

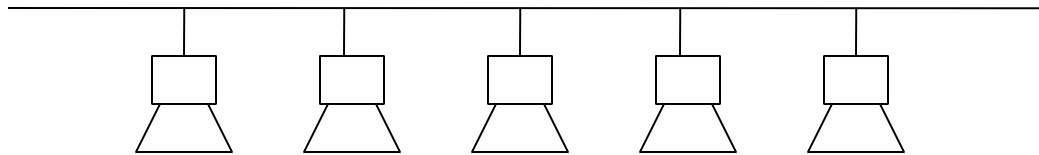
LPR 2005/2006

Lezione 8

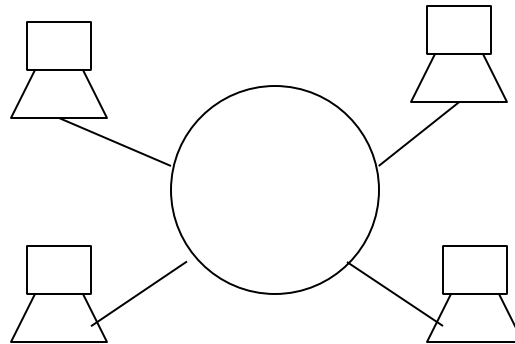
- Classfull IP Addressing
- Subnetting
- CIDR: Classless Addressing
- RMI callbacks
- Esercizi

INTERCONNESSIONE DI RETI: IL LIVELLO IP

- *Rete elementare* = gli host sono connessi *direttamente mediante un mezzo fisico*, ad esempio un cavo o una fibra.
- Tecnologie disponibili:
 - *Ethernet* (cavo coassiale condiviso, bus)

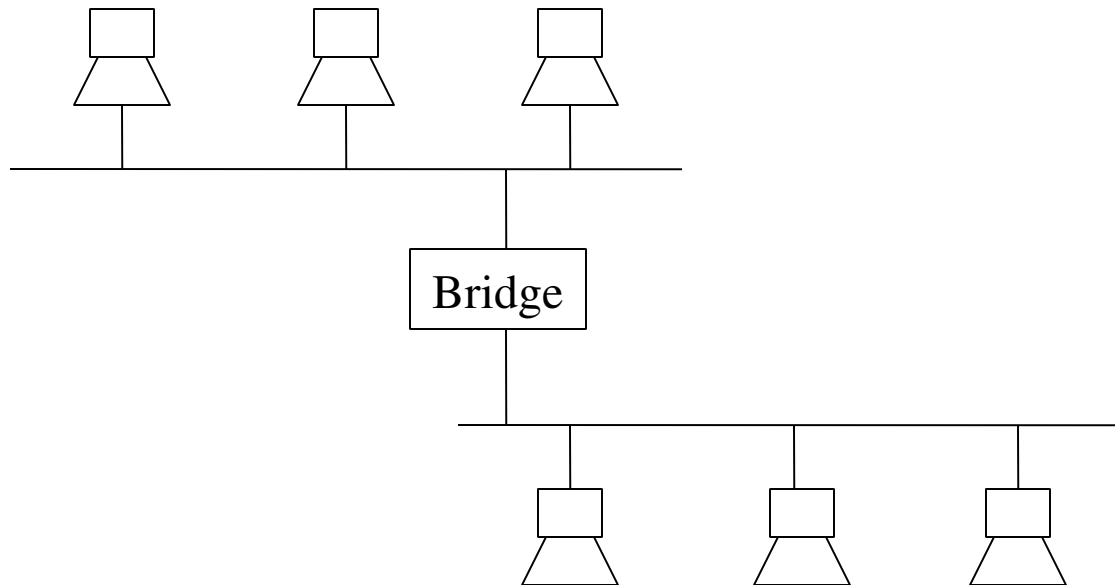


- *Token Ring* (FDDI, anello condiviso)



INTERCONNESSIONE DI RETI: IL LIVELLO IP

Rete locale estesa = insieme di reti elementari interconnesso mediante un insieme di commutatori (*bridge*).



Bridge = è in grado di eseguire solo il *livello link* ed il *livello fisico* dello stack TCP-IP

INTERCONNESSIONE DI RETI: IL LIVELLO IP

La costruzione di reti estese mediante bridges e commutatori è *limitata* da diversi fattori

- *scalabilità*: una rete estesa non può contenere un numero troppo elevato di hosts
- *eterogeneità limitata*: una rete estesa utilizza una unica tecnologia

