LEZIONE 13:

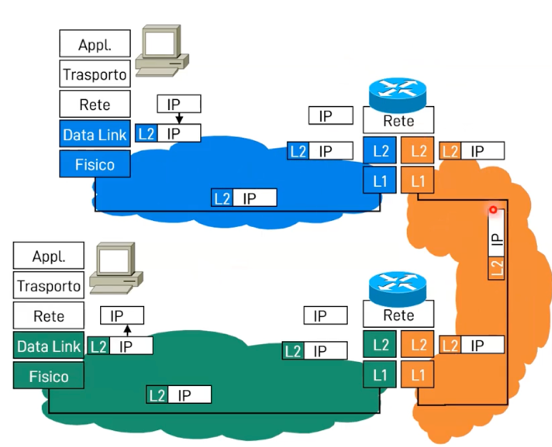
**IV IL LIVELLO DI RETE**

PARTE 1: Livello di rete e protocollo Pv4

4.1 VISIONE DI INSIEME:

L’obiettivo è quello di spostare pacchetti dal mittente al destinatario e di far comunicare host che stanno su reti diverse. Il mittente ha una sua applicazione che si interfaccia con il livello di trasporto, che usa i servizi del livello di rete per trasferire i propri segmenti

E’ interessante capire che ci sono protocolli di rete in tutti gli host e anche in tutti i router attraverso cui transitano i pacchetti. I router di fatto ispezionano tutti gli header e tutte le protocol control information del livello e attraverso ciò capiscono che decisioni prendere per inoltrare il pacchetto verso la prossima destinazione



!!! reti diverse potrebbe usare internamente protocolli di Livello 2 o Livello 3 diversi. In questo caso, è compito del router che interfaccia una rete ad un'altra il farle comunicare.

CONSEGNA DIRETTA E INDIRETTA: la consegna può essere sia diretta che indiretta. Può succedere che un router non sia direttamente connesso ad una rete, come nell’esempio il router A non è connesso direttamente a quello C.

LE FUNZIONI DEL LIVELLO DI RETE sono 2:

* INOLTRO (forwarding): è un’operazione locale con scala locale, ovvero un router sposta un datagramma da una porta di ingresso alla porta di rete di uscita.
* INSTRADAMENTO (rounting): è un’operazione locale con scala glocale perché permette di determinare il percorso che un pacchetto deve seguire.

4.2 COME E’ FATTO UN ROUTER: DATA PLANE E CONTROL PLANE

La distinzione tra la parte del router che esegue l’instradamento e la parte di inoltro comporta una distinzione tra:

* DATA PLANE (“piano dati”): è una funzione locale ad ogni router che determina come inoltrare un datagramma da una porta di ingresso ad una di uscita. Questa è una funzione di forwarding
* CONTROL PLANE: è una logica globale di rete che determina come instradare un datagramma in un percorso end-to-end, cioè dall’host mittente all’host destinatario. Questa è una funzione di routing.

Si può chiaramente evidenziare la distinzione TR control plane e dataplane;

* 1. Il control plane: parte superiore che rappresenta l’algoritmi di routing che popolano le tabelle
  2. Il data plane: parte inferiore con le tabelle e le operazioni di inoltro eseguite in funzione dei dati di quelle tabelle

