Politecnico di Milano – Sede di Cremona

A.A. 2005/06

Corso di FONDAMENTI DI RETI DI TELECOMUNICAZIONI

Martino De Marco
(demarco@cremona.polimi.it)

Parte 4

RETI IP

Indice

- Introduzione
- Lo stack protocollare
- Indirizzamento e instradamento
- Architettura di Internet

Le origini

- 1965: ARPA sponsorizza la ricerca “cooperative network of time-sharing computer”
- 1969: primi host di ARPANET (collegamento tra UCLA, SRI, UCSB e University of Utah)
- ’70s: ARPANET è un successo, l’e-mail diventa immediatamente l’applicazione più utilizzata
- 1973: ARPANET diventa internazionale, con la connessione allo UCL (Londra) e Royal Radar Establishment (Norvegia)
- 1975: test di TCP utilizzando un link via satellite tra UCL e Stati Uniti

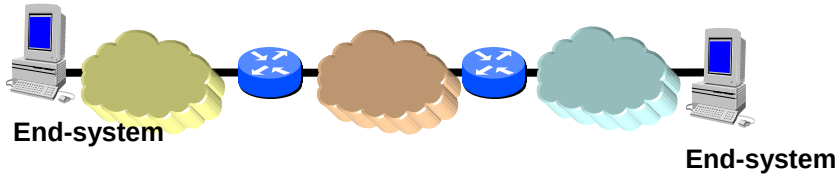
La storia

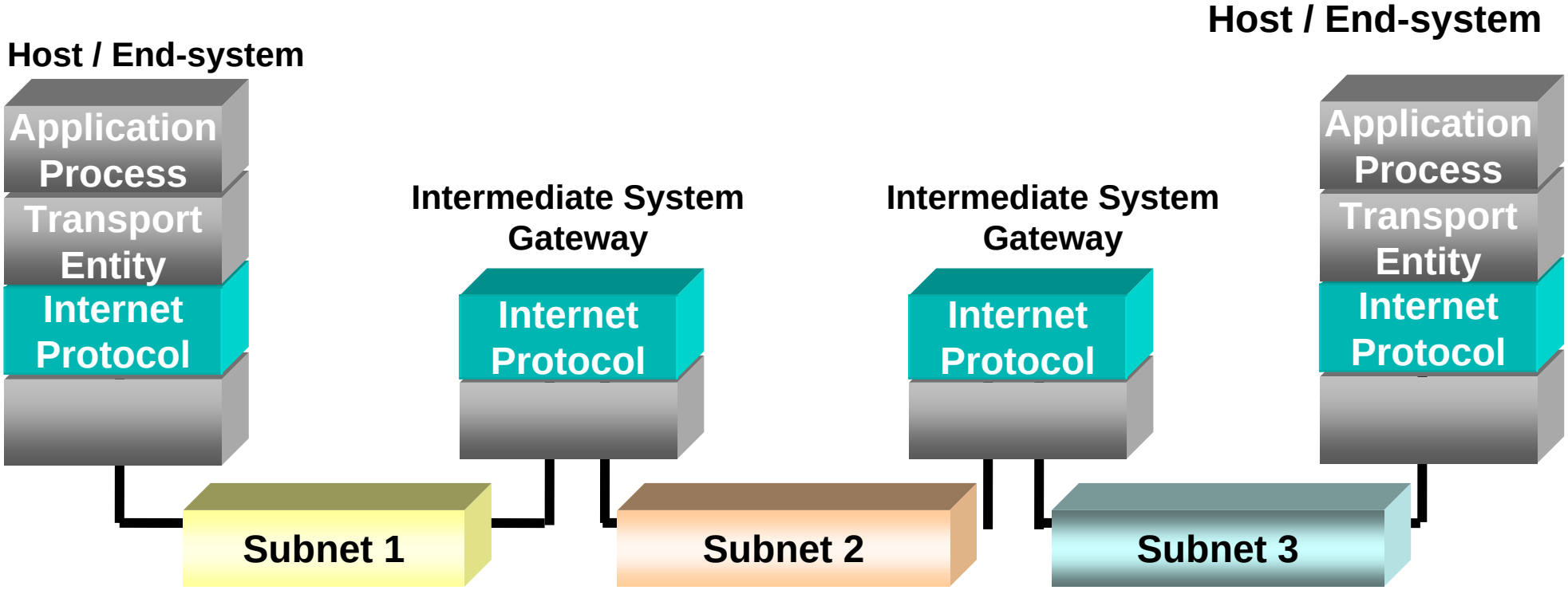
- Inizi ’80s: DARPA finanzia lo sviluppo del Berkeley UNIX e decide di implementare nativamente TCP/IP
- 1982: L’insieme di reti costituenti ARPANET è articolato come una “internet”
- Fine ’80s: NSFNET diviene il backbone di Internet
- 1990: ARPANET è “decommissioned”
- 1993: Mosaic diviene disponibile
- 1995: L’era degli ISP e dell’uso commerciale di Internet
- Oggi: Internet è “*THE NET*”