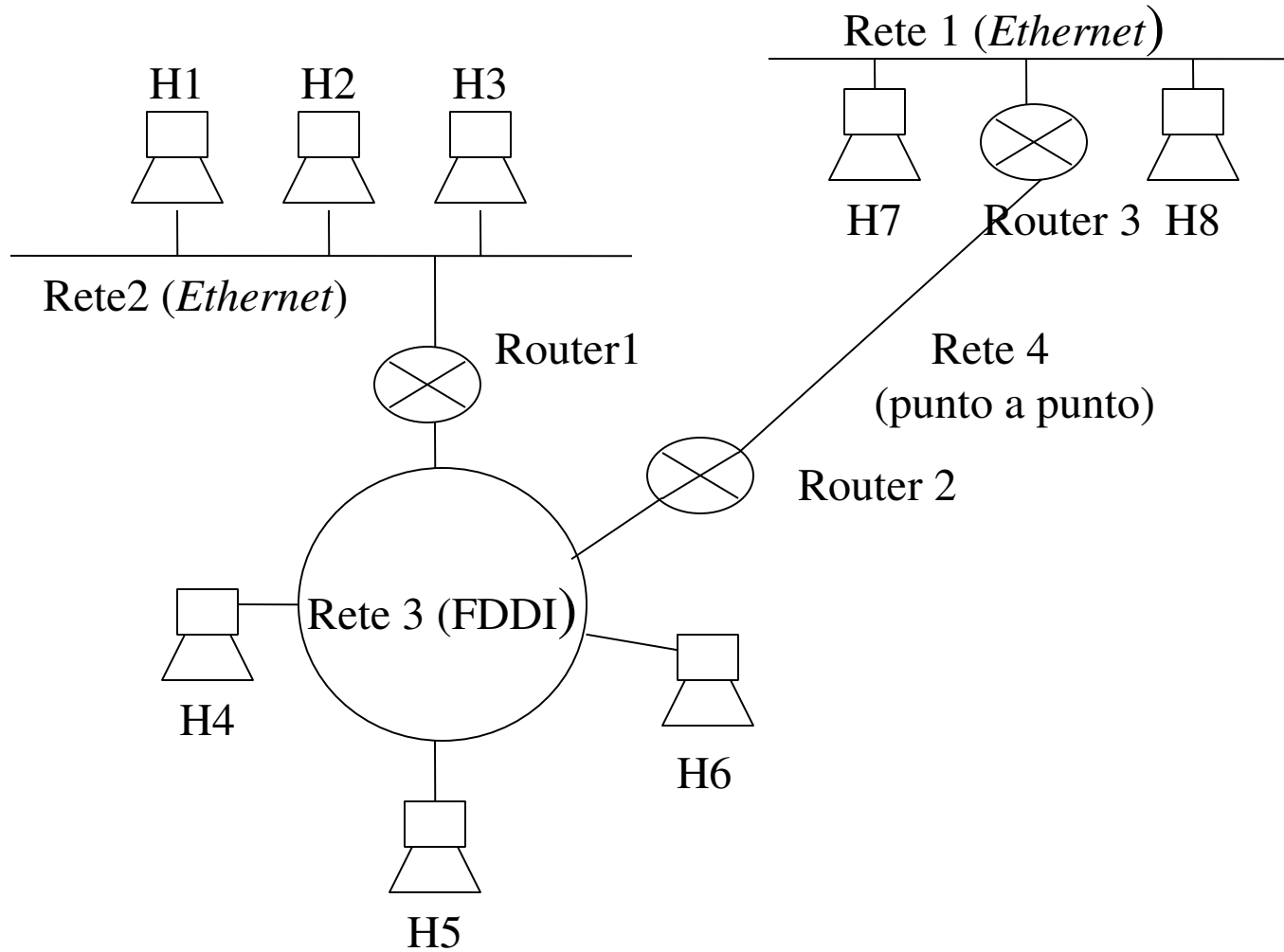
⇒

la costruzione di una rete eterogenea di grosse dimensioni (internet) richiede l'utilizzo di diverse tecnologie

