LIVELLO DI RETE: PIANO DEI DATI

Servizi e protocolli del Livello di Rete

Differenza tra livello di trasporto e livello di rete

- Il livello di trasporto viene interpretato dagli end-system;
  End-System: Nel gergo delle reti, un computer, un telefono o un dispositivo Internet delle cose connesso a una rete di computer viene talvolta definito sistema finale o stazione finale, perché si trova ai margini della rete.
- Il livello di rete viene interpretato da tutti i nodi della rete che siano host o router.

Cosa fa il livello di rete: trasporta il segmento dal host mittente al host destinatario.

Mittente: incapsula il segmento, facendolo diventare un datagramma, e lo passa al livello di collegamento.

Destinatario: decapsula il datagramma e manda il segmento al livello di trasporto.

Cosa fa il router:

- Analizza i campi del pacchetto IP (datagramma);
- Effettua l'operazione di inoltro (forwarding) per garantire una comunicazione end-to-end tra end system per il pacchetto. Grazie alla funzione di instradamento, possiamo avere un percorso che collega al sorgente alla destinazione.

Funzioni chiave dello Strato di Rete

1. Forwarding: muove il pacchetto dall'interfaccia input di un router alla corretta interfaccia output del router.
2. Routing: determina il percorso del pacchetto dalla sorgente al destinatario. Il percorso migliore viene determinato dagli algoritmi di routing.

Suddivisione del Livello di Rete

Le funzionalità del Livello di Rete sono suddivise tra due diversi piani:

1. **Piano Dati**: si trova al di sotto del piano di controllo. Effettua operazioni di valenza locale per ogni singolo router, in modo particolare operazioni che portano ad un inoltro corretto dei pacchetti (prende un datagramma in arrivo su un'interfaccia e lo fa raggiungere all'interno del router all'interfaccia corretta di uscita verso cui deve essere inviato);
2. **Piano di Controllo**: si occupa di tutte le operazioni che hanno una logica network-wide; quindi, una logica che riguarda la rete come sistema.
   Una tipica funzione del piano di controllo è l'instradamento, che ha una valenza globale. Quindi i vari router devono comunicare tra di loro per fare in modo che le tabelle di

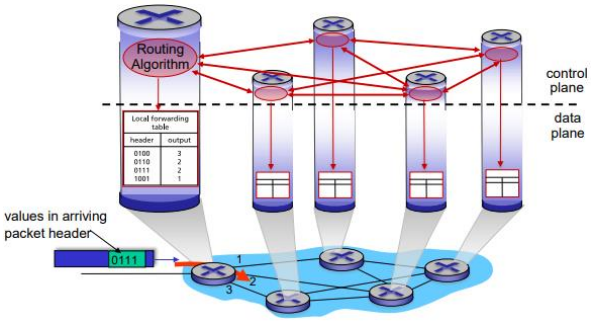
inoltro siano riempite correttamente per fare in modo che un pacchetto inviato da end-system per un altro end-system segua un percorso tra i due end-system. L'instradamento viene gestito da algoritmi di routing. Un'altra funzione del piano di controllo è anche uno scambio di messaggi, non i soliti messaggi generati dagli utenti, ma dei messaggi di controllo per fare in modo che la rete funzioni come dovrebbe.

Ci sono due approcci:

1. Algoritmi di routing tradizionali implementati in tutti i router. Permette quindi un instradamento sensato per i pacchetti. Questo approccio viene utilizzato per la stragrande maggioranza.
2. Software-defined networking, il piano di controllo viene tolto da tutti i dispositivi ma viene centralizzato in un dispositivo chiamato controllore. Si ha quindi un dispositivo centralizzato in grado di comunicare con tutti i router, che non hanno più un piano di controllo nel caso più generale possibile, per poter definire le tabelle di instradamento per tutti i router e successivamente le tabelle verranno inviate verso tutti i nodi, ovvero tutti i router. Non verrà affrontato nel corso.

Approccio tradizionale – Per-router control plane

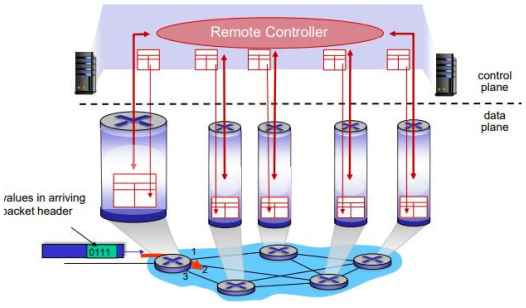
Cosa si intende con approccio tradizionale?



Sotto c'è la parte relativa al piano dei dati. Tutti i router hanno un piano di controllo. I router si scambiano messaggi tra di loro per capire che percorso intraprendere per arrivare alle varie destinazioni. Significa che l'algoritmo di routing ha l'obiettivo di capire come popolare la tabella di inoltro e da un punto di vista locale ogni qualvolta un pacchetto entra in un'interfaccia del router andrà a cercare un match nella tabella di inoltro per far in modo che vada nell'interfaccia di output corretta.

- Separazione tra piano dei dati e piano di controllo;
- La tabella di inoltro si trova nel piano dei dati, ma si interfaccia anche verso la logica che si ha al piano di controllo poiché è proprio esso che popola la tabella di inoltro.