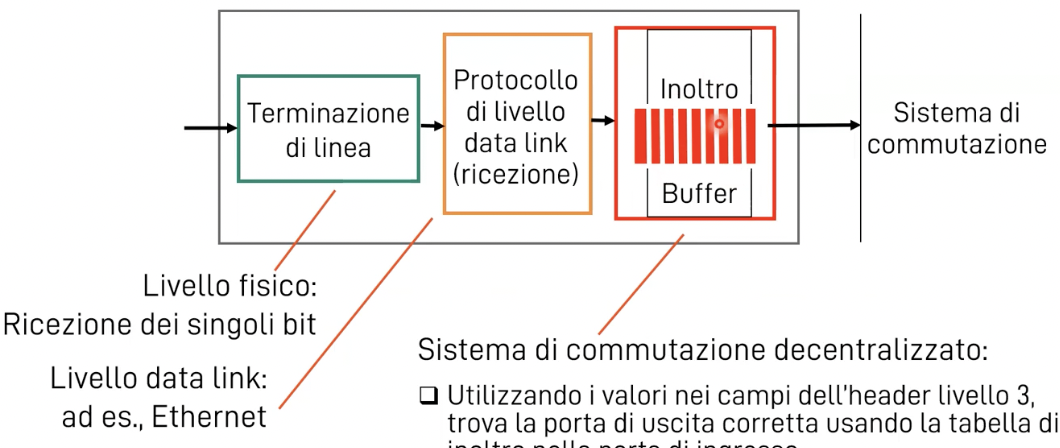
ARCHITETTURE DEI ROUTER:

Nella struttura dei router sono presenti delle porte di ingresso, un sistema di commutazione che serve a fare il forwarding cioè a prendere i pacchetti di ingresso e a portarli verso la porta di uscita che si affaccia verso la rete che si vuole raggiungere, e delle porte di uscita. Il sistema di commutazione è collegato al processore solitamente che implementa per esempio la logica del control plane. In base a se il processore viene usato o meno il tempo di esecuzione dell’operazione cambia: senza processore le operazioni richiedono nanosecondi mentre con il processore millesecondi. Quello che fa inoltre il processore è gestire algoritmi di instradamento.

FUNZIONI DELLE PORTE DI INGRESSO

Sono le seguenti:

* Ricevere i pacchetti (terminazione di linea): c’è dell’elettronica che serve a fare in modo che l’energia che viaggia per i cavi venga ricevuta dall’elettronica del ricevitore.
* Questo crea un insieme di bit letti del protocollo di livello data link (protocollo di livello 2 di collegamento) e questa lettura consente ai pacchetti di togliere l’incapsulamento di livello 2 e di mettere il datagramma nella coda dei pacchetti di ingresso.
* E’ quindi presente un buffer che porta al sistema di commutazione. Questo deve essere decentralizzato: utilizzando i valori nei campi dell’header livello 3 trova la porta di uscita corretta usando la tabella di inoltro nella porta di ingresso. L’obiettivo di tutto questo, ovvero del fatto che ci sia un “*match plus action*”, cioè un’operazione che richiede di trovare una corrispondenza tra l’header del pacchetto e la porta di uscita è quello di non introdurre ritardi ulteriori, prendendo una decisione alla stessa velocità con cui la porta di ingresse riceve i dati. Questo potrebbe non essere sempre possibile: in questo caso ciò comporta accodamento nel buffer, ovvero se i datagrammi arrivano più in fretta del tasso di smistamento del commutatore.



!! i buffer servono a dare tempo al sistema.

SISTEMI DI COMMUTAZIONE:

Traferiscono pacchetti dall’ingresso all’uscita appropriate con una certa efficienza chiamata TASSO DI COMMUTAZIONE, ovvero la frequenza alla quale i pacchetti vengono trasferiti dagli ingressi alle uscite. Questo è spesso misurato come multiplo della velocità di commutazione. Con N ingressi, di solito vorremo una commutazione N volte più veloce della comunicazione per riuscire a gestire il traffico per n PORTE sena che nessuna della porta veda un calo della velocità di comunicazione.

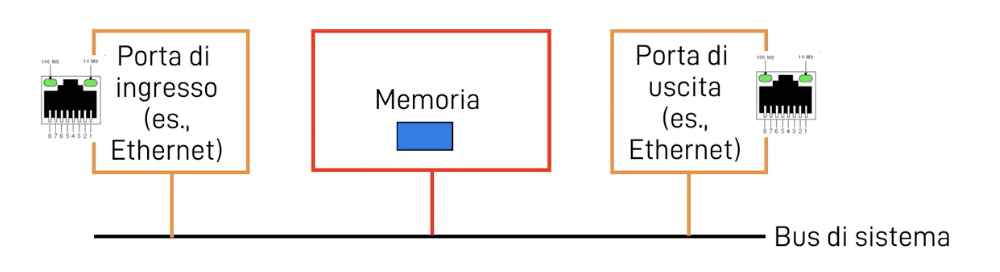
Ci sono 3 tipi di strutture per realizzare ciò: a memoria, a bus e a matrice.