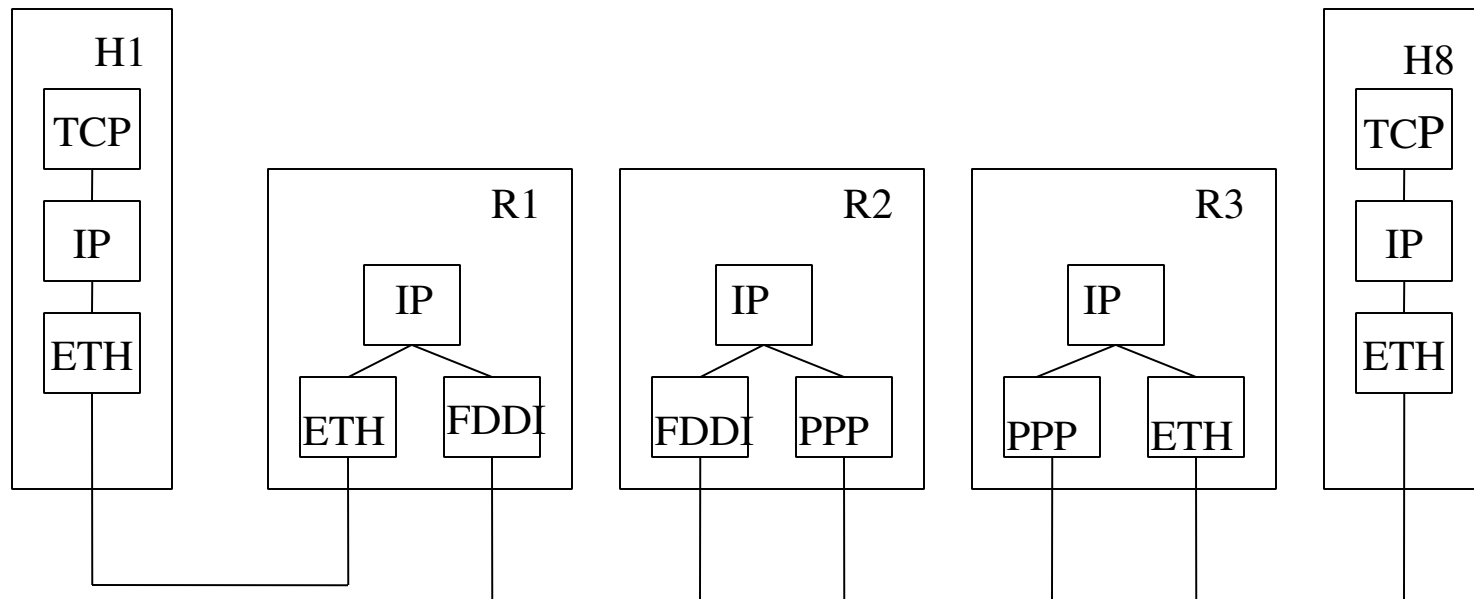
⇒

protocollo IP(Internet Protocol): permette l'instradamento di messaggi tra reti estese diverse utilizzando componenti specializzate (*routers o gateway*)

INTERNETWORKS (INTERNETS)



INTERNETWORKS (INTERNETS)



INTERNETWORKS (INTERNETS)

- internetwork(internet) = insieme di reti qualsiasi interconnesse che utilizzano un servizio di consegna di pacchetti tra hosts

- *Esempi:*

una grossa azienda con molte sedi interconnesse le LAN presenti nelle diverse sedi e costruisce una propria *internet privata*

Internet (con l'iniziale maiuscola): la rete mondiale a cui sono interconnesse quasi tutte le reti

- *Terminologia:*

rete fisica = rete a connessione diretta oppure rete commutata che utilizza una unica tecnologia

rete logica = rete (internet) costruita interconnettendo un insieme di reti fisiche

INDIRIZZAMENTO IP

Schemi di indirizzamento IP

- Classful Addressing
- Subnetting
- Classless Addressing *CIDR*

- *Classfull Addressing:*

ogni indirizzo IP rappresenta una gerarchia a due livelli

la prima parte dell'indirizzo *identifica la rete fisica* a cui appartiene l'host individuato da quell'indirizzo

la seconda parte individua *l'host*

171.69.210.245
└──────────┬──────────┘
rete host

CLASSFULL ADDRESSING

Classfull Addressing

Classe A

0	Network A.	Host Address
---	------------	--------------

Classe B

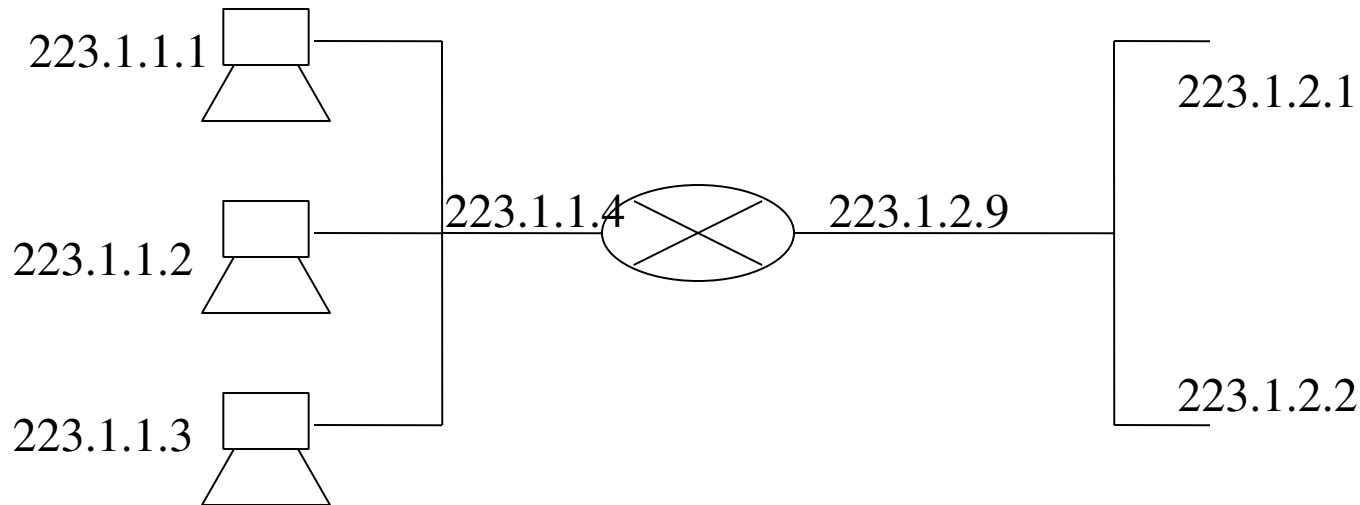
10	Network Address	Host Address
----	-----------------	--------------