Software-defined networking



Non abbiamo un piano di controllo distribuito in ognuno dei singoli nodi, ma abbiamo un unico piano di controllo centralizzato che è eseguito in un controllore remoto. Esso si interfaccia con i vari router per ricevere informazioni sul loro stato in modo tale da poter ricostruire la topologia complessiva della rete e può andare in modo ottimo a riempire le tabelle di inoltro di tutti i router e poi vengono inviate ai vari router e così ognuno riempire la propria tabella locale.

Network service model

A Livello di Rete si potrebbero implementare diversi servizi molto sofisticati.

Un **datagramma** offre servizi:

- Spedizione garantita (TCP)
- Spedizione garantita sotto i 40 msec di ritardo

Sono due dei servizi che potrebbero essere offerti relativi al singolo datagramma.

Un **flow di datagrammi** (A livello di rete, un flusso di datagrammi potrebbe essere tutti i datagrammi che hanno lo stesso indirizzo IP di sorgente e lo stesso indirizzo IP di destinazione. Tutti i datagrammi che hanno qualcosa in comune) offre i servizi:

- È possibile effettuare una consegna in ordine, fatta da TCP che ragiona per flussi a livello di trasporto. Una consegna garantita per il flusso oppure un limite ai cambiamenti che si possono avere nello spaziamento tra pacchetti.

Digressione sullo spaziamento tra pacchetti: quando noi mandiamo pacchetti in rete, tipicamente i pacchetti vengono immessi con una determinata frequenza, cadenzati, il problema è che poi i pacchetti viaggiano nella rete hanno una frequenza può cambiare; quindi, la distanza tra i pacchetti può cambiare: possono avvicinarsi e allontanarsi (variazione della frequenza istantanea). Questa variazione si chiama Jitter.

Definizione formale di Jitter: il jitter è la variazione della latenza di una rete nel tempo. La latenza causa ritardi nei pacchetti di dati che viaggiano su una rete, ma il jitter si verifica quando questi pacchetti di rete arrivano in un ordine diverso da quello previso dall'utente.

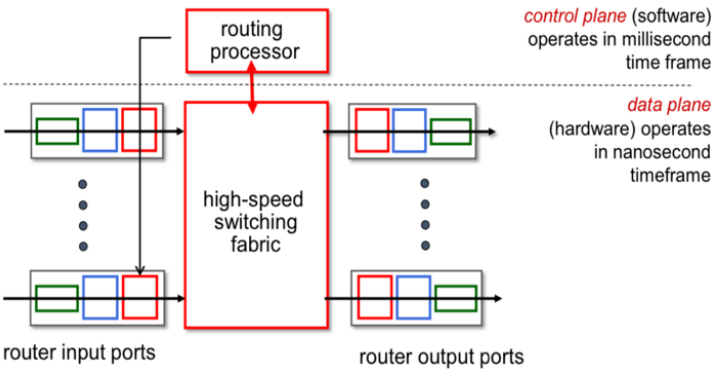
A livello di rete, in realtà, il modello di servizio è "**best effort**", che vuol dire "Io faccio del mio meglio a livello di rete per consegnare i datagrammi ma se non riesco a farlo pazienza".

- Non c'è una garanzia sull'avere successo nella spedizione di datagrammi al destinatario;
- Non c'è una garanzia sull'essere in orario e l'ordine della spedizione
- Non c'è una garanzia sulla banda disponibile per il flow

Viene adottato tutt'ora nelle reti.

Architettura di un Router

Come è fatto un router a livello di piano dati?



Una serie di **porte input/interfacce input** e **porte/interfacce output**

Tra le porte input e le porte output, abbiamo un box che prende nome di **High-speed switching fabric**; il compito di questa componente interconnette le porte di input alle porte di output. È fondamentale, perché in generale abbiamo che dentro abbiamo una rete che viene implementata in HW e permette di interconnettere le porte di input alle porte di output.

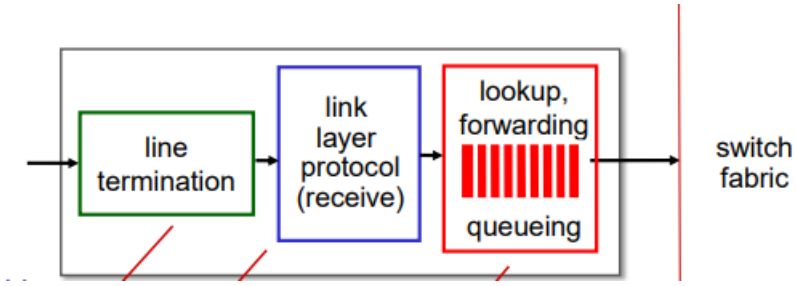
La digressione spiega che high-speed switching fabric ha la necessità di una implementazione HW poiché ha la necessità di essere molto veloce e performante per le operazioni che svolge.

Digressione su implementazione di HW: Perché in HW? Il piano dei dati ha requisiti molto più stringenti in fatto di tempo per effettuare le operazioni rispetto al piano di controllo. Al livello di piano dati, ogni volta che entra un pacchetto in un router, devono essere svolti tutte le operazioni, che portano il pacchetto dati ad accordarsi, molto rapidamente per la quantità di pacchetti che entrano nel router. Per questo è necessario effettuare delle implementazioni HW, quindi dei circuiti performanti che effettuano queste operazioni molto rapidamente. Si arriva all'ordine dei nanosecondi. Il piano di controllo, che effettuare operazioni come scambio di messaggi, calcolo delle rotte per mezzo di algoritmi di instradamento etc., è implementato SW e le operazioni si effettuano nell'ordine dei millisecondi.