IP e l'Internet philosophy

- I principi fondamentali
  - Backward compatibility con le infrastrutture esistenti
  - "*The goal is connectivity, the tool is the Internet Protocol, and the intelligence is end to end rather than hidden in the network*"
  - Funzionalità *end-to-end* possono essere realizzate al meglio da protocolli end-to-end
  - Nessun proprietario, nessun controllo centralizzato, nessuno ha potere di vita o di morte
- Punti fermi della soluzione
  - hardware e software eterogenei
  - Scalabilità
  - Simplicità
  - Packet switching realizzato da nodi *store&forward*
  - Univocità dell'indirizzamento a livello network

Un paradigma vincente





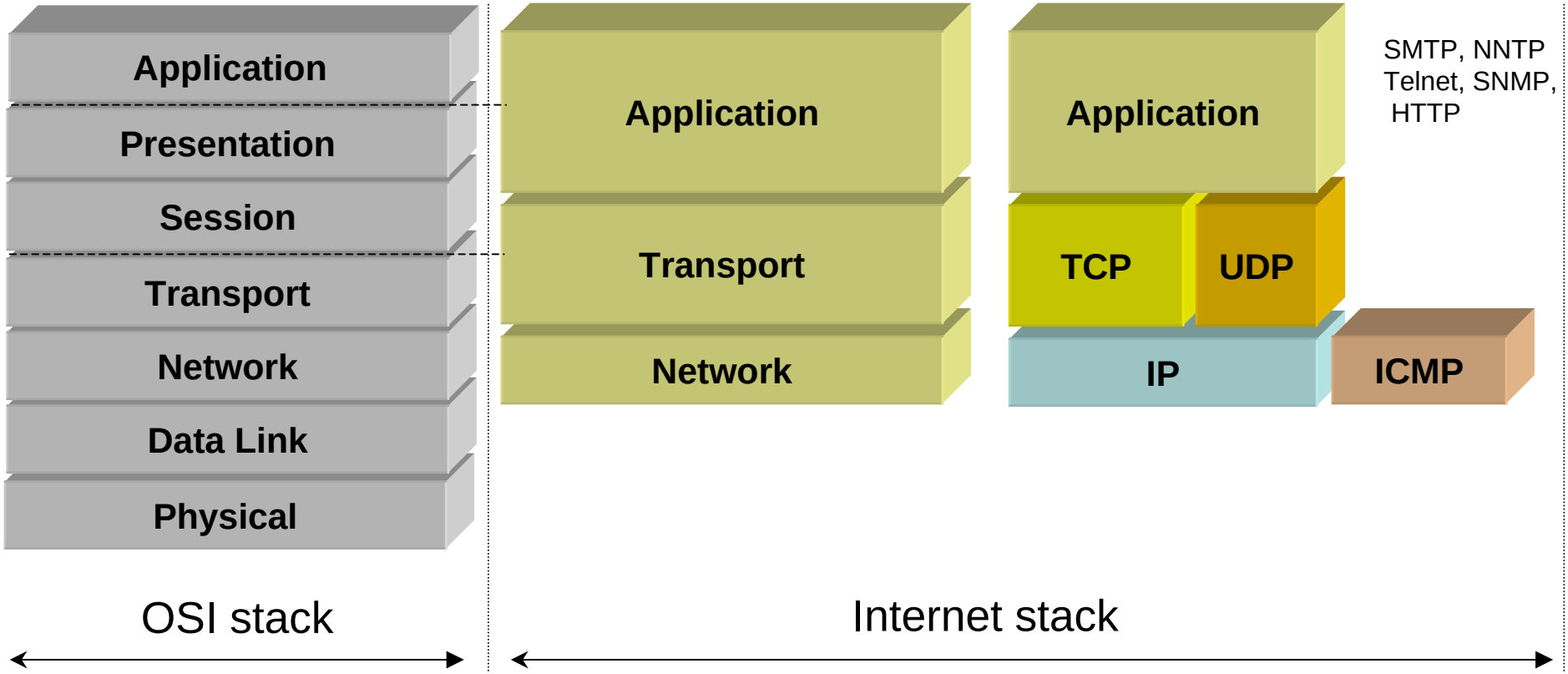
Indice

- Introduzione
- Lo stack protocollare
- Indirizzamento e
- Architettura di Int

  - OSI e Internet stack
  - Network Layer
    - IP
  - Transport layer
    - UDP
    - TCP