Classe C

100	Network address	Host Address
-----	-----------------	--------------

CLASSFULL ADDRESSING

- Tutti gli hosts ed i routers che condividono lo stesso indirizzo di rete sono connessi alla stessa rete fisica



- ogni rete fisica connessa ad Internet ha almeno un router che è connesso ad un'altra rete fisica
- un router possiede un insieme di *interfacce di rete*

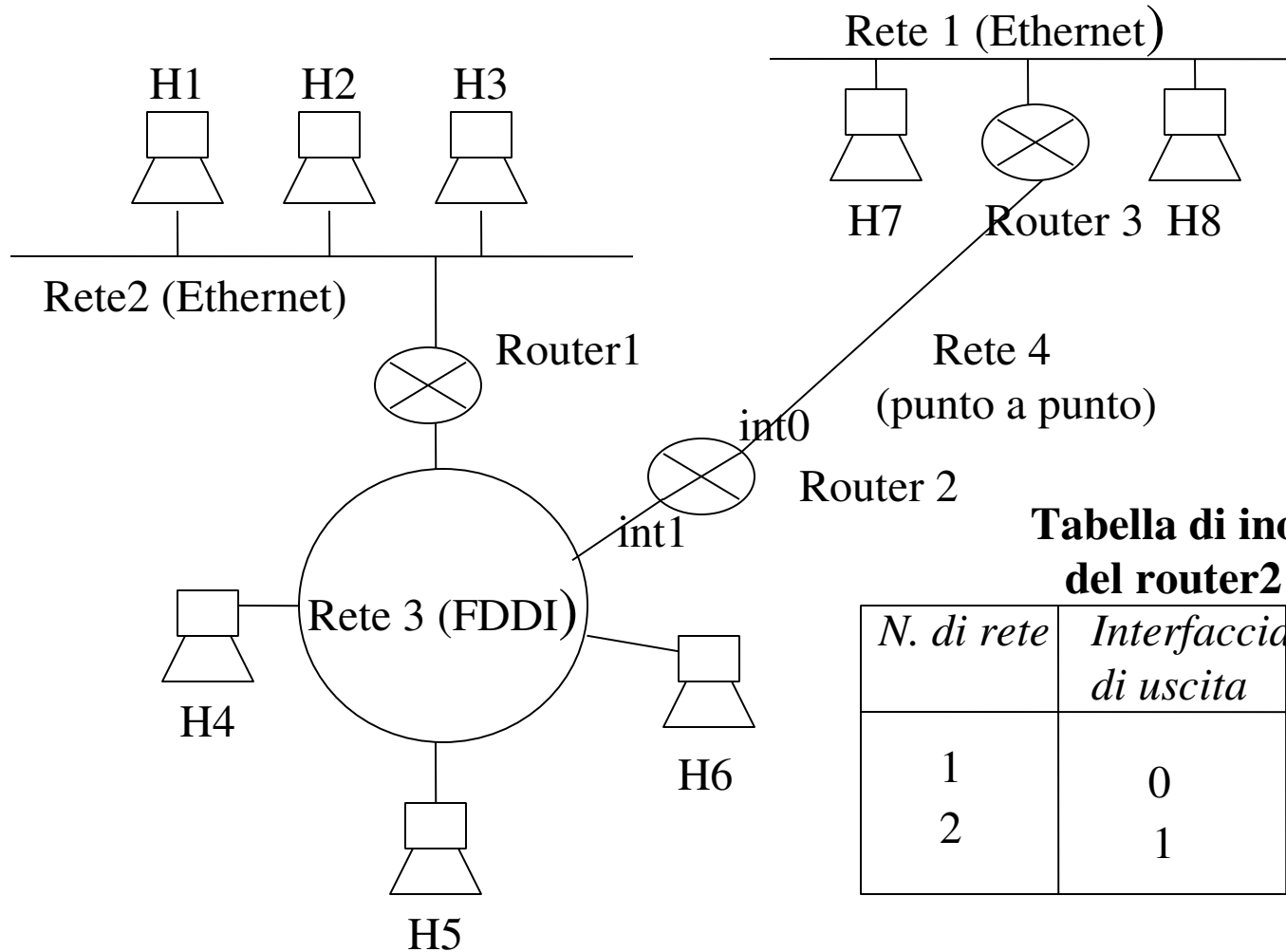
INOLTRO DI PACCHETTI

- *inoltro* = meccanismo utilizzato da routers ed hosts per decidere su quale porta di uscita inviare un pacchetto ricevuto su una porta di ingresso
- il meccanismo di inoltro sfrutta un insieme di *tabelle di inoltro* che contengono un insieme di coppie

