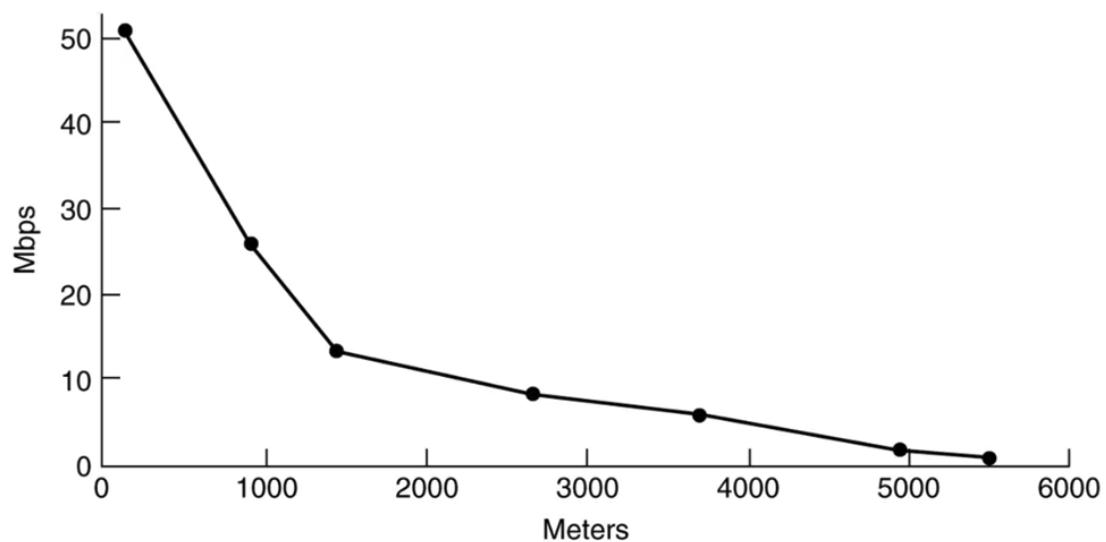
Nell'esempio abbiamo due telefoni in due edifici diversi.

LOCAL LOOP

- Modem telefonici
- DSL
- Fiber to the X (Fibra).

I modem non sono quasi più utilizzati. Ora si usano i router.

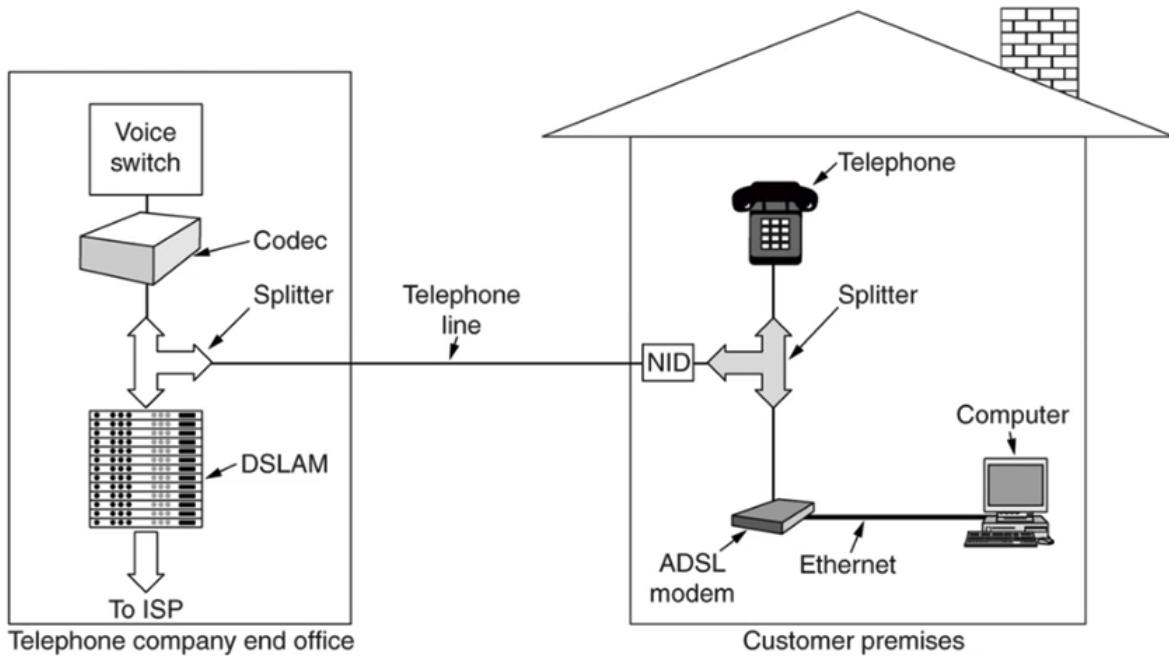
DSL



Più si è distanti dalla centrale più la banda scarseggia.

Ci sono 256 canali di 4kHz, uno per la voce, per l'upstream e per la downstream.

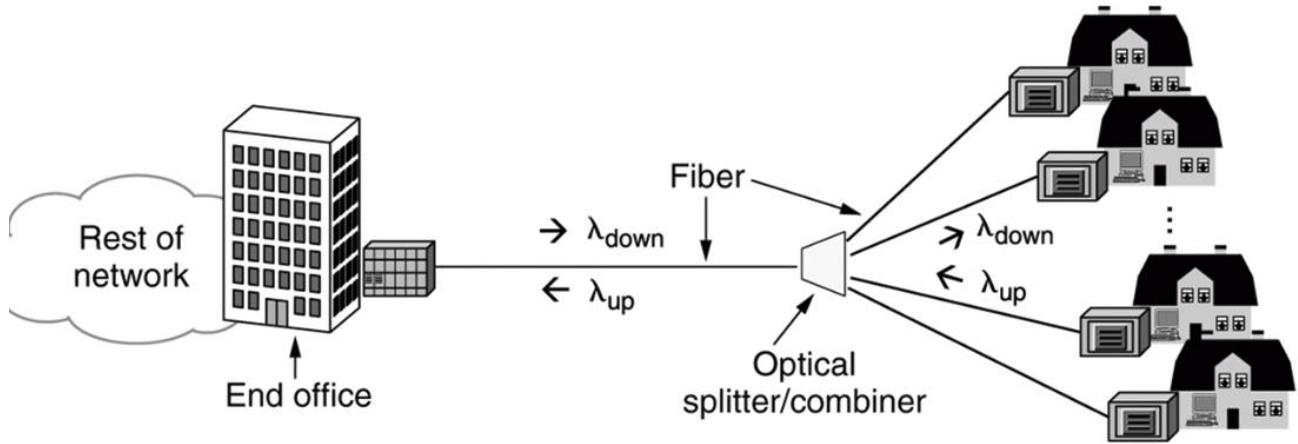
La A in ADSL sta per Asymmetric: ci sono più download che upload.



A typical ADSL equipment configuration.

Nelle centrali ci sono gli apparati DSLAM mentre a casa, tramite linea telefonica abbiamo un telefono collegato all'ADSL.

FIBRA



Ci sono utenti che fanno uplink e downlink.
Nei contratti telefonici bisogna vedere sempre la minima banda garantita che sarà sicuramente quella disponibile a priori.

TRUNKS E MULTIPLEXING

Questo concetto è usato per la digitalizzazione dei segnali voce.

DIGITALIZZAZIONE DEI SEGNALI VOCE

Tecnica TDM:

- Conversione da analogico a digitale.
- Usa un codec per digitizzare segnali analogici.

Il segnale viene ricreato dai sample facendo play sulla parte temporale.

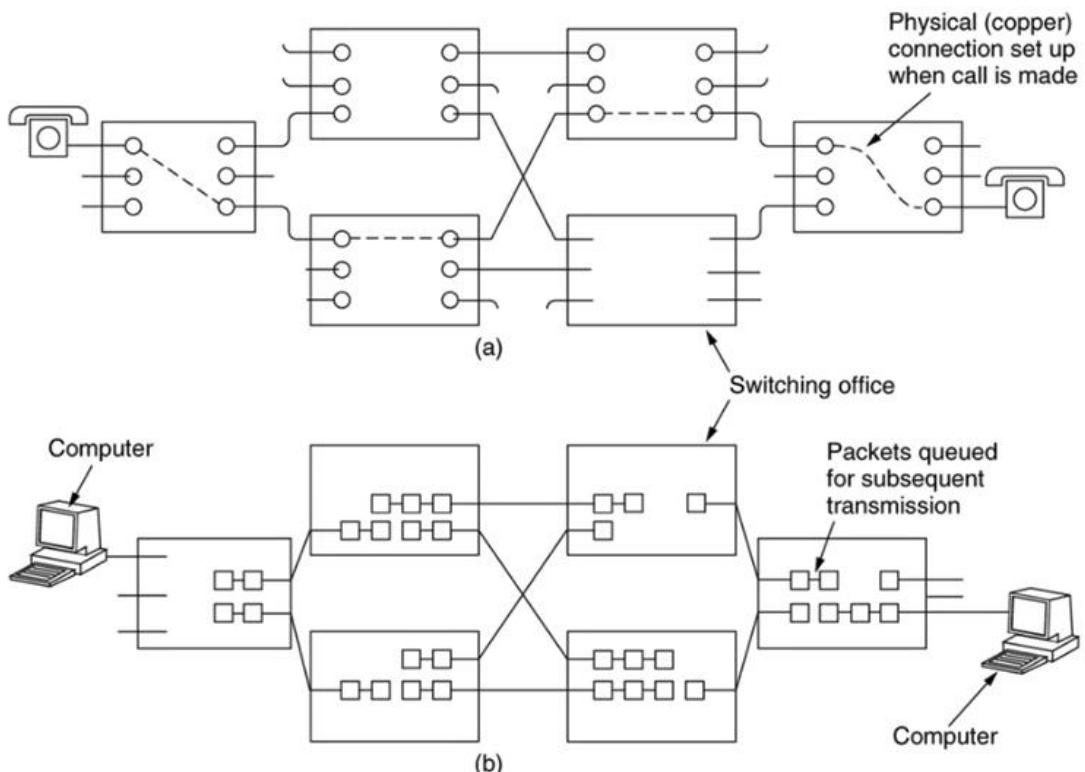
SWITCHING

Le tecnologie di trasmissione sono dipendenti dallo switching.

È una parte principale dei sistemi telefonici.

Attualmente ci sono due tecniche di switching:

- Quella **circuit switching** che si usa nei sistemi telefonici tradizionali, sia fissi che mobili
- **Packet switching**, basata sulla tecnologia VoIP.

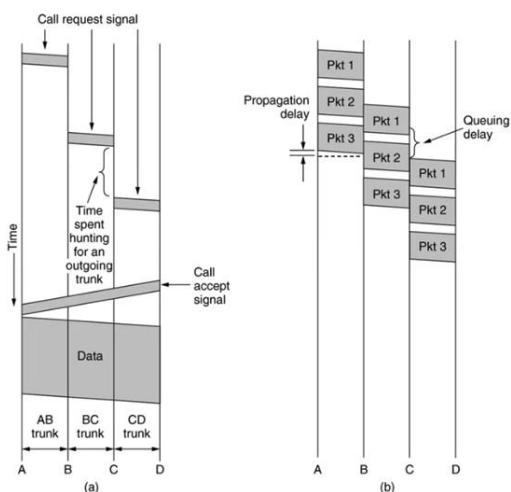


- A) Circuit : Ci deve essere una connessione in rame quando viene effettuata una chiamata da un telefono all'altro.
La connessione attraversa vari circuiti fino al telefono di destinazione.

- Mentre la parte di sotto è un esempio di PacketS: abbiamo 2 pc: i pacchetti possono attraversare una qualsiasi strada random, viene memorizzato un pacchetto da qualche parte per la trasmissione successiva.

Dal punto di vista temporale vediamo che per il CS c'è il segnale di call request che innesca la chiamata e in questa fase c'è il tempo per cercare una linea di uscita e dopo averla trovata la chiamata viene effettuata. Quando il segnale viene propagato dall'altro lato avviene la comunicazione dati.

Quando si parla di PackSwitching il tutto viaggia in modo ordinato, ci possono essere ritardi con delle code. Senza dubbio però è molto più veloce del CS



Timing of events in (a) circuit switching, (b) packet switching.

Item	Circuit switched	Packet switched
Call setup	Required	Not needed
Dedicated physical path	Yes	No
Each packet follows the same route	Yes	No
Packets arrive in order	Yes	No
Is a switch crash fatal	Yes	No
Bandwidth available	Fixed	Dynamic
Time of possible congestion	At setup time	On every packet
Potentially wasted bandwidth	Yes	No
Store-and-forward transmission	No	Yes
Charging	Per minute	Per byte

- Nella parte CS abbiamo una necessità di fare una chiamata mentre nel PS no.
- Nel caso del CS abbiamo bisogno di un percorso fisico (rame) per la circolazione del dato.
- Nel CS ciascun pacchetto segue la stessa strada mentre il PS no, i pacchetti viaggiano random.
- Nel CS i pacchetti arrivano in ordine, nel PS no.
- Se si rompe uno switch nel CS ci sono problemi, nel PS no perché se si rompe ci sarà una rotta alternativa su cui passare
- La larghezza di banda disponibile col CS è fissa, con il PS è dinamica o adattiva
- Nei PS la congestione può avvenire su ciascun pacchetto e c'è uno spreco di banda perché c'è una locazione fissa di banda nei CS.
- Il pagamento per il CS è per minuto, per PS per byte.

RETI CELLULARI

I dispositivi sono distinti per generazioni.

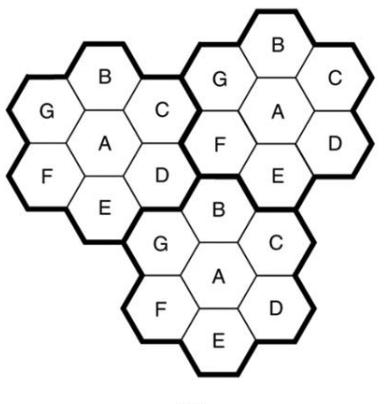
Inizialmente avevamo reti 1G, 2G, 3G che fornivano voce analogica e digitale.

Quando si parla di 4G la parte di CS è sparita, non c'è più la fase di “*call setup*”. Sono state aggiunte tecniche di trasmissione a livello IP e femtocelle.

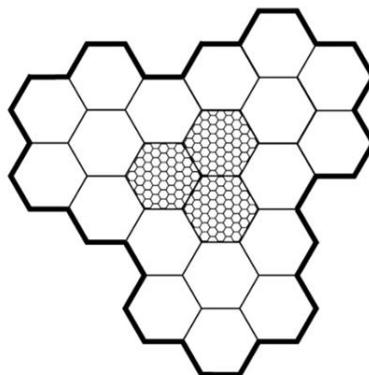
Il 5G si sta espandendo ora con trasmissione di 20 Gigabit.

La latenza è un problema delle reti cellulari.

Common Concepts: Cells, Handoff, Paging



(a)



(b)

(a) Frequencies are not reused in adjacent cells. (b) To add more users, smaller cells can be used.

Le frequenze non sono usate in celle adiacenti (come vediamo in figura “a”). Per estendere la funzionalità delle reti, sono usate più celle.

GESTIONE DELLE CHIAMATE

Chiamate in uscita: accendere il telefono, comporre il numero e chiamare. Il telefono trasmette il numero chiamato e la sua identità al canale di accesso. La stazione base cerca un canale per la chiamata.

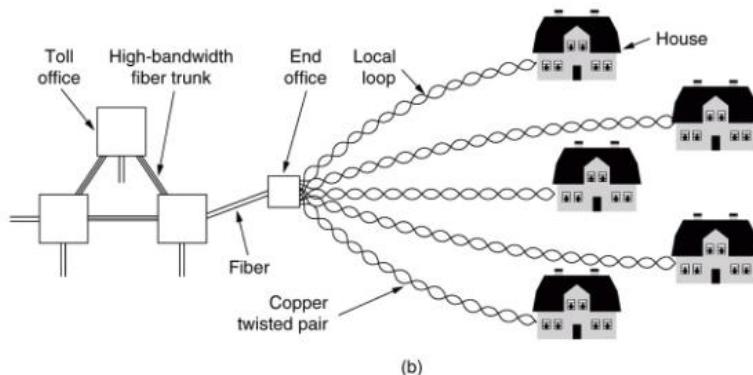
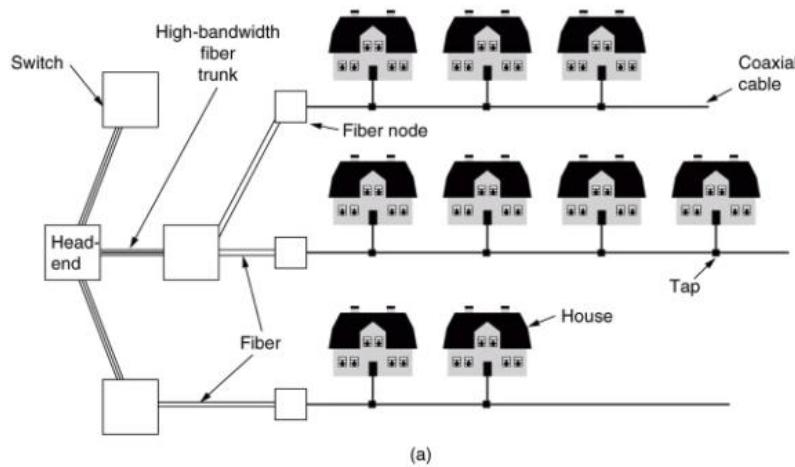
Chiamate in ingresso: telefoni in “idle” continuamente in ascolto per catturare i messaggi diretti a sé stessi. I pacchetti mandati alla stazioni vengono inviati in broadcast sul canale di *paging*. Quando rispondiamo al telefono si passa all’altro canale e si sente lo squillo.

LEZ 5

RETI CABLATE

Sono viste come il futuro delle reti ad accesso a banda larga. Anche la fibra ottica è cablata.

Molte persone avevano servizi tv, telefoni e internet cablati.



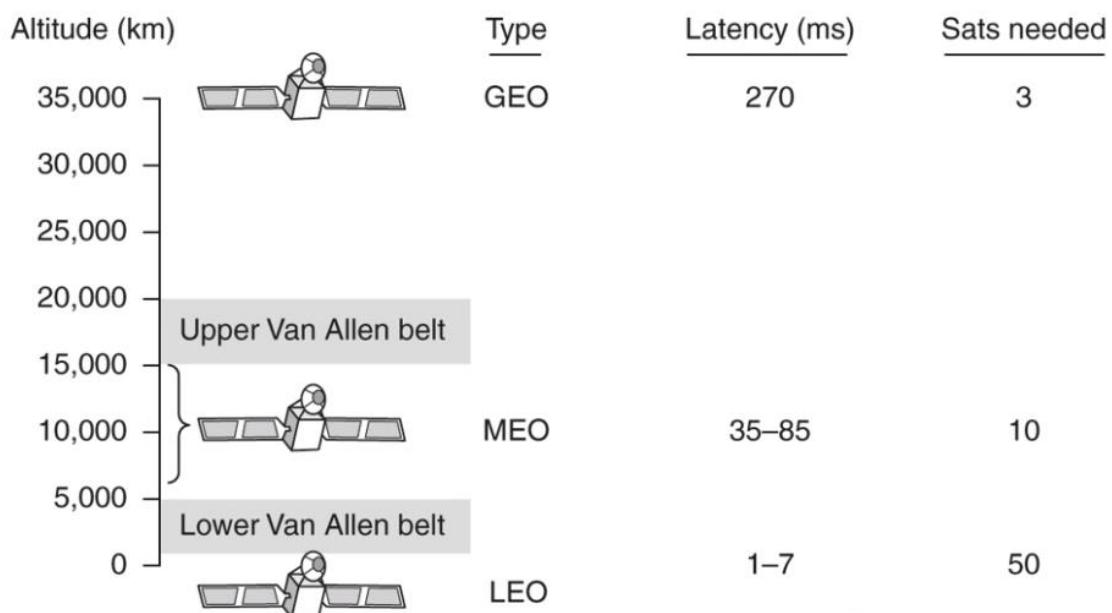
(a) Hybrid Fiber-Coax cable network. (b) The fixed phone system.

Su una frequenza ci sono diversi servizi, abbiamo la parte tv, radio fm e la parte di dati.

DOCSIS

È uno standard per le reti cablate, infatti i possessori di reti cablate devono avere un modem cablato DOCSIS. Dal cavo alla rete di casa abbiamo la connessione Ethernet, i pc non hanno quindi una porta DOCSIS ma Ethernet.

COMUNICAZIONI SATELLITARI



Sono le alternative alle reti in fibra ottica.

La comunicazione satellitare include l'altitudine, il tempo di andata e ritorno (latenza) e quanti satelliti servono per una copertura efficiente e globale. I satelliti più in alto hanno maggiore latenza ma ne servono meno per una copertura totale.

SATELLITI GEOSTAZIONARI

Le principali bande satellitari:

Band	Downlink	Uplink	Bandwidth	Problems
L	1.5 GHz	1.6 GHz	15 MHz	Low bandwidth; crowded
S	1.9 GHz	2.2 GHz	70 MHz	Low bandwidth; crowded
C	4.0 GHz	6.0 GHz	500 MHz	Terrestrial interference
Ku	11 GHz	14 GHz	500 MHz	Rain
Ka	20 GHz	30 GHz	3500 MHz	Rain, equipment cost

Ku è usato anche da *Sky*

SATELLITI MEO

Si trovano ad altitudini basse, sono lenti in longitudine, devono essere controllati movimento per movimento, hanno una piccola copertura e richiedono minore potenza trasmissiva per essere raggiunti. Solitamente sono usati per sistemi di navigazione.

SATELLITE LEO

Ciascun satellite ha 4 vicini.

Iridium è un provider di reti satelliti *LEO*.

RETI DI ACCESSO TERRESTRI: CAVO, FIBRA E ADSL

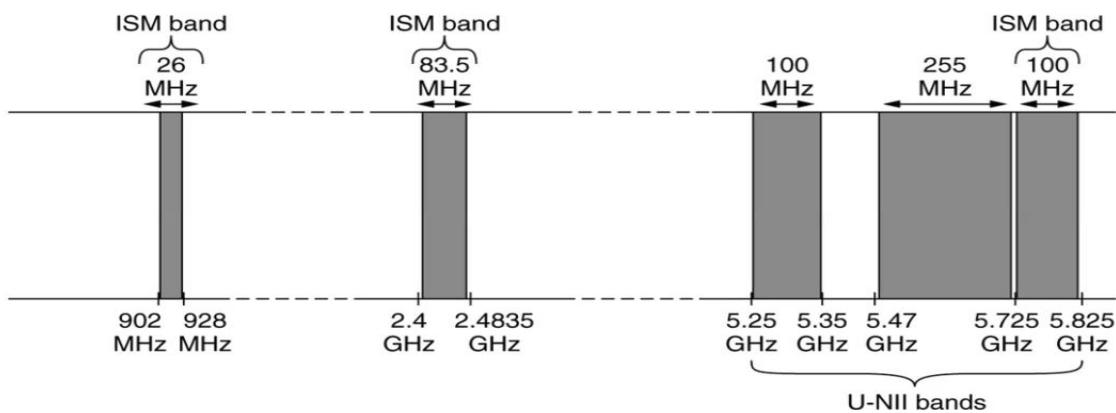
- Similarità:
 - Forniscono servizi simili
 - Hanno prezzo simili

- Usano la fibra per quanto riguarda le backbone.
 - Differenze:
 - impattano di più sull'ultimo miglio (0 km).
 - La velocità di banda è differente
 - Differiscono in massima velocità
 - Disponibilità
 - Sicurezza
-

RETI TERRESTRI VS SATELLITARI

Le reti satellitari sono ancora poco diffuse, anche se sono facili da implementare. Sono piazzate dove le infrastrutture terrestre sono poco sviluppate. Sono molto utili quando il broadcasting è essenziale. Le reti satellitari stanno avendo un crescente interesse circa l'accesso Internet durante i voli aerei.

ALLOCAZIONE DELLO SPETTRO DI BANDA



ISM and U-NII bands used in the United States by wireless devices.

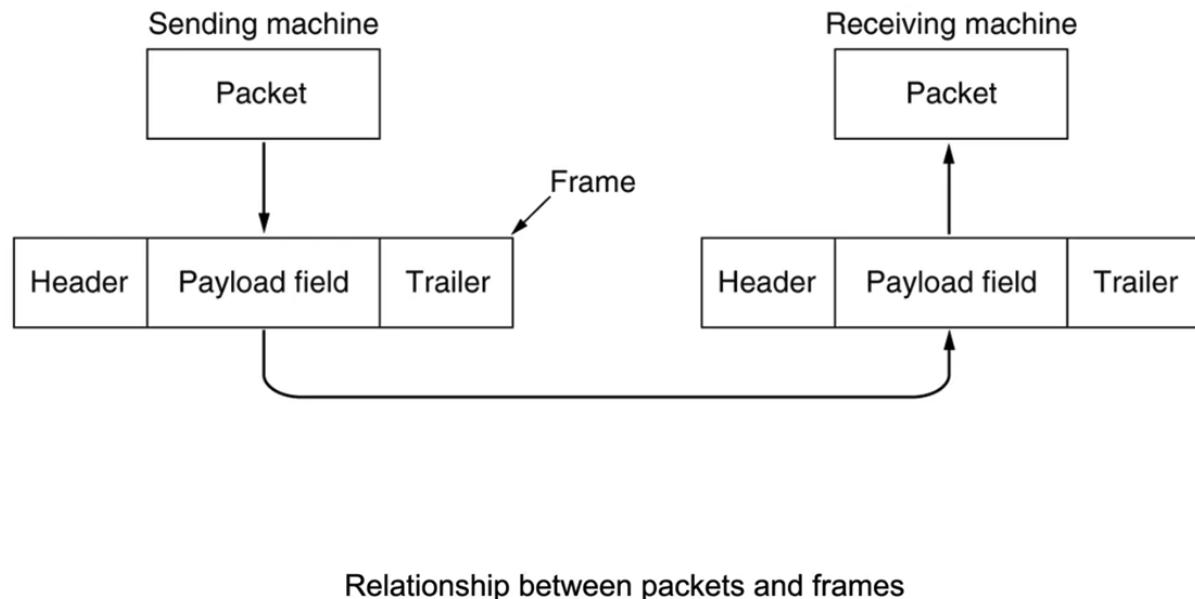
L'allocazione dello spettro di banda è gestito dalle nazioni.

RETI CELLULARI

Ci sono decisioni politiche e di marketing che impattano sulla crescita delle reti cellulari.

Capitolo 3

DATA LINK LAYER DESIGN ISSUES



Abbiamo una macchina che invia info e a una che le riceve. Un pacchetto viene incapsulato in frame che diventa un campo del *payload* e viene messo un

header e un trailer che viene trasportato alla macchina ricevente.

Consente comunicazioni affidabili ed efficienti tra due macchine adiacenti (connesse da cavo coassiale, doppino, linea telefonica). I protocolli del livello DL devono tener conto che:

- Ci sono errori e disturbi occasionali.
- Il canale ha una *data rate finito*.
- C'è un ritardo nella propagazione.

Il percorso reale può cambiare da protocollo a protocollo. Ci sono 3 servizi:

- Servizi senza ACKNOWLEDGE e senza connessione
 - Servizi senza connessione ma con ACK
 - Connection-oriented con ACK
-

CONTROLLO DEGLI ERRORI

Tutti i *frame* sono consegnati a livello di rete e devono arrivare in modo corretto.

Richiede il frame di ACK e uso di timer.

CONTROLLO DEL FLUSSO

Se ci sono troppi dati potrebbero perdere.

Bisogna controllare che l'invio dei frame di trasmissione siano a un ritmo accettabile dal ricevente.
Si può fare in due modi:

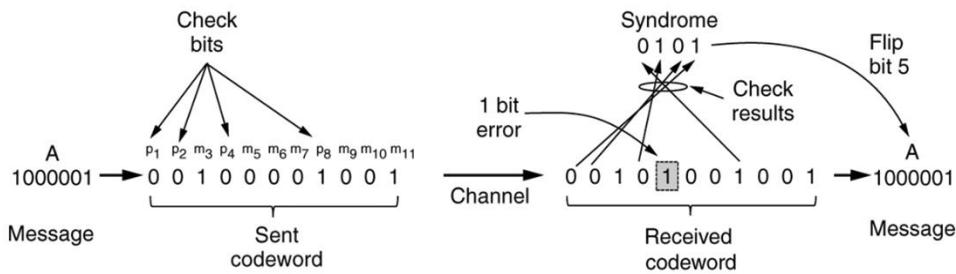
- Basato su feedback: chi riceve invia delle info a chi ha inviato dandogli il permesso di mandare altri dati oppure ricevente informa quello che invia sul suo stato
 - Basato su rate: è implementato a livello interno dei protocolli. Chi invia può trasmettere i dati e non c'è necessità che il ricevente dia un feedback.
-

ERROR DETECTION E CORREZIONI

- Codici a correzione di errore: sono meccanismi che includono info aggiuntive in modo che chi riceve può dedurre qual è stato il dato realmente trasmesso. Se c'è un errore il meccanismo grazie a info ridondanti ricava il dato perso.
 - Ci sono meccanismi che capiscono che c'è stato un errore MA non quale e quindi si deve richiedere la ritrasmissione.
-

CODICI DI CORREZIONE DI ERRORE

HAMMING:



Abbiamo il messaggio $A \rightarrow 1000001$ e vengono aggiunti 4 bit ai 7 di base. Questi vengono aggiunti a un canale che potrebbe avere un errore (ad esempio m_5 da 0 diventa 1) e grazie al meccanismo il sistema lo riconverte e il messaggio A viene recepito correttamente, dopo aver eliminato i 4 bit aggiunti prima.

STRUTTURA DATI DI UN FRAME

Un frame ha un campo *kind*, *seq* e *ack* racchiusi in un *HEADER* e poi un campo *INFO*.



- Kind: tipo di frame (dati, controllo).
- Seq: numero progressivo del frame.
- Ack: info legate all'*acknowledgement*.
- Info: pacchetto di livello Network completo.

Lez 6

CODICE DI CORREZIONE D'ERRORE

Si usano poiché ci sono elementi che modificano le prestazioni come ad esempio le condizioni atmosferiche.

Ci sono codice di correzione diversi a seconda dei metodi di comunicazione (fibra, satellite, cavo, ecc...).

La discesa della qualità non sempre influenza ciò che l'utente guarda, come nel caso delle TV satellitari.

Altro codice a correzione d'errore è quello di *Reed-Solomon e LDPC*. Sono entrambi codici a blocchi lineari, il secondo è utile quando ci sono frame di grossa taglia e sono i migliori poiché funzionano meglio degli altri codici.