*È costituito da un certo numero di connessioni molto elevato e vengono implementati delle reti di commutazione che non sono delle full mesh ma hanno delle topologie specifiche che richiedono un numero minore di collegamenti ma garantiscono una proprietà fondamentale per questa tipologia di rete HW, ovvero che sia **NON BLOCCANTE**: se io voglio mandare un pacchetto da una interfaccia input ad una interfaccia output, devo essere in grado di farlo nonostante tutto quello che sta accadendo alle altre interfacce. La rete deve permettermi di farlo senza che io debba fermarmi perché le altre connessioni sono occupati da altri pacchetti.*

Definizione di High-Speed Switching Fabric: un dispositivo HW molto performante nel nostro router che permette di connettere le porte di input alle porte di output.

Porte di input



Arriva il datagramma, un insieme di bit, alla porta di input:

1. **Terminazione di linea (line termination):** ha il compito di ricevere correttamente i bit che sono trasmessi sul mio canale e farli diventare delle informazioni leggibili dai livelli più alti. Potrebbe essere una porta Ethernet per esempio.

2. Una volta ricevuti i bit, li si passa al **livello di collegamento**.
3. Dopo la terminazione di tutte le operazioni previste dal livello di collegamento, il **datagramma viene decapsulato e arriva al "box rosso"** in cui vengono svolte tutte le operazioni del livello di rete: il box si chiama **decentralized switching**.
    i. **Look-up**: utilizza il valore del header, controlla la porta di output utilizzando la tabella di forwarding nella input port memory ("match plus action").
    Si adotta un metodo in IP chiamato "**destination-based forwarding":** prendo una decisione sull'interfaccia in output per il mio datagramma basandosi solo sull'indirizzo di IP di destinazione del mio datagramma. Ci sono tante altre possibilità in realtà, ma per come funziona IP avremo solo una scelta della porta di output basato sull'indirizzo IP di destinazione del mio datagramma.
    ii. **Obiettivo**: il processamento dei pacchetti a livello di rete nella porta interfaccia input deve avvenire a "line speed" (introdurre il minor impatto possibile in fatto di interfacce input in particolare nel ritardo, limitare l'accodamento dei pacchetti. Non è sempre possibile, perché nel momento in cui arrivano tantissimi pacchetti ho dell'accodamento con conseguenti ritardi e perdite, ma si vuole cercare di minimizzare e si cerca di scaricare le code).
    iii. **Input port queueing**: se il datagramma arriva prima della rate di forwarding nello switch fabric. I pacchetti si possono accodare in questo buffer.

Switching fabrics

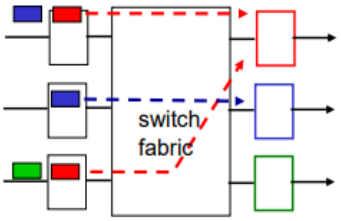
Come abbiamo detto prima, l'obiettivo è anche quello di limitare il più possibile l'accodamento dei pacchetti. Come è possibile questa cosa? Dimensionando lo switching fabric in modo tale che sia sensato.

Cosa fa il SF: Trasferisce il pacchetto dalla porta di input al link di output corretto.

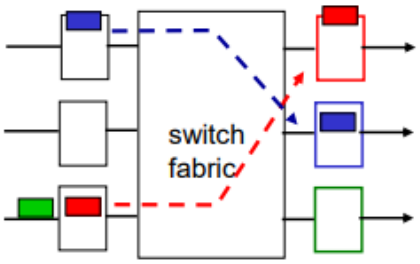
Ci possono essere due problemi:

1. **Switching rate**: dimensionare lo switching fabric in modo tale che le porte di input non diventino un bottleneck.
   Esempio: Se un router ha 32 porte, solitamente la velocità di switching è 3,2 Tbit per avere line speed perché solitamente le porte possono avere una velocità di 100 Gb al secondo.

2. **Head-of-the-line (HDL) blocking**: un problema che porta ad aumentare il tempo in cui un datagramma accordato in una coda in una porta di ingresso necessita per essere inviato alla sua porta di output. Questo si verifica quando davanti si ha un pacchetto in coda che blocca la comunicazione.



output port contention: only one red datagram can be transferred. lower red packet is *blocked*

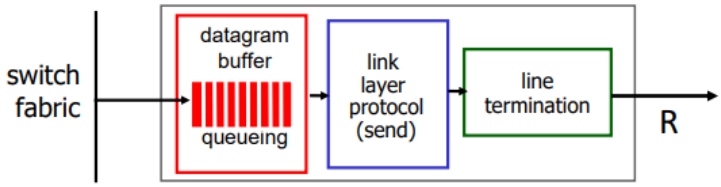


one packet time later: green packet has experienced *HOL blocking*

In questo esempio abbiamo una situazione in cui sulla porta in alto abbiamo un pacchetto rosso e un pacchetto blu accodato, la porta centrale ha un pacchetto blu, e la porta sotto ha un pacchetto rosso e un pacchetto verde accodato.