(numero di rete, interfaccia di uscita, prossimo router)

- *indirizzamento gerarchico* consente di
 - nelle tabelle dei routers vengono memorizzati solo gli indirizzi delle reti, piuttosto che gli indirizzi dei singoli hosts della rete.
 - Il router associato alla rete destinataria invia il messaggio all'host selezionato.
- *indirizzamento gerarchico*= aumento della *scalabilità del sistema*

INTERNETWORKS (INTERNETS)



**Tabella di inoltro
del router2**

<i>N. di rete</i>	<i>Interfaccia di uscita</i>	<i>Next router</i>
1	0	Router3
2	1	Router1

INOLTRO DEI PACCHETTI

Algoritmo di inoltra eseguito dal nodo N (host o router):

NetworkAdd = indirizzo di rete contenuto nel pacchetto da inoltrare

- *Networkadd è uguale all' indirizzo di rete di una delle interfacce di N*
⇒ il destinatario si trova sulla stessa rete del mittente, il pacchetto può essere consegnato direttamente dal livello link
- *Networkadd è contenuto nella tabella di inoltra di N*
⇒ il pacchetto va consegnato al router *next router*
- *altrimenti*
il pacchetto va consegnato ad un *default router*
(ad esempio un host può avere un default router a cui consegnare tutti i pacchetti non destinati alla rete locale su cui tale host è connesso)

INOLTRO DEI PACCHETTI

- la trasmissione di un pacchetto IP su una rete fisica avviene mediante il livello link
- ogni nodo connesso ad una rete fisica possiede un indirizzo detto indirizzo fisico o *MAC (media acces control)*
- quando un pacchetto IP viene spedito su una rete fisica va utilizzato il MAC del nodo destinazione
- traduzione indirizzo IP–MAC address avviene mediante *ARP (Address Resolution Protocol)*
- Pacchetto IP incapsulato in un frame a livello link che contiene il MAC address del prossimo router o dell'host destinatario

CLASSFULL ADDRESSING: SVANTAGGI

Svantaggi: suddivisione rigida in poche classi \Rightarrow inefficienza nell'assegnazione degli indirizzi IP

Esempio: una struttura universitaria possiede due reti dipartimentale con 180 hosts, una con 260 hosts: sono necessari

- due indirizzi in classe C diversi per le prime due reti (inutilizzati 152 indirizzi IP)
- Un indirizzo in classe B per la terza rete (inutilizzati più di 64000 indirizzi)

Gli indirizzi in classe B sono molto richiesti:

è difficile prevedere l'espansione futura di una rete con un centinaio di hosts \Rightarrow

Si richiede un indirizzo in classe B per evitare di riconfigurare in seguito gli indirizzi IP di tutti gli hosts

CLASSFULL ADDRESSING:SVANTAGGI

- *Classfull Addressing*: un indirizzo di rete diverso per ogni rete fisica connessa
- Dimensione delle tabelle di inoltro è proporzionale al numero di reti fisiche della rete: situazione accettabile solo per internetworks di dimensioni modeste
- Tempo di instradamento proporzionale alla dimensione delle tabelle di inoltro
- Classfull Addressing: alternative
 - *Subnetting*: assegna un numero di rete unico per un insieme di reti fisiche
 - *CIDR (Classless InterDomain Routing)*: aggrega più indirizzi di rete assegnati ad uno stesso sistema in un unico indirizzo visibile dall'esterno

SUBNETTING

- Idea base: uno stesso indirizzo di rete NA (di classe A, B, o C) individua una *rete logica* formata da più *reti fisiche* (*subnets*)
- La rete logica fa parte di una stessa organizzazione (esempio: una struttura universitaria, una grossa azienda,..) \Rightarrow tutte le reti fisiche sono gestite dallo stesso *border router*
- I routers esterni hanno visibilità solo della rete logica (di NA), non delle reti fisiche che ne fanno parte

\Rightarrow

le tabelle di instradamento contengono gli indirizzi IP delle reti logiche(NA), non delle sottoreti