Ci sono anche codici di errore che non si usano per le trasmissioni di dati a livello di reti ma altri che si possono usare per altri contesti come ad esempio i codici *fontana*: sono usati in contesti dove è fondamentale la ridondanza delle informazione, non è possibile richiedere la ritrasmissione come l'invio di segnali nascosti nello spam e il ricevente non può richiedere al mittente di rinviarlo, sono strutturati in modo che se si perdono messaggi sono facilmente ricostruibili.

Il detecting ovviamente segnala gli errori:

- Parità
- Checksum
- CRCs

Sono metodi di somma che permettono la risoluzione di errori.

Esempio di detecting con bit di parità

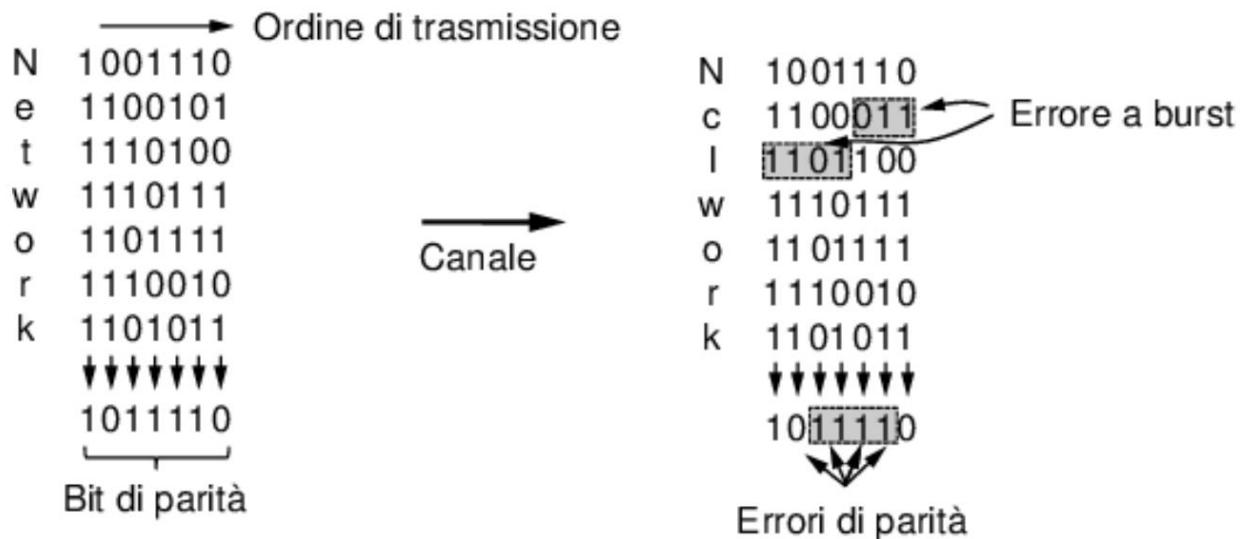


Figura 3.8 Interleaving dei bit di parità per la rilevazione di un errore a burst.

Grazie a ciò si capisce che c'è stato un errore.

ELEMENTI DI BASE DEI PROTOCOLLI DATA LINK

Le assunzioni si basano sui modelli di comunicazione.
Ci sono 3 protocolli:

1.Utopia: non c'è bisogno del flow control e dell'error correction

2.Aaggiunta del controllo di flusso col meccanismo di fermare la trasmissione e aspettare

3.Aaggiunta della correzione dell'errore

Chi gestisce il dato viaggiante sono processi indipendenti: c'è un processo che fa qualcosa, un altro che fa altro e così via e la comunicazione avviene passando i messaggi tra i vari livelli.

La comunicazione è monodirezionale e le macchine usano servizi connection-oriented.

Questa ovviamente è un'assunzione: le macchine non crashano (ovviamente non è vero, spesso è il contrario).

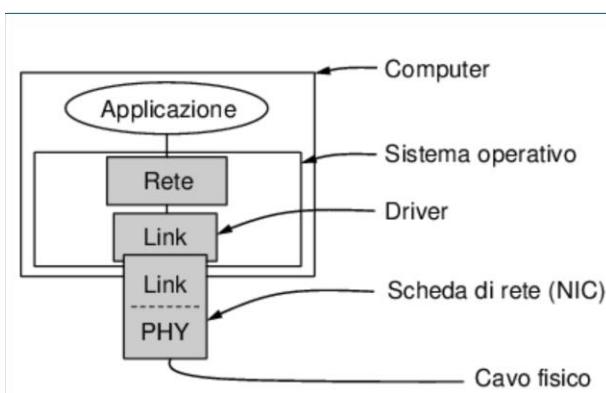


Figura 3.10 Implementazione dei livelli fisico, data link e di rete.

Esistono diversi moduli che se stanno sul livello di rete vengono chiamate applicazioni, se sono sotto

vengono chiamate SO, driver ecc... Ma sono comunque applicazioni.

Anche le schede di rete hanno del SW interno (*firmware*).

Ci interessa fare trasmissioni di dati bidirezionali e migliorare l'efficienza del *link layer*: possiamo inviare simultaneamente frame anche prima di aver ricevuto l'ACK.

TRASMISSIONE BIDIREZIONALE E FRAME MULTIPLO

- Trasmissioni bidirezionali: *piggybacking (a cavalluccio)*: si usa lo stesso link per i dati in entrambe le direzioni
 - Si aggiunge un certo delay sugli ACK che vengono dati in uscita: il pacchetto di ACK può arrivare anche nel frame successivo e chi deve ricevere l'ACK è disposto ad aspettare perché sa che arriverà nel prossimo frame.

I vantaggi di questa tecnica (*piggynacking*) sono una gestione migliore della larghezza di manda e minore carico sulla gestione del processo dal lato del ricevente.

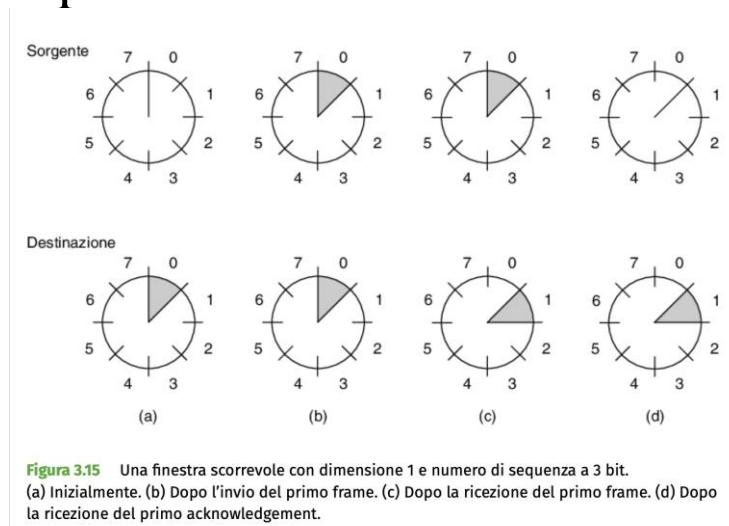
Lo svantaggio del piggybacking è stabilire il giusto tempo in cui il link layer deve aspettare l'ACK. Se l'ACK non arriverà mai ci sarà ovviamente un malfunzionamento.

CONCETTO DI FRAMING

In ogni istante di tempo succedono 3 cose:

- Chi invia ha a sua disposizione dei numeri di sequenza che corrispondono ai frame consentiti da inviare.
- I frame si possono inviare solo in una finestra temporale (sliding window)
- Chi riceve mantiene una finestra di ricezione che corrisponde a un insieme di set che può accettare.

In breve chi **invia** ha un elenco dei frame che può mandare, i **frame** si possono mandare in un tempo specifico e chi **riceve** mantiene aperta una finestra di ricezione che corrisponde ai frame che gli è consentito aspettare.



Capitolo 4

IL SOTTOLIVELLO MAC

È il sottolivello del livello DL.

Capiremo come funziona il sottolivello che controlla l'accesso al mezzo trasmissivo.

Il problema delle reti broadcast è a quale stazione assegnare il mezzo trasmissivo in caso di più richieste simultanee. È compito dei protocolli del sottolivello MAC decidere chi deve essere il prossimo a trasmettere su un canale broadcast.

I protocolli MAC possono assegnare i canali attraverso un'allocazione:

- Statica: l'allocazione è fissata in anticipo
- Dinamica: allocazione in base alle esigenze del momento.

IL PROBLEMA DELL'ALLOCAZIONE DEL CANALE

Il canale può essere su un mezzo fisico o radio.

Ci sono due allocazioni: statiche e dinamiche.

ALLOCAZIONE STATICÀ

Non funziona bene quando c'è un traffico con picchi molto alti e le performance sono negative quando il traffico più alto rispetto al traffico medio ha un rapporto di 1000:1. La maggior parte dei canali

potrebbero essere in attesa per la maggior parte del tempo. Tutti i sistemi fanno traffico di punta, tranne i dispositivi IOT.

Ci sono problemi da risolvere e l'allocazione dinamica del canale prova a risolvere i problemi di cui sopra.

Di fatto l'allocazione statica non viene utilizzata.

LE ASSUNZIONI PER L'ALLOCAZIONE DINAMICA DEL CANALE

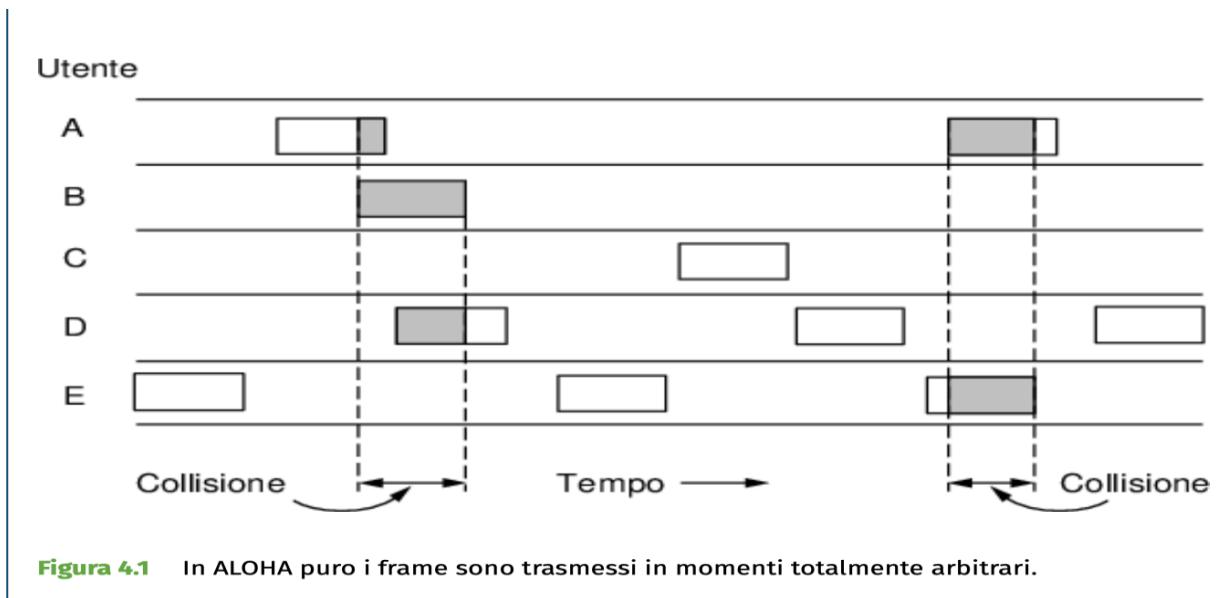
Si assume che il traffico sia indipendente e che c'è un solo canale. Si può capire se c'è il sensing (capire se siamo connessi su quel canale) e osservare le collisioni.

PROTOCOLLI PER ACCESSO MULTIPLO

- ALOHA
- Carrier Sense Multiple Access
- Wireless LAN Protocol

ALOHA

Non si usa molto. ci sono più utenti che condividono un mezzo trasmissivo e mandano dei frame, man mano che si va avanti nel tempo i pacchetti viaggiano e si possono generare molteplici collisioni (i pacchetti viaggiano in contemporanea).



Il throughput decresce quando si generano collisioni:

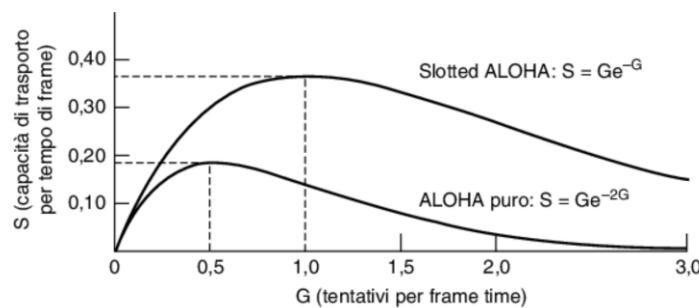


Figura 4.3 La capacità di trasporto in funzione del traffico per un sistema ALOHA.

CARRIER SENSE MULTIPLE ACCESS (CSMA)

Ci sono diverse versioni:

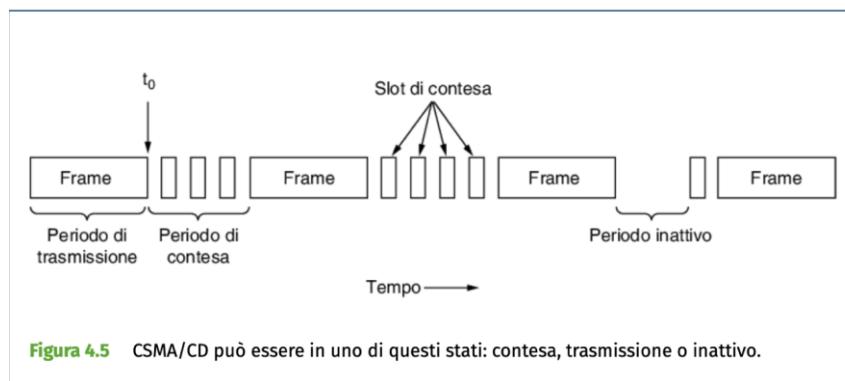
- Persistente a 1: quando una stazione deve trasmettere, ascolta il canale: se è occupato aspetta e poi trasmette, se è libero trasmette.
- Non persistente: quando una stazione deve trasmettere, ascolta il canale: se è occupato attende che si liberi, quindi aspetta un tempo random e ripete il procedimento. Se è libero trasmette.

- Persistente a P: ascolta il canale: se è occupato aspetta il next slot e ricomincia. Se è libero: con probabilità p trasmette subito; con probabilità $1-p$ aspetta il prossimo slot e ricomincia.

Quella che usiamo al giorno d'oggi per le reti Ethernet LAN è il **CSMA/CD** dove CD sta per *collision detection*.

Vediamo come funziona la parte di Collision Detection che può essere usata in trasmissione, in stato di idle e condivisione del mezzo.

C'è un periodo di trasmissione del frame, c'è poi un periodo di contesa: la scheda di rete cerca di capire se c'è qualcun altro sulla rete, dopo il controllo il frame viene inviato e viaggia ma viene rifatto il controllo della collision:



WIRELESS LAN PROTOCOLS

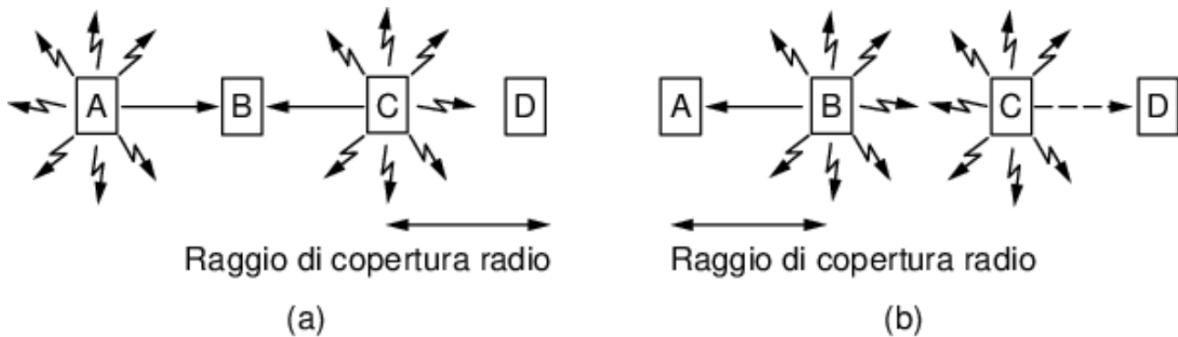


Figura 4.11 Una LAN wireless. (a) A e C sono terminali nascosti quando trasmettono a B. (b) B e C sono terminali esposti quando trasmettono ad A e D.

Una variante è il protocollo MACA dove il nodo A invia un messaggio di RTS (pronto a inviare) a B e il nodo B risponde con un CTS (Clear To Send) che è pronto a ricevere.

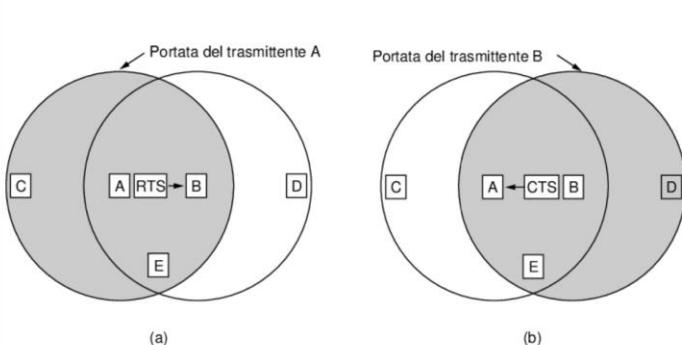


Figura 4.12 Il protocollo MACA. (a) A invia un frame RTS a B. (b) B risponde ad A con un frame CTS.

ARCHITETTURA CLASSICA ETHERNET

Ogni computer ha un'interfaccia di rete che comunica sul mezzo.

Il *Transceiver* è un dispositivo non più utilizzato.

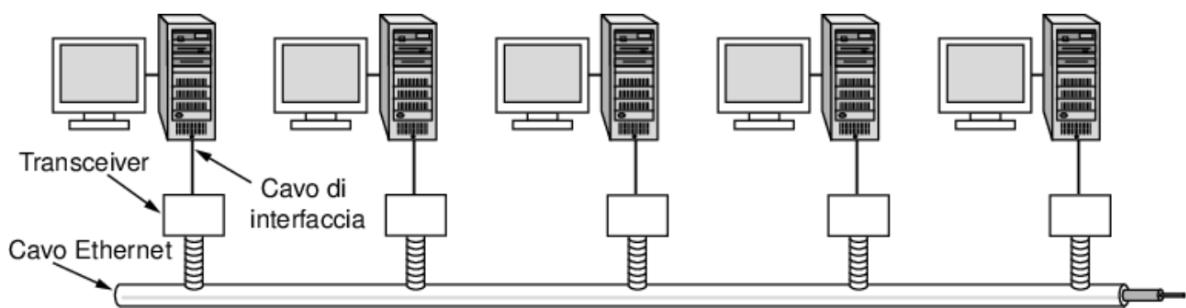


Figura 4.13 Architettura di Ethernet classica.

CLASSIC ETHERNET MAC SUBLAYER PROTOCOL

Byte	8	6	6	2	0-1500	0-46	4	
(a)	Preamble	Indirizzo di destinazione	Indirizzo d'origine	Type	Dati » »	Pad	Check-sum	
(b)	Preamble	S o F	Indirizzo di destinazione	Indirizzo d'origine	Length	Dati » »	Pad	Check-sum

Figura 4.14 Formati di frame. (a) Ethernet DIX. (b) IEEE 802.3.

Dettagli che non ci interessano (...).

C’è un preambolo, un indirizzo di destinazione, di sorgente, un campo lunghezza, dati (va da 0 a 1500 che è il valore maggiore che si può trasmettere in un’Ethernet).

PERFORMANCE DELL’ETHERNET

A seconda del numero di stazioni e dell’efficienza del canale le prestazioni variano.

L'efficienza è più alta quando il numero di stazioni che trasmettono è più basso.

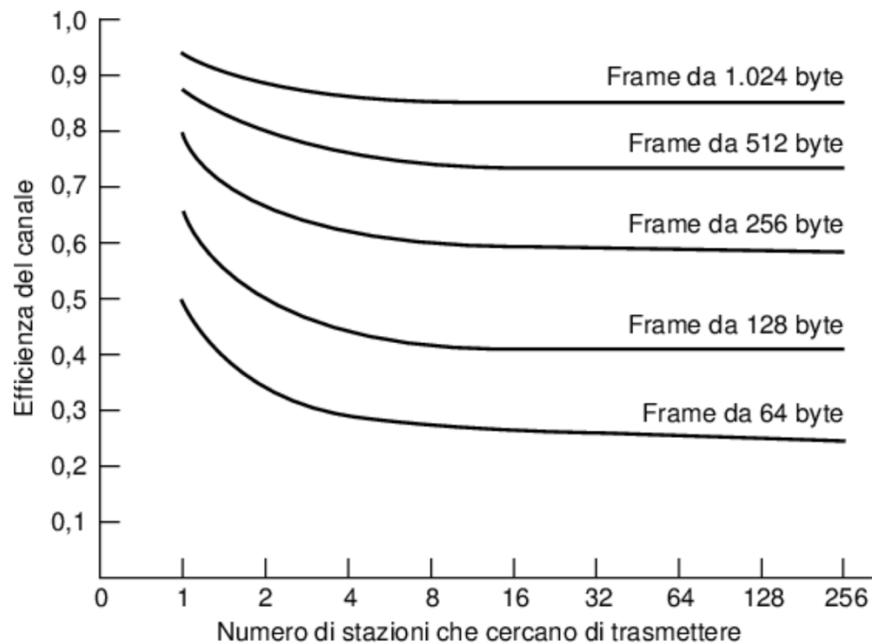
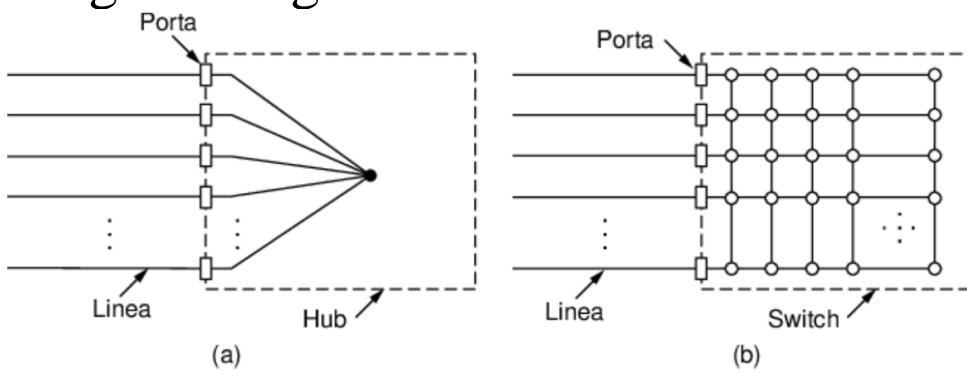


Figura 4.16 Efficienza di Ethernet a 10 Mbps con slot da 64 byte.

SWITCHED ETHERNET

- Parte Hub: Abbiamo un certo numero di linee che confluiscono in delle porte che vengono immesse in un unico Hub.
- Tecnologia switch: le linee e le porte sono ben schematizzate e ordinate. È implementata per gestire i guasti.



GIGABIT EHTERNET

C’è un meccanismo di interconnessione con switch o hub che sono interconnessi con altri switch e tra di loro sono connessi su più piani.

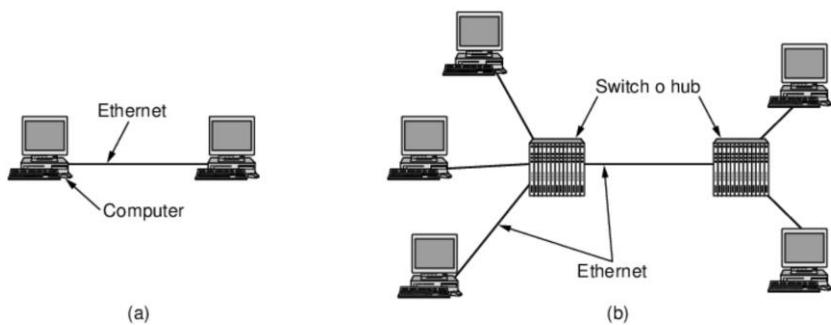


Figura 4.20 (a) Una Ethernet a due stazioni. (b) Una Ethernet a più stazioni.

Nome	Cavo	Lunghezza max. segmento	Vantaggi
1000Base-SX	Fibra ottica	550 m	Fibra multimodale (50, 62,5 mm)
1000Base-LX	Fibra ottica	5.000 m	Fibra mono (10 mm) o multimediale (50, 62,5 mm)
1000Base-CX	2 coppie di STP	25 m	Doppino schermato
1000Base-T	4 coppie di UTP	100 m	UTP standard in Categoria 5

Figura 4.21 Cablatura gigabit Ethernet.

La fibra funziona meglio.

10-GIGABIT ETHERNET

Attivano fino a 40 km di lunghezza.

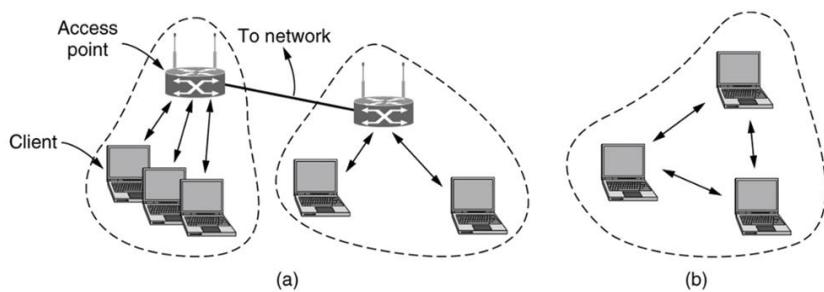
Le reti MAN (Metropolitan Area Network) usano questa rete.

Ci sono differenze sostanziali in ambito di massimo segmento.

Lez 7

WIRELESS LANs → 802.11

Ci sono diversi client che si collegano all'access point, questa modalità si chiama infrastruttura.
Se collego pc tra loro tramite Wi-Fi si dice modalità ad-hoc.



Quando attiviamo il wireless nei nostri dispositivi mobile, viene caricato il driver nel kernel anche se nei dispositivi moderni è integrato. Dopo, fatto ciò, viene messo su il livello di rete e il dispositivo guarda all'interno di un DB locale se tra tutte le reti salvate c'è un Access Point che sta irradiando via wireless una connessione. Infatti i SO per evitare di inserire pswd a ogni collegamento, salvano pswd e rete dopo il primo accesso.

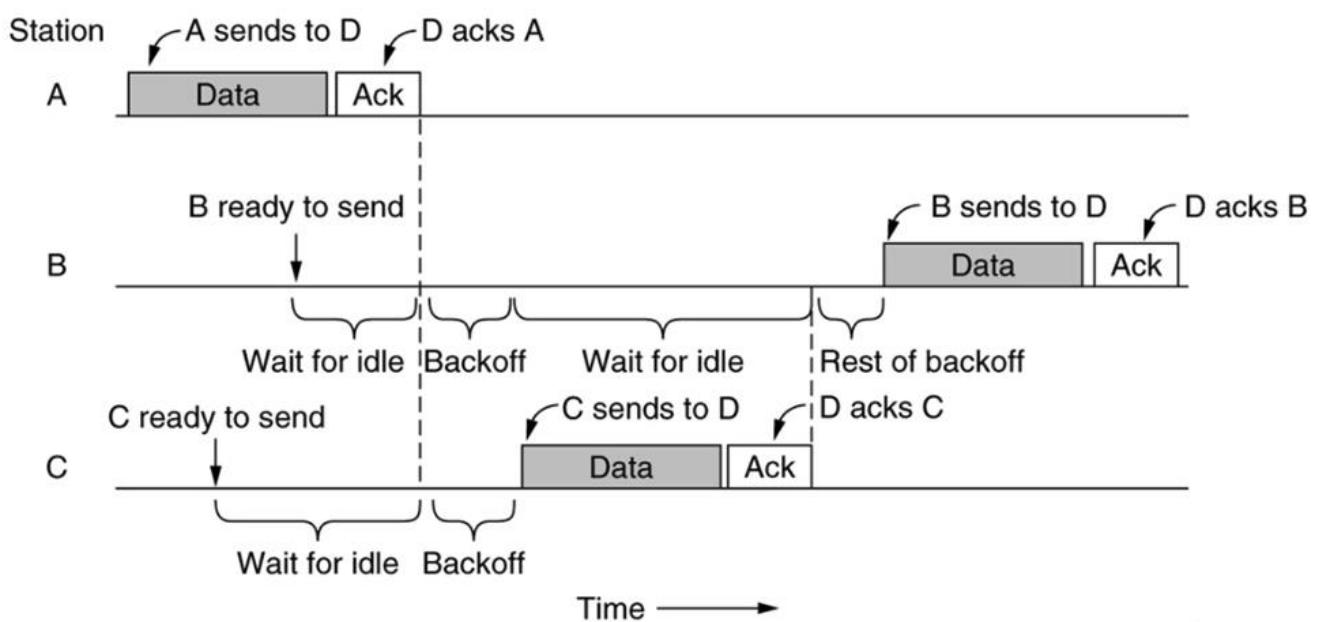
Questa funzionalità è comodo ma pericoloso: i sistemi wireless non sono molto smart: non appena vedono una rete nell'elenco salvato si collegano in automatico e passa le credenziali.

802.11 MAC SUBLAYER PROTOCOL

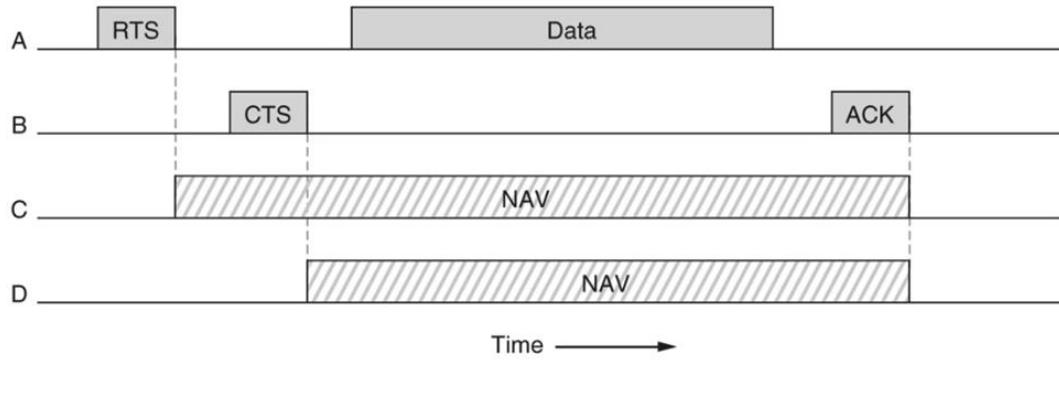
Il CSMA/CA (*collision avoidance*) è un meccanismo semplice. Esempio:

Ci sono 3 stazioni: A manda dati a D, lo stesso B e C. Chi comincia a trasmettere, A, manda i dati, resta in attesa dell'ACK da parte di D e nel frattempo B e C erano pronti e hanno atteso senza far nulla (idle).