  - Altri utili protocolli

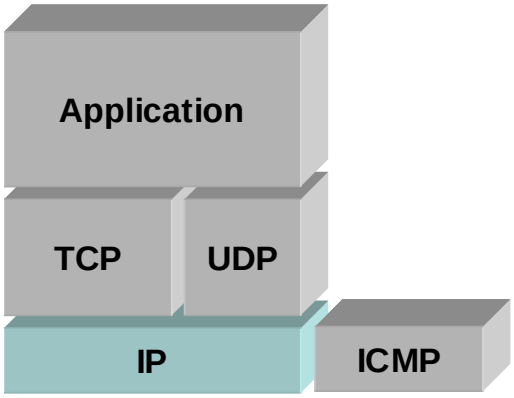
OSI vs Internet stack

- *Internet stack is "light" and "condensed"*



Network layer: Internet Protocol (IP)

- Corrisponde al Livello 3 del modello OSI
- Definito da RFC
- Network independent
- Protocollo Connectionless:
  - best effort
  - unreliable data delivery
- Segmentation e reassembly di datagrammi
- Indirizzamento
- Instradamento
- Rilevazione e notifica di errori



Task IP

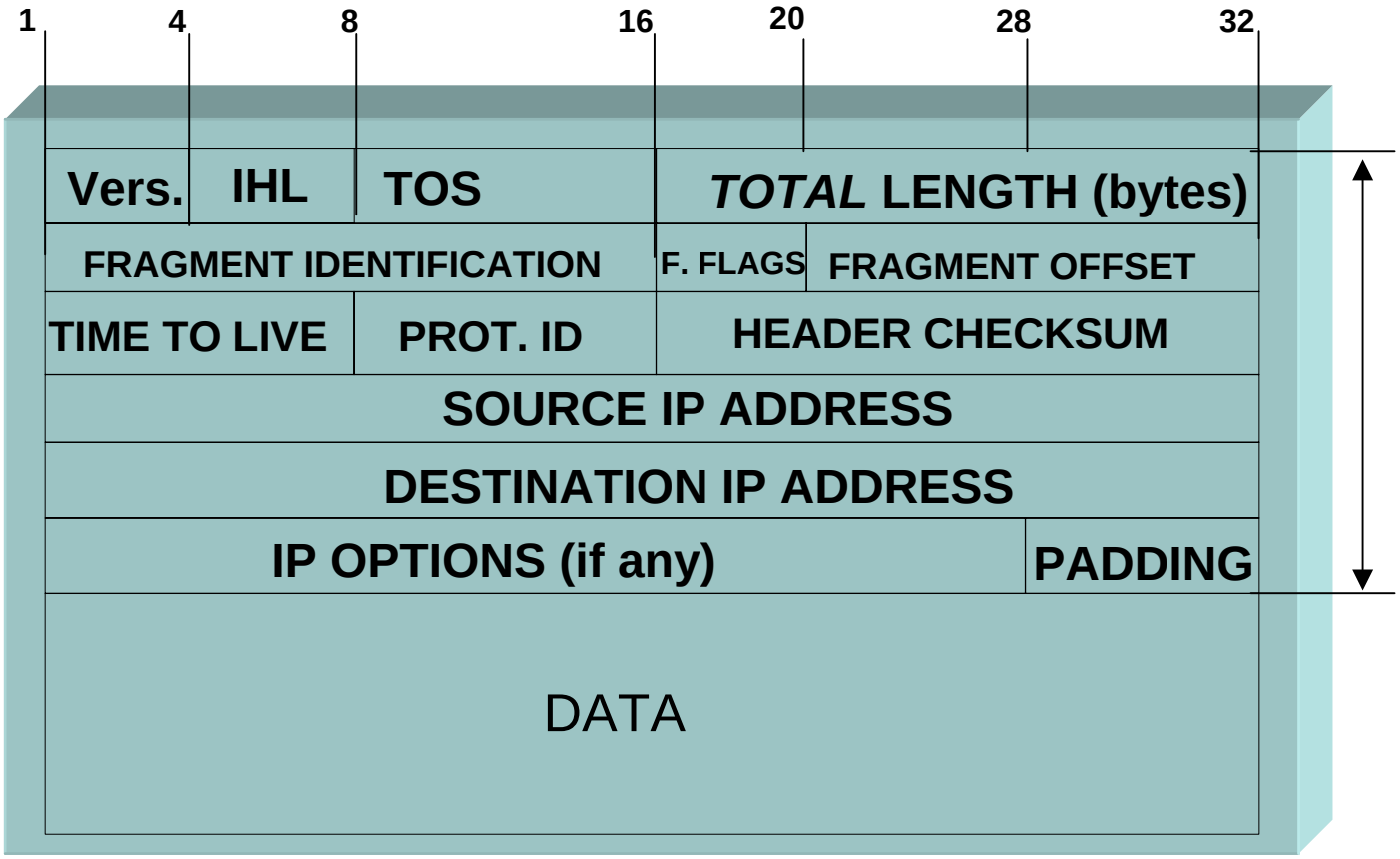
- Riceve i dati ed i parametri di trasferimento dal protocollo di livello trasporto
- Costruisce datagrammi
  - Determina il valore corretto dell'header checksum
- Prende decisioni di instradamento
  - Determina il prossimo nodo sul percorso verso la destinazione
- Prepara i dati per la trasmissione
- Interagisce con il protocollo di livello inferiore
- Verifica eventuali errori di trasmissione nell'header