\Rightarrow

diminuisce la dimensione delle tabelle di instradamento

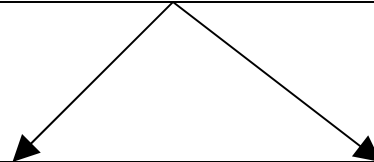
SUBNETTING

Come individuare le singole reti fisiche che fanno parte di una rete logica ?
alcuni bits riservati per l'indirizzo dell'host vengono utilizzati per individuare la sottorete

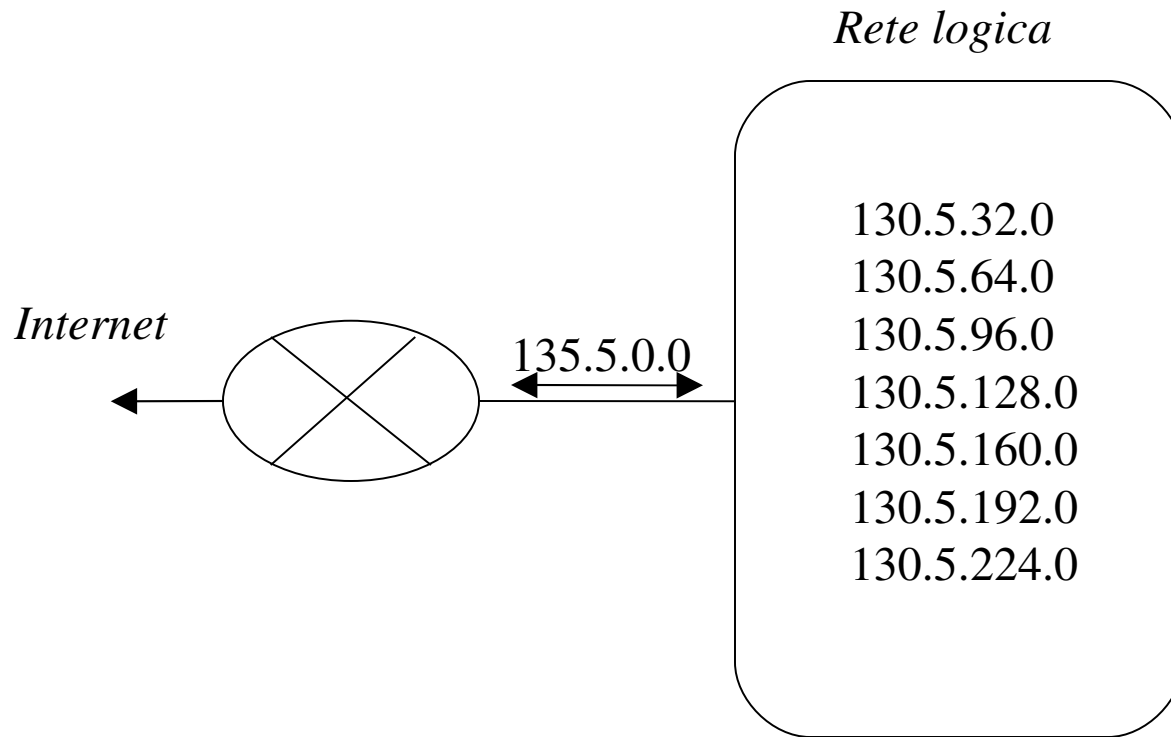
Classfull Addressing: gerarchia a due livelli



Subnetting: gerarchia a tre livelli

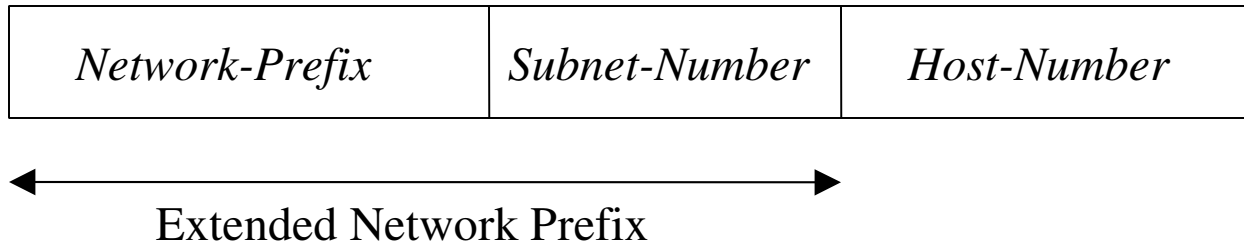


SUBNETTING



- un singolo indirizzo in classe B (130.5.0.0) per la rete logica
- il *terzo ottetto* individua la sottorete

SUBNETTING



- *Network Prefix*: utilizzato dai routers esterni
- *Extended Network Prefix*: utilizzato dai routers interni
- *Problema*: come individuare i bit che fanno parte dell'extended network prefix?