Una cosa fondamentale è che non posso avere due pacchetti che provengono da due code di due interfacce input diverse allo stesso tempo. In questo caso abbiamo due pacchetti rossi che devono andare alla stessa interfaccia output e questo non può avvenire contemporaneamente altrimenti si avrebbe una collisione di pacchetti e non arriverebbero correttamente. Quindi solo uno dei due passi in questo round. Immaginiamo che sia il pacchetto nell'interfaccia sopra vada per prima e quello blu va in quello di uscita. Il pacchetto rosso sotto sta ancora aspettando che il pacchetto rosso stia venendo trasmesso tramite lo switching fabric. Però ho ancora il pacchetto verde dietro di me che sta aspettando e che in realtà potrebbe andare se non ci fosse il rosso davanti. Questo è il concetto di HOL blocking. Il pacchetto verde ha perso un round che poteva utilizzare per andare, ma viene bloccato da un pacchetto rosso davanti.

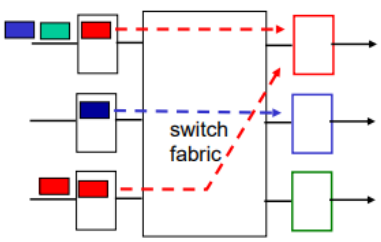
Output port queueing – Gestione di accodamento e perdite



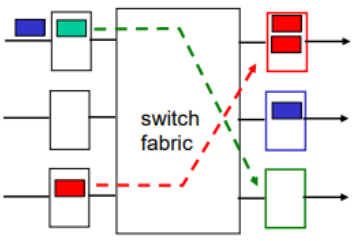
- Se ho un accodamento al buffer, avviene nel box rosso.
- Dalla mia coda viene preso il datagramma che viene incapsulato in una trama di livello di collegamento e viene inviato come sequenza di bit sul mio link di uscita.

Problema: come viene gestito l'accodamento? Come vengono gestite le perdite?

- L'accodamento può portare al buffer overflow e scartare pacchetti. In rete esistono i datagrammi che hanno priorità diverse. (**Drop policy**) Vengono adottate delle politiche di dropping che dipendono dalla tipologia di datagrammi che ho davanti. Secondo



at *t*, packets more from input to output



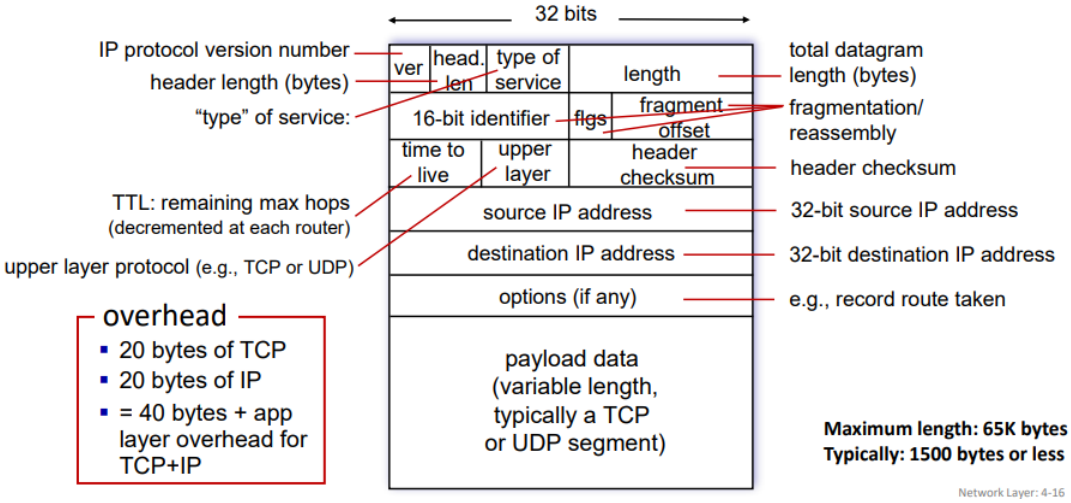
one packet time later

queste politiche, si decidono di droppare prima il livello di buffer overflow dei pacchetti di bassa priorità e cerco di evitare lo scarto di pacchetti di alta priorità.

- Le **discipline di scheduling** scelgono quali pacchetti scartare e non scartare in caso di accodamento eccessivo. Ce ne sono tante che vanno a scegliere in modo intelligente quale pacchetto trasmettere sulla mia interfaccia e si adottano politiche che portano ad avere la banda utilizzata maggiormente per i pacchetti di alta priorità (Priority scheduling: i pacchetti con una priorità più alta, utilizzando maggiormente la banda e hanno in questo modo una performance migliore).

Un esempio: FIFO, ma è molto basic.

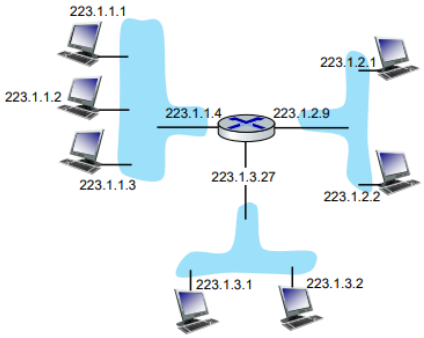
IP Datagram format



Indirizzamento IP

Definizione Indirizzo IP: identificatore di 32 bit associato a ogni interfaccia di un host o di un router. Attenzione! Non è assegnato ad host o router, ma alle loro interfacce.