Immagine che contiene testo, strumento scrittorio, screenshot

Descrizione generata automaticamente

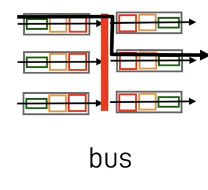
* COMMUTAZIONE A MEMORIA:

E’ tra le prime generazioni di router, ovvero computer tradizionali dove la commutazione è controllata dalla CPU e dove i pacchetti vengono copiati nella memoria del computer. La velocità di commutazione era limitata dalla banda dati della memoria -> la memoria, in questo modo, veniva usata due volte per datagramma: una volta per l’ingresso e una per l’uscita per cui servivano due accessi dal bus del sistema e ciò era poco efficiente



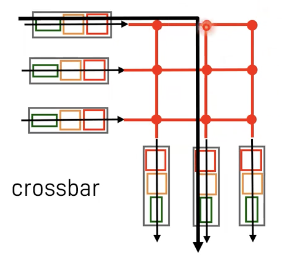
* COMMUTAZIONE A BUS:

In questo caso le porte di ingresso e di uscita condividono un bus dati interno al router per trasmettere i datagrammi dagli ingressi alle uscite. Come per tutti i bus esiste il problema della contesa del bus, ovvero la velocità di commutazione è limitata dalla banda del bus, cioè dalla velocità di trasferimento dati sul bus. Questo è un bus che di base è molto veloce.



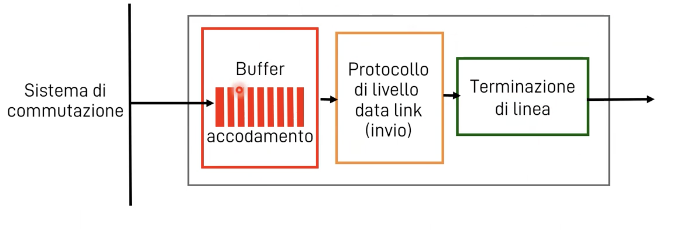
* COMMUTAZIONE A MATRICE

Questa permette di interconnettere delle linee di ingresso e delle linee di uscita. Questa è ispirata ai primi commutatori telefonici elettro-meccanici e è usata anche per connettere i processori nei sistemi di calcolo multiprocessore. Questa supera i limiti di velocità di commutazione a bus perché si usa un collegamento elettrico a seconda di come è fatta la matrice perchè questa consente un certo parallelismo. Il suo design è avanzato: è presente una frammentazione dei datagrammi in celle di lunghezza fissa che transitano poi attraverso la matrice.



FUNZIONI DELLE PORTE DI USCITA:

In questo caso, se i datagrammi arrivano dal commutatore più in fretta di quanto le porte di uscita possano smaltirli, questi potrebbero essere scartati se le memorie si riempiono. La porta funziona comunque nello stesso modo: c’è un buffer che consente l’accodamento dei datagrammi. Dopo di che è presente un protocollo di livello data link per l’incapsulamento a livello 2 e la trasmissione e infine è presenta la terminazione di linea che trasferisce il datagramma.