Appena capiscono che qualcuno sta trasmettendo aspettano e appena il canale torna libero potrebbero trasmettere insieme ma si genererebbe una collisione e per ovviare ciascuna stazione che era pronta a trasmettere, introduce un ritardo casuale (*backoff*), ad esempio nell'esempio la stazione C ha un *backoff* minore rispetto a B e trasmette prima. B oltre al backoff deve anche aspettare la fine di C, c'è di nuovo una piccola backoff e poi riesce a trasmettere a D.



Oltre ad avere un sensing fisico, si ha il concetto di NAV: quando una stazione trasmette, dice anche per quanto tempo trasmetterà, in modo che gli altri sapranno per quanto tempo stare in idle.



Virtual channel sensing using CSMA/CA

LA STRUTTURA DEL FRAME 802.11

Ha una parte di 11 bit, con vari campi.

Ci sono 3 indirizzi, oltre a chi manda e chi riceve, c'è anche l'Access Point o un intermediario qualsiasi.

Bytes	2	2	6	6	6	2	0–2312	4			
Bits	Frame control	Duration	Address 1 (recipient)	Address 2 (transmitter)	Address 3	Sequence	Data	Check sequence			
	Version = 00	Type = 10	Subtype = 0000	To DS	From DS	More frag.	Retry	Pwr. mgt.	More data	Protected	Order

Tutto ciò serve a darci *SERVIZI*:

- Il protocollo 802.11 accetta connessioni e associazioni e gira i dati a chi di dovere.
- Sicurezza e privacy
- Posso dare priorità ai servizi e controllare la potenza trasmissiva per evitare problematiche, come ad esempio quando ci sono condizioni atmosferiche sfavorevoli.

Lez 8

BLUETOOTH

È un protocollo molto usato. A seconda delle versioni arriva a un certo raggio.

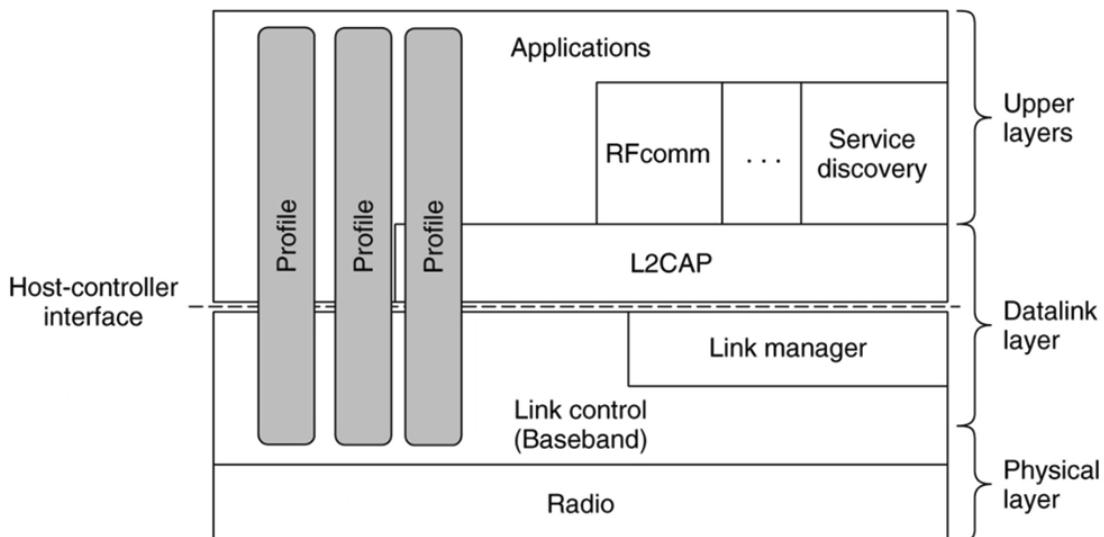
Vengono chiamate Personal Area Network.

Al momento siamo alla versione Bluetooth 5.

È un protocollo data link. A livello radio trasporta “qualcosa”: audio, file, connessione remota ecc...

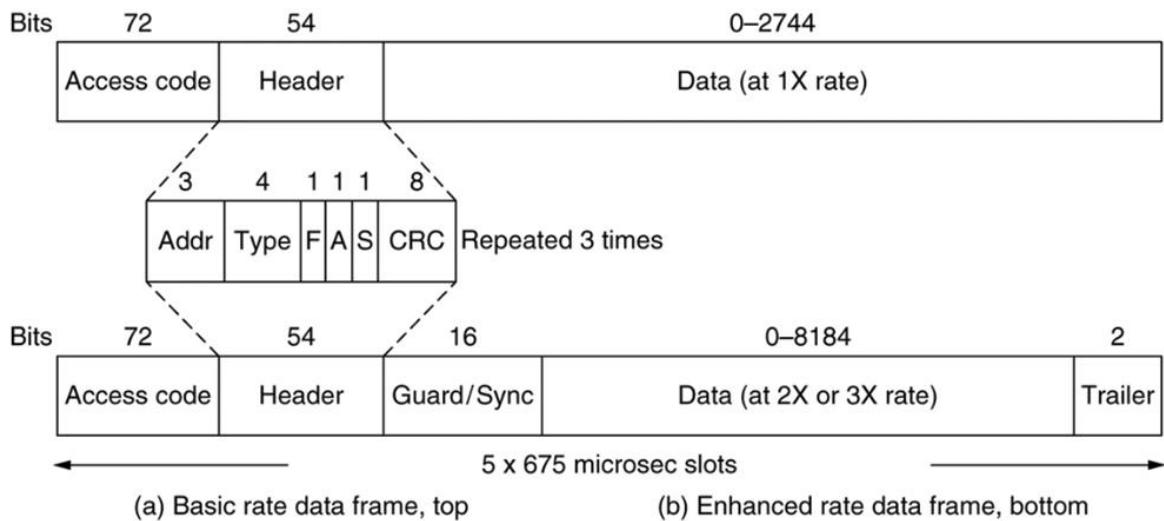
ARCHITETTURA

Abbiamo un trasmettitore al centro (master) e vari client (slave) e uno Bridge Slave che fa da ponte tra due reti.



The Bluetooth protocol architecture

The Bluetooth Frame Structure



Typical Bluetooth data frame at (a) basic and (b) enhanced data rates.

BLUETOOTH 5

Supporta i dispositivi IOT.

La velocità è passata da 1MB a 2MB.

Il range è arrivato fino a 40 metri.

Il consumo energetico è diminuito ed è aumentato il raggio d'azione. Inoltre è stata aumentata la sicurezza.

STANDARD DOCSIS

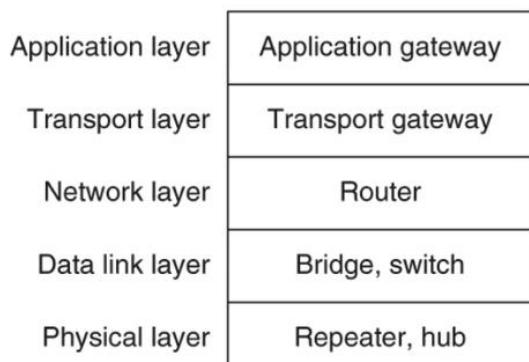
Il sottolivello MAC si occupa di fare l'allocazione del canale, della qualità del servizio.

Per quanto riguarda il ranging c'è stato un particolare improvment.

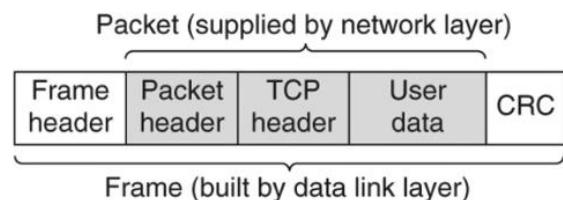
Tutti i meccanismi di comunicazione, in particolari quelli radio, vanno a influire sulla bassa latenza che è un problema quando c'è necessità di fare comunicazioni in real time e affidabili.

Il docsis non è usato in Italia.

RIPETITORI, HUBS, SWITCH, ROUTER, GATEWAY



(a)



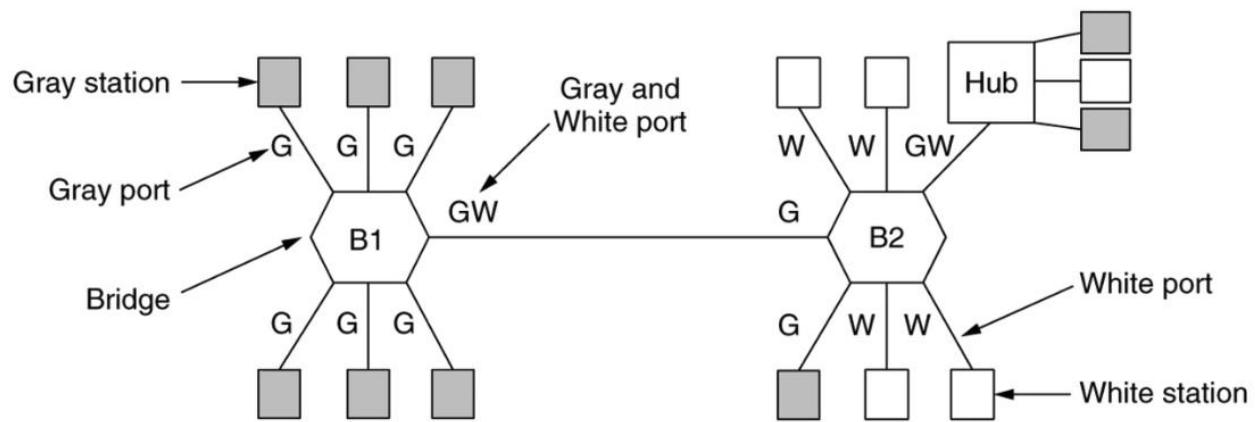
(b)

VLAN standard 802.1Q

Tutti i dispositivi sono connessi a una presa e vengono collegate a uno switch di piano che consentono di creare dei percorsi virtuali (reti locali virtuali) che non si vedono tra loro.

Ci sono due bridge B1 e B2 e due VLAN (bianco e grigie): nonostante siano su 2 reti diverse vengono viste come una sola rete.

La VLAN estende la rete locale e crea dei segmenti isolati. Solitamente si creano VLAN per evitare uso non autorizzato di blocchi di rete.



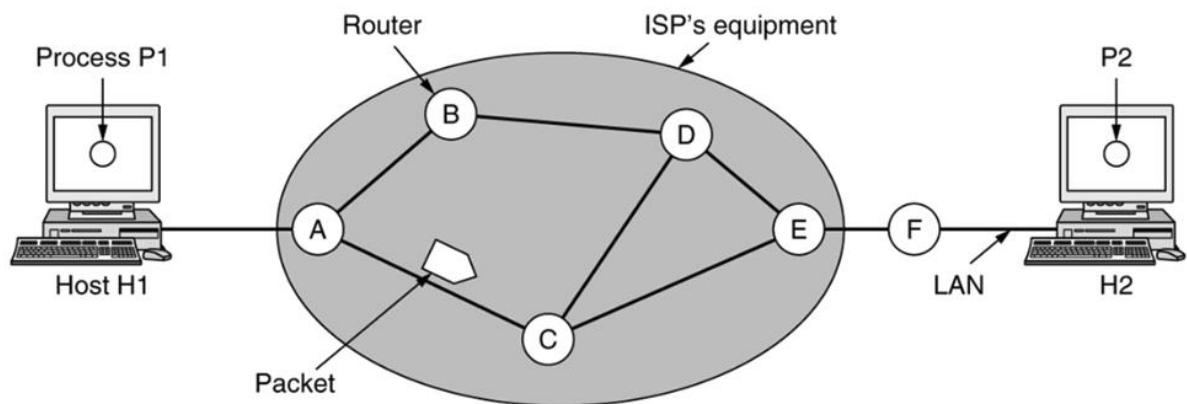
Lez 9

NETWORK LAYER

Garantisce il funzionamento di Internet.

Ci sono diverse problematiche:

- Abbiamo un insieme di router che prendono info e la devono instradare ma prima di farlo la memorizzano.



Un processo P1 sta su un host e deve comunicare con un altro host altrove, magari su una LAN.

I computer comunicano grazie ai processi che parlano dei protocolli ed erogano dei servizi.

La “A” potrebbe essere un router di casa che a sua volta è interconnessa con vari dispositivi. Il router F consegna il pacchetto all’host2 con P2.

Ogni router, prende il pacchetto, lo memorizza.

SERVIZI A LIVELLO DI TRASPORTO

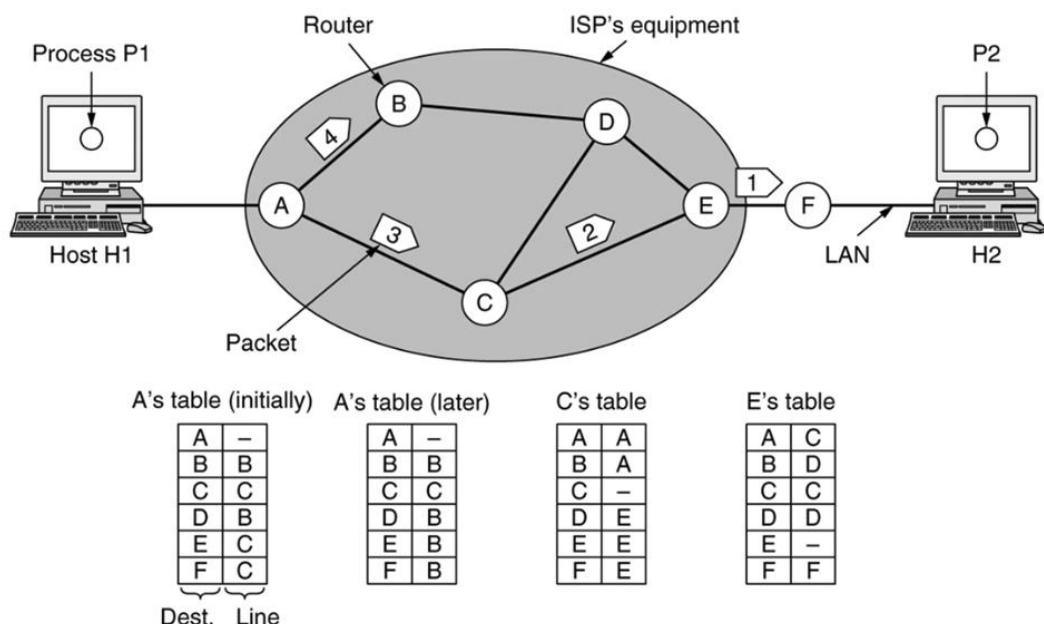
I servizi sono indipendenti dalla tecnologia che usa il router.

Il livello di trasporto nasconde il numero, tipo e topologia dei router che attraversa.

Gli indirizzi di rete sono disponibili a livello di trasporto per usare un unico livello di numerazione.

È possibile che ci sono reti che usano lo stesso numbering plan? No se abbiamo indirizzi pubblici, si se privati. (Domanda Esame).¹

IMPLEMENTARE SERVIZI SENZA CONNESSIONI



Ogni router sa cosa fare: se gli arriva un pacchetto con un determinato indirizzo sa come gestirlo.

¹ DomandaEsame1

Se un router ha un pacchetto 192.168.1.1 sa che deve girarli sull’interfaccia interna, tutto il resto deve mandarli al gateway.

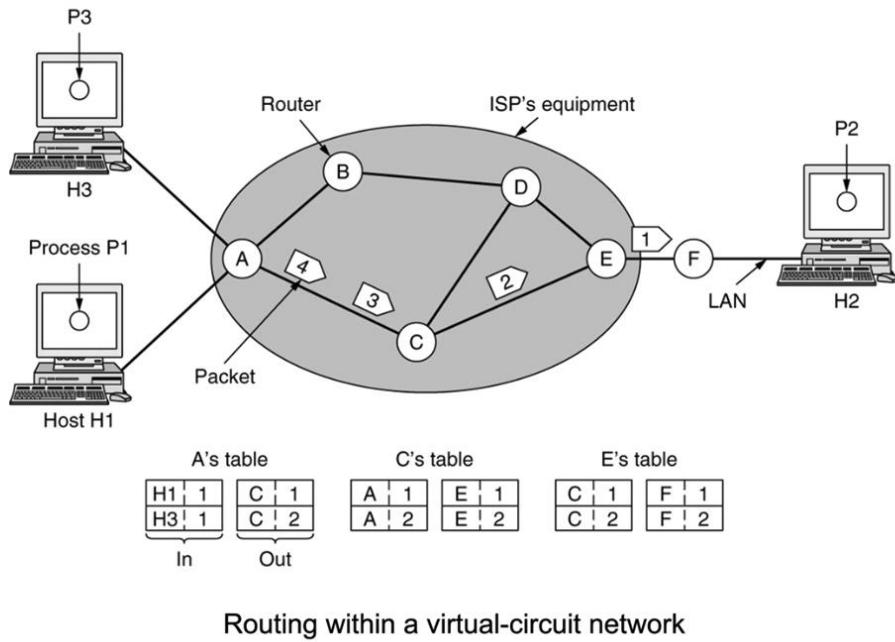
Se abbiamo una connessione col nostro vicino di casa (Telecom-Fastweb), i router sapranno come raggiungere la propria rete interna, quella internet e avranno un altro pezzo nella tabella di routing che ci dirà dove andare.

Quindi il router ogni istante ha una tabella di routing.

In figura:

- Guardiamo la tabella del router A all’inizio: abbiamo la destinazione e come arrivarci. Se siamo nel nodo A non devo passare nulla per arrivare ad A. Per arrivare a B devo instradare al nodo B, lo stesso se da A voglio passare per C vado su C direttamente. Se il router A ha un pacchetto per il router D, ci arrivo tramite B. In questo caso il servizio è senza connessione perché il pacchetto potrebbe cambiare strada.
-

SERVIZI CONNECTION-ORIENTED



RASSEGNA TRA PROBLEMATICA, DATAGRAM NETWORK E VIRTUAL CIRCUIT NETWORK

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Lez 10

CONFRONTO TRA RETI A CIRCUITO VIRTUALE E RETI DATAGRAM

Problema	Rete datagram	Rete a circuito virtuale
Problema	Rete datagram	Rete a circuito virtuale
Impostazione del circuito	Non necessaria	Necessaria
Indirizzamento	Ogni pacchetto contiene gli indirizzi completi di sorgente e destinazione	Ogni pacchetto contiene un breve identificatore di circuito virtuale
Informazioni di stato	I router non mantengono informazioni di stato sulle connessioni	Ogni circuito virtuale richiede spazio nella tabella del router per la connessione
Routing	Ogni pacchetto è inoltrato in maniera indipendente	Il percorso viene scelto durante l'impostazione del circuito e tutti i pacchetti lo seguono
Effetto del malfunzionamento di un router	Nessuno, tranne i pacchetti persi perché in transito durante il malfunzionamento	Tutti i circuiti virtuali che passano dal router vengono chiusi
Qualità di servizio	Difficile	Facile se abbastanza risorse possono essere allocate in anticipo a ogni circuito virtuale
Controllo della congestione	Difficile	Facile se abbastanza risorse possono essere allocate in anticipo a ogni circuito virtuale

ALGORITMI DI ROUTING

Vengono usati a livello di autonomous system per connettersi alle reti.

- Principio di ottimalità: rimozione dei cicli per poter raggiungere tutti gli host. Le reti però molto spesso contengono cicli. A livello ottimale, se riuscissimo ad avere un sink tree (senza cicli) sarebbe meglio.
- Shortest path: algoritmo di Dijkstra. Calcola il percorso più breve su un grafo in termini di pesi e distanza.
- Flooding: è una tecnica dove si inonda una rete locale, possono crearsi collisioni a livelli bassi (non a livello di rete).
- Distance vector routing:
- Link state routing: a seconda dello stato (congestioni, malfunzionamento, delay elevato) del link si prendono delle decisioni. Si possono applicare politiche di instradamento a seconda delle circostanze.

C'è possibilità di fare routing gerarchico: potrebbero esserci reti dedicate dove ognuna ha una priorità di rete a seconda delle gerarchie imposte.

- Routing Broadcast: in alcune applicazioni, gli host hanno bisogno di inviare messaggi a molti o a tutti gli altri host. La trasmissione contemporanea di un pacchetto a tutte le destinazioni è chiamata *broadcasting*. Molto spesso usato nelle sale di gaming. Un metodo broadcast poco efficiente e molto costoso è quello in cui la sorgente invia un pacchetto distinto a ogni destinazione. Una soluzione efficiente è il routing *multideestination* dove ogni pacchetto ha una lista delle destinazioni desiderate.
- Routing multicast: alcune app, come giochi, o dirette video trasmessi a più punti, inviano pacchetti a più destinazioni. Inviare pacchetti distinti a ogni destinazione è costoso. È quindi necessario trovare il modo di inviare messaggi a gruppi ben definiti: col multicast. Il multicast ha un metodo che consente di creare e distruggere gruppi e identificare quali router vi appartengono. I multicast inviano pacchetti lungo gli spanning tree per consegnarli ai membri di un gruppo facendo uso efficiente di banda larga.
- Routing anycast: a volte i nodi forniscono un servizio, quale l'orda del giorno o la distribuzione di contenuti, per i quali l'aspetto essenziale è ottenere la giusta info e non il nodo che viene contattato. Nell'anycast un pacchetto è consegnato al membro più vicino di un gruppo.

ROUTING BASATO SULLO STATO DEI LINK

Un router cerca di capire chi sono le reti vicine facendo un'operazione di discovery e acquisisce info sugli indirizzi di rete.

Dopo ciò, capisce che ci sono nodi vicini e può misurare alcuni aspetti (costo, delay, tempo di interconnessione, distanza, ...) e setta delle metriche per ciascun vicino.

Dopo aver imparato queste info, prepara i pacchetti da instradare tenendo conto di quello che ha appreso.

Siccome ciascun pacchetto sa quale sarà il prossimo router, se ci sono più vicini, vengono costruiti tenendo conto le info acquisite in precedenza.

Calcola inoltre il percorso minimo verso altri router.

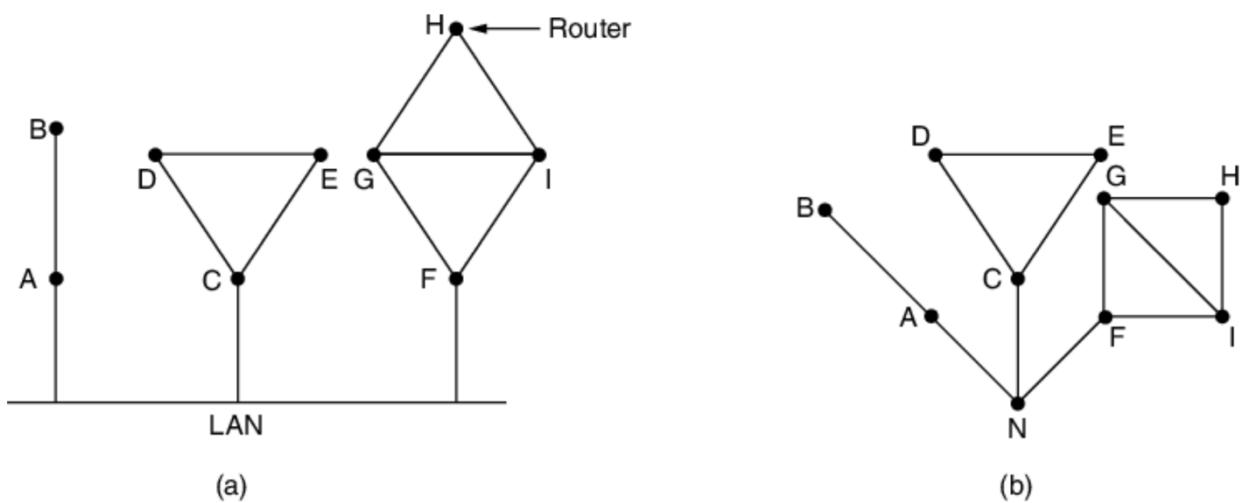


Figura 5.11 (a) Nove router e una LAN broadcast. (b) Un modello di (a) tramite un grafo.

Un router H deve comunicare con 9 router e una LAN, aggiunge un nodo aggiuntivo (N) tramite il

quale A, C, F possano comunicare, passando da un modello broadcast a modello di grafo.

ROUTING GERARCHICO

Immaginiamo sedi (*region*) di azienda o università al cui interno ci sono dei router .

Il router 1A comunica con sé stesso direttamente.

Per arrivare a 1B fa un solo hop, lo stesso per 1C.

Se da router 1A è diretta da router 2A ci arrivo tramite router 1B facendo 2 hop (salti)...

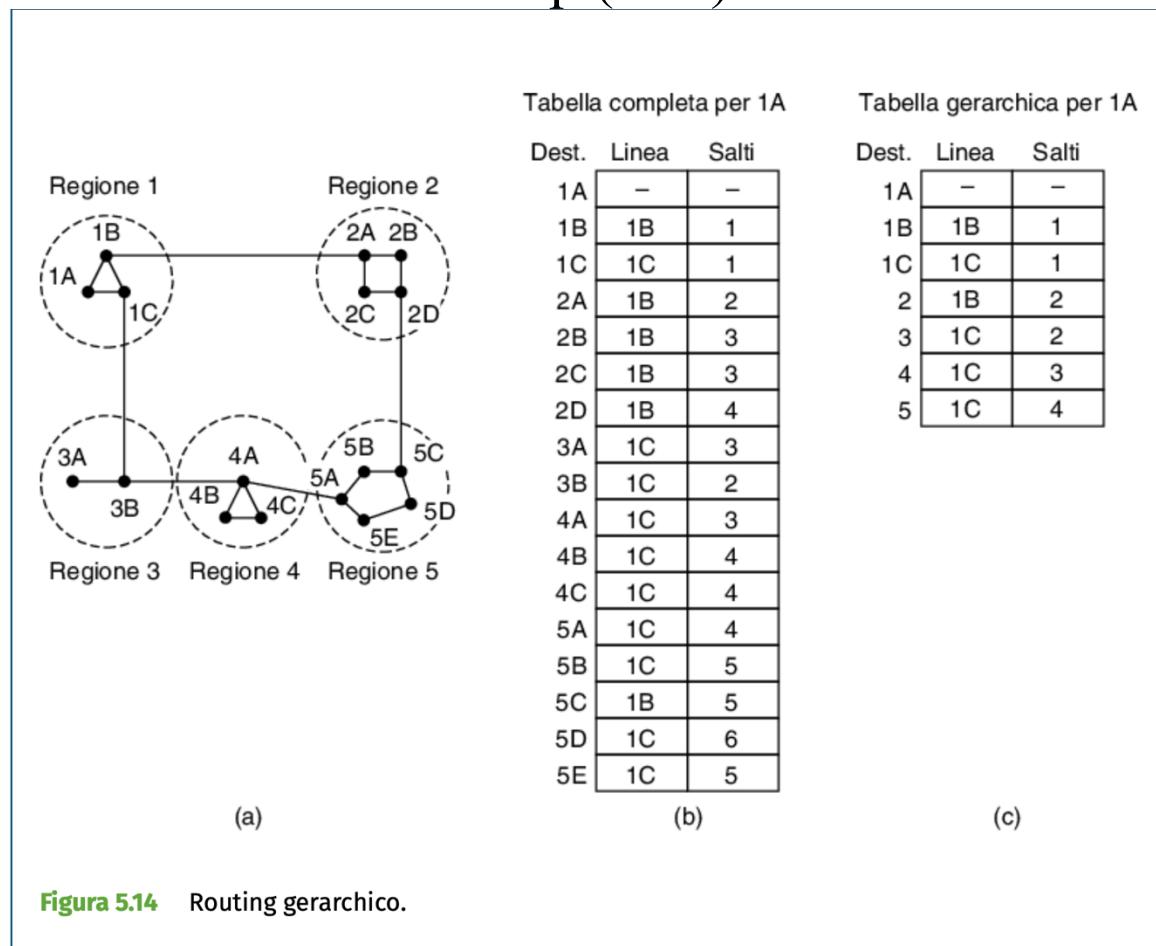
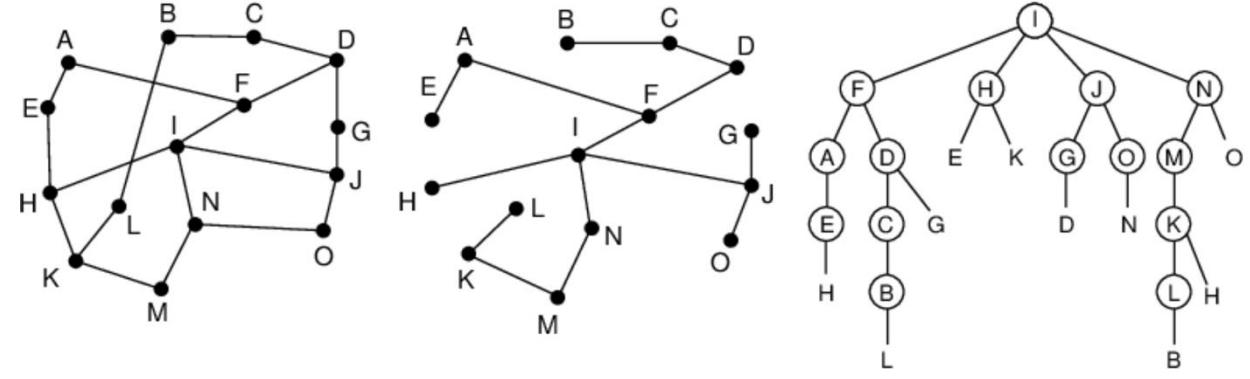


Figura 5.14 Routing gerarchico.

- **BROADCAST ROUTING:** un nodo trasmette e tutti ricevono.



- ROUTING MULTICAST: sta sopra.
 - ROUTING ANYCAST: il routing viene fatto verso singoli nodi di un gruppo. È un multicast ristretto a un singolo elemento.