Esempio: un router, 3 interfacce e 3 indirizzi IP.

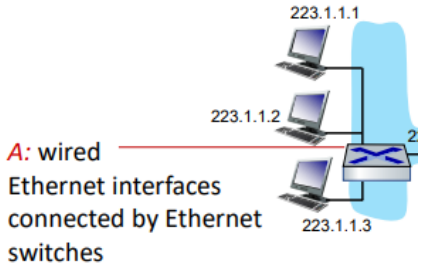


Definizione Interfaccia: connessione tra host/router e il link fisico. Solitamente un host ha uno o due interfacce.

Esempio: wired Ethernet, wireless 802.11.

Come sono rappresentati gli indirizzi IP?

Binario in 32 bit, ma per una rappresentazione più compatta si usa una notazione decimale puntata.
Esempio: 223.1.1.1 = 11011111 00000001 00000001 00000001



Come sono assegnati gli indirizzi IP nella rete?

Le interfacce sono collegate in qualche modo, che non abbiano in mezzo un router.

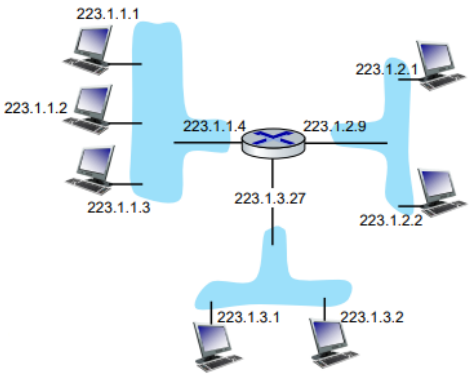
Esempi: le interfacce Ethernet sono connessi da switch Ethernet mentre le interfacce wireless WiFi sono connessi da un WiFi base station.

Tutte le interfacce di dispositivi di rete che sono fisicamente raggiungibili senza avere in mezzo un router, fanno parte dello stesso subnet (sottorete).

Subnet

Definizione di Sottorete: un insieme di tutte quelle interfacce di dispositivi che possono raggiungersi tra di loro senza avere di mezzo un router.

Esempio: abbiamo tre sottoreti.

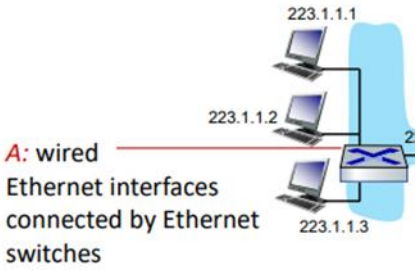


Struttura dell'Indirizzo IP

L'indirizzo IP ha una struttura per il quale è diviso in **subnet part + host part.**

Subnet part: bit più significativi più a sinistra. Dispositivi appartenenti alla stessa sottorete hanno gli stessi bit significativi.

Host part: bit rimanenti a destra. Ogni dispositivo ne ha uno distinto.

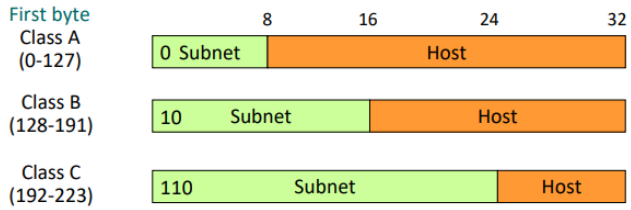


Esempio:
Subnet part: 223.1.1
Host part: 1 oppure 2 oppure 3

Indirizzi subnet/24: I primi 24 bit più a sinistra riguardano la parte di sottorete, gli ultimi 8 bit sono la parte di host.
/24 è una rappresentazione per indicare quanti bit a sinistra riguarda la parte di sottorete e quanti bit a destra riguardano la parte di host.

Assegnamento degli indirizzi IP

1. **CIDR (Classless InterDomain Routing)**
   Metodo per il quale la porzione di sottorete può avere una lunghezza arbitraria, rappresentata attraverso /X. Si guardano i primi X bit per capire quale parte di questo indirizzo IP va per la parte di sottorete.
   Esempio: 11001000 00010111 00010000 00000000 = 200.23.16.0/23
2. **Rappresentazione per mezzo di subnet mask**
   Definisco indirizzo di 32 bit in cui ho valore 1 solo per i bit relativi alla parte di sottorete e ho valore 0 solo per tutti i bit per la parte di host. Molto flessibile.
   Esempio: 11111111 111111111 111111110 00000000 = 255.255.254.0
3. **Rappresentazione iniziale: Classful**
   Metodo dei primi indirizzi IP. Non era necessario avere nessuna subnet mask, ma era poco flessibili. Gli indirizzi IP erano suddivisi in diverse classi e si guardava i primi bit a sinistra per vedere a che classe appartiene l'indirizzo.