MECCANISMO DI SCHEDULING

Scheduling è la decisione di quale pacchetto deve essere trasmesso su un link di uscita. Si può fare in diversi modi. Come già detto, lo scheduling FIFO è il più usale e consiste nell’inviare i pacchetti in ordine di arriva nella coda. In questo modo è sempre possibile che il sistema rimanga saturo e in questo caso viene applicata una politica di scarto: se la coda è piena si deve decidere di scartare i pacchetti. I pacchetti possono essere scartati in maniera diversa:

1. Tail drop: scarto i pacchetti in arrivo
2. Prority drop: scarto o cancello i pacchetti basandomi su un livello di priorità
3. Random drop: scarto o cancello i pacchetti casualmente.
   1. IL PROTOCOLLO IP
      1. Il formato dei datagrammi:

l livello di rete in internet si posiziona sotto il livello di trasporto e sopra il livello di collegamento. Il protocollo tipico che si usa a livello di rete è il protocollo IP con le sue convenzioni sugli indirizzi, il suo formato dei datagrammi, le sue politiche di gestione dei pacchetti. Questo protocollo si serve di altri servizi quali il protocollo ICMP che serve per ottenere a livello di rete informazioni su errori e scambio di informazioni tra router e notifiche sui cambiamenti. Questi protocolli stabiliscono inoltre delle regole di inoltro per la gestione dei pacchetti. Le regole di inoltro hanno bisogno di dati che vengono dati dai protocolli di routing (RIP, OSPF, BGP) che stabiliscono la porta di uscita adatta. Questi sono protocolli che fanno parlare le macchine in Internet e cercano di capire quale sia il percorso migliore per popolare le tabelle di instradamento.

FORMATO DEL DATAGRAMMA IP (IPv4).

Nell parte inferiore sono presenti i dati, segmento del livello di trasporto, ma non necessariamente perché ci sono dei protocolli che mandano le informazioni direttamente attraverso il protocollo IP senza passare attraverso il livello di trasporto. Il segmento dati viene incapsulato insieme all’interno del datagramma IP. IP aggiungerà poi delle proprie informazioni di controllo (HEADER). Due informazioni molto importanti sono l’indirizzo IP del nodo sorgente e di quello di destinazione, oltre che altre informazioni di supporto come la lunghezza, il tipo di servizio, la lunghezza dell’header, la versione, l’ID del datagramma, il Time-to-Live…

CAMPI DEL DATAGRAMMA IP:

* VER (4 bit): il numero di versione iP (indici che sono solo 4 o 6)
* LUNGHEZZA HEADER (4 bit): indica il numero di parole di 32 bit nell’header (= 5 parole di 32 bit se non ci sono opzioni -> l’header di IP ha quindi una lunghezza tipica di 20 BYTE)
* TIPO DI SERVIZIO (8 bit): è la classe di servizio del datagramma che serve a incrementare un trattamento diverso dei pacchetti sulla base del contenuto che essi portano. Si può usare potenzialmente per fare funzioni avanzate (DiffServ e Explicit Congestion Notification (ECN))
* LUNGHEZZA TOTALE (16 bit): è il numero totale di byte nel datagramma includendo sia l’header che i dati
* IDENTIFICAZIONE ID DEL DATAGRAMMA (16 bit): è il numero (sequenziale di solito) assegnato al datagramma che è usato per raccogliere eventuali frammenti multipli e riassemblare il datagramma complessivo
* FLAG (3 bit): campo in cui i bit specificano se il datagramma è un frammento di un datagramma più lungo, se è così, dicono inoltre se questo è l’ultimo frammento o no.
* OFFSET DEL FRAMMENTO (13 bit): indica in quale punto del datagramma originale va inserito questo frammento. Di solito è espresso in multipli di 8 Byte.
* TIME TO LIVE (TTL) (8 bit): è un intero inizializzato dal mittente che viene ridotto di 1 da ogni router per cui passa il datagramma. Inoltre, se raggiunge il valore zero, il router scarta il datagramma e invia un messaggio di notifica al mittente.
* PROTOCOLLO SUPERIORE (8 bit): specifica quale protocollo di livello superiore aspettarsi all’inizio dei dati incapsulati nel datagramma (es: 6 = nel campo dati del datagramma IP c’è un segmento TCP, 17 = nel campo dati del datagramma IP c’è un segmento UDP)
* CHECKSUM DELL’HEADER (16 bit): come in UDP, si calcola il complemento a 1 della somma con riporto di tutte le parole di 16 bit dell’header.
* INDIRIZZO IP SORGENTE (32 bit): è l’indirizzo IP del mittente INIZIALE
* INDIRIZZO IP DESTINAZIONE (32 bit): è l’indirizzo IP della destinazione FINALE
* OPZIONI IP: campo nella maggior parte dei casi vuoto che però può essere usato per controllare l’elaborazione e l’instradamento dei datagrammi
* PADDING: si possono inserire dei bit a zero aggiunti se le opzioni non terminano ad un multiplo di 32 bit, in modo che l’header sia un multiplo di 32 bit.