SUBNETTING

Subnet Mask: individua l'extended network prefix

10000010	00000101	00000101	00011001
Network prefix	Subnet number	Host number	
11111111	11111111	11111111	00000000

Maschera di sottorete (255.255.255.0)

I bit della maschera posti ad 1 individuano il network prefix

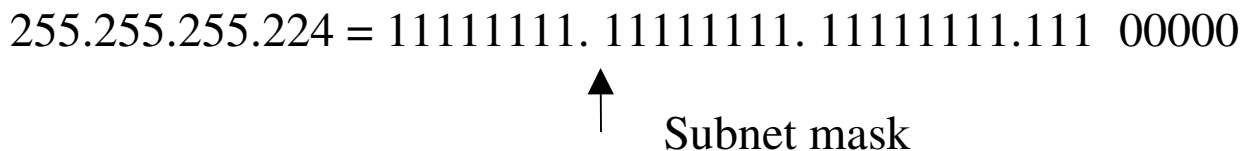
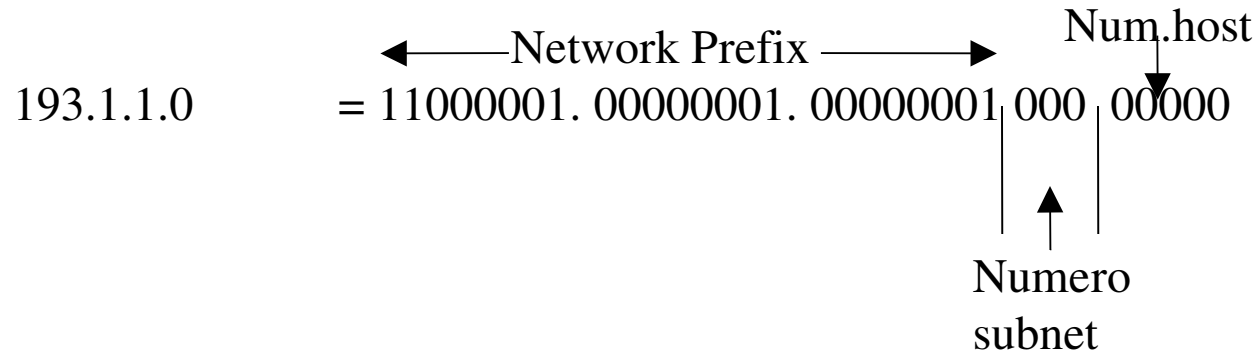
L'AND bit a bit tra un indirizzo e la maschera restituisce l'extended network prefix

SUBNETTING: ESEMPIO

Una struttura universitaria possiede 6 reti. La rete più grande contiene 25 hosts.
All'organizzazione viene assegnato il numero di rete (in classe C) 193.1.1.0.

Per l'individuazione delle 6 reti sono necessari 3 bits \Rightarrow lunghezza dell'extended network prefix = 27

Numero di bits per 'host' = 5 (si possono specificare $32 - 2$ hosts diversi)



CIDR: INSTRADAMENTO SENZA CLASSI

Classless Interdomain Routing: aggrega gruppi di indirizzi IP contigui in un unico indirizzo IP (proposto a partire dal 1990)

Esempio: Per contenere il problema dell'uso inefficiente degli indirizzi IP, ad una organizzazione vengono assegnati 16 indirizzi in classe C contigui, ad esempio gli indirizzi nell'intervallo *192.4.16 -192.4.31*, piuttosto che un indirizzo in classe B.

⇒

i primi 20 bits di tutti gli indirizzi nell'intervallo sono uguali a

11000000 00000100 0001

CIDR considera l'insieme di reti dell'organizzazione come un'unica rete di indirizzo 11000000 00000100 0001.

CIDR: INSTRADAMENTO SENZA CLASSI

- CIDR: i routers esterni all'organizzazione vedono le reti di un'organizzazione come una unica rete aggregata (*supernetting*)
- Il numero di bits necessari per individuare la rete aggregata *varia a seconda* della sua dimensione

⇒

poiché i routers esterni utilizzano per l'instradamento l'indirizzo della rete aggregata, devono gestire indirizzi con network-prefix di *dimensione variabile*

⇒

classless routing= routing che non utilizza le classi per l'instradamento

CIDR: INSTRADAMENTO SENZA CLASSI

- Utilizzato da OSPF e BGP
- Utilizzato nei backbone routers
- Notazione utilizzata `xxx.xxx.xxx.xxx/n` dove `n` e' il numero di 1 nella maschera
- Esempio: `192.168.12.0/23` corrisponde alla maschera di rete `255.255.254.0`

RMI :

IL MECCANISMO DELLE CALLBACK

Meccanismo RMI

- comunicazione unidirezionale (dal client al server)
- *comunicazione sincrona, rendez voz esteso*: il client invoca un metodo remoto e si blocca finchè il metodo non termina