Entità IP in un gateway

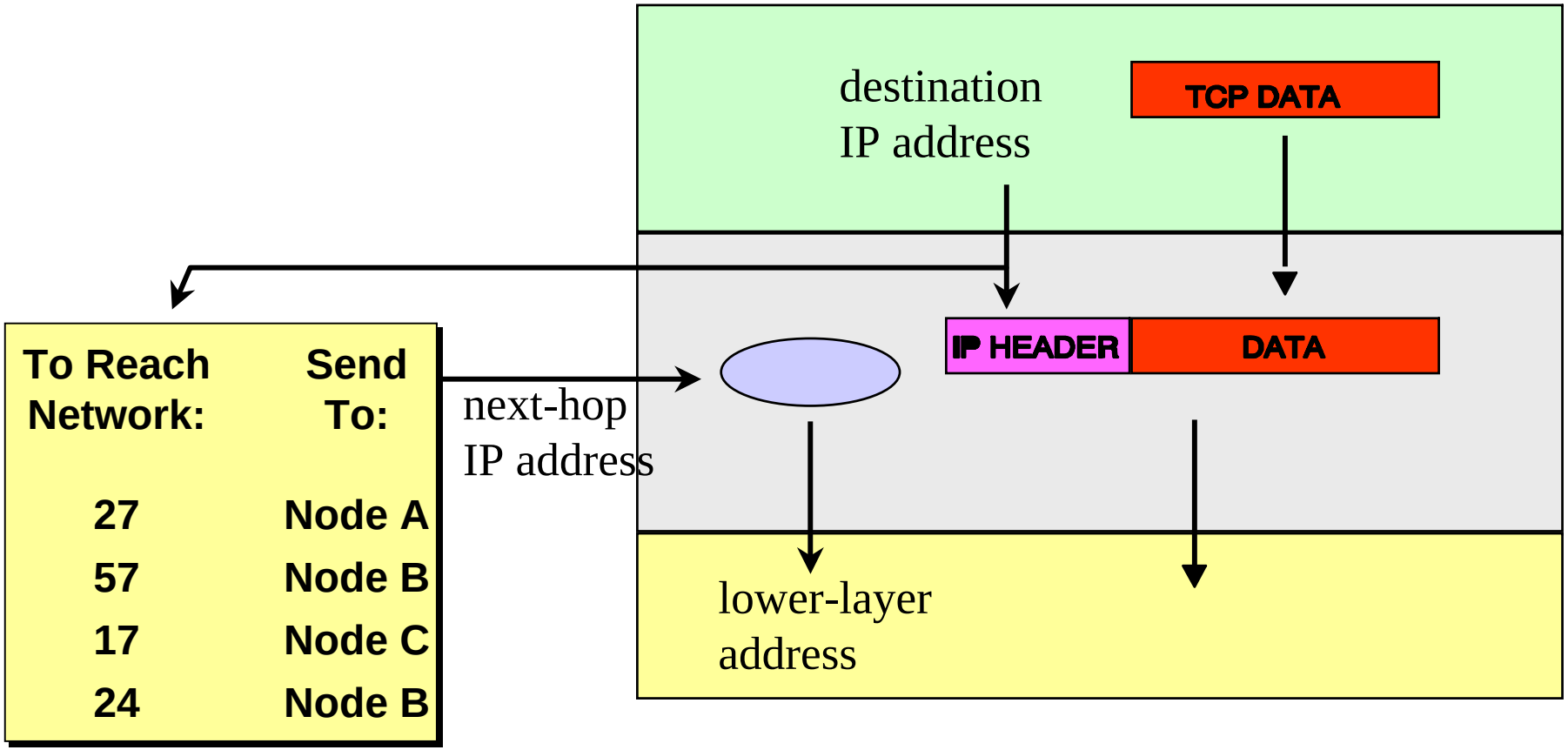
- Verifica la correttezza dell'*header checksum* (*error control*)
- Verifica l'*IP version*
- Decrementa il valore del TIME TO LIVE di 1
- Implementa le funzioni specificate dalle opzioni
- Determina il next hop e l'interfaccia di uscita del datagramma
- Fragmenta i datagrammi se necessario