MTU e FRAMMENTAZIONE DATAGRAMMI

Ciascun tipo di hardware specifica un **limite di dati che una trasmissione può trasportare** (Maximum Transmission Unit, MTU). Per cui ciascun datagramma deve contenere al più un numero di Byte pari alla MTU.

LEZIONE 14° a

**INDIRIZZI IP**

Un indirizzo IP è una stringa di 32 bit associata ad **un’interfaccia di rete, ovvero alla connessione tra l’host (o il router) e il collegamento fisico che trasporta i pacchetti**. I router, ad esempio, hanno molte interfacce di rete sulle quali si affacciano mentre gli host hanno 1 o 2 interfacce ciascuna delle quali ha un indirizzo IP diverso. Inoltre, gli host e i router devono avere indirizzi compatibili: devono usare le stesse convenzioni di indirizzo. Ogni indirizzo IP pubblicamente raggiungibile deve essere unico e ogni interfaccia ha almeno un indirizzo IP.

Quando si invia un pacchetto su Internet, il software che implementa i protocolli di rete lato mittente deve specificare:

1. il proprio indirizzo IP a 32 bit, ovvero il **SOURCE ADDRESS**
2. l’indirizzo IP del ricevitore a 32 bit, cioè il **DESTINATION ADDRESS**

A questo punto i router hanno tutte le informazioni necessarie per prendere decisioni di inoltro e instradamento basandosi unicamente sul destination address.

La notazione che si usa per gli indirizzi IP è la **DOTTED DECIMAL NOTATION**: ogni sezione da 8 bit è espressa come un intero decimale, vengono usati solo numeri positivi separati da punti:

* 00000000 -> 0
* 00000001 -> 1
* 00000010 -> 2
* ….
* 11111111 -> 255

Quindi la gamma di indirizzi disponibili va da 0.0.0.0 a 255.255.255.255

**GERARCHIA DEGLI INDIRIZZI IP**

Gli indirizzi IP sono gerarchicamente divisi in **due parti**:

* Un *prefisso (indirizzo di rete)* che identifica la **rete alla quale l’host allacciato** (network ID, o **NetID**).
* Un *suffisso (indirizzo di host)* che identifica **un’interfaccia di rete (di un host) allacciata a quella rete** (**HostID**). ! ogni interfaccia di rete ha un indirizzo IP unico su quella rete.

COME DISTINGUERE NETID e HOSTID? In passato erano presenti assegnazioni statiche che ora sono obsolete, ovvero erano presenti diverse classi di indirizzi con diverse lunghezze dei prefissi:

Immagine che contiene tavolo

Descrizione generata automaticamenteImmagine che contiene testo

Descrizione generata automaticamente

Ovviamente queste sono dimensioni infelici perché il **problema è che sono fisse**. Questo comportava che gli indirizzi e lo spazio di indirizzamento in internet non venivano usati bene. Inoltre, un altro problema è che esistevano anche la classe D (primi bit = 1110) riservata per 268435456 indirizzi multicast e anche una classe E (primi bit = 1111) riservata per 268435456 indirizzi non assegnati. Per cui alla fine chi chiedeva uno spazio di indirizzamento voleva ottenere classi A e B per accomodare le crescite future per cui alla fine lo spazio degli indirizzi fu presto esaurito.