GESTIONE DEL TRAFFICO A LIVELLO DI RETE

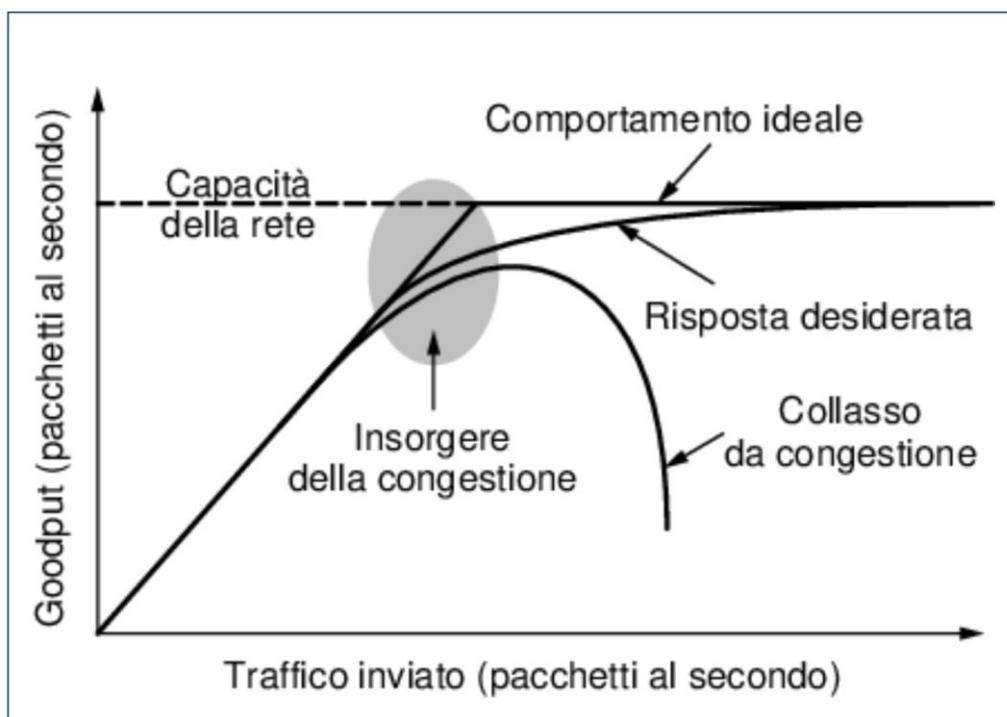
Ci sono congestioni e bisogna gestirle.

- Si possono implementare meccanismi di routing che tengono conto del traffico.
 - Ci sono politiche di ammissione del traffico.
 - Load & Traffic shaping: modellazione di una linea su un limite (Traffic Shaping).²

² DomandaEsame2

CONGESTIONE

Una rete ha una capacità nominale. La velocità di trasferimento collassa quando si creano congestioni e aumentano gli indici di pacchetti persi e la latenza



Fattori che possono determinare la congestione:

- Troppi pochi buffer nel router.
- Processore del router troppo lento.
- Linea di trasmissione troppo lenta.

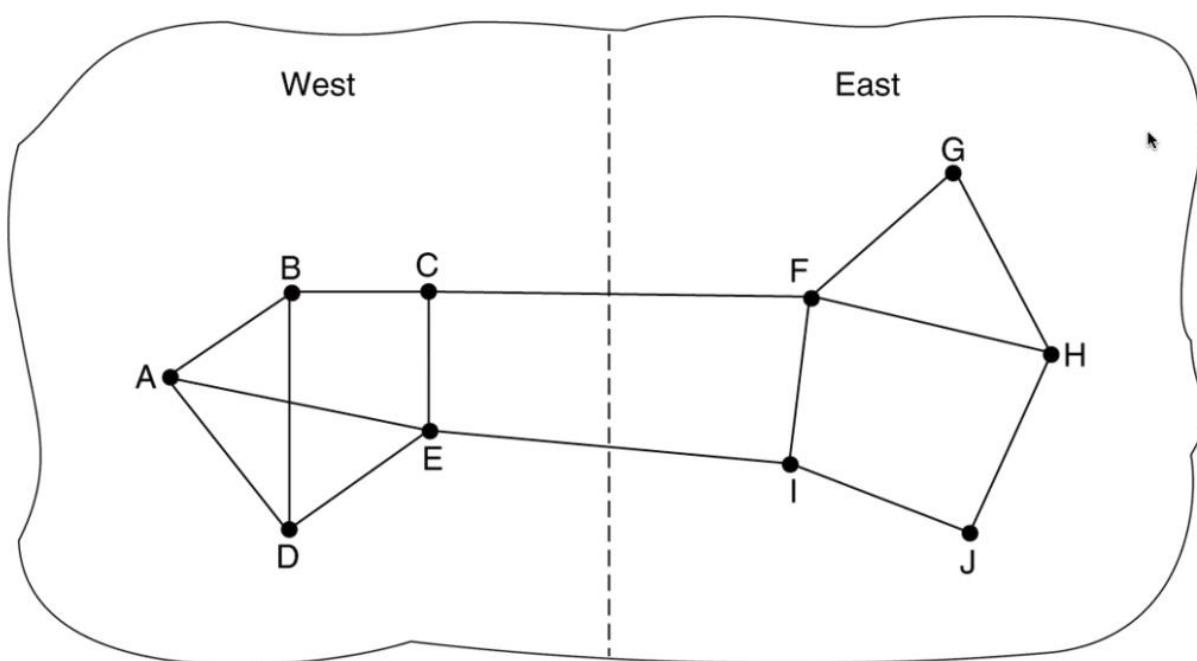
La congestione tende a propagarsi ai suoi vicini.

Lez 11

ROUTING TRAFFIC-AWARE

L'intelligenza sta nel capire dove deve essere instradato il traffico e se le due connessioni sono in grado di sopportare il carico.

Quello che cambia sono gli algoritmi che si applicano a tipologie di questo tipo.



ADMISSION CONTROL

Ci sono reti congestionate per varia natura: crash, maltempo, malfunzionamenti...

Si possono applicare algoritmi che tengono conto dei tratti congestionati per comunicare da una rete all'altra efficientemente.

TRAFFIC SHAPING

Applicato dai gestori telefonici soprattutto.

Shape significa forma. L'idea è quella di modellare il traffico secondo un modello. Se compro la fibra e per questioni commerciali il gestore ha deciso che la linea è di 2.5 gigabit ma ci danno come massima linea possibile 1 gigabit in down e 200 mega in up, quella è la conseguenza dello shaping: ci viene modellata la connessione per gestire al meglio tutta la struttura di rete. È come se avessimo un tubo e lo "strozzassimo" in modo tale che da quella fonte passa solo una determinata quantità di informazioni.

Viene fatto ciò perché non c'è abbastanza banda per tutti

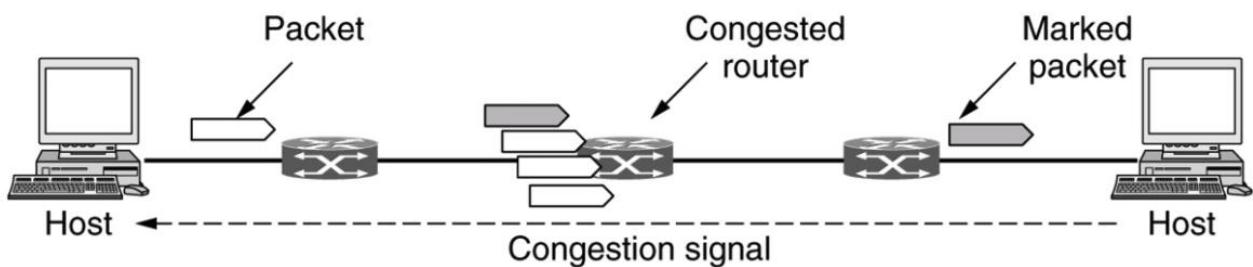
NOTIFICA ESPLICITA DI CONGESTIONE³

La congestione può stare sul router o sul link.

Soltanamente viene usato un pacchetto con un marcitore per dire che viene da un router congestionato. La congestione sul router può dipendere dalla RAM o dalla CPU utilizzata, un tempo

³ DomandaEsame3

nei router si inseriva la stessa quantità di RAM che avevano i computer. Per creare una congestione sul proprio router bisogna prima di tutto capire qual è il protocollo del dispositivo e poi si può far fare computazione al router per rallentarlo.



QUALITA' DEL SERVIZIO E APPLICAZIONI QoE

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

CATEGORIE QUALITA' DI SERVIZIO ED ESEMPI

Per la parte telefonica serve un bit rate costante.
Per la videoconferenza serve una parte variabile real time di bit rate.

Per un film on demand serve un bit rate non-real time.
Per trasferire file serve un bit rate disponibile.

OVERPROVISIONING

Significa avere più rete di quella che serve.

È una soluzione costosa perché si valuta la situazione nel caso peggiore, cosa che raramente potrebbe accadere.

Si progetta una rete asintoticamente maggiore rispetto all'utilizzo che verrà fatto.

Un esempio può essere un sistema di videosorveglianza che necessita di una qualità video-audio estremamente ottima.

Le problematiche per gestire la qualità del servizio è che c'è necessità per le app di avere un indirizzamento particolare e c'è bisogno di qualcosa che regoli il traffico in quella rete: la rete deve conoscere bene ciò che le arriva e prepararsi.

SCHEDULING DEI PACCHETTI

Il router deve poter gestire diverse grandezze di banda, avere CPU veloci e un buffer ampio.

Gli algoritmi di scheduling usati sono:

- FIFO
- Queue
- Coda pesata.

INTERNETWORKING

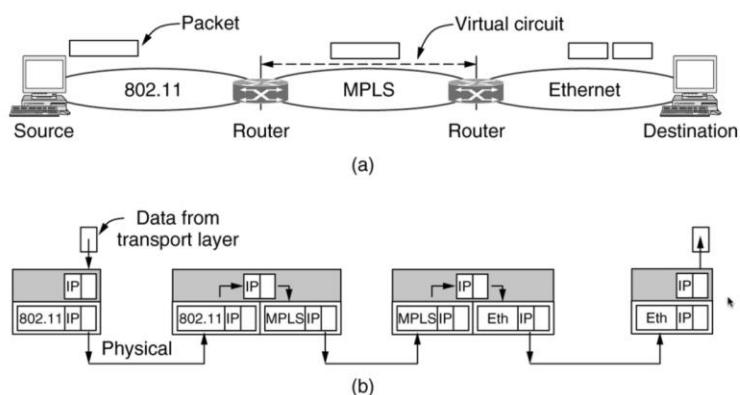
Le reti differiscono tra loro almeno per protocollo e mezzo di trasmissione.

Per far vedere più reti serve implementare internetwork routing tra reti.

COME DIFFERISCONO LE RETI

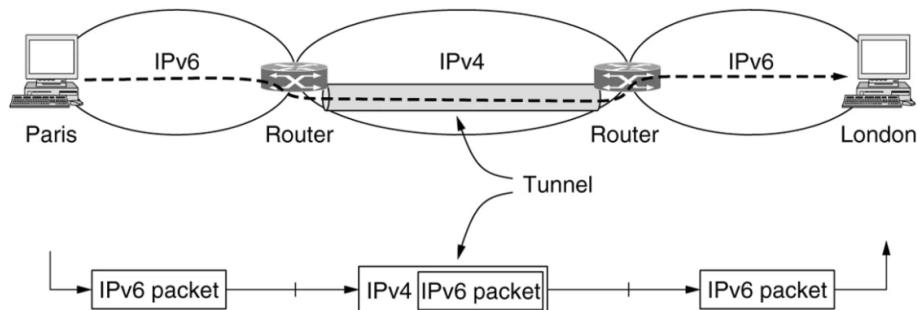
Item	Some Possibilities
Service offered	Connectionless versus connection oriented
Addressing	Different sizes, flat or hierarchical
Broadcasting	Present or absent (also multicast)
Packet size	Every network has its own maximum
Ordering	Ordered and unordered delivery
Quality of service	Present or absent; many different kinds
Reliability	Different levels of loss
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, packet, byte, or not at all

Connecting Heterogeneous Networks



(a) A packet crossing different networks. (b) Network and link layer protocol processing.

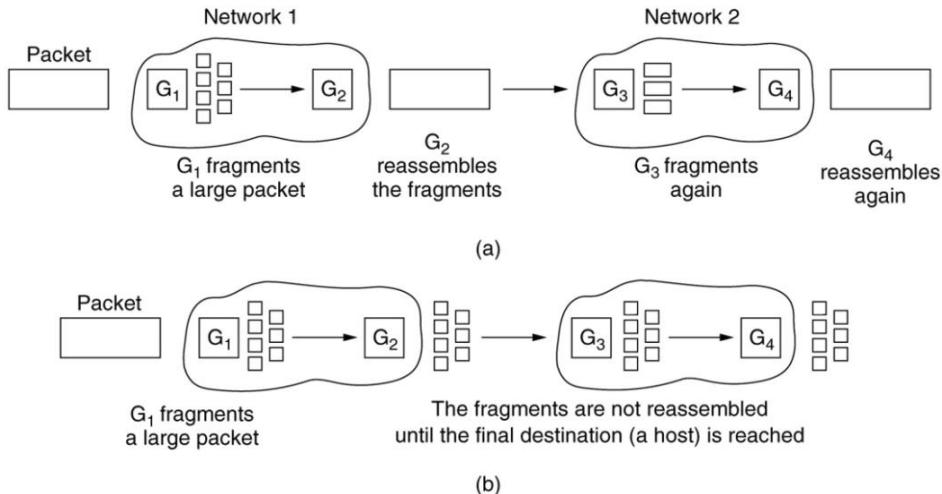
Connecting Endpoints Across Heterogeneous Networks (1 of 2)



Viene applicato l'incapsulamento: viene veicolato il traffico IPv6 all'interno di pacchetti IPv4. È la stessa idea della VPN.

FRAMMENTAZIONE

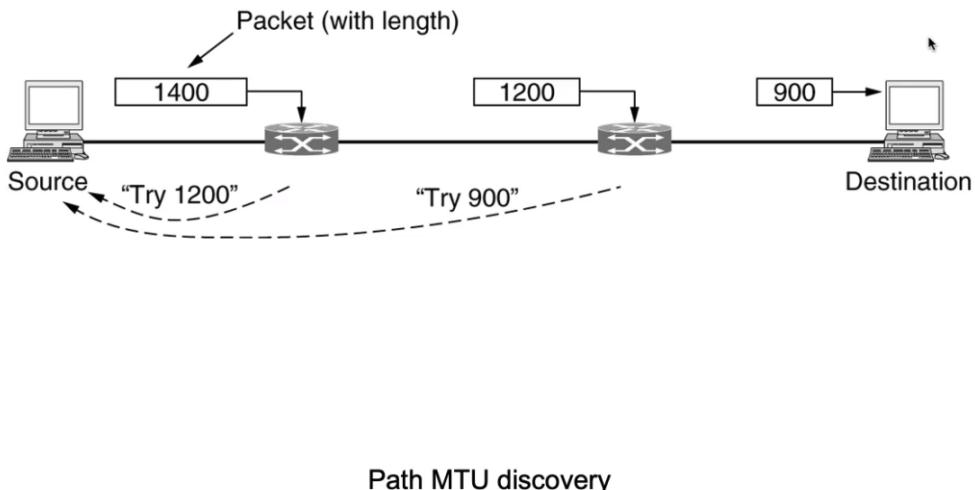
- Framm trasparente: posso avere un pacchetto che viene frammentato in piccoli pezzi dal gateway1 che li riassembra e li ritrasmette a g3.
- Framm. Non trasparente: il g1 frammenta i pacchetti e li passa a g2 che li passa a g3 che non li riassembra e le ripassa a g4



(a) Transparent fragmentation. (b) Nontransparent fragmentation.

Posso avere delle grandezze di datasize diverse. Nella frammentazione non trasparente in uscita ho n pacchetti che dipendono dal comportamento dei router.

IL MTU è un parametro che indica la più grande quantità di dati che può gestire il singolo frammento.



Ci sono algoritmi per capire qual è l'MTU sulle varie interfacce. Sulla Ethernet è la migliore.

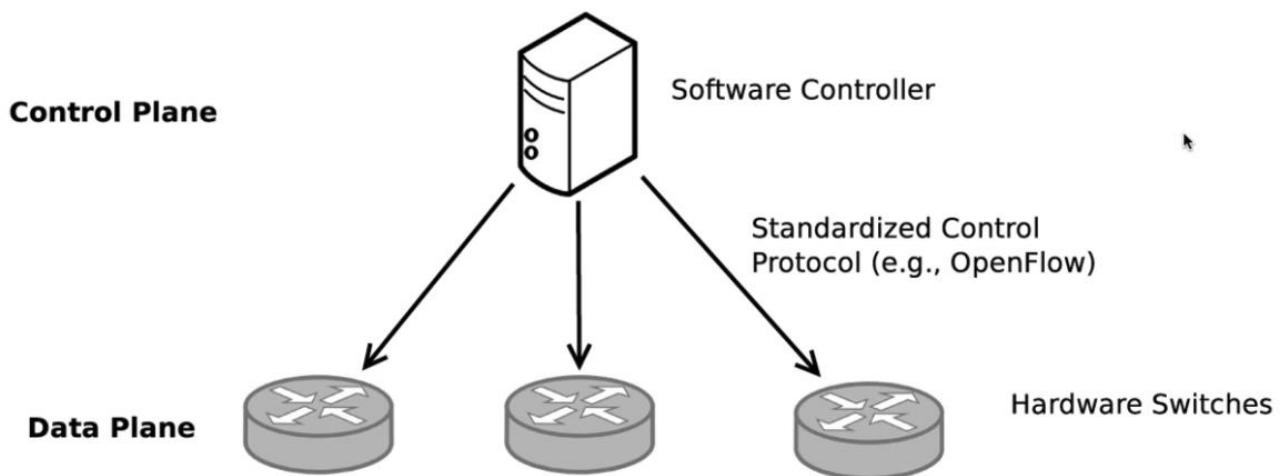
SOFTWARE-DEFINED NETWORKING

Con delle macchine virtuali possiamo simulare processori, RAM ecc...

Immaginiamo di avere una serie di protocolli e una serie di HW (SDN data plane) e dei meccanismi per fare telemetria di rete.

La parte di dataplane di un SDN riguarda l'HW mentre la control plane è SW.

Con la similitudine della virtualizzazione, la parte di SW che implementa la macchina virtuale è nell'SDN control plane. L'idea è che posso avere delle reti virtuali



IL LIVELLO DI RETE SU INTERNET

Ci sono 10 principi:

- Essere sicuri che funzioni
- Renderlo semplice
- Fare scelte chiare
- Trarre vantaggio dalla modularità

- Aspettarsi eterogeneità (reti diverse)
- Cercare un buon design
- Pensare alla scalabilità
- Considerare performance e costi
- Essere severi quando si invia e tolleranti quando si riceve
- Evitare opzioni e parametri statici.

Abbiamo la versione 4 del protocollo IP.
Si è passato da IPv4 a IPv6.

Lez 12

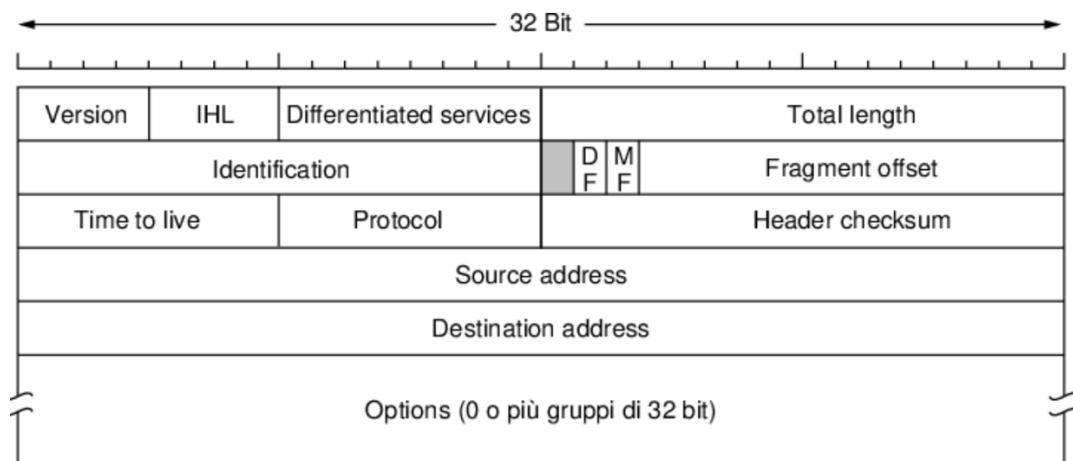
IL LIVELLO DI RETE IN INTERNET

L'IP è un protocollo datagram (non connesso e non affidabile) che opera come segue:

- Riceve i dati dal livello transport e li incapsula in pacchetti.
- Instrada i pacchetti sulla subnet, frammentandoli magari lungo il percorso.
- Arrivati a destinazione:
 - Riassembra i frammenti in pacchetti.
 - Estraie i dati del livello trasporto
 - Consegna al livello trasporto i dati arrivati.

È un livello che consente di fare applicazioni rete e consente di avere dispositivi e canali di comunicazione eterogenei.

PROTOCOLLO IPv4



Come si è ovviato al fatto che gli indirizzi IP sono pochi? (domanda esame)⁴

- IPv6
- Indirizzi privati (da soli funzionano ma non completamente, hanno necessità di avere un'altra tecnologia, un router che ha un indirizzo pubblico e fare il NAT)

Il NAT fa vedere a internet gli indirizzi che per definizione sono nascosti. Nessun router ruoterà una rotta con una classe privata ma grazie al NAT è possibile farlo. I nostri cellulari hanno tutti il NAT ad esempio.

Grazie al campo *destination address* possiamo oscurare dei siti internet e renderli inaccessibili.

Con IPv4 possiamo:

- Attivare opzioni di sicurezza.
- Definisce un insieme di host su cui il pacchetto deve transitare, se un host non è disponibile il pacchetto non arriverà.
- Elenco dei router che devono fare il percorso.
- Fa sì che ogni router aggiunga il proprio IP. In breve è comando *traceroute*.
- Fa sì che ogni router aggiunga indirizzo e ora.

⁴ DomandaEsame4

INDIRIZZI IP

- Prefissi: un blocco contiguo di spazi di indirizzi IP
- Sottoreti
- CIDR
- NAT
- PAT

Si parla di NAT ma in realtà è PAT: se da internet dobbiamo mostrare la webcam con un IP privato bisogna fare un NAT tra l'indirizzo intero e l'indirizzo IP pubblico del router, ricordando che se cambiamo IP non funziona.

PREFISSI: abbiamo 32 bit a disposizione (IPv4): alcuni bit avranno una parte dell'indirizzo di rete e l'altra verrà usata per la parte di host (ha 8 zeri: ho 8 bit usabili per indirizzare degli host $2^8 = 256$ host).

Ci sono alcuni indirizzi *speciali* dove l'indirizzo 0 è l'indirizzo di rete e il 255 identifica i *broadcast*.

Gli indirizzi utilizzabili dagli host sono quindi 254. Se faccio ulteriori sottoreti perdo altri indirizzi).

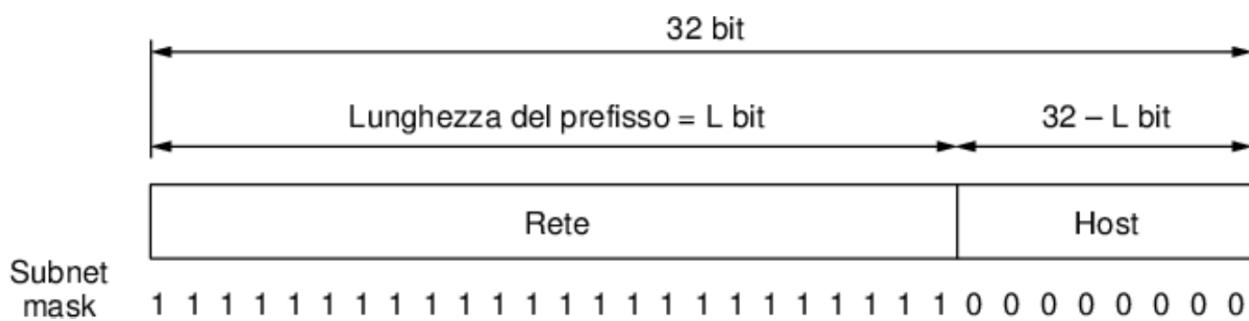


Figura 5.48 Un prefisso IP e una subnet mask.

SOTTORETI

Partendo da destra: immaginiamo una rete con un blocco (16) di classe C: 128.208.0.0 significa che la parte di rete è data dai primi 2 byte. La rete è suddivisa, grazie al border gateway, in più sottoreti.

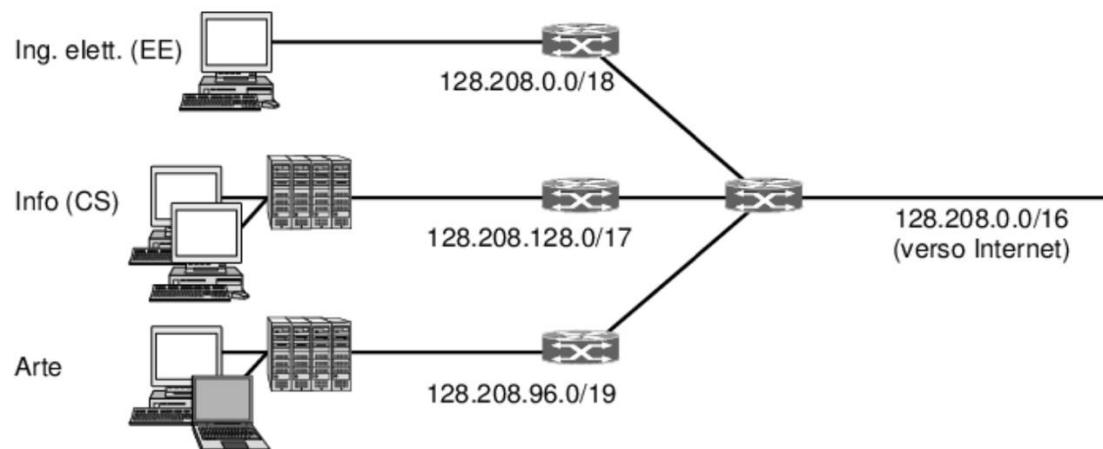


Figura 5.49 Suddivisione di un prefisso IP in reti separate tramite subnetting.

CIDR

Es: Cambridge con prefisso 192.24.0.0/21.

Se faccio 32 bit – 21 ho 2^{11} indirizzi IP usabili (2048)

Edimburgo: $/22 \rightarrow 32-22 = 2^{10} = 1024$ IP usabili.

Università	Primo indirizzo	Ultimo indirizzo	Totale indirizzi	Prefisso
Cambridge	194.24.0.0	194.24.7.255	2.048	194.24.0.0/21
Edimburgo	194.24.8.0	194.24.11.255	1.024	194.24.8.0/22
(disponibili)	194.24.12.0	194.24.15.255	1.024	194.24.12.0/22
Oxford	194.24.16.0	194.24.31.255	4.096	194.24.16.0/20

Quando ho un indirizzo aggregante, oltre alle reti che splitta, ci sono anche gli indirizzi *available*.

CLASSFUL ADDRESSING⁵

Le reti sono divise in classi che vanno da A ad E.

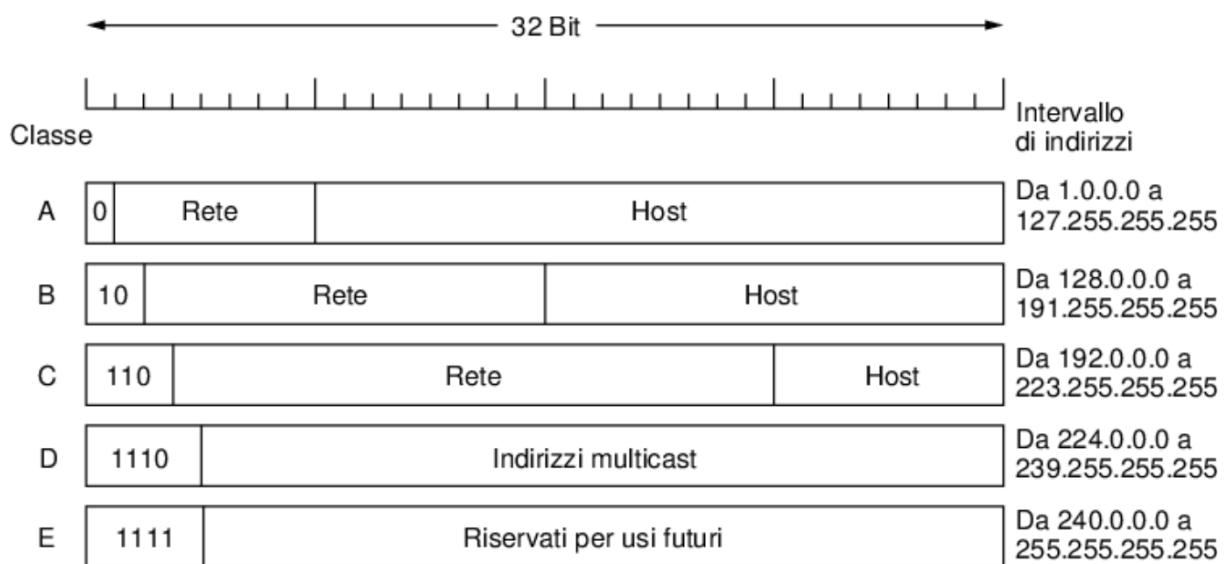


Figura 5.53 Formati degli indirizzi IP.

⁵ DomandaEsame5 molto importante.

Abbiamo una parte di rete e di host.

Analizzando gli indirizzi di classe C abbiamo 24 bit per la classe di rete

L'indirizzo di loopback (localhost → 127.0.0.1) sono ad esempio di classe A.

NAT

Problema: si ha a disposizione un numero N di macchine e un numero M di IP, con M<N.

È necessario quindi un sistema per consentire alle restanti macchine di collegarsi alla rete.

Nell'ambito delle reti informatiche, il network address translation (abbreviato in NAT e traducibile in italiano con “traduzione degli indirizzi di rete”) è un meccanismo che permette di modificare l'indirizzo IP dei pacchetti in transito attraverso apparati di rete - come router o firewall - all'interno di una comunicazione in corso tra due o più nodi della rete.

Il principio del network address translation

Il meccanismo del NAT è diventato fondamentale nel momento in cui gli indirizzi IP del protocollo IPv4 hanno cominciato a scarseggiare. L'aumento esponenziale dei dispositivi connessi alla Rete ha fatto sì che gli indirizzi disponibili in questo spazio di rete (4 miliardi e 300 milioni circa) si esaurissero in fretta. Il NAT è stato quindi utilizzato per poter nascondere

dietro un unico indirizzo pubblico decine e decine di indirizzi privati (e quindi altrettanti device).