IP – formato del datagramma

IHL: *IP Header length* (in parole di 4 ottetti)
PROT.ID: *Protocol Identifier* assegnato dallo IANA (RFC1700)
F. FLAGS: *Fragmentation Flags* (Don't Fragment, More Fragments)
IP OPTIONS: *Loose Source Routing, Strict Source Routing*



Inoltro dei datagrammi



Frammentazione

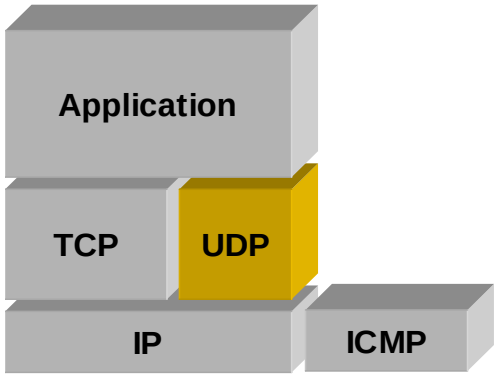
- Se un datagramma ha dimensione superiore alla *Maximum Transfer Unit* (MTU) consentita dal livello inferiore, il nodo IP:
  - Frammenta il datagramma
  - Copia l'header del datagramma originario in ogni frammento
  - Imposta i bits FRAGMENT CONTROL in tutti gli header dei frammenti (*Flag* + *Fragment offset*)
- Il riassemblaggio viene realizzato solo dall'host "destinatario finale" del datagramma

Riassemblaggio

- L’entità IP di destinazione:
  - Memorizza tutti i frammenti nel *reassembly buffer*
  - Posiziona ogni frammento nel buffer in accordo con il suo valore di offset
  - Verifica la ricezione di tutti i frammenti (tramite il conteggio dei byte ricevuti e dei valori del fragment offset)
  - Scarta tutti I frammenti ricevuti se entro un certo timeout non è in grado di riassemblare il datagramma originario