Per questo si decise di cambiare strategia. Innanzitutto, si nominò **un’autorità chiamata ICANN (Internet Corporation for Assigned Names and Numbers)**, autorità creata per **gestire l’assegnazione di indirizzi ed arbitrare eventuali dispute tra molteplici pretendenti**. ICANN non gestisce gli indirizzi direttamente, ovvero non assegna i prefissi direttamente ma gestisce e autorizza diverse entità dette “**registar**” a farlo.

Per risolvere il problema dell’indirizzamento con classi si è deciso di creare un indirizzamento in cui sono state eliminate completamente le classi in modo da ridurre il problema dell'esaurimento degli indirizzi IP e c’è stata una suddivisione tra prefisso e suffisso completamente arbitraria.

**MASCHERE DI RETE:**

Collegato al fatto che esiste un indirizzo IP che un certo numero di bit di NetID è il concetto di maschera. Le maschere di rete sono ciò che consente di **conoscere il limite tra prefisso e suffisso**. Siccome il limite cambia, il modo per saperlo è **memorizzare 32 bit dove gli unici bit a 1 sono quelli del prefisso**. Ovviamente questi bit a 1 sono i primi N bit (e non sono sparsi all’interno di questa maschera).

In IP questa informazione è chiamata **SUBNET MACK** e solo il suo prefisso serve ai router per prendere decisioni di inoltro. Una maschera permette quindi di estrarre il prefisso da un indirizzo attraverso l’uso di un AND logico bit a bit. L’and estrae un prefisso e permette di andare a leggere la tabella: la rete di destinazione è infatti in una **tabella di inoltro**, ovvero una **lista di indirizzi di rete (Ni) e delle maschere corrispondenti (Mi)**. Per mandare pacchetti a D si usa quindi l’interfaccia per cui D &Mi == Ni.

**NOTAZIONE CIDR**

E' una notazione che permette di sintetizzare tutte le informazioni relative ad un indirizzo e alla propria maschera. In pratica, consideriamo ancora prefissi di 26 bit e 6 bit per l’HostID. Nella maschera in questo caso sono presenti 26 bit a 1 seguiti da 6 zeri.

* + Notazione binaria: 11111111.11111111.11111111.11000000
  + In notazione decimale: 255.255.255.192

Con la notazione CIDR si può sintetizzare scrivendo: ddd.ddd.ddd.ddd/m

Quindi la solita **notazione decimale** **per l’indirizzo + /m dove m indica il numero di bit a 1 nella maschera.**

**INOLTRO CON CIDR**

Quando riceve un datagramma da inoltrare, un router deve decidere con quale interfaccia inoltrarlo. **L’interfaccia usata dipende dall’indirizzo IP di destinazione e dalle regole che il router ha**. La corrispondenza tra le due cose viene identificata usando la **netmask**. Consideriamo la tabella di inoltro qui sotto e l’IP di destinazione 200.23.19.7: calcoliamo l’AND bit a bit con una maschera avente 23 bit a 1 (/23). Quello che si ottiene:

200.23.19.7 & 255.255.254.0 (/23) = 200.23.18.0 -> eth1

Immagine che contiene tavolo

Descrizione generata automaticamente

E se ci sono corrispondenza multiple? Consideriamo ancora la destinazione. 200.23.19.7 e la nuova tabella sotto. La tabella contiene elementi con net,mask diverse:

* 200.23.19.7 & 255.255.240.0 (/20) = 200.23.16.0 -> OK!
* 200.23.19.7 & 255.255.254.0 (/23) = 200.23.18.0 -> OK!

Regola: **si usa il prefisso più lungo** (“longest prefix matching”). In questo caso il datagramma viene quindi inoltrato verso eht1.

Immagine che contiene tavolo

Descrizione generata automaticamente