Classe A: sottorete (0-127) N° limitato di sottoreti N° molto elevato di host	Classe B: sottorete (128-191) N° medio di sottoreti N° medio di host	Classe C: sottorete (192-223) N° molto elevato di sottoreti N° limitato di sottoreti

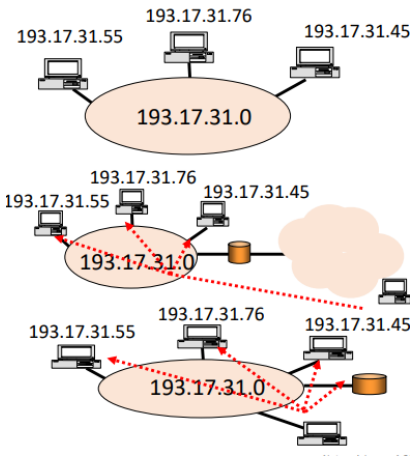
Questo è molto più semplice, non ho la necessità di una subnet mask, ma io ho solo questo schema ed è molto limitante.

Indirizzi IP speciali

- **Subnet address**: parte di host con tutti 0
  E.g. 193.17.31.0/24

- **Direct broadcast address:** parte di host con tutti 1 (non vengono usati per sicurezza)
  E.g. 193.17.31.255

- **Limited broadcast address:** tutti 1 (255.255.255.255)
  Broadcast per tutti gli host della sottorete dalla quale sto inviando questo pacchetto, ma questo genere di pacchetti non può superare il router.
  Utilizzato da un protocollo.



Forwarding di tipo destination-based

Ci specifica esattamente come funziona l'inoltro all'interno dei router.
Supponiamo di avere una tabella di inoltro all'interno del nostro router fatto in questo modo:

	Destination IP Address Range	Link interface
Subnet mask: \21	11001000 00010111 00010*** ********	0
Subnet mask: \24	11001000 00010111 00011000 ********	1
Subnet mask: \21	11001000 00010111 00011*** ********	2
	otherwise	3

Tipicamente all'interno di router non abbiamo singoli indirizzi IP ma raggruppamenti di indirizzi IP, tanti IP associati ad una subnet mask. Per ognuno di questi range di indirizzi IP, ho un'interfaccia di output con il quale farò match. Solitamente ho anche una default route che serve in caso un indirizzo IP non matcha con nessun'altra riga.
Esempio:

1. 11001000 00010111 00010110 10100001
   A quale indirizzo matcha? Con la prima riga e andrà all'interfaccia 0.
2. 11001000 00010111 00011000 10101010
   A quale indirizzo matcha? Matcha sia con la seconda riga che con la terza.
   È una cosa che si verifica molto spesso. Si segue la regola chiamata **longest prefix match.** In questo esempio, quindi, vince la seconda riga e andrà all'interfaccia 1.
   Funziona sempre.

Longest prefix match: quando più regole matchano, quella che vince è quella che porta ad un match con un numero più alto di bit. Quindi il prefisso più lungo che matcha, vince.

Come si ottengono gli indirizzi IP?

1. ***Come gli host possono ottenere gli indirizzi IP dalla rete in cui sono collegati?***
   Ci sono due possibilità:
   a. **Lo configuro staticamente;**
   b. **DHCP: Dynamic Host Configuration Protocol**; assegnamento dinamico dal server "plug-and-play". L'indirizzo IP viene assegnato automaticamente mediante questo protocollo. L'indirizzo IP viene assegnato per un tempo limitato e poi vi viene tolto, magari viene anche riassegnato a qualcun altro. Permette quindi il riuso degli indirizzi, che vengono allocati solo quando se ne ha il bisogno.

DHCP richiede quattro messaggi:

i. **DHCP discover;**
ii. **DHCP offer;**
iii. **DHCP request;**

iv. **DHCP ack.**

Come funziona DHCP?

Immaginiamo che ci colleghiamo ad una rete mediante un access point WiFi e sul nostro computer viene eseguito sempre un client DHCP. Da qualche parte nella rete è presente un DHCP server che svolge il compito di assegnare gli indirizzi IP in modo dinamico. Il DHCP server è un nodo funzionale, ad esempio il router a casa nostra è implementato un DHCP server e assegna indirizzi IP ai dispositivi che si connettono.

Come avviene l'assegnazione?

È un processo molto rapido che si effettua ogni volta prima di accedere alla rete.

Primo passaggio [opzionale] DHCP discover: Il client che arriva manda un DHCP discover message che richiede l'assegnazione di un indirizzo IP. "C'è un server DHCP in questa rete?". Include alcune informazioni:

Indirizzo IP di sorgente: 0.0.0.0 (host broadcast);
Porta DHCP client: 68 (specificando questa porta, il DHCP server capisce che mi sto riferendo a lui perché è una porta adibita per il DHCP server);
Indirizzo IP di destinazione: 255.255.255.255 (Indirizzo di broadcast perché io non so l'indirizzo del DHCP ancora, ma so che se c'è, è in ascolto sulla porta 68 perché è la porta per questo tipo di cose).
yiaddr: 0.0.0.0 (Non ho idea di quale sarà l'indirizzo IP che mi sarà assegnato);
Transaction ID: viene associato a coppie di messaggi discover-offer oppure request-ack.

Secondo passaggio [opzionale] DHCP offer: il DHCP server si identifica e assegna al client un indirizzo IP che può usare.
Questi due passaggi possono essere evitati se un client ha già un indirizzo IP e vuole rinnovarne l'utilizzo quando è prossimo alla scadenza.