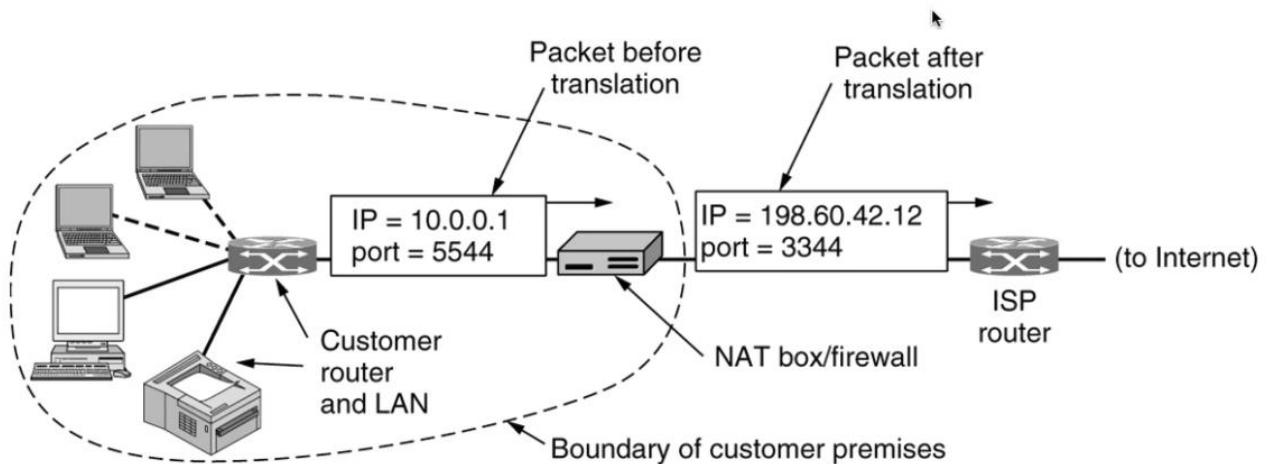
Nonostante l'introduzione del protocollo IPv6, questa tattica di “mascheramento” degli indirizzi privati è molto utilizzata sia per semplice comodità sia perché capace di garantire un certo numero di vantaggi.

Il NAT utilizza un gateway con almeno un'interfaccia connessa alla rete interna e almeno un'interfaccia di rete connessa al web (e quindi all'esterno della LAN). Grazie a questo “portale”, tutti i dispositivi connessi alla LAN potranno accedere a Internet utilizzando, e quindi “consumando”, un singolo indirizzo IP pubblico: quello del gateway stesso. In questo caso il gateway si occupa di fare una traslazione - o traduzione - degli indirizzi IP dei pacchetti in transito attraverso il router stesso: quando un terminale della rete interna effettua una richiesta verso Internet, l'indirizzo IP di partenza risultante sarà quello, pubblico, del gateway stesso e non quello privato del dispositivo che effettua la richiesta; nel caso di pacchetti in entrata, invece, il gateway si occupa di reindirizzare il traffico verso i dispositivi di destinazione appartenenti alla rete modificando l'IP di destinazione di conseguenza.

Lez 13

Abbiamo visto che la tecnica NAT è usata per arginare il problema di scarsità di indirizzi IPv4.
Ma quali sono le problematiche del NAT?

- Problemi di sicurezza e privacy
- Performance: alto consumo di corrente. Il *natting* è fatto da pc, router che impiegano molte energie.



IPv6

È vista come una caratteristica ancora, non è ben diffusa per l'uso comune privato.

- Supporta miliardi di host.
- Riduce le tabelle di routing: significa gestire in memoria strutture dati più semplici.
- Semplifica il protocollo
- Miglior sicurezza
- Attenzione al tipo di servizio.

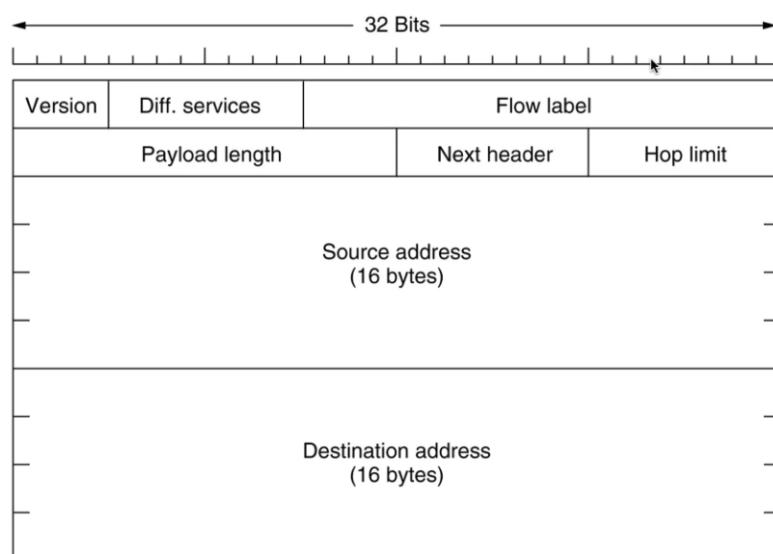
- Permette coesistenza di vecchi e nuovi protocolli.
- Permette evoluzioni future di protocolli.
- Roaming dell'host senza cambiare indirizzo.

C'è da dire che IPv4 non potrà mai sparire: se più indirizzi comunicano ma già uno solo è IPv4 allora serve considerarlo. Inoltre i dispositivi IoT usano IPv4.

Gli *improvements* sono:

- Indirizzi più lunghi
- Headers più semplici
- Grande avanzamento nella parte di sicurezza
- *Miglior QoS (quality of service)*.

The Main IPv6 Header



The IPv6 fixed header (required)

Extension header	Description
Hop-by-hop options	Miscellaneous information for routers
Destination options	Additional information for the destination
Routing	Loose list of routers to visit
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender's identity
Encrypted security payload	Information about the encrypted contents

IPv6 extension headers

Con IPv6 possiamo ridurre la frammentazione perché abbiamo indirizzi più lunghi.

I pacchetti IPv6 contengono info sul routing, al contrario degli IPv4.

PROTOCOLLI DI CONTROLLO

- *ICMP* → *internet control message protocol (ping)*.
- *ARP* → *address resolution protocol*.
- *DHCP* → *dynamic host configuration protocol*.

Se una rete non dà il DHCP dovrei conoscere i parametri di rete come possiamo fare?
Grazie al protocollo ARP⁶.

⁶ DomandaEsame6

ICMP

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo and echo reply	Check if a machine is alive
Timestamp request/reply	Same as Echo, but with timestamp
Router advertisement/solicitation	Find a nearby router

Ha una serie di messaggi, alcuni molto usati come *echo request e echo reply* per vedere se una macchina è attiva, oppure il *timestamp request o il time exceeded* usato col *traceroute*.

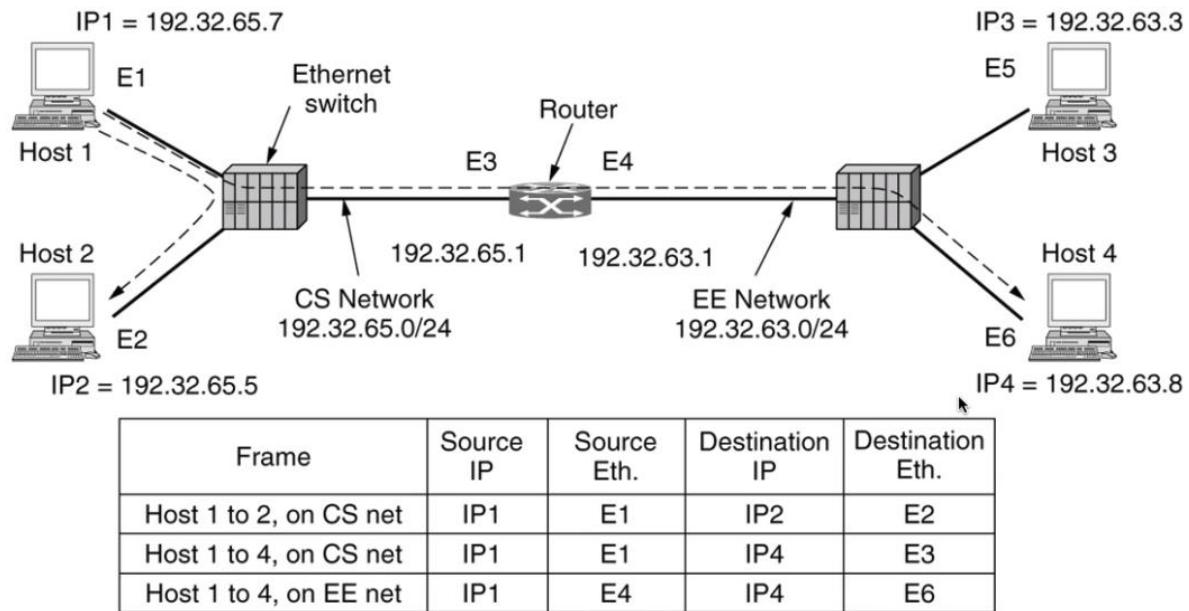
Abbiamo altri messaggi di servizio, come l'ultimo in tabella, che serve per fare discovery di router che ci sono su una rete locale.

I tempi di risposta sono in *millisecondi*.

PROTOCOLLO ARP

Il mittente inoltra in broadcast una richiesta ARP per l'IP di interesse. La macchina con quell'IP risponde col suo indirizzo MAC. Per evitare la risoluzione IP→MAC per ogni pacchetto, le coppie IP:MAC ottenute vengono registrate nella cache locale.

Il comando *arp* di Unix mostra tale cache.



Two switched Ethernet LANs joined by a router

DHCP

Viene annunciato il proprio indirizzo, subnet, DNS e gateway di default.

Bisogna far attenzione a come viene usato poiché è semplice violare la configurazione di rete di una struttura.

OSPF → INTERIOR GATEWAY ROUTING PROTOCOL

Può essere implementato da chiunque, usato nelle reti delle aziende.

L'OSPF ha diverse caratteristiche:

- Open source

- Supporta diverse metriche di distanza
- Dinamico
- Sicuro
- Supporta routing in base ai tipi di servizi
- Supporta sistemi gerarchici
- Supporta il bilanciamento del carico

OSPF supporta reti con accessi multipli.

Domanda esame? Dove si usa OSPF?

Si usa all'interno di 1 autonomum system. Quelli usati all'esterno sono i border gateway protocol.

5 MESSAGGI OSPF

Message type	Description
Hello	Used to discover who the neighbors are
Link state update	Provides the sender's costs to its neighbors
Link state ack	Acknowledges link state update
Database description	Announces which updates the sender has
Link state request	Requests information from the partner

- Hello fa discovery di vicini
- Link state up: chi invia il suo stato ai vicini fornisce i suoi costi
- Link state ack: il ricevente manda un ACK
- Database description: elenco degli aggiornamenti

- Link state request: il router chiede info a un suo partner.
-

BGP: EXTERIOR GATEWAY ROUTING PROTOCOL

Il protocollo *Border Gateway Protocol* è un protocollo di routing dinamico usato per connettere tra loro più router che appartengono a AS diversi.

Il BGP è il protocollo di routing per antonomasia, sul quale è basato il funzionamento di internet. Esso funziona attraverso la gestione di una tabella di prefissi (reti IP) che forniscono info sulla raggiungibilità delle diverse reti tra più AS. Questo protocollo non si basa su ampiezza di banda ecc, ma prende decisioni di instradamento basandosi su specifiche politiche.

La motivazione che spinge un'organizzazione all'uso del BGP per scambiare le proprie *route* in internet è la necessità di avere una rete *multihomed*.

Per rete *multihomed* si intende una rete che acquista transito da più provider e deve garantire la raggiungibilità dall'esterno dei propri servizi, tramite un set di IP pubblici. Senza il BGP se mettessimo un servizio online su un IP, se il link dovesse cadere, diventerebbe irraggiungibile per le altre reti.

Invece usando il BGP gli indirizzi saranno raggiungibili tramite qualsiasi link verso il quale siano state propagate le *route*.

➤ Ricordiamo che un AS è un'infrastruttura di rete sotto il controllo organizzativo di un unico ente. A livello mondiale a ciascun AS è associato un identificativo detto AS Number e rilasciato dal RIPE (IANA).

CRITICITA' E REQUISITI BGP

Per propagare in internet la propria *route* via BGP è necessario essere AS riconosciuto e avere una propria classe di IP pubblici.

- Possible routing constraints

- Do not carry commercial traffic on the educational network
- Never send traffic from the Pentagon on a route through Iraq
- Use TeliaSonera instead of Verizon because it is cheaper
- Don't use AT&T in Australia because performance is poor
- Traffic starting or ending at Apple should not transit Google

Lez 14

Un indirizzo IP usato è assegnato a un host, ma non è detto che la macchina sia funzionante. Per vedere se è attivo possiamo usare diverse tecniche:

- Ping (però potrebbe esserci un firewall di mezzo)
- Traceroute (ma potrebbe essere bannato e possiamo vederlo fino a un certo router)
- Abbiamo il servizio *whois*, un sistema che interroga un database, ed è applicabile su un dominio o su un IP. Se c'è un servizio(coppia IP-porta) che risponde a un IP significa che è attivo.

Il modo è provare tutte le porte per vedere se ci sono porte aperte, cioè servizi in ascolto.

INGEGNERIA DEL TRAFFICO TRA I DOMINI

Bisogna fare attenzione a configurare quei parametri e configurazione dei protocolli di reti per gestirli in modo corretto ed evitare congestione.

Se c'è traffico tra domini c'è in entrata e in uscita: dobbiamo scegliere le rotte per controllare il traffico in entrata.

Se ci sono sistemi critici in un autonomum system e ci sono rotte che vanno verso specifici IP dovremmo gestire le rotte in modo particolare dando degli attributi locali.

Bisogna inoltre prevedere come il traffico lascia la rete: immaginiamo un server di streaming video di un'università: quello è traffico in uscita dalla nostra

rete. Bisogna evitare che il traffico in uscita venga instradato su un traffico diverso in modo da non creare congestioni e rallentamenti. Ovviamente è sottinteso che bisogna avere più *uplink* oppure implementare un *traffic shaping*(*dedicare una certa quantità di banda a un servizio*).

INTERNET MULTICASTING

È identificata dagli indirizzi IP di classe D.

Ogni classe D identifica un gruppo di host.

Abbiamo 28 bit per identificare i gruppi e possiamo avere 250 milioni di gruppi esistenti allo stesso tempo. Un processo invia un pacchetto a un indirizzo di classe D e ci sarà un modo che prova a inviare un pacchetto a tutti i membri del gruppo a cui è riferito, non c'è garanzia che i pacchetti vengano consegnati. La parte multicast è solitamente disabilitata sui router e viene usata a livello gaming.

POLICY AT NETWORK LAYER

Ogni stato usa policy differenti a livello rete.

Ci possono essere dispute tra *vicini* e avere un'interruzione per un mancato pagamento.

Generalmente dovrebbero esserci canoni ben chiari:

- Non si ritarda il traffico
- Non lo si blocca
- Non si paga per la priorità

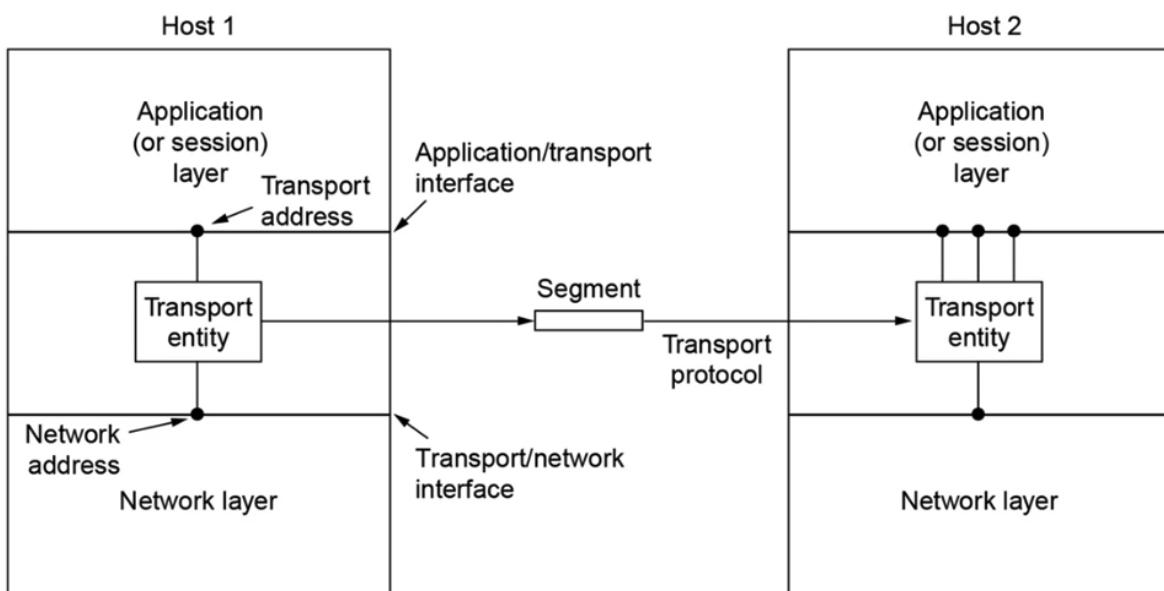
Lez 15 → LIVELLO DI TRASPORTO

I protocolli di trasporto devono:

- Definire la modalità di indirizzamento a livello di trasporto.
 - Gestire gli errori e il flusso.
- I problemi sono simili del livello DL, ma il canale fisico in questo caso è l'intera sottorete.

Il servizio di trasporto fornisce servizi agli strati superiori e usa servizi forniti a livelli inferiori.

Dal punto di vista storico, senza le socket di Berkeley, lo sviluppo sarebbe stato molto lento.



The network, transport, and application layers

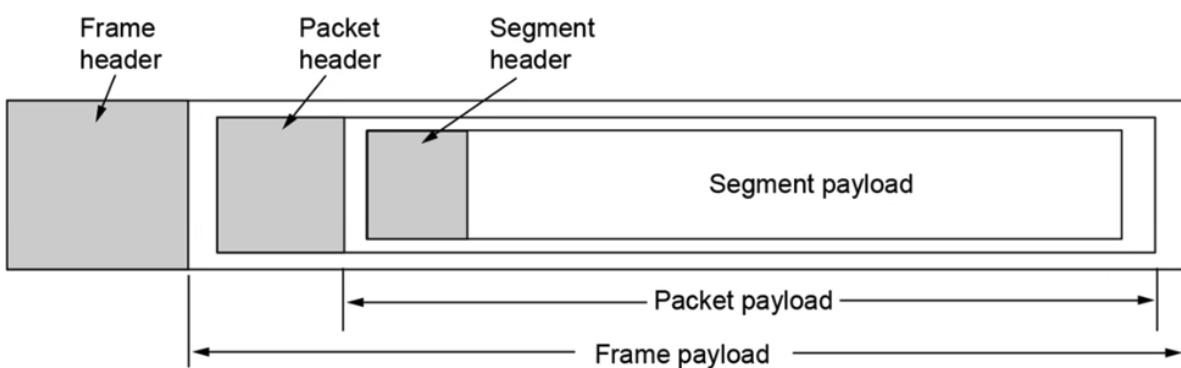
Nella parte inferiore abbiamo il livello di rete e in quello superiore abbiamo il livello applicazione o sessione.

Se sono a livello di trasporto ho una coppia di info (IP-Porta), a livello di rete avrò solo un indirizzo di rete.

Tra i vari livelli esistono le interfacce.

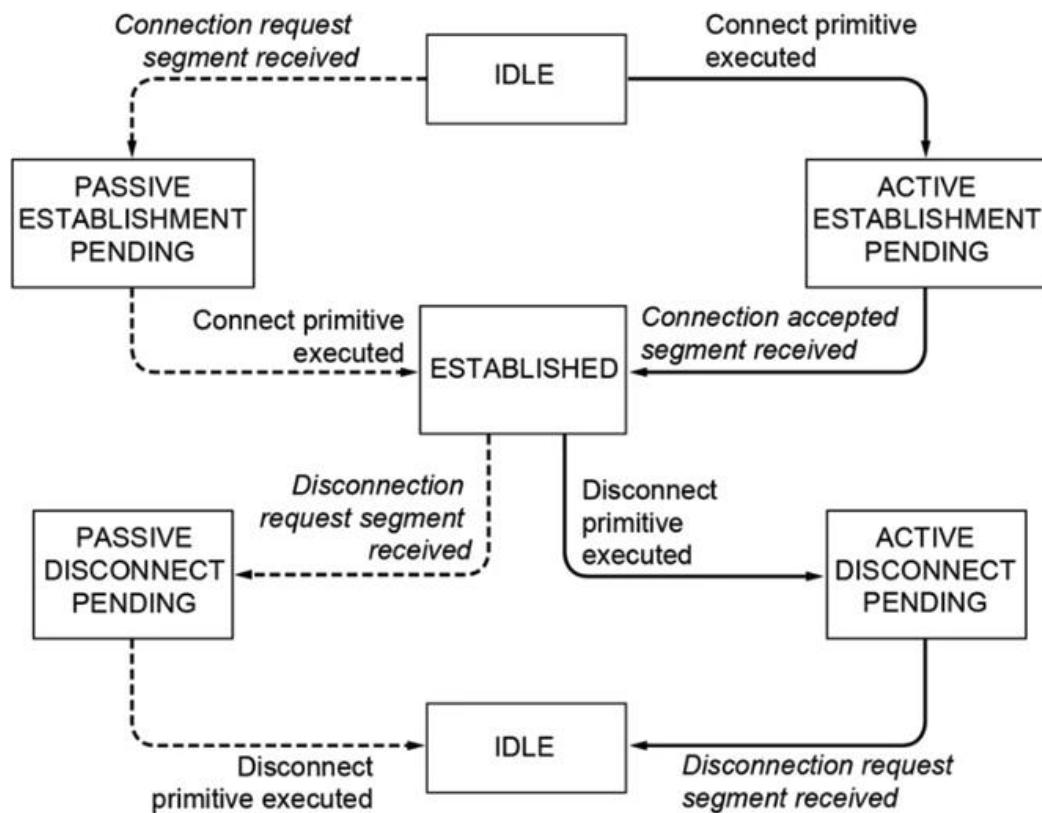
PRIMITIVE DEI SERVIZI DI TRASPORTO

- Listen: non invia nessun pacchetto. Resta in attesa fin quando qualche processo non si connette.
- Connect: invia un pacchetto di *connection request*. Prova in maniera attiva a stabilire una connessione.
- Send: invia un pacchetto col dato da mandare.
- Receive: resta in attesa finche non arrivano pack DATA.
- Disconnect: invia un pacchetto di *disconnect req*. Chiede che venga rilasciata una connessione.



Quando mi collego a un sito e faccio una *get*, questa parte di protocollo si trova in *segment*. Per ottenere il semgent payload ci deve essere una parte di packet e una di frame (incapsulamento). A secondo del framing cambia il mezzo di trasmissione.

BERKELEY SOCKET



Ci sono stati in cui si passa, alcuni relativi al server e altri al client. Ci sono stati dove ci sono scambi di dati.

Primitive	Meaning
SOCKET	Create a new communication endpoint
BIND	Associate a local address with a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Passively establish an incoming connection
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

Esempio di socket

```
#define SERVER_PORT 8080      /* arbitrary, but client & server must agree */
#define BUF_SIZE 4096    /* block transfer size */

int main(int argc, char **argv)
{
    int c, s, bytes;
    char buf[BUF_SIZE];    /* buffer for incoming file */
    struct hostent *h;      /* info about server */
    struct sockaddr_in channel; /* holds IP address */

    if (argc != 3) {printf("Usage: client server-name file-name\n"); exit (-1);}
    h = gethostbyname(argv[1]); /* look up host's IP address */
    if (!h) {printf("gethostbyname failed to locate %s\n", argv[1]); exit (-1);}

    /* rest of program omitted for brevity */
}
```

La *gethostbyname* andrà a buon fine quando ci sarà una corrispondenza stringa-IP data dal DNS: grazie al *resolver* che interroga i *nameserver* del pc: se sul pc non c'è un nameserver la *gethostbyname* fallisce.

```
s = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
if (s < 0) {printf("socket called failed\n"); exit (-1);}
memset(&channel, 0, sizeof(channel));
channel.sin_family= AF_INET;
memcpy(&channel.sin_addr.s_addr, h->h_addr, h->h_length);
channel.sin_port= htons(SERVER_PORT);
c = connect(s, (struct sockaddr *) &channel, sizeof(channel));
if (c < 0) {printf("connect failed\n"); exit (-1);}
```

Il client inizializza una socket (s).
Channel è di tipo sockadd_in.

La *memset* pulisce *channel*.

Valorizziamo il campo *channel.sin_family* e *sin_port*.

Facciamo poi la *connect()*.

Se la *connect* è minore di 0 fallisce: il server non è attivo, il pc è disconnesso, l'IP è errato...

La *connect* non usa DNS.

```
/* Connection is now established. Send file name including 0 byte at end. */
write(s, argv[2], strlen(argv[2])+1);

/* Go get the file and write it to standard output. */
while (1) {
    bytes = read(s, buf, BUF_SIZE); /* read from socket */
    if (bytes <= 0) exit(0);        /* check for end of file */
    write(1, buf, bytes); /* write to standard output */
}
}
```

Arrivati qui significa che il client ha risposto e scriviamo sul socket col file descriptor (S) che è un intero e andiamo a scrivere come primo argomento il nome del file, seguito da \0.

Entriamo poi in un loop infinito dove si va a leggere il file e si scrive un carattere alla volta.

-----PARTE SERVER-----

```
#define SERVER_PORT 8080          /* arbitrary, but client & server must agree */
#define BUF_SIZE 4096    /* block transfer size */
#define QUEUE_SIZE 10

int main(int argc, char *argv[])
{
    int s, b, l, fd, sa, bytes, on = 1;
    char buf[BUF_SIZE]; /* buffer for outgoing file */
    struct sockaddr_in channel; /* holds IP address */
```

```

/* Build address structure to bind to socket. */
memset(&channel, 0, sizeof(channel)); /* zero channel */
channel.sin_family = AF_INET;
channel.sin_addr.s_addr = htonl(INADDR_ANY);
channel.sin_port = htons(SERVER_PORT);

/* Passive open. Wait for connection. */
s = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP); /* create socket */
if (s < 0) {printf("socket failed\n"); exit(-1);}
setsockopt(s, SOL_SOCKET, SO_REUSEADDR, (char *) &on, sizeof(on));

```

Il server prepara la struttura dati channel.

```

b = bind(s, (struct sockaddr *) &channel, sizeof(channel));
if (b < 0) {printf("bind failed\n"); exit(-1);}

l = listen(s, QUEUE_SIZE); /* specify queue size */
if (l < 0) {printf("listen failed\n"); exit(-1);}

```

Con la *bind()* collega il socket all'indirizzo e con la listen specifica la lunghezza della coda.

Finora il server non ha usato il DNS. Il DNS serve a quelli che non “ricordano” l’IP, come il client.

```

/* Socket is now set up and bound. Wait for connection and process it. */
while (1) {
    sa = accept(s, 0, 0); /* block for connection request */
    if (sa < 0) {printf("accept failed\n"); exit(-1);}

    read(sa, buf, BUF_SIZE); /* read file name from socket */

    /* Get and return the file. */
    fd = open(buf, O_RDONLY); /* open the file to be sent back */
    if (fd < 0) {printf("open failed\n"); exit(-1);}

```

Viene effettuata la accept() che può fallire se non ci sono più risorse o se la coda è piena.

Si fa poi la read, leggendo un buf_size alla volta.

```
        while (1) {
            bytes = read(fd, buf, BUF_SIZE); /* read from file */
            if (bytes <= 0) break; /* check for end of file */
            write(sa, buf, bytes); /* write bytes to socket */
        }
        close(fd);      /* close file */
        close(sa);      /* close connection */
    }
}
```

Legge il contenuto del file.

Esame: un server può funzionare senza DNS? Ovvio.
È usato solo se c'è una *gethostbyname*...⁷

⁷ DomandaEsame7

Lez 16

Domanda esame:⁸

Se devo mettere un dispositivo nuovo in rete di cosa ho bisogno?

- Un mezzo di comunicazione (interfaccia di rete, scheda di rete)
 - Router
 - Implementare il livello di rete e di trasporto.
-

ELEMENTI DEI PROTOCOLLI DI TRASPORTO

Quello che è vista come la chiave del livello di trasporto è il *3-wayhandshake*.

La parte di gestione della connessione, degli errori e di multiplexing usano gli stessi algoritmi che si usano per i *datalink*, perché quindi viene ricostruita questa parte anche per il livello di trasporto?

- La ridondanza introdotta al livello di trasporto e poi duplicata a livello applicativo serve perché c'è un link *end-to-end* mentre a livello datalink c'è un controllo singolo su ogni dispositivo di connessione.

⁸ DomandaEsame8

SIMILITUDINE DATALINK E LIVELLO DI TRASPORTO

Il data link layer ha una singola connessione e un punto di ingresso e di uscita.

A livello trasporto c'è un'astrazione superiore poiché non c'è un singolo canale ma le frecce della parte B della foto danno l'idea di più canali.

L'altra differenza è che a livello data link abbiamo il *mac address*, a livello di trasporto invece necessitiamo di due coppie: IP-porta sorgente e IP-porta destinazione.

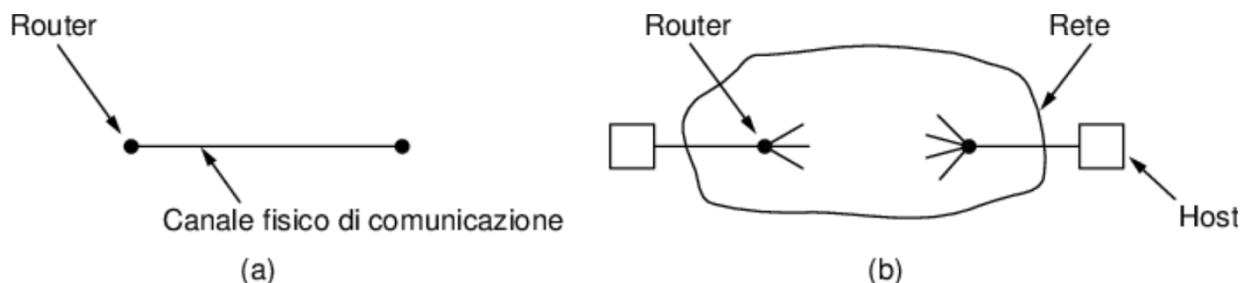


Figura6.7 (a) Ambiente del livello data link. (b) Ambiente del livello di trasporto.

ADDRESSING

Un host trasmette, con una porta di uscita che è collegata al NSAP (network service access points).

Il punto di accesso all'indirizzo di rete è l'indirizzo di rete è l'IP.