In molte applicazioni

- il client è interessato ad un evento che si verifica sul server e notifica il suo interesse al server (ad esempio utilizzando RMI)
- il server registra che il client è interessato in quell'evento
- quando l'evento si verifica, il server lo notifica ai clients interessati l'accadimento dell'evento

RMI :

IL MECCANISMO DELLE CALLBACK

Esempi di applicazioni:

- un utente partecipa ad una chat e vuol essere avvertito quando un nuovo utente entra nella 'chat room'
- lo stato di un gioco multiplayer viene gestito da un server. I giocatori notificano al server le modifiche allo stato del gioco. Ogni giocatore deve essere avvertito quando lo stato del gioco subisce delle modifiche.
- gestione distribuita di un'asta: un insieme di utenti partecipa ad un'asta distribuita. Ogni volta che un utente fa una nuova offerta, tutti i partecipanti all'asta devono essere avvertiti

RMI :

IL MECCANISMO DELLE CALLBACK

Soluzioni possibili per realizzare nel server un servizio di notifica di eventi al client:

- *Polling*: il client interroga ripetutamente il server, per verificare se l'evento atteso si è verificato. L'interrogazione avviene mediante *l'invocazione di un metodo remoto* (mediante RMI)

Svantaggio: alto costo per l'uso non efficiente delle risorse del sistema

- *Registrazione* dei clients interessati agli eventi e successiva notifica (asincrona) del verificarsi dell'evento al client da parte del server

Problema: quale meccanismo utilizza il server per risvegliare il client?

RMI :

IL MECCANISMO DELLE CALLBACK

- E' possibile utilizzare RMI sia per l'invocazione client-server(registrazione del client) che per quella server-client(notifica del verificarsi di un evento) utilizzando il *meccanismo delle callback*
- Il server definisce una interfaccia remota *ServerInterface* che definisce un metodo remoto che può essere utilizzato dal client per registrarsi
- Il client definisce una interfaccia remota *ClientInterface* che definisce un metodo remoto utilizzato dal server per notificare un evento al client
- Il server ha a disposizione la *ClientInterface* e lo stub relativo
- Il client ha a disposizione *la ServerInterface* e lo stub relativo

RMI :

IL MECCANISMO DELLE CALLBACK

- Il client, al momento della registrazione sul server, passa al server un riferimento RC ad un oggetto che implementa la *ClientInterface*
- Il server memorizza RC in una sua struttura dati (ad esempio, un vector)
- al momento della notifica, il server utilizza RC per invocare il metodo remoto di notifica definito dal client.
- In questo modo rendo ‘simmetrico’ il meccanismo di RMI, ma...
il client non registra l’oggetto remoto in un rmiregistry, ma passa un riferimento a tale oggetto al server, al momento della registrazione

ESERCIZIO: GESTIONE FORUM

Si vuole implementare un sistema che implementi un servizio per la gestione di forum in rete. Un forum e' caratterizzato da un argomento su cui diversi utenti possono scambiarsi opinioni via rete. Il sistema deve prevedere un server RMI che fornisca le seguenti funzionalita':

- a) apertura di un nuovo forum, di cui e' specificato l'argomento (esempio: giardinaggio)
- b) inserimento di un nuovo messaggio indirizzato ad un forum identificato dall'argomento (es: e' tempo di piantare le viole, indirizzato al forum giardinaggio)
- c) reperimento dell'ultimo messaggio inviato ad un forum di cui e' specificato l'argomento.

ESERCIZIO: GESTIONE FORUM

Si devono definire almeno le seguenti classi:

- una interfaccia *ForumInterface*, che definisce i metodi del server che possono essere invocati in remoto
- una classe *Forum* che implementa la struttura dati relativa ad un singolo forum. Il numero massimo di messaggi per ogni forum e' definito staticamente ed e' uguale per ogni forum.
- una classe *Forumimpl* che implementa i metodi definiti nella interfaccia *ForumInterface*. Il numero massimo di forum che possono essere gestiti da *Server* e' determinato staticamente.
- Una classe *ForumObj* che attiva una istanza dell'oggetto remoto *Forumimpl*
- una classe *Client* che interagisce con l'utente ed, in base alle richieste dell'utente, invoca i metodi di *Server*.
- una classe *Messaggio* che implementa la struttura dati che descrive il messaggio inviato al forum .

ESERCIZIO: CALLBACS

Nella lezione precedente è stato assegnato un esercizio per la gestione elettronica di una elezione a cui partecipano un numero prefissato di candidati. Si chiedeva di realizzare un server RMI che consentisse al client di votare un candidato e di richiedere il numero di voti ottenuti dai candidati fino ad un certo punto.

Modificare l'esercizio in modo che il server *notifichi ogni nuovo voto ricevuto* a tutti i clients che hanno votato fino a quel momento. La registrazione dei clients sul server avviene nel momento del voto.