Terzo passaggio DHCP request: il client sa che questo messaggio è diretto a lui per via del transaction ID, che è lo stesso di quello presente nella DHCP discover. Sa quale indirizzo può usare e fa una richiesta. Source rimane 0.0.0.0 perché la richiesta non è ancora stata completata, destinazione è ancora broadcast. Indirizzo IP rimane quello assegnato dal DHCP server. Il DHCP server capisce che il client vuole usare l'indirizzo IP che ha assegnato prima.

Quarto passaggio DHCP ACK: questo è messaggio è solo una ACK come conferma da parte del DHCP server. Infatti, ha la stessa transaction ID di prima, che era quello del discover/offer + 1. DHCP può essere utilizzato anche per assegnare il nome e l'indirizzo IP di DNS server locali o anche per indicare il valore della subnet mask.

2. Come le reti possono ottenere un pool di indirizzi IP da poter assegnare agli host che si collegano a loro?

Esempio: Come fa la Bicocca ad avere indirizzi IP da assegnare ai suoi host?
Un ISP ha assegnato uno spazio di indirizzamento di una sottorete. Si effettua una operazione di subnetting. L'ISP magari deve fornire connettività a diverse reti che sono collegate a lui e quindi estende la parte di sottorete utilizzando dei bit che magari sono per l'indirizzo di host.

Definizione di subnetting:

Il subnetting è un processo utilizzato nelle reti informatiche per suddividere una rete IP più grande in reti più piccole, chiamate subnet. Questo viene fatto per vari motivi, tra cui il controllo del traffico di rete, l'isolamento di sezioni specifiche della rete e la gestione più efficiente delle risorse di indirizzamento IP.
Ogni subnet è una sottorete che ha uno spazio di indirizzamento disgiunto.

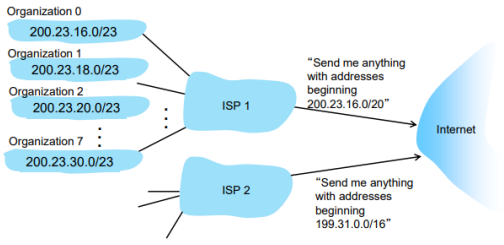
Esempio: blocco dell'ISP 11001000 00010111 00010000 00000000 = 200.23.16.0/20
Invece di utilizzare uno /20, ISP utilizza /23 per le varie organizzazioni. In questo modo ISP alloca il suo spazio di indirizzamento per otto organizzazioni diverse.

Organization 0	11001000 00010111 0001000**0**	00000000	200.23.16.0/23
Organization 1	11001000 00010111 0001001**0**	00000000	200.23.18.0/23
Organization 2	11001000 00010111 0001010**0**	00000000	200.23.20.0/23
…	…..	….	….
Organization 7	11001000 00010111 0001111**0**	00000000	200.23.30.0/23

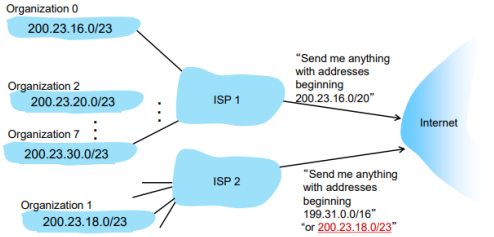
L'organizzazione 0 può effettuare nuovamente un subnetting ed estendere a /26 se c'è la necessità.

Route aggregation

L'ISP ha uno spazio di indirizzamento ben specifico e infatti può chiedere a internet di mandare a lui tutti i pacchetti che abbiano un indirizzo IP di destinazione appartenente allo spazio di indirizzamento dell'ISP. Lo spazio di indirizzamento sarà quindi messo nella tabella di inoltro. Verrà poi effettuato un ulteriore inoltro alle varie organizzazioni.



Attraverso lo spazio di indirizzamento dell'ISP, per tutte le 8 organizzazioni corrisponderà una sola riga della tabella di inoltro. Successivamente ISP inoltrerà correttamente i pacchetti.



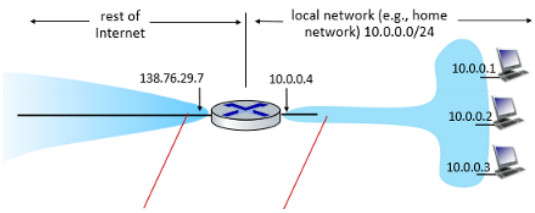
Supponiamo che l'organizzazione 1 si muova dall'ISP 1 all'ISP 2. Non cambia effettivamente nulla, bisogna solo che l'ISP 2 oltre a dire che gli si deve mandare tutti i pacchetti con quella sottorete, bisogna anche inviargli

anche il traffico della rete dello spazio di indirizzamento dell'Organizzazione 1 (200.23.18.9/23).
Si dovrà solo aggiungere questa regola al router.

Come si ottiene il pool di indirizzi IP?
Vengono assegnati dall'ICANN: Internet Corporation for Assigned Names and Numbers.
Sono esauriti gli indirizzi IP disponibili, ma NAT risolve il problema della terminazione degli indirizzi IPv4. C'è anche IPv6 con indirizzi IP da 128 bit ma allo stato attuale non può ancora essere utilizzato.