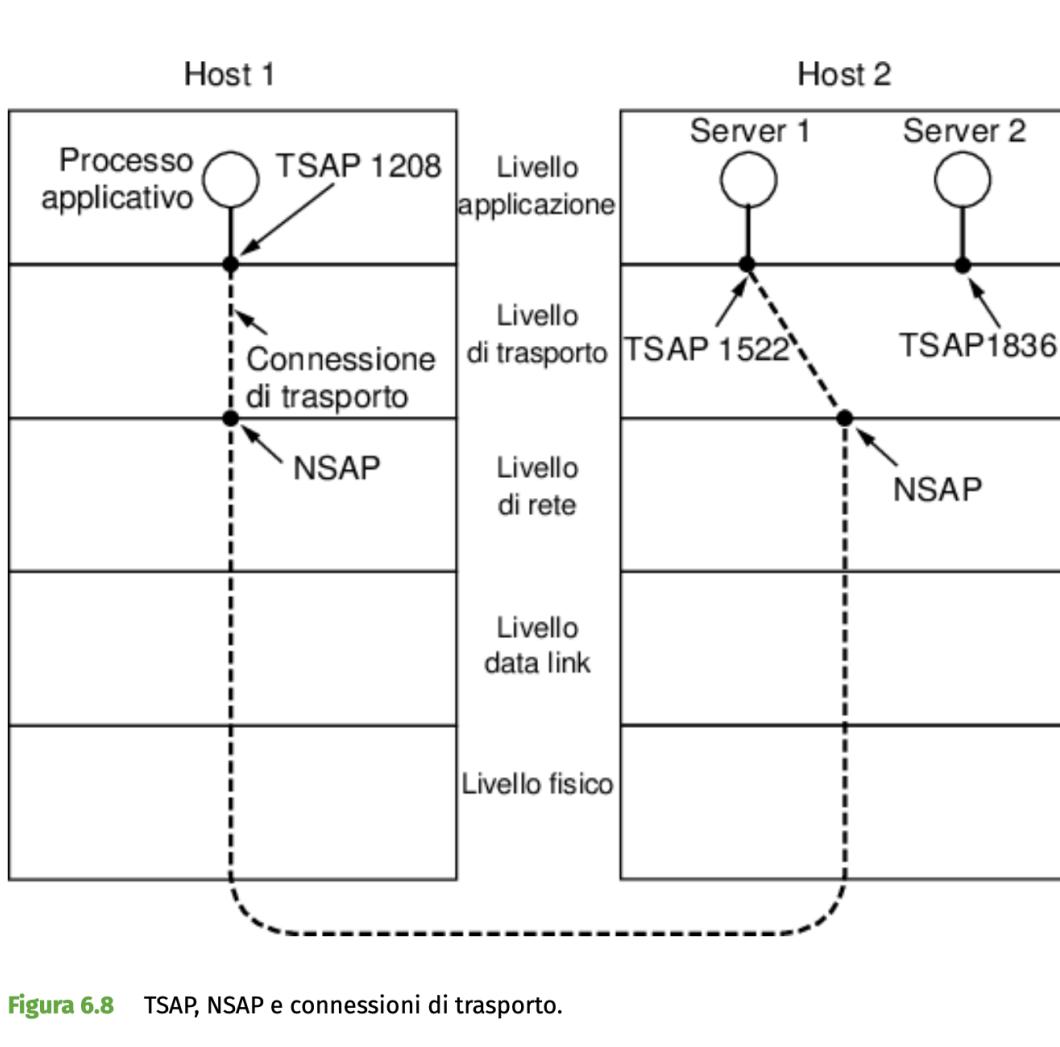


Figura 6.8 TSAP, NSAP e connessioni di trasporto.

!!il meccanismo funzionante per la mail è il DNS.

INETD

Meccanismo utile per controllare le risorse: ha un ruolo di “portiere”: quando un host richiede una connessione “lancia” il server interessato, senza di esso dovremmo avere in memoria in stato *running* tutti i server. Potrebbe capitare ad esempio che il server di stampa resti in *running* per un tempo indefinito e magari lo usiamo solo una volta ogni tanto...

STABILIRE UNA CONNESSIONE

Ci sono tecniche per restringere il TTL dei pacchetti poiché alcuni pacchetti potrebbero viaggiare senza mai arrivare e rallenterebbero il sistema (*starvation*).

3-WAY-HANDSHAKE

In a) c'è un h1 che manda una connection REQ mettendo un numero di serie, l'h2 dà l'ACK a quel numero di sequenza e a sua volta manda un numero di sequenza y. L'h1 manda a sua volta un pacchetto dati che contiene sia il numero x che y : così l'h2 è sicuro che anche la sua risposta sia arrivata. Può succedere che ci potrebbe essere una Conn Req duplicata (b) e quando arriva l'ACK dall'host2, l'h1 la rigetta. Ci potrebbe essere un caso in cui, oltre avere un duplicato di Conn Req anche un duplicato di ACK (c).

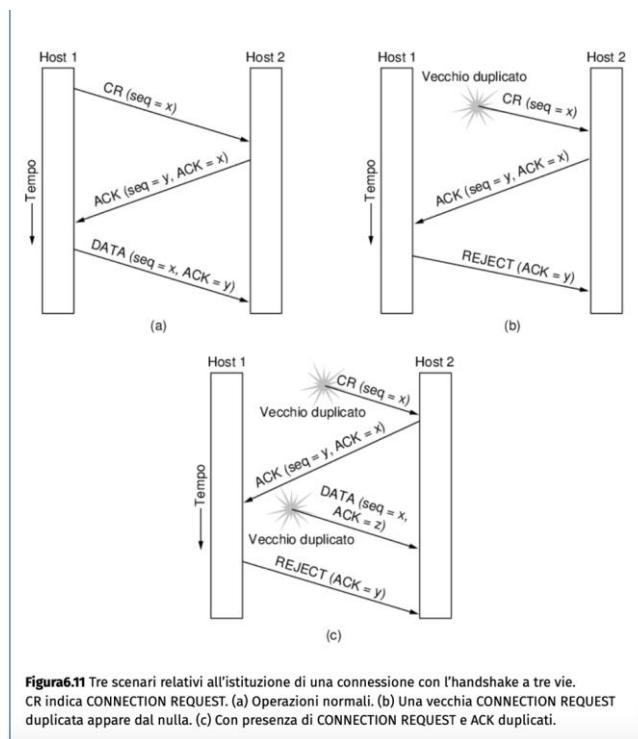


Figura 6.11 Tre scenari relativi all'istituzione di una connessione con l'handshake a tre vie.
CR indica CONNECTION REQUEST. (a) Operazioni normali. (b) Una vecchia CONNECTION REQUEST duplicata appare dal nulla. (c) Con presenza di CONNECTION REQUEST e ACK duplicati.

RILASCIARE UNA CONNESSIONE

Potremmo avere dei problemi poiché la disconnessione potrebbe essere fatta asimmetricamente (ognuno lascia la connessione senza contare dell'altro) o simmetrica (si disconnettono solo se sono d'accordo entrambi). L'h1 crea una richiesta di connessione, h2 dà un ACK, si scambiano i dati e a un certo punto l'h2 si disconnette: dalla disconnect req in poi, i dati che stavano transitando da h1 a h2 non arrivano più. Questa cosa è stata formalizzata con problema dei 2 eserciti:

- L'esercito blu deve attaccare il bianco, se attaccassero asimmetricamente perderebbero poiché sono in meno
- Se attaccassero simmetricamente vincerebbero.

Il problema è che si devono accordare, con un emissario che scende e passa per una zona avversaria per accordarsi con gli alleati. Questo emissario è la similitudine del pacchetto che potrebbe essere perso, proprio come l'emissario potrebbe essere catturato se passa per l'esercito bianco...

Supponendo che arrivi e dà il messaggio, la parte 1 aspetta la conferma e quindi l'emissario deve ritornare per dare conferma, rischiando nuovamente di essere catturato e non arrivare mai a destinazione.

La parte 2 ora vorrebbe avere conferma della ricezione della conferma da parte 1: si innesca un

procedimento ciclico irrisolvibile: se non si prende una decisione *timestamp* non si risolve il problema.

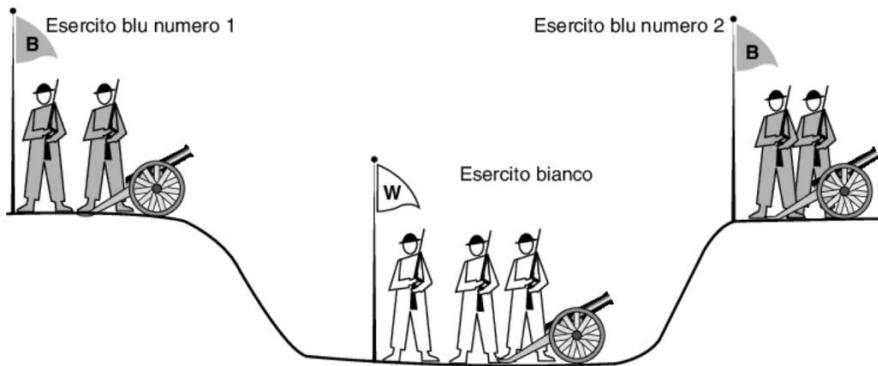


Figura 6.13 Il problema dei due eserciti.

La soluzione in generale è quindi avere un *timer* per evitare pacchetti pendenti che non sono mai arrivati. I *timer* possono funzionare se e solo se hanno una sorgente temporale valida e affidabile: devo avere dei *timestamp* (*marcatura temporale*).

È stato infatti inventato un protocollo *NTP* per aggiornare e sincronizzare l’orario delle nostre macchine.

INTERNET TRANSPORT PROTOCOLS: UDP
Progettato per inviare datagram IP senza stabilire una connessione.

Ricordiamo che ICMP è parallelo a IP mentre UDP e TCP sono “fratelli”.

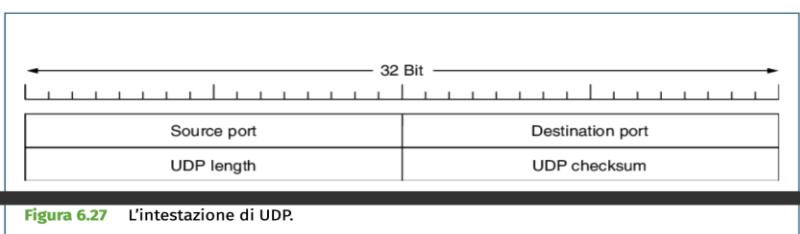


Figura 6.27 L'intestazione di UDP.

Le porte sono 16 bit mentre gli indirizzi sono 2^{32} .

REMOTE PROCEDURE CALL

Tramite rete posso invocare un processo su un altro computer. È inutilizzata ormai.

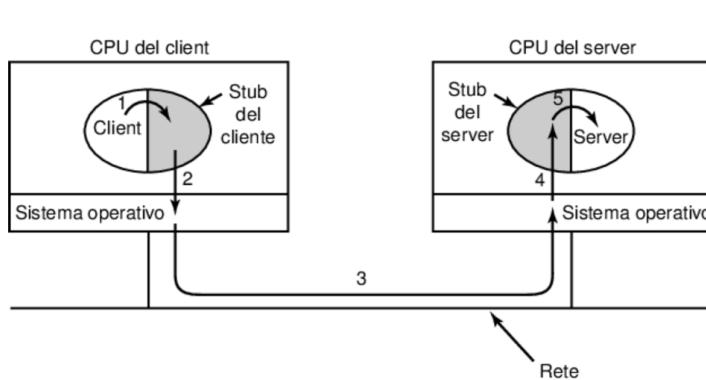


Figura6.29 I passaggi per effettuare una chiamata a procedura remota. Gli stub sono ombreggiati.

REAL TIME TRANSPORT PROTOCOL

Molto usato per lo streaming.

Il data link di solito è *ethernet*. Il kernel del SO dialoga con la parte di trasporto.

C'è il concetto di frame: la parte di payload del framing viene suddivisa in pacchetti.

Gli RTP header contengono il concetto di timestamp, sincronizzazione e numero di sequenza.

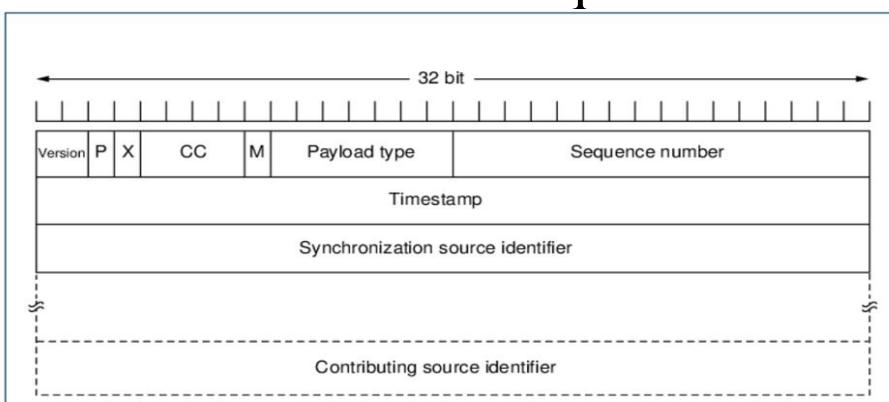
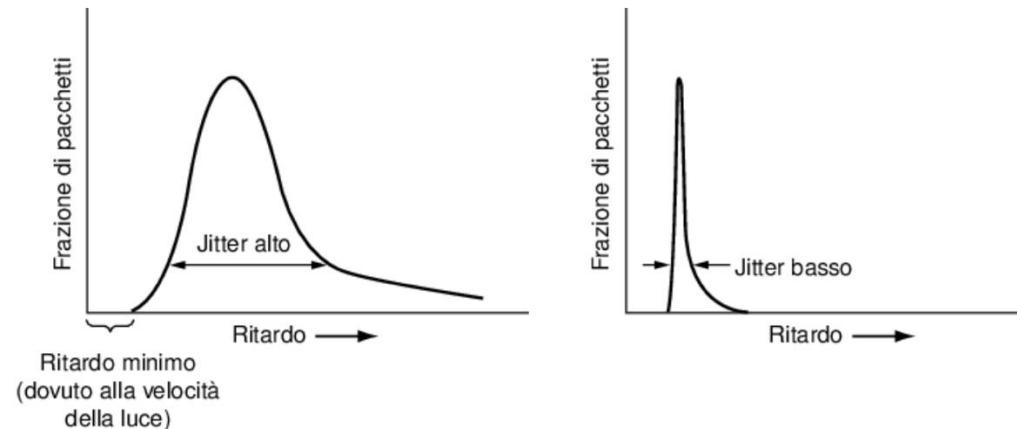


Figura6.31 L'intestazione RTP.

Se c'è una rete con alto *jitter*, si amplifica il delay. Questo succede quando abbiamo connessioni satellitari con un grosso delay, in quel caso i decoder devono comportarsi in maniera opportuna.



PROTOCOLLO TCP

È il meccanismo di trasporto più utilizzato. Progettato per fornire un flusso affidabile end-to-end su una rete inaffidabile.

Porta	Protocollo	Utilizzo
20, 21	FTP	Trasferimento di file
22	SSH	Login remoto, rimpiazzamento di Telnet
25	SMTP	Posta elettronica
80	HTTP	World Wide Web
110	POP-3	Accesso remoto alla posta elettronica
143	IMAP	Accesso remoto alla posta elettronica
443	HTTPS	Web sicuro (HTTP su SSL/TLS)
543	RTSP	Controllo di riproduttori multimediali
631	IPP	Condivisione di stampanti

Figura 6.34 Alcune porte assegnate.

Le porte sono assegnate dall'istituzione IANA.

Per vedere chi c'è su una porta effettivamente si usa il comando *curl o telnet seguito da un sitoweb*.

Domanda esame: MTU ethernet 1500.⁹

Lez 17

Si usa UDP o TCP di più? Visto che c'è una grande migrazione verso i servizi IP per l'intrattenimento, UDP sta ritornando in auge. Questo implica un maggiore *overhead* lato client poiché le caratteristiche che non dà UDP devono essere garantite a livelli superiore.

INTERNET TRANSPORT PROTOCOL: TCP

Esistono 700 servizi registrati allo IANA. Ma non c'è certezza che sulla porta 443 ad esempio ci sia HTTPS ma ci potrebbe essere un server mail o un traffico HTTP. Supponiamo che per qualche motivo si voglia mettere un server HTTPS sulla porta 80. Di default il browser andrà sulla porta 443, quando vado sull'URL dobbiamo specificare porta e servizio:

http://nomeservizio.com:num_porta.

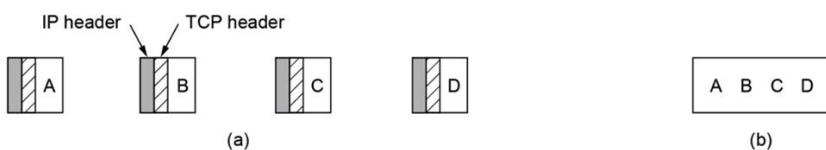
<https://dominio:80>.

Se ho un dominio e voglio sapere quali sono i servizi attivi si prova a fare una connessione su tutte le porte. Se a seguito di questa scansione non c'è nessuna porta aperta, siamo sicuri che sia così o potrebbe esserci un'altra cosa?

⁹ DomandaEsame9

- Ogni server può decidere a chi rispondere. Se a seguito della scansione ottengo un elenco di porte sono raggiungibili, se così non fosse ci potrebbe essere un servizio che potrebbe essere configurato su una *subnet specifica*.
- Potrebbe esserci un *firewall* di mezzo.

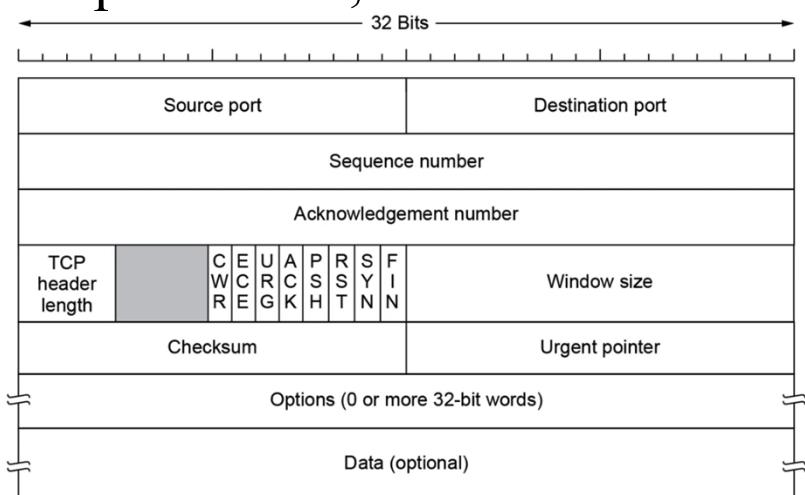
In base all'incapsulamento abbiamo vari header.



(a) Four 512-byte segments sent as separate IP datagrams. (b) The 2048 bytes of data delivered to the application in a single READ call.

HEADER TCP

Gli indirizzi degli Host non ci sono perché sono nell'header precedente, nell'header IP che lo contiene.



La potenza di TCP sta nell'uso dei flag molto potenti, come quello *URG(urgente)*, *RST(reset)*, *FIN(fine)*, *PSH (push, come fflush)*, *SIN e ACK (per sincronizzazione)*.

Gli indirizzi quindi sono a livello superiore e non nell'header.

DOMANDA: quanto sono lunghi gli IP nell'header TCP? Non ci stanno. Per le porte invece ho 16bit.¹⁰

“ il comando *nmap* ci dice le porte aperte che ci sono, se la porta è TCP o UDP”

```
bash-3.2# nmap elearning.uniparthenope.it
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-16 11:42 CET
Nmap scan report for elearning.uniparthenope.it (192.167.9.76)
Host is up (0.0028s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
3306/tcp  open  mysql
```

¹⁰ DomandaEsame10

Lez 18

Quando si parla di TCP si parla di segmenti.

Quando ci colleghiamo a Instagram, cosa succede dal punto di vista del TCP. Se registrassi l'accesso tramite app o browser, cosa accade dal punto di vista dei segmenti TCP?

- Si crea un canale identificato da ip porta sorgente/destinazione e il protocollo (*quintupla*).
- Il client richiede la connessione al server sulla porta 443, il DNS risolve il dominio e con la 3-way-handshake si instaura una connessione...
- Quando facciamo un'attività di questo tipo, abbiamo una sequenza di segmenti che per poter essere “consumati” a livello superiori, devono essere messi insieme. Ci sono molti segmenti che quindi devono essere messi insieme così può ricostruire il flusso di dati e si ottiene il pezzo di comunicazione (accesso al servizio o altri aspetti). Sono comunicazioni full-duplex (partono dati da client e dal server). Ci sono diversi modi per tirar fuori i vari servizi: se intercettassimo quello che esce dal nostro pc con dei programmi, otterrei tanti flussi di dati con tanti segmenti che appartengono a diversi protocolli e programmi (ad esempio i browser usano HTTPS). Da quel flusso come faccio a distinguere i vari servizi (mail, teams, netflix...) da quell'unico flusso di dati?

Prendo la quintupla e rimetto insieme i segmenti per creare i flussi: ci sono diversi modi, con wireshark e da commandline (*TCP Flow*). Se applico una metodologia di questo tipo posso vedere dei protocolli “strani” come un *cryptomining* e agire di conseguenza.

HEADER TCP

L’header IP è grande 20 byte, quello TCP è 20 byte di base. In realtà ci sono parti opzionali.

Quando si usano normali applicazioni, i flag TCP non sono molto usati. I programmati si devono attenere a quello che c’è scritto nei protocolli o possono fare altro?

- Nessuno vieta di manipolare questi bit. *Ad esempio il sin flooding.*
- Per evitare ciò si instaura un firewall che controlla il flusso (ad esempio conta quanti flag ci sono in più ecc...) in modo da scartare o bannare i pacchetti. Il firewall deve agire a livello di rete ma non sull’host poiché è computazione e crea overhead.

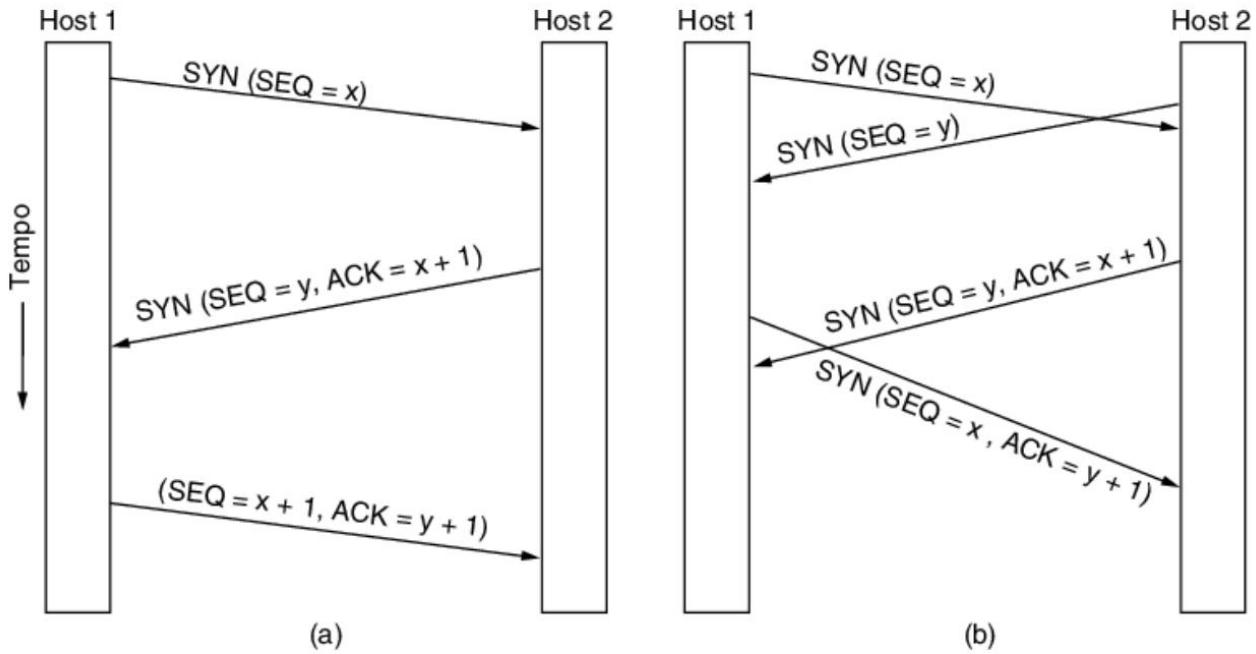


Figura 6.37 (a) Instaurazione di una connessione TCP nel caso normale. (b) Instaurazione simultanea da entrambi i lati.

Nel TCP chi consuma una connessione è chi instaura la connessione (*client*).

Ecco perché le connessioni domestiche sono ADSL (asimmetriche).

MODELLO DELLA GESTIONE DELLA CONNESSIONE TCP

Stato	Descrizione
CLOSED	Nessuna connessione è attiva o in sospeso
LISTEN	Il server è in attesa di una chiamata in ingresso
SYN RCVD	È arrivata una richiesta di connessione; in attesa di ACK
SYN SENT	L'applicazione ha iniziato ad aprire una connessione

Stato	Descrizione
ESTABLISHED	Il normale stato di trasferimento dei dati
FIN WAIT 1	L'applicazione ha detto di aver terminato
FIN WAIT 2	L'altro lato ha accettato il rilascio
TIME WAIT	Attende la scadenza di tutti i pacchetti
CLOSING	Entrambi i lati hanno cercato di chiudere contemporaneamente
CLOSE WAIT	L'altro lato ha iniziato il rilascio
LAST ACK	Attende la scadenza di tutti i pacchetti

- C'è lo stato di closed dove non ci sono connessioni pendenti.
- Listen: server in attesa
- SYM RCVD: richiesta di connessione arrivata e si è in attesa dell'ACK
- SYN SNT: app ha aperto una connessione
- (vedi la tabella su).

Con *netstat -a* possiamo vedere questi flag.

ESAME: in quali degli stati del TCP ci sono dati che transito? Quando la connessione è stabilita ma anche durante il 3-way-hsk.

TCP SLIDING WINDOW

I socket usano I buffer(uno che scrive e uno che legge) che sono di una certa grandezza. A un certo punto il buffer si satura e si crea un ACK.

I buffer vengono scritti e letti (consumati).

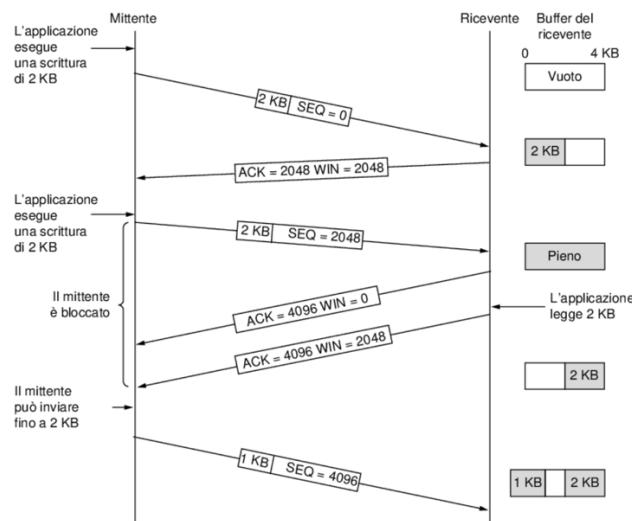


Figura 6.40 Gestione della finestra in TCP.

TCP TIME MANAGEMENT

TCP da solo non funziona, viaggia su IP che a loro volta viaggiano su datalink.

Se siamo su una rete ethenet la densità di probabilità di ACK è più stretta (veloce) mentre quando andiamo a misurare la densità di probabilità su TCP abbiamo probabilità più basse e spalmate sul tempo.

TCP è più lento perché passa su n datalink e alcuni possono essere rallentati.

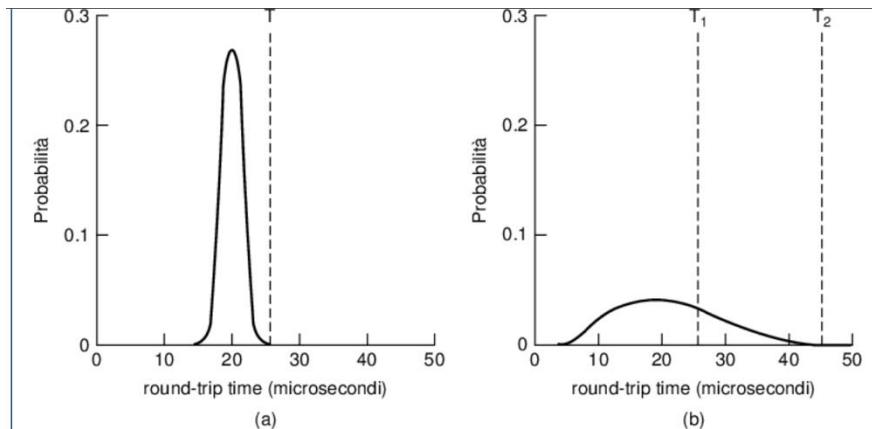


Figura 6.42 (a) Funzione di densità di probabilità dei tempi di arrivo degli acknowledgement nel livello data link. (b) Funzione di densità di probabilità dei tempi di arrivo degli acknowledgement TCP.

TCP CONGESTION CONTROL

Che succede quando siamo ci sono link veloci o lenti?

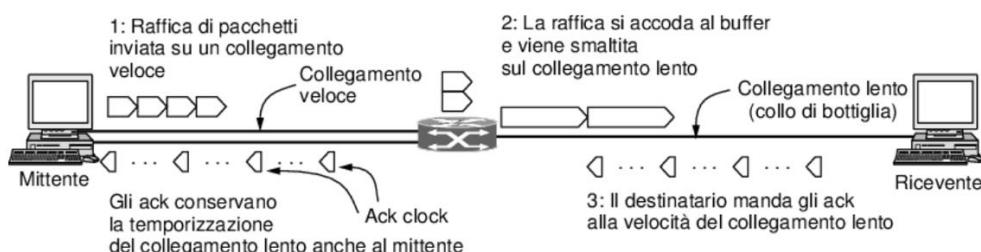


Figura 6.43 Una raffica di pacchetti da una sorgente e l'ack clock di ritorno.

Se ho una linea veloce e a un certo punto ho un collo di bottiglia (alta densità), la linea veloce deve adattarsi a quella lenta per sincronizzare e comparare i tempi.

TCP ha una partenza lenta (incrementa la finestra di trasmissione lentamente) ma se dovesse succedere una problematica di qualsiasi genere, immediatamente c'è un abbassamento del *rate* e si ricomincia lentamente a ritrasmettere.

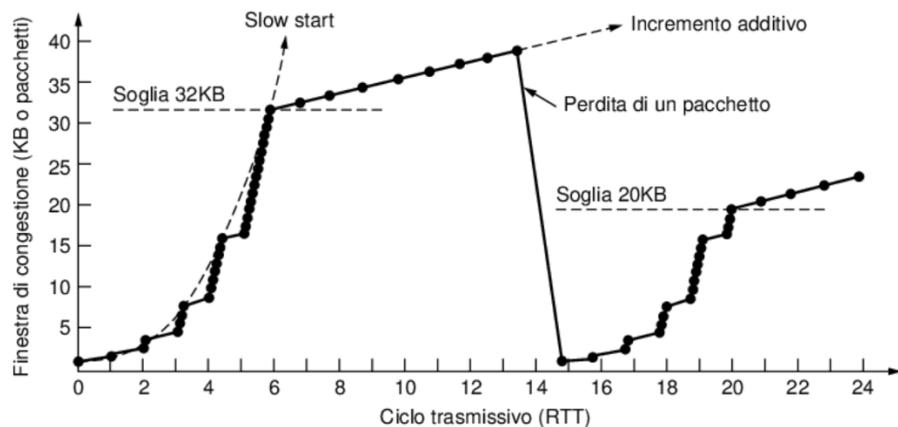
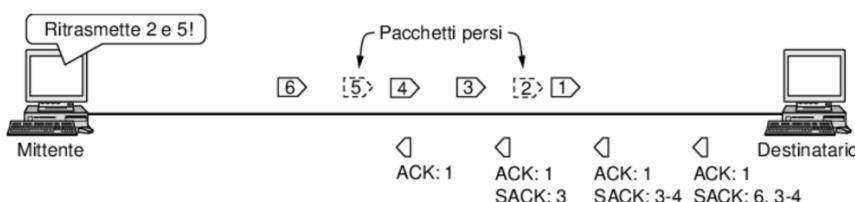


Figura 6.46 Slow start seguito da incremento additivo in TCP Tahoe.

...

Un altro concetto utile per problemi di congestione è il fatto che potrei mandare ACK selettivi: si è perso qualche pacchetto, chi deve inviare ritrasmette solo quelli persi e si richiede la ritrasmissione.



Il sender vede che non gli sono arrivati degli ACK su determinati pacchetti e quindi li può ritrasmettere.

FUTURO DI TCP

TCP si è evoluto molto, ogni SO faceva una sua implementazione.

TCP ha ancora dei problemi, come quello della sicurezza che non viene risolto dal TCP, solo IP SEC gestisce la parte di sicurezza, altrimenti viene demandato al livello applicativo come *https*, *tls*, *http*... C'è un approccio conservativo “se le cose funzionano, lasciamole andare”.

!! PER QUANTO RIGUARDA l'HEADER: Ci sono altri 4bit (vuoti) che non sono utilizzati, questo significa che nonostante TCP si sia evoluto non è stato considerato questo spazio. !!

PROBLEMI DI PERFORMANCES NELLE RETI

Chi fa reti dovrebbe vedere se i suoi dispositivi sono sovraccarichi e perché.

C'è il problema del sovraccarico sincrono: un pezzo di rete manda delle tempeste di *broadcast* (succede quando connetto un cavo di rete a due porte ethernet e genero una tempesta).

MISURARE LE PERFORMANCE

Bisogna misurare il throughput e capire dove esistono dei colli di bottiglia. Sostanzialmente la velocità è meno rilevante della performance di rete: gli user

vogliono la *quality of experience* senza preoccuparsi tanto della velocità della rete (ad esempio si preferisce vedere un film di alta qualità senza pensare alla velocità). Si può “simulare” una velocità anche grazie a meccanismi di *cache*.

Le best practices sono:

- Usare più connessioni TCP per saturare la capacità di accesso.
- Vedere i tempi in parallelo per vedere come funzionano destinazioni multiple.

Misurare la QoE:

- Misurare la felicità di un utente col servizio.
- Considerazione umana.
- Feedback utenti.

STRUTTURA PER HOST VELOCI

Se abbiamo una connessione di 2.5gigabit ma il nostro host ne supporta 1 è uno spreco.

In un host si dovrebbe minimizzare i *content switches* perché genera overhead. Inoltre, invece di fare il recover, si deve evitare la congestione ed evitare timeout.

In breve: le componenti importanti delle connessioni moderne sono ormai gli host.

Lez 19 → Livello Applicazione

Siamo sulla parte più alta della pila, quello applicazione.

Il livello applicazione consente di usare Internet e come primo protocollo applicativo vediamo il DNS.

DNS

Permette di “non ricordarci staticamente l’associazione nome a dominio – indirizzo IP”.

Se un DNS non è disponibile o settato male, la maggior parte delle applicazioni non funzionano perché fanno query DNS.

Altre invece non la usano, come ad esempio *Skype*, prima che passasse a *Microsoft*.

!! !!ESAME: ¹¹qual è il protocollo di base per poter usare internet? Il DNS perché senza di esso è complicato usare Internet. (ricordare la diff. Tra internet e web).

Protocolli applicativi come *http* usano il DNS: su un singolo IP è possibile mettere più di un webserver che eroga più siti di più domini e questa cosa viene implementata grazie a una modifica del protocollo http dove il client in maniera esplicita menziona l’host da visionare “host:nomesito.it”. !!!!

¹¹ DomandaEsame11

Supponiamo di fare una query http verso un indirizzo IP: che succede?

- Su quell'IP ci sarà un servizio http che risponde.
 - Se non mettiamo https forse non veniamo redirectati.
 - Non appare il sito perché ci sarà un host primario che ha una *document root* che dà un messaggio di errore che non abbiamo permessi.
 - Se non ci sono virtual host allora se scrivo un IP senza passare per DNS funziona.
-

STORIA

Con ARPANET c'era un file *hosts.txt* che era una lista con tutte le associazioni Host-IP.

Nel 1983 fu inventato il DNS:

- Schema gerarchico di nomi.
- Mapping dei nomi degli Host (FQDN¹²) con gli IP.
- Database distribuito con lo schema di *naming*.

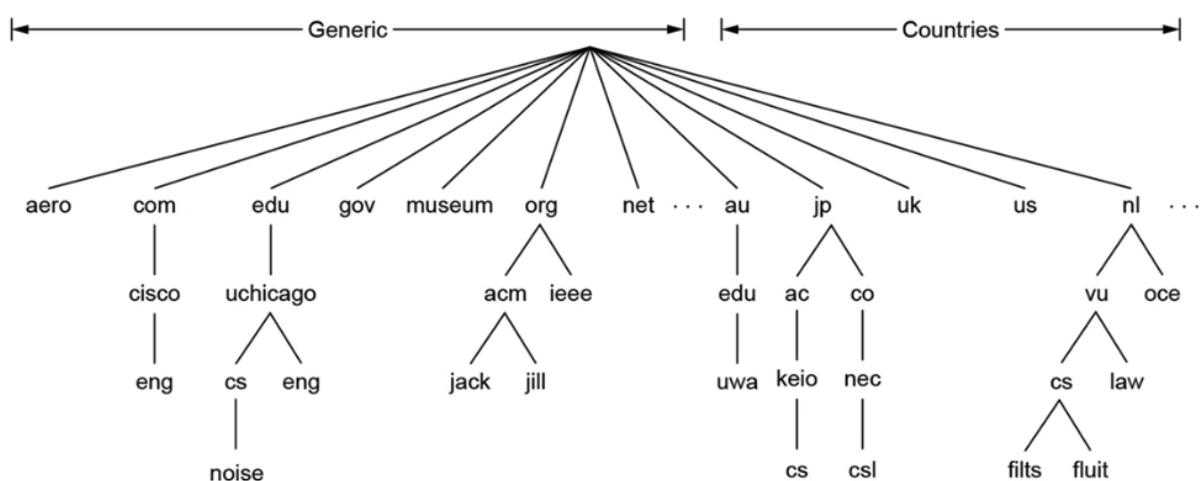
Funzionamento:

- Il resolver manda una query contenente il nome al dns
- Il dns resolver fa una *lookup ricorsiva* del nome, facendo una serie di richieste a più DNS resolver.

¹² fully qualified domain name

- La risoluzione si ferma quando siamo arrivati ad avere l'IP del nome che volevamo risolvere.
 - La gethostbyname riceve i risultati del processo
 - Domande e risposte viaggiano in pacchetti UDP
 - I programmi possono usare gli IP per comunicare con gli host che rispondono al nome DNS che hanno risolto.

DNS NAME SPACE E GERARCHIA



Ricordiamo che il punto è la *radice*.

I domini sono formati da *parte generica e dai Paesi*.

Vediamo la struttura gerarchica:

- Ad esempio abbiamo il dominio di 1° livello com
 - Il 2° livello cisco
 - Il 3° livello eng

I nodi foglia li possiamo vedere o come un host o un sottodomainio.

Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No
aero	Air transport	2001	Yes
biz	Businesses	2001	No
coop	Cooperatives	2001	Yes
info	Informational	2002	No
museum	Museums	2002	Yes
name	People	2002	No
pro	Professionals	2002	Yes
cat	Catalan	2005	Yes
jobs	Employment	2005	Yes
mobi	Mobile devices	2005	Yes
tel	Contact details	2005	Yes
travel	Travel industry	2005	Yes
xxx	Sex industry	2010	No

Nel 2020 sono diventati più di 1200 i gTLDs (*global Top Domains*).

Come funziona se una persona vuole intestarsi un dominio?¹³



Ci si registra all'ICANN.

¹³ DomandaEsame13

DNS QUERY E RISPOSTE

Ci sono domande e risposte fatte dai sistemi.

Vediamo i principali tipi di record di una risorsa DNS.
Se facciamo una richiesta a un IPv4 si fa una richiesta
“A”, se IPv6 “AAAA”.

Il DNS può essere interrogato per diverse cose.

Vedremo che quando dobbiamo inviare una mail a un dominio, viene fatta una query DNS ricorsiva, al fine di trovare il *mail exchange* che gestisce il dominio della mail (MX).

Col DNS posso sapere il nome a dominio di una sottozona (NS): ogni dominio quindi può essere identificato con una zona, ogni zona ha una sottozona (ad esempio la zona “.it” ha infinite sottozoni cioè tutti i siti “.it”).

CNAME è una sorta di *alias*.

TXT sono commenti.

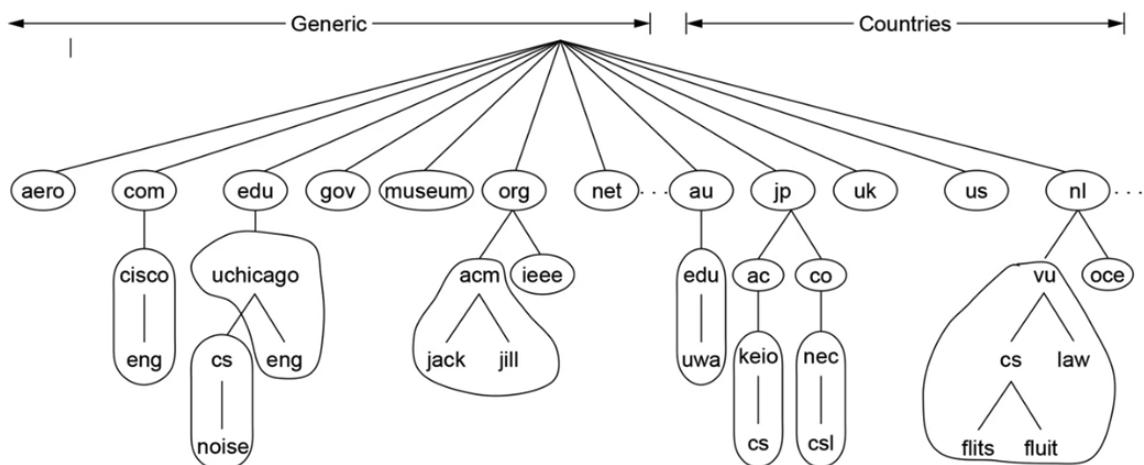
SPF specifica parametri al sistema di posta elettronica
PTR è una sorta di reverse lookup.

SOA è una struttura dati contenente dei campi che spiegano dopo quanto tempo viene aggiornato il dominio in secondi e il suo responsabile (importante).

Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

RISOLUZIONE DI NOMI

In figura vediamo le zone e sottozone.



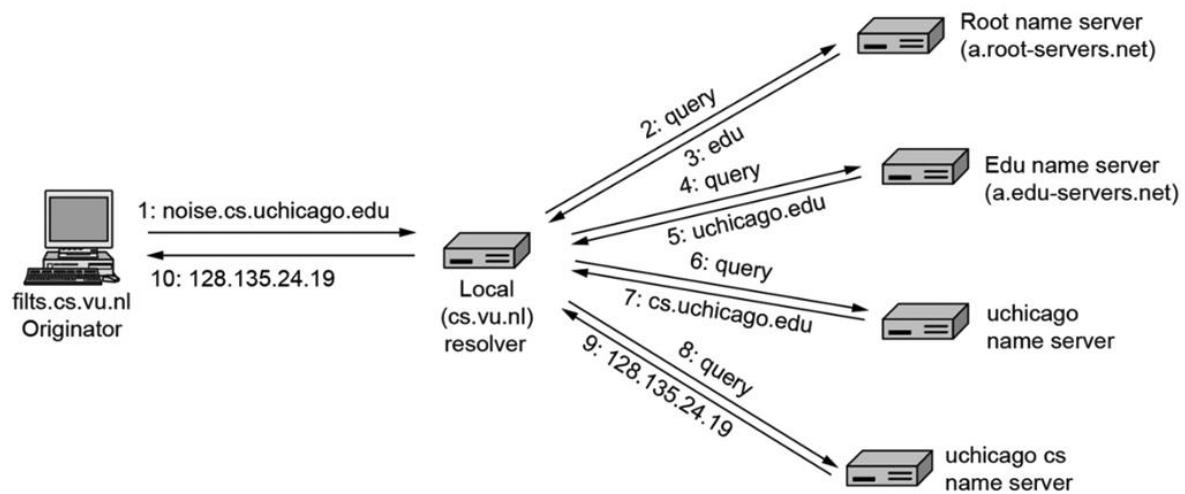
Part of the DNS name space divided into zones (which are circled)

Chi fa la richiesta vuole sapere info relative a
noise.cs.unicicago.edu. come funziona?

- Il computer da cui si fa richiesta ha una configurazione passata dal DHCP. Ha un IP, DNS e subnet mask. Il dns in questo caso è un computer, il *resolver*.
- La prima query il computer la fa al suo resolver. Il resolver potrebbe essere anche sul pc locale.
- Si comincia dal punto che è la *root*. Si chiede al dominio punto qual è il nameserver che gestisce il top level domain. Il DNS dirà chi gestisce il dominio di primo livello e restituisce la risposta al resovler.
- Il resolver va a interrogare all'edu server chi gestisce uchicaho.edu e gli risponderà con “cs.uchicago.edu”
- A questo punto la query è fatta all'host che è il name server, c’è una query di tipo A, e avrà una risposta finale.

DOMANDA ESAME: Il dns che fornisce il DHCP è un IP o un nome completo a dominio? IP. ¹⁴

¹⁴ DomandaEsame14



Example of a resolver looking up a remote name in 10 steps

Lez 20 → ELECTRONIC MAIL

È il protocollo più vecchio “sopravvissuto” a livello applicativo. C’era prima del DNS.

È rimasto il mezzo preferito ed ha ricevuto molti upgrade a riguardo.

La *pec*, posta elettronica certificata, è nata in Italia. Prima, quando non c’era il DNS, quando si inviava una mail bisognava specificare l’instradamento della mail, usava una sintassi diversa da quella attuale proprio per specificare la parte host dell’e-mail.

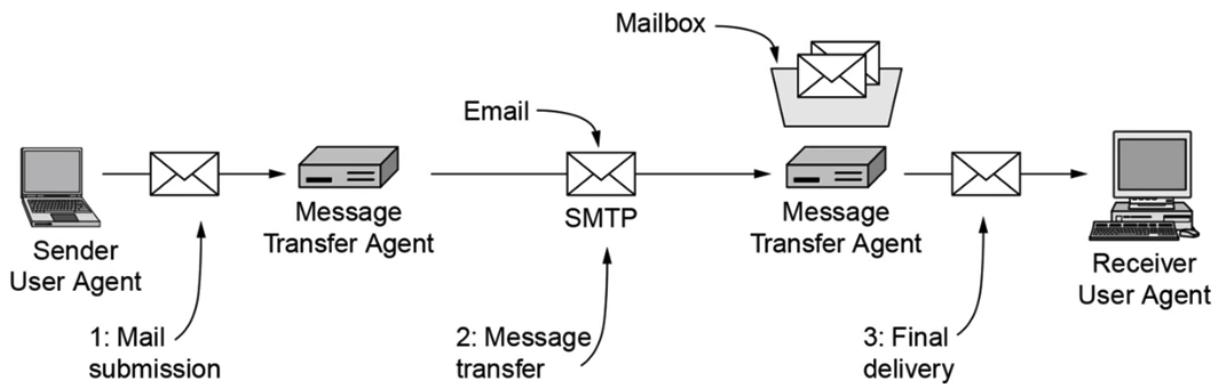
Ci sono dei programmi, *user agent*, o interfacce web dedicate per questo servizio.

La posta elettronica è un servizio asincrono: un server potrebbe essere non raggiungibile per qualche motivo e arriva in ritardo. È un servizio affidabile poiché al 95% verranno consegnate poiché ci sono specifiche che contattano periodicamente il server di posta finché essa non viene consegnata.

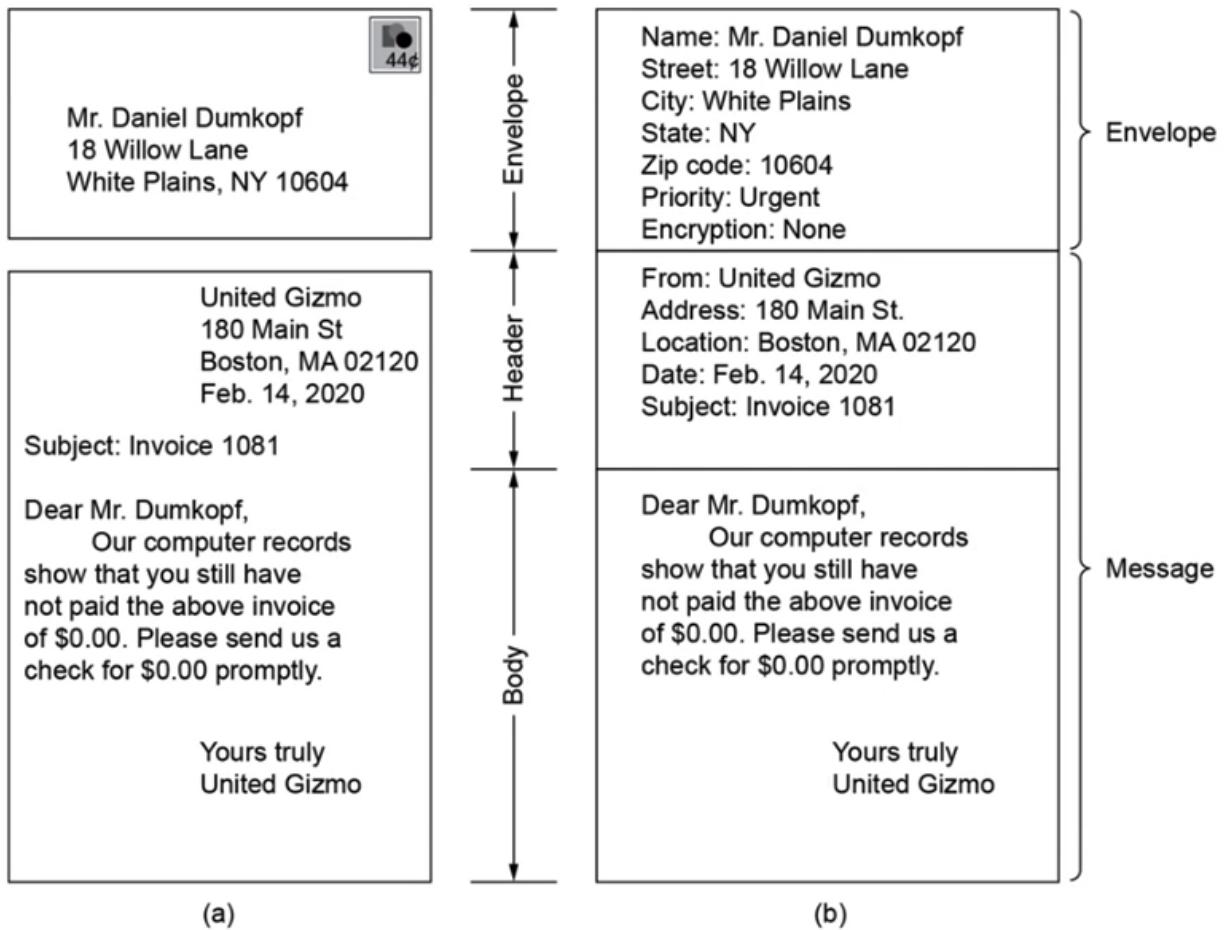
ARCHITETTURA E SERVIZI

C’è un mittente e un destinatario. Il mittente invia la mail all’MTA, l’MTA fa il parsing all’elenco dei destinatari. Il protocollo SMTP effettua il trasferimento vero e proprio. Una volta arrivato all’MTA di destinazione, avviene la consegna finale. Il protocollo SMTP è a 7bit. I messaggi sono ASCII puro e la mailbox non è altro che una sequenza di

blocchi testuali. Viene formattata in un certo modo, così *l'user agent* capisce dove inizia e finisce la mail. Il fatto che la mailbox sia a 7bit porta alcuni problemi: dove ci sono i file di singoli user con le loro mailbox, se qualcuno può accedere a quel server, può vedere tutte le mailbox e cambiare il contenuto.

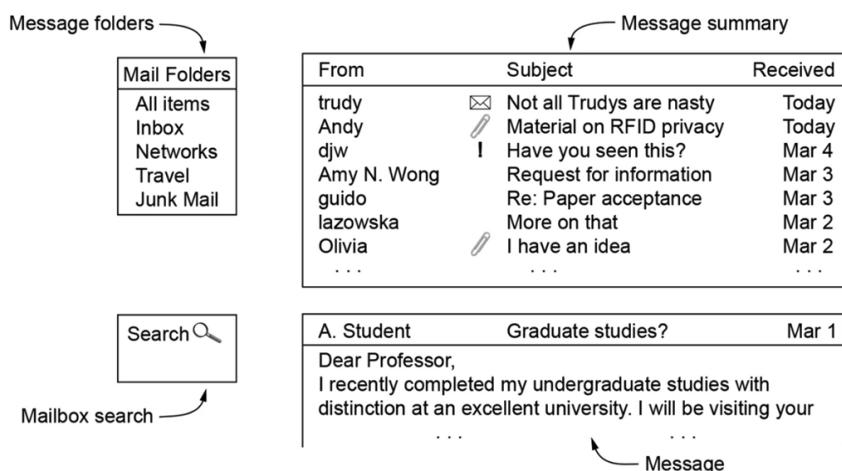


Come i messaggi di posta cartacea, anche l'e-mail ha una “busta” con l’indirizzo e il mittente, un header e un body. La parte header dell’e-mail è parte già della mail. Tecnicamente la parte di *envelope* viene messa dai programmi automaticamente.



Envelopes and messages. (a) Paper mail. (b) Electronic mail.

USER AGENT
È il programma o interfaccia di posta elettronica.



Ogni interfaccia o programma mail ha un’opzione nascosta che consente di vedere l’*header* della posta elettronica.

Guardando l’*header* possiamo vedere dov’è stata inviata la mail e tutti gli host che ha attraversato.

Con alcuni userAgent, se inviamo una mail con un IP locale, vedendo l’*header* si rischia di risalire all’IP: se siamo sulla stessa rete con un meccanismo di *spoofing* potremmo “rubare” il *Mac Address* del mittente.

FORMATI DEI MESSAGGI

RFC 5322 → è l’internet message format.

Abbiamo detto che la mail lavora a 7bit e non trasferisce dati binari ma c’è la possibilità di condividere immagini e allegati: questa cosa potrebbe sembrare un controsenso ma grazie al *MIME* è possibile aggiungere allegati multimediali o file .zip alla mail; la versione sicura di *MIME* è *S/MIME*

Se scriviamo una mail in cinese, i caratteri cinesi non sono codificabili in ASCII puro in 7 bit ma sono in UNICODE a 16bit entra in funzione la parte *MIME*.

Ci sarà poi una procedura SW che converte ciò che non è a 7bit in 7bit e viene codificato in un formato *based 64*: è un modo per trasferire informazioni binarie in caratteri ASCII.

Vediamo alcuni header *RFC 5322*:

Header	Meaning
To:	Email address(es) of primary recipient(s)
Cc:	Email address(es) of secondary recipient(s)
Bcc:	Email address(es) for blind carbon copies
From:	Person or people who created the message
Sender:	Email address of the actual sender
Received:	Line added by each transfer agent along the route
Return-Path:	Can be used to identify a path back to the sender

- Nel campo *sender* potrei scrivere ciò che voglio se so come farlo: essendo ASCII potrei cambiare il mittente.
- *Received*: ci sono almeno 2 campi: chi riceve e chi consegna.
- *Return-Path*: è messo ugual al *from* cioè chi invia la mail.

Header	Meaning
Date:	The date and time the message was sent
Reply-To:	Email address to which replies should be sent
Message-Id:	Unique number for referencing this message later
In-Reply-To:	Message-Id of the message to which this is a reply
References:	Other relevant Message-Ids
Keywords:	User-chosen keywords
Subject:	Short summary of the message for the one-line display

Il campo *data* è modificabile: potrei scriverlo appositamente errato per simulare un invio di una mail in una data differente.

Header	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Human-readable string telling what is in the message
Content-Id:	Unique identifier
Content-Transfer-Encoding:	How the body is wrapped for transmission
Content-Type:	Type and format of the content

MIME CONTENT TYPES AND SUBTYPES

Type	Example subtypes	Description
text	plain, html, xml, css	Text in various formats
image	gif, jpeg, tiff	Pictures
audio	basic, mpeg, mp4	Sounds
video	mpeg, mp4, quicktime	Movies
font	otf, ttf Fonts for typesetting	
model	vrml	3D model
application	octet-stream, pdf, javascript, zip	Data produced by applications
message	http, RFC 822	Encapsulated message
multipart	mixed, alternative, parallel, digest	Combination of multiple types

I messaggi contengono una parte testuale, immagini, audio... questo viene “riassunto” con *multipart*.

MESSAGE TRANSFER

SMTP è un protocollo a livello applicativo e ha delle keyword per i programmi che usano questo protocollo. Una di quelle più usate è *SIZE*: succede quando il messaggio inviato è troppo grande.

Keyword	Description
AUTH	Client authentication
BINARYMIME	Server accepts binary messages
CHUNKING	Server accepts large messages in chunks
SIZE	Check message size before trying to send
STARTTLS	Switch to secure transport (TLS; see Chap. 8)
UTF8SMTP	Internationalized addresses

Un'altra è AUTH dove il client chiede l'autenticazione.

STARTTLS comincia una comunicazione in chiaro sulla porta 25 e fa *un'autoswitch* sulla versione sicura di TLS.

CONSEGNA FINALE

La consegna viene effettuata usando dei protocolli specifici ad esempio POP(*post office protocol*) e IMAP che ha migliorato il protocollo POP.

La porta della mail è 25 mentre di IMAP è 143. Il nostro programma di posta, oltre a essere un client SMTP (quando invio una mail parlo protocollo SMTP), è anche un client IMAP o POP.

Il client si collega al server e invia una serie di comandi. Ci sono alternative webmail più veloci.

Command	Description
CAPABILITY	List server capabilities
STARTTLS	Start secure transport (TLS; see Chap. 8)
LOGIN	Log on to server
AUTHENTICATE	Log on with other method
SELECT	Select a folder
EXAMINE	Select a read-only folder
CREATE	Create a folder
DELETE	Delete a folder
RENAME	Rename a folder
SUBSCRIBE	Add folder to active set
UNSUBSCRIBE	Remove folder from active set
LIST	List the available folders
LSUB	List the active folders
STATUS	Get the status of a folder
APPEND	Add a message to a folder
CHECK	Get a checkpoint of a folder
FETCH	Get messages from a folder
SEARCH	Find messages in a folder
STORE	Alter message flags
COPY	Make a copy of a message in a folder
EXPUNGE	Remove messages flagged for deletion
UID	Issue commands using unique identifiers
NOOP	Do nothing
CLOSE	Remove flagged messages and close folder
LOGOUT	Log out and close connection

IMAP (version 4) commands

Lez 21

IL WORLD WIDE WEB

Ci sono oggetti statici e pagine dinamiche, grazie ai vari strumenti di programmazione dei siti web.

Il classico esempio sono i social: nella mainpage prima del login non c'è nulla, entrati poi c'è un mondo completamente dinamico.

ARCHITETTURA

A sinistra abbiamo l'oggetto del nostro navigare (pagine web) poi il web browser al centro e altri computer, tra cui il web server a destra.

Ci sono una serie di altri server che a volte non “vediamo”, solo facendo attenzione al codice sorgente possiamo notarli.

Quando facciamo richieste http o https a un web server, abbiamo altre richieste fatte autonomamente dal sito che stiamo navigando. Questo comporta un rallentamento della pagina che stiamo visitando.

Ovviamente ciò che vediamo è renderizzato in modo diverso a seconda dello schermo del dispositivo.

C'è quindi la fase di fetch dove vengono prese le pagine e il browser ce le fa visualizzare. Una singola

richiesta in effetti si traduce in molteplici richieste che non sono soltanto su quella pagina che vediamo. Per avere una sequenza di pagine coerenti bisogna fare tante richieste.

CLIENT SIDE

Alcune domande da porsi:

- Dov'è localizzata una determinata pagina?
- Come può essere acceduta e come si chiama?
- “Perché con un browser visualizzo pagine in modo corretto e con altri no?”
 - ! Poiché i motori di rendering potrebbero essere poco potenti e i linguaggi di scripting possono essere poco noti.

➤ Come si capisce dove è localizzata una pagina?

Usiamo whois → Ci sono due strade:

- Dominio: mi dice chi è il responsabile.
- IP: è fondamentale.

Un IP potrebbe ospitare più siti web, se ho un sito web come faccio a capire se su quell'IP c'è solo il server per noi?

- Ho un sito web, ricavo l'IP, gli accedo manualmente, se quello che mi compare non è il mio sito web ma una pagina diversa vuol dire che il nostro sito è un virtual host.

È difficile stabilire dov'è collocato un sito e addirittura visualizzare la stessa cosa...

Esempio: ho due pc collegati su una pagina, avrò due pubblicità diverse poiché anche da dove accediamo c'è una visualizzazione del sito differente.

È molto semplice implementare uno script con un userAgent che in base a chi accede fa visualizzare info diverse.

PASSI CHE VENGONO ESEGUITI QUANDO SI SELEZIONA UN LINK

- Clicco sul link
- Il browser stabilisce l'url.
- Viene fatta una query DNS e viene ricevuta una risposta.
- Il browser fa una connessione TCP.
- Invia una richiesta http per la pagina.
- Il server manda la pagina con una risposta http.
- Il browser, se serve, non solo prende quello che aveva chiesto ma potrebbe prelevare anche altre URL.
- Il browser mostra la pagina.
- Le connessioni TCP sono rilasciate.