NAT: Network Address Translation

Obiettivo: andare a fare in modo che tutti i dispositivi appartenenti ad una rete locale condividano un unico indirizzo IPv4 ogni qualvolta internet è coinvolto nella comunicazione.
Nasce come soluzione tampone alla mancanza di Indirizzi IP in quando IPv6 non è ancora possibile utilizzarlo.



Sull'interfaccia verso il mondo esterno, con il NAT, si ha che tutti i dispositivi della rete locale condividano quell'indirizzo IP.

Con il NAT, nasce **la differenza tra indirizzi IP pubblici** (indirizzi che vengono assegnare alle interfacce connesse a Internet. Esempio: 138.76.29.7) **e indirizzi IP privati** (indirizzi che vengono assegnate alle interfacce appartenente alle reti locali.

Esempio: 10.0.0.1, 10.0.0.2, 10.0.0.3, 10.0.0.4).

Indirizzi IP privati: Ci sono tre diversi spazi di indirizzamento IP per gli indirizzi IP privati. Questi spazi di indirizzamento di indirizzi IP per sottoreti sono esclusivamente assegnabili alle reti locali (Prefissi 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16). Inoltre, questi indirizzi, essendo uguali per ogni rete locale, hanno appunto una **valenza solo locale**. Quando ci si interfaccia verso Internet, si ha un indirizzo IP diverso.

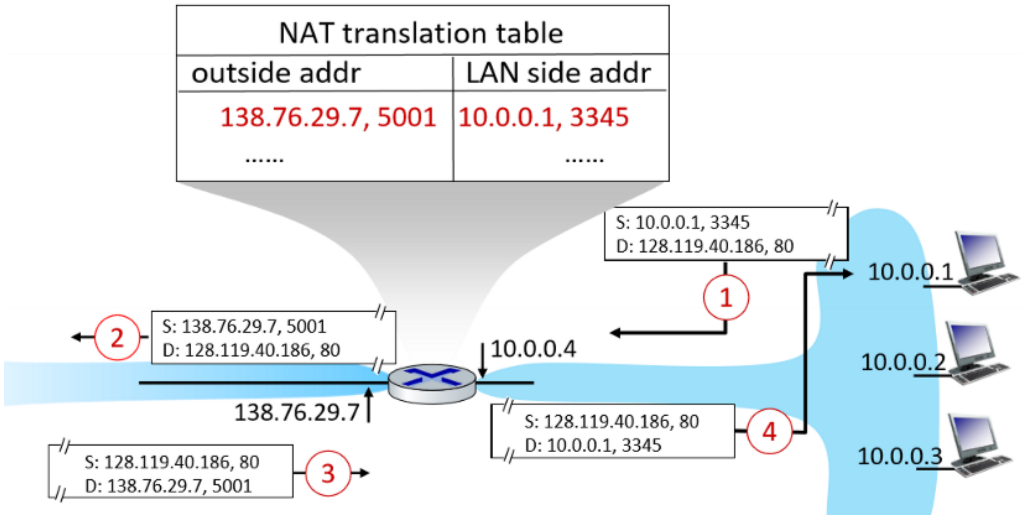
Vantaggi:

- **Riduciamo il numero di indirizzi IP pubblici** necessari per interfacciarci con il mondo esterno;
- È possibile effettuare delle **modifiche all'interno di una rete locale** senza impattare sul mondo esterno;
- È possibile **cambiare ISP** senza dover modificare l'indirizzo IP nella rete locale;
- **Sicurezza** poiché tutti i dispositivi sotto NAT non possono essere raggiunti direttamente dal mondo esterno.

Come funziona il NAT?

Effettua una traslazione degli indirizzi. L'operazione di natting viene effettuato da un router: per tutti i datagrammi che vanno dai nodi della rete locale verso il mondo esterno, che quindi avranno un indirizzo IP sorgente che è privato e una specifica porta definita, il NAT sostituisce l'indirizzo IP sorgente privato con l'indirizzo IP del NAT e con una nuova porta scelta in modo randomico.

Il datagramma viaggerà nella rete del mondo esterno e quando il server/client in risposta invierà un datagramma riceverà esattamente l'indirizzo IP del NAT e questa nuova porta come indirizzo di destinazione. L'associazione tra queste diverse coppie viene mantenuta in una tabella nella NAT translation table e viene interrogato ogni volta che si dovrà effettuare questa operazione di traslazione degli indirizzi. Una volta che un datagramma arriva dal mondo esterno al NAT, essa fa l'operazione inversa e sostituisce il (NAT IP address, new port #) con l'indirizzo privato (Source IP address, port #) salvato nella NAT table.



La entry è univocamente collegata alla coppia poiché ogni porta randomica lo rende unico. Non possono esserci due entry uguali. Se abbiamo due datagrammi provenienti da due sorgenti diverse che vanno in una destinazione comune, non avranno mai entry uguali.

Controverso:

- I router dovrebbero interpretare solo fino al livello 3, ma con il NAT i router dovrebbero decapsulare fino al livello di trasporto. Rompe l'indipendenza tra gli stati;
- IPv6 è stato inventato appositamente per risolvere il problema della mancanza degli indirizzi; tuttavia, è ancora tutt'altro che banale provare a risolverlo con IPv6.

NAT funziona anche per 4G/5G. **NAT è nata come soluzione tampone**, ma effettivamente non si sa quando lo toglieranno, probabilmente mai.