Invece il server:

- Accetta la connessione TCP da un client (browser)
- Va sul filesystem e prende i file chiesti dalla query del client.
- Prende il file dal disco.
- Invia i contenuti del file al client
- Rilascia la connessione TCP

I server moderni hanno molte caratteristiche, come quella di generare contenuti dinamici, come i social network appunto.

Problemi: se ho un dispositivo vecchio, se faccio troppe richieste avrò un collo di bottiglia.

Quando si fa un design di un webserver si devono usare dischi veloci e fare caching a livello di webserver.

L'altro problema è che i server processano una richiesta alla volta ma è stato risolto coi server multithread.

Lez 22

HHTP e HTTPS

Sono i protocolli più diffusi e usati in questa era.
Gli header si scambiano entrambi i dati, il client scambia col server ma non c'è nessuna garanzia di integrità dei dati.

METODI http E HTTPS

- Head: legge l'header di una pagina
- Get: legge una pagina web
- POST: aggiunge a una pagina web
- PUT: memorizza una pagina
- DELETE: rimuove la pagina web
- TRACE: visualizza richiesta in ingresso
- CONNECT: si collega tramite proxy
- OPTIONS: richiede opzioni di una pagina

CODICI STATUS

- Se tutto ok è 200.
- Se è ok ma non c'è contenuto 204.
- Pagina cached ancora valida 304.
- Pagina non trovata 404.
- Errore server 500.

http MESSAGE HEADERS

- Referrer: il sito conosce da dove arrivo.

- User agent: info su piattaforma e browser.
 - Accept-Language: lingue che il client può gestire.
 - Date: data in cui è spedito il messaggio.
 - Simile al MIME per la mail è *content type*, *content len...*
 - *Last modified: data e ora in cui è stata modificata una pagina.*
-

CACHING HTTP

Se devo implementare politiche di caching, se ho un file che è un elemento di una pagina, se quel file ha un *last modified* corrispondente a quello della cache, il client non va prende di nuovo la risorsa ma sfrutta la cache. Nel caso in cui la cache sia scaduta si fa la *get* esterna oppure si usa la memoria.

I meccanismi di caching possono essere trasparenti come i *firewall*. Alcuni gestori implementano cache trasparenti che controllano le nostre operazioni (il proxy implementa le politiche di caching).

Il proxy è usato sia per controllare gli accessi a internet che per sfruttare il principio spazio-temporale della cache.

VERSIONI HTTP

Ogni versione è supportata da diversi browser. Ad oggi la versione di http è la 3.

La versione minima è la 1.1 che permette di comunicare, ma i server rispondono anche alle 1.0.

Inizialmente http era *connectionless* ma grazie ad alcuni parametri tipo *keep-alive* è cambiato. Si potrebbero sfruttare delle caratteristiche di una pagina “innocua” fatta in HTML5 che potrebbe svolgere diverse attività come *embedding dei sistemi multimediali* (*tipo twitch*).

WEB PRIVACY

Il browser ha un fingerprinting (approfondire) I cookie storano info che verranno poi lette da terzi per fare un tracking: dei sistemi potrebbero tracciare le nostre navigazioni per le pubblicità.

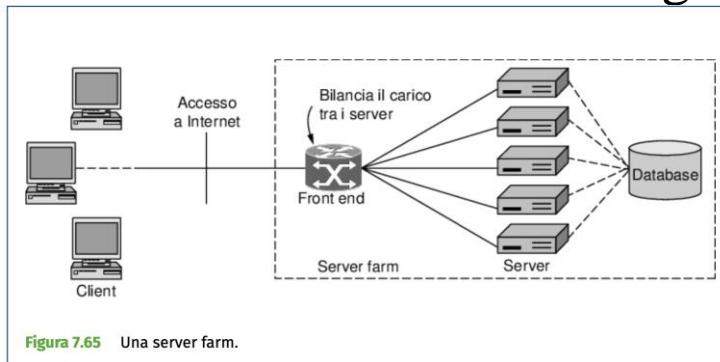
È facile e pericoloso creare un fingerprint di un browser.

CONTENT DELIVERY

I contenuti sono quelli più pesanti: mentre il web è stata la killer application di anni fa, al giorno d'oggi le killer application sono le IPTV per lo streaming. Ci sono alcune reti che si occupano della parte di content delivery, come *facebook* che ha la rete *fbcdn*.

SERVER FARM

La server farm ha un router di frontiera che bilancia il carico su più server. Immaginiamo di avere un sito importante che riceve molte richieste esterne: il router di frontiera smista il traffico su più di un server che eroga lo stesso servizio, anche se la server farm potrebbe fornire più servizi. Ci sono più hw che rispondono allo stesso sito web. I contenuti sono presi da un database, tutti i server sono connessi ad esso. C’è quindi il gateway della server farm che smista del traffico a più di un’entità fisica di quel server per fare *load balancing*. Non è una cosa facile poiché ci vorrebbe una visione lato gateway che riesca a capire quante richieste arrivano, tenendo traccia di quante richieste ha dato a tutti i server, ci potrebbero essere però richieste più pesanti di altre. Per ottimizzare quindi dovrebbe avere anche visione del carico di CPU dove nei sistemi unix possiamo vederlo col comando *uptime*. Per vedere questa info andrebbe fatto uno scripting sul gateway che oltre a smistare le richieste ai server dovrebbe gestire anche la CPU.



SERVER FARM E WEB PROXY

Un’azienda ha una serie di computer connessi che deve implementare politiche di controllo della navigazione, disattiva la gestione diretta di internet e la configura solamente tramite un proxy. La proxy ha una cache: tutte le pagine richieste sono nella cache e si evita di fare una richiesta esterna, evitando di intasare il link di uscita dell’organizzazione. Così si ottiene il controllo *whitelist e blacklist* della navigazione degli utenti e l’*improvement dello speedup del sistema grazie al sistema di caching.*

CONTENT DELIVERY NETWORKS

Supponiamo di voler accedere al sito microsoft.com che è il nodo più alto. L’idea è implementare meccanismi di distribuzione geografica (un nodo in Asia, uno in Europa...) che consente a chi si trova in zone a livello Internet più vicine in Asia di non andare sugli altri ma ovviamente di vedere la versione più vicina localizzata. La richiesta quindi non viene fatta al CDN server originario ma a quello più vicino. Il nodo centrale fa il delivery ai vari nodi, l’utente ovviamente non deve fare nessuna operazione intermedia. Questa operazione di distribuzione geografica è possibile vederla grazie al comando *traceroute*.

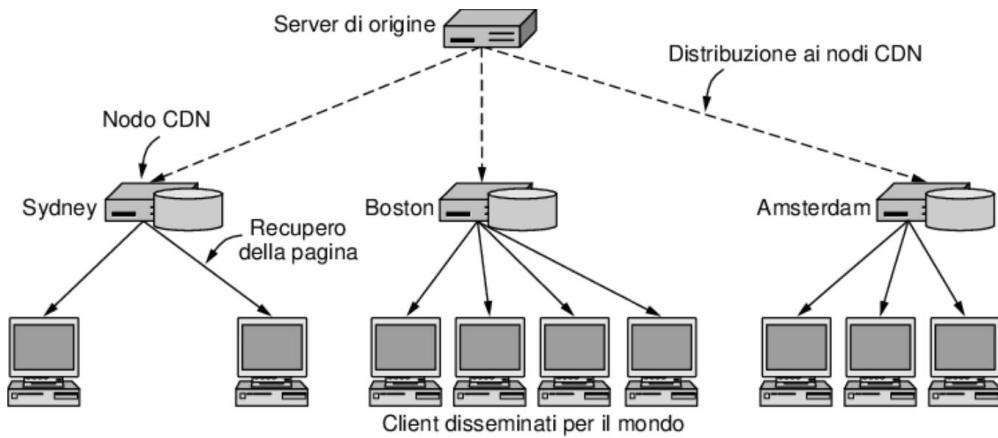


Figura 7.67 Albero di distribuzione CDN.

Quando un computer fa una query a un sito, la prima info è fornita dal DNS. Se il Dns fornisce l'IP del server più vicino allora accade quello descritto sopra, ciò significa che il CDN dipende dal DNS strettamente.

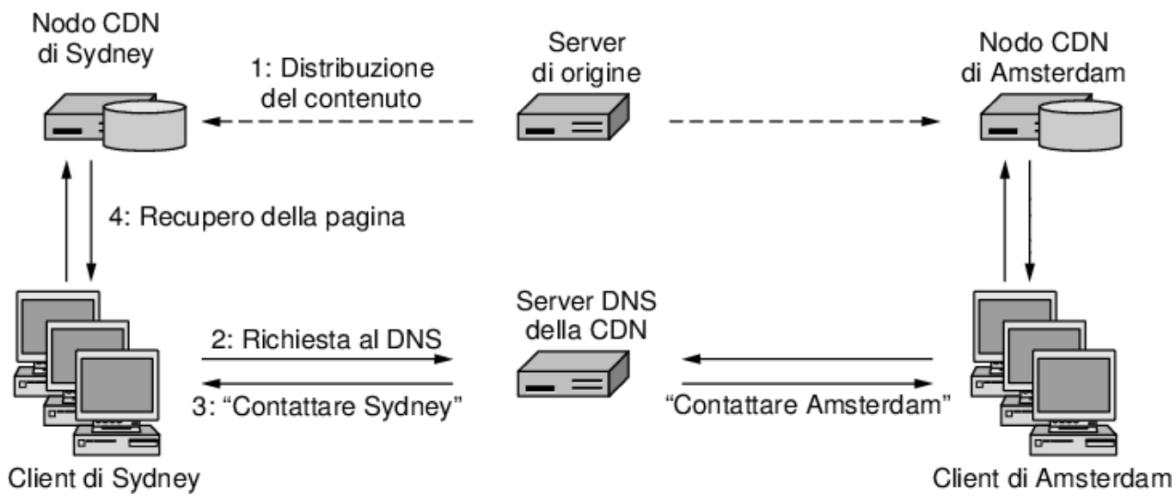


Figura 7.68 Indirizzamento dei client verso i nodi CDN più vicini tramite DNS.

RETI P2P

Le reti p2p si differenziano rispetto all’architettura client/server per il ruolo che hanno all’interno dell’architettura.

In client/server le architetture sono sbilanciate perché uno chiede e l’altro serve. In particolare notiamo lo sbilanciamento quando il server cade: il servizio non potrà essere più utilizzato.

L’alternativa è usare un modello p2p dove tutte le entità hanno la stessa responsabilità.
Ogni nodo deve funzionare sia da client che da server.

Ogni nodo fornisce servizio agli altri nodi e richiede servizi agli altri nodi. In un’architettura del genere, se uno dei nodi dovesse cadere, il servizio potrà essere gestito dagli altri nodi rimanenti.

Ad esempio: il nodo A chiede servizio sia a B che C, se B cade, il nodo A chiede a C. A a sua volta deve però fornire il servizio. L’idea delle reti p2p è quella di sfruttare la ridondanza delle risorse per offrire una maggiore qualità del servizio.

Questo ovviamente ha un costo.

BITTORRENT → ci sono 3 problemi da risolvere:

1. Come un peer trova un contenuto
2. Come replicare il contenuto tra peer
3. Come incoraggiare i peer a convivere.

TROVARE CONTENUTI IN BITTORRENT

BT crea un file torrent che contiene un tracker cioè l'IP porta di un server che contiene le info di tutti i peer che hanno una parte o tutto il file che stiamo cercando. Inoltre contiene un elenco di *chunk* ovvero le parti in cui è diviso il contenuto.

All'interno del file torrent ho quindi queste info. Il tracker mantiene le info *dello swarm*: *tutti i peer che contengono il file con i quali devo fare up/download*.

Secondo questo schema BT implementa un'architettura ibrida.

REPLICARE IL CONTENUTO

Quando uno swarm è appena formato, alcuni peer (*seeder*) devono necessariamente avere tutti i chunk.

Per evitare che tutti i peer scarichino i chunk nello stesso ordine, i peer si scambiano le liste di chunk e scelgono di scaricare prima i chunk più

difficili da trovare. Questo comportamento porta in breve tempo ad un'ampia disponibilità dei chunk.

CONDIVIDERE I CONTENUTI

Il protocollo deve favorire chi fa upload.

Chi fa solo download viene chiamato *leecher* e ha il servizio peggiore perché ostacola il funzionamento della p2p.

Ogni peer seleziona in modo casuale gli altri peer, scegliendo quelli che offrono migliori prestazioni. Più peer contribuisce al download di un contributo più potrà aspettarsi chunk in cambio.

EVOLUZIONE DI INTERNET

Si è evoluto su vari fronti:

- DNS: prima usava i file hosts.
- ROUTER: sono più performanti.
- Linee di trasmissioni: prima non erano a 56k ma erano molto lente.

L'architettura originaria internet aveva pochi host e comunicazioni e ora l'architettura internet ha diversi provider di comunicazione, con quelli che comprano cloud e cdn.

Lez 23

SICUREZZA DELLE RETI

C’è un legame stretto tra reti, SO e sicurezza poiché sono 3 elementi dipendenti gli uni dagli altri.

Chi implementa la sicurezza sono i SO e chi la mette a “rischio” sono proprio le reti.

Ci sono infinite possibilità di cause di problemi di sicurezza:

Profilo	Scopo
Studente	Divertirsi a leggere le e-mail degli altri
Cracker	Provare i sistemi di sicurezza di qualcuno; rubare dati
Venditore	Dichiarare di rappresentare tutta l’Europa, non solo Andorra
Multinazionale	Scoprire il piano di marketing strategico del concorrente
Ex-impiegato	Vendetta per il licenziamento
Contabile	Sottrarre soldi a una società
Agente di borsa	Negare un accordo fatto al cliente via e-mail
Ladro di identità	Rubare numeri di carte di credito per rivenderli
Governo	Scoprire i segreti militari o industriali del nemico
Terrorista	Rubare i progetti di un’arma biologica

Di questi tempi a questo elenco potremmo aggiungere anche *estremisti politici* o *negazionisti*...

Si parla già di sicurezza a livello di navigazione aerea o navale poiché si parla di alcuni guasti aerei causati da presunti *hacker*.

Alcuni progettisti di aerei usano la stessa rete sia per la parte passeggeri che di gestione.

FONDAMENTI DI SICUREZZA

La triade che forma la sicurezza è la C.I.A.

- Confidenzialità: i dati non sono visti da non autorizzati
- Integrità: solo i possessori possono modificarli
- *Availability*

Quando andiamo su un server *https*, il server non sa chi siamo noi, almeno che non ci identifichiamo ma il browser sa con chi stiamo comunicando.

Un'altra questione è il *principio di non ripudio*: «< ho fatto qualcosa ed è impossibile negarlo>>. Nascono quindi le firme digitali.

Ci sono altri principi da rispettare:

- Minore autorità possibile.
- Separazione privilegi.
- Principio di *open design*.
- Accettabilità psicologica.

Se proteggo qualcosa e il costo della protezione è più alto di quello che sto facendo, significa che sto proteggendo qualcosa di innocuo.

Al momento non esiste una classificazione della priorità di sicurezza di una rete.

PRINCIPI DI ATTACCO

Chi attacca ha delle sfide che deve risolvere, come provare a entrare in un sistema. Per poter rubare dei dati deve entrare in un sistema e devo avere strumenti che permettono ciò.

Fare un attacco significa violare la triade.

Quando ci sono attacchi DDoS si viola il principio della disponibilità.

I passi di un attaccante sono:

- Avere più info possibile
- Provare a fare attività di intercettazione.
- Distruggere tutto.

ALCUNE SOLUZIONI

- Capire cosa può fare un attaccante.
- Monitorare la rete.
- Capire le zone del sistema più delicate con problematiche di confidenzialità. Se ci sono dati particolari è bene tenerne conto.
- Considerare l'uso di crittografia o chiave pubblica
- Uso di firme digitali e gestione delle chiavi.
- Vedere i meccanismi di autenticazione sicura.
Non limitarsi a username o password ma considerare la parte di *biometria*.
- Affrontare il problema della sicurezza mail: protocollo più usato ma anche più attaccato.

GLI INGREDIENTI DEI POSSIBILI ATTACCHI

- *Reconnaissance*: rubare informazioni di un'organizzazione.
 - Scansione delle porte.
 - Traceroute.
 - Tramite il DNS possiamo fare attività che non danno nell'occhio
 - La FIN SCAN manda un pacchetto di chiusura della comunicazione: come se dicessimmo a una persona di avere una raccomandata e andarsene senza dargliela, ovviamente la persona aprirà per averla. La scansione FIN SCAN è molto efficiente.
- Sniffing e snooping: quando siamo su una rete switchata, per prendere i pacchetti di tutte le macchine bisogna mettere l'interfaccia di rete di uno dei computer i modalità *promiscua* (*tcpdump* o *wireshark*). Per capire se c'è uno sniffer sulla rete potrei vedere se le interfacce di quei sistemi sono messi in modalità promiscua.
 - Potrei prendere il MAC Address di qualcuno e rubarlo (**SOLO SE SULLA STESSA RETE!**). Per proteggersi da ciò si potrebbe attivare il MA solo da una porta: se ho il pc connesso su una porta ethernet possiamo

configurare un campo in modo che quel MA può passare solo da quella porta fisica.

- Spoofing: SMTP spoofing per la mail. DNS spoofing. TCP spoofing. Tutti sfruttano il fatto di trovarsi in mezzo *man in the middle* tra servizio e utente. Questo fa capire quanto è importante avere attenzione.
- Disruption: interruzione del servizio. Implica una questione progettuale: se devo gestire un sito inutile posso avere un web server anche su un telefono, se devo gestire qualcosa di importante devo considerare il *size* dell'hw. Gli attacchi di tipo DDos non sono veri attacchi ma sono causati dall'overflow di hw. Una macchina non potrebbe rispondere per crash, memoria piena o attacchi di *SYN FLOODING*.
- Ci sono ovviamente attacchi “concordati”: tante persone si collegano contemporaneamente a un sito per farlo cadere.

➤ESAME: Per difendersi da attacchi DDos:¹⁵ a livello di border gateway applicando un filtro

¹⁶ Domanda Esame

che non accettano più connessioni dalla stessa parte. Il problema di questi attacchi è dovuto alla rete e ai server. Concretamente posso mettere meccanismi a livello *firewall* che capiscono se c'è qualcosa di anomalo. La protezione quindi è fatta a livello firewall e border gateway: si capisce se ci sono connessioni che arrivano da un solo punto o sono dirette a un solo punto.

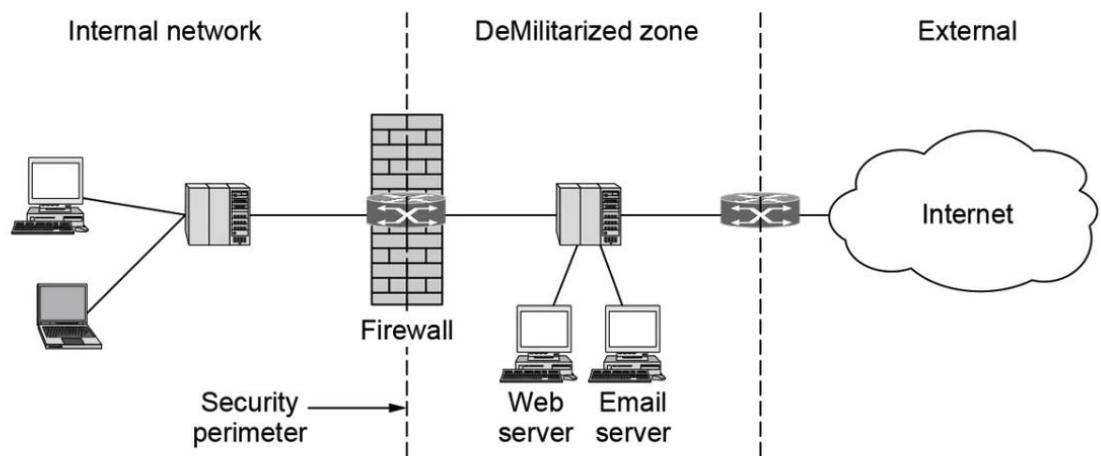
►È importante anche guardare il lato energetico.

Ci sono dei protocolli, in particolare NTP (sincronizza il tempo) con fattori di amplificazione altissimi: in fatti il GARR sconsiglia di far uscire verso fuori il protocollo NTP. In alcune università si usa o NTP locale oppure non si può. Si poteva pensare che il peggiore fosse BitTorrent ma in realtà è più basso del DNS e NTP.

FIREWALLS

Un **firewall** è un dispositivo che ha lo scopo di connettere due o più reti diverse e che viene attraversato dal traffico che va da una rete all'altra; tali reti possono appartenere a diverse zone di sicurezza: alcune sono più fidate e necessitano di maggiore protezione e altre sono più rischiose.

Il firewall implementa quindi **policy di sicurezza attraverso le quali viene determinato il traffico che può transitare da una rete all'altra e quello che deve essere bloccato**, in quanto rischioso per la rete o le reti che devono essere protette.



INTRUSION DETECTION E PREVENTION

- **IPS & IDS:** sistema di *detection* delle intrusioni. Idealmente capiscono che c'è un attacco e lo fermano prima di causare danni. Possono essere host-based o network-based.
- L'idea è usare il principio di difesa in profondità ma è qualcosa di invasivo a livello privacy.

CRITTOGRAFIA (approfondire)

Abbiamo un messaggio da inviare, testo in chiaro, che deve essere protetto da un metodo crittografico che prende in input una chiave e l'output è chiamato *testo cifrato*.

2 PRINCIPI FONDAMENTALI CRITTOGRAFICI

1. *Ridondanza*: i messaggi devono contenere ridondanza
 2. *Freshness*: alcuni metodi sono necessari per evitare i *replay attack*.
-

SUBSTITUTION CIPHERS

Ogni gruppo di lettere è rimpiazzato da un'altra lettera o da un altro gruppo di lettere.

Un esempio è quello di “Caesar”: *la lettera più comune è la “e”, seguita poi da “t”, “o”, “a”...*

Th, in, er, re sono combinazioni comuni.

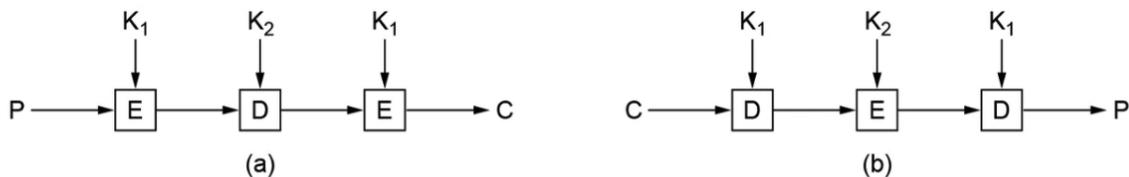
The, ing, and sono combinazioni comuni a tre lettere.

ALGORITMO DI CHIAVE SIMMETRICA

Bisogna avere dei meccanismi che mischiano il bit iniziale. Il più usato è lo standard DES e quello AES che è alla base dei vari meccanismi VPA.

IDEA DEL DES

Testo in chiaro P il cui output è un cybertest C e ci sono 3 chiavi k₁,k₂,k₃. Per encryption si usa k₁, per la decryption la k₂.



Si prende il testo, lo si cifra, decifra e lo si cifra di nuovo per avere il testo crittografato.

Per avere il testo da crittografato a testo chiaro si fa al contrario.

Il DES è abbastanza debole e quindi si è passato all'AES.

AES

È lo standard de facto della cifratura simmetrica.

L'algoritmo viene chiamato symmetric block cipher.

Si possono usare chiavi di diversa grandezza, di 128, 192 e 256 bit.

L'hw non sempre riusciva a implementare l'AES.

Il nome tecnico dell'algoritmo è *Rijndael*: usa sia sostituzioni che permutazioni sui round successivi.

Funziona molto bene ed è veloce, al contrario del DES
CHIPER MODES

Name	Position	Bonus
A d a m s , L e s l i e	C l e r k	\$ 1 0
B l a c k , R o b i n	B o s s	\$ 5 0 0 , 0 0 0
C o l l i n s , K i m	M a n a g e r	\$ 1 0 0 , 0 0 0
D a v i s , B o b b i e	J a n i t o r	\$ 5

Senza cifratura potrei sostituire i bonus perché non c'è nessuna dipendenza tra i blocchi. Questo deve essere evitato.

Esistono diverse modalità di concatenazione dei blocchi, come il *CBC*:

- Il testo in chiaro è diviso in blocchi.
 - In output abbiamo i cifrati relativo ai blocchi.
 - L'algoritmo di cifratura è sempre lo stesso.

ALGORITMI DI CHIAVE PUBBLICA

Il problema delle chiavi simmetriche è che se inviamo un messaggio, dobbiamo anche inviare la chiave.

Grazie a RSA (“how to share a secret”) si è risolto il problema.

RSA sfrutta la complessità, attraverso un algoritmo di complessità esponenziale: “se ho un numero grandissimo (1024) dico che è il prodotto di 2 numeri primi p e q ma non dico quali sono. Per scoprirlo

faccio operazioni interattive per n volte finchè non lo trovo.

Grazie a questo algoritmo, due peer ottengono un valore uguale segreto che possono usare.

Lez 24

La firma digitale serve a “*non negare il contenuto*” e verificare l’identità del mittente. Una persona non può inviarsi un messaggio da solo per simulare un evento. Ci sono diversi modi di firma: *message digests*, *sh1*, *shA*: sono modi per calcolare un valore univoco, se cambia un solo bit cambia l’hash. Esiste una versione diversa del Message Digests che è quello di farlo autenticato: posso usare un algoritmo di hash ma usare anche una *password*, si chiama *hash autenticato*: posso generare un hash solo se conosco un *segreto* (*password*). Questo consente di non far verificare a chiunque la firma. È usato per i numeri di serie dei software: la funzione di *unlock* del sw che sblocca un numero di serie è una funzione di hash autenticato.

SIMBOLOGIA CHIAVI

- A: Alice (*sender*).
- B: Bob (*recipient*).
- P: Testo.
- BB: autorità centrale.

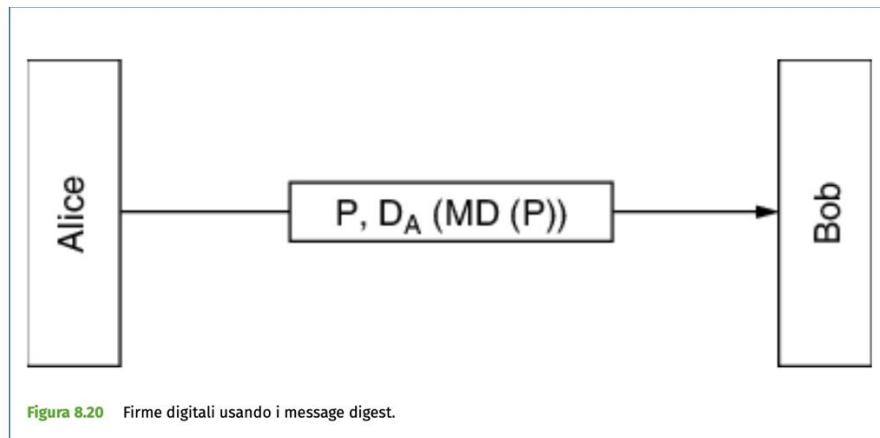
- T : timestamp
- R_a : random number scelto da A: sono in realtà numeri pseudocasuali.
- MD: Message Digest.
- D_a : Chiave privata (*decrypt*).
- E_a : chiave pubblica (*encrypt*).

MESSAGE DIGEST

Dato un messaggio in chiaro P possiamo calcolare il *digest* ma dato il *digest*, è impossibile definire il messaggio P .

Dato un messaggio P nessuno può trovare un altro messaggio che generi lo stesso *hash* (*collizione*).

Procedura: mando un messaggio P in chiaro, l'hash firmato:



!! Può firmare chi ha la chiave *privata*. !!

Ci sono modi per “imbrogliare”: chi garantisce che il destinatario è effettivamente lui? Potrebbe esserci *un*

man in the middle che mostra una *fake page*: A manderà un messaggio usando la chiave pubblica pensando fosse B ma in realtà il *man* decifra il messaggio e lo ricifra mandandolo a B.

CERTIFICATI

- Per ovviare a ciò sono stati creati i *certificati digitali*.

Per firmare molti contenuti si fa un hash per avere un valore e si firma quell'hash per risparmiare computazione.

Lo standard usato dai certificati è X.509.

➤ Come funziona? PKI (Catena di certificati)

Chi firma potrebbe essere un'autorità di livello minore (tipo il comune): si genera quindi una catena di certificati dove ognuno firma quello del livello inferiore.

AUTENTICAZIONE

C'è una directory: se A vuole comunicare con B chiede la chiave pubblica di B alla directory, una volta ottenuta, A cifra un messaggio E con il mittente e un numero casuale, lo manda a B. B chiede alla directory

chi è il mittente: quando B ha la chiave pubblica di A a sua volta manda un messaggio cifrato ad A con la chiave pubblica di A con lo stesso numero casuale di A, un numero casuale B e un K_s che è una chiave di sessione. Dopo averlo inviato ad A, può decifrare il messaggio e una volta fatto ciò vede lo stesso valore casuale R_a e la chiave di sessione che per ultimo manda una chiave di sessione cifrando il valore casuale di B. K_s e R_b sono note a B e quindi può decifrare il messaggio e hanno così fatto una *mutua autenticazione*.

SICUREZZA DELLE COMUNICAZIONI

- IPsec è a livello IP. Ci costringe a usare un client *IPSEC*. Viene usata un hash autenticato.
 - VPN: crea un percorso virtuale sicuro usando meccanismi di crittografia.
-

SICUREZZA DELLE MAIL

S/MIME è la parte di autenticazione mail.

- PGP: *pretty good privacy: meccanismo di comunicazione sicuro*.

SMIME garantisce la sicurezza dei messaggi di mail. Supporta diversi algoritmi di crittografia. I client di posta lo supportano.

La PEC è un'alternativa (italiana): posta elettronica certificata che specifica se la mail è stata consegnata.

TLS : TRANSPORT LAYER SECURITY

TLS prima si chiamava SSL.

Viene dall'antenato di *firefox* (*netscape*).

SSL crea una connessione sicura tra due socket: ha 4 operazioni che può essere implementata con algoritmi a parte:

1. Data integrity.
2. Secret communication.
3. Autenticazione del server dal client.
4. Negoziazione delle chiavi e dei parametri tra client e server.

Soltamente, se parliamo di http usato con SSL siamo sulla porta 443 ma possiamo farlo su qualunque porta. Quando usiamo SSL siamo a livello di trasporto.

Se devo parlare con qualcuno devo conoscere la sua lingua. Se vi sono dei dialetti dovrei conoscere anche quelli: se ci sono quindi parametri che usa un

protocollo devo concordare dei parametri (es avere stessa chiave).