Transport layer: User Datagram Protocol

- Connectionless service
- Unreliable

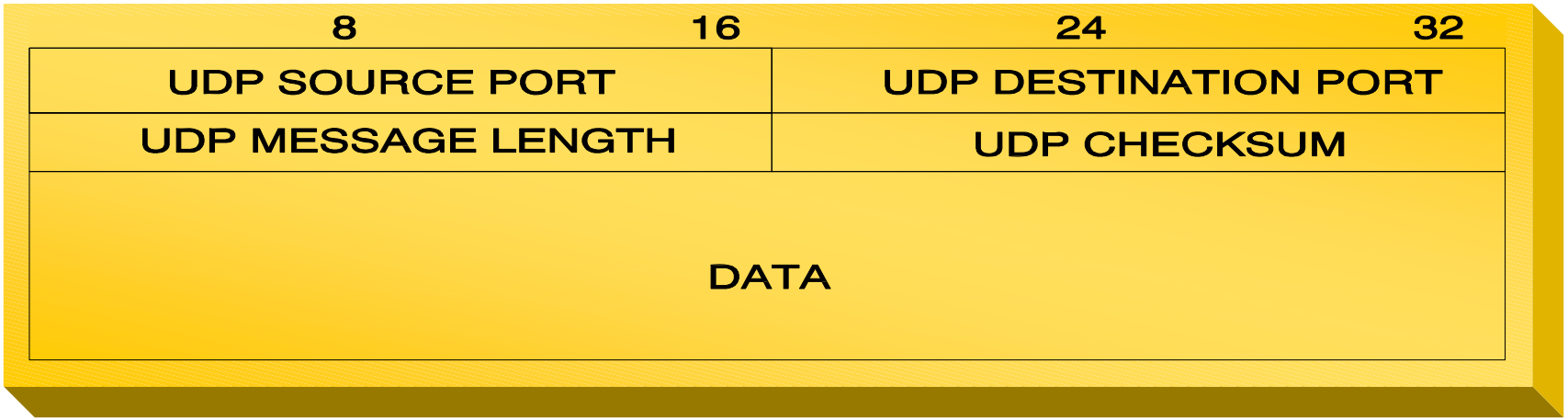


- UDP <u>implementa</u>:
  - Multiplexing per mezzo dei socket
  - Controllo d'errore sull'header opzionale
  - Verifica della corretta identità del destinatario, utilizzando uno pseudoheader nel calcolo della checksum
  - UDP è adatto ad applicazioni di tipo transazionale (SNMP, DNS)
  - Preferibile nei casi in cui I meccanismi di controllo di flusso del TCP non consentono di garantire un limite al ritardo (RTCP)

- UDP <u>non supporta</u>:
  - Consegna dei dati in sequenza
  - Rilevazione di datagrammi persi o duplicati
  - Controllo di flusso

Formato dei messaggi UDP

- UDP SOURCE PORT (16 bits), *port* (=applic.) dal quale provengono i dati
- UDP DEST. PORT (16 bits): *port* al quale consegnare i dati sull'host remoto
- UDP MESSAGE LEN. (16 bits): ottetti del datagramma incluso l'header UDP
- UDP CHECKSUM (16 bits): checksum (opzionale), 0 se non utilizzata



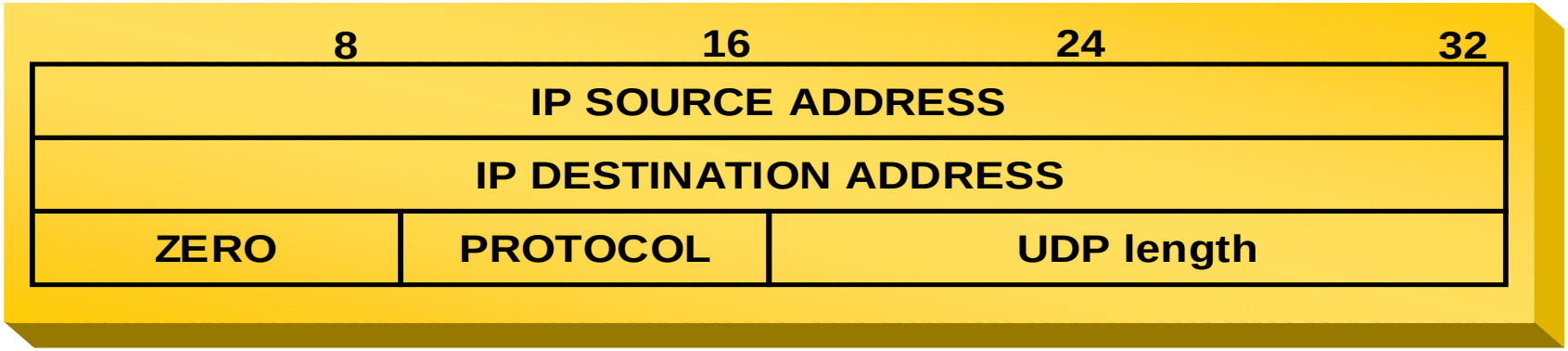
UDP ports

No.	Port
0	Reserved
7	Echo
9	Discard
11	Active users
13	Daytime
15	Netstat
17	Quote
19	Chargen
37	Time
42	Host NS
43	whois
525	Time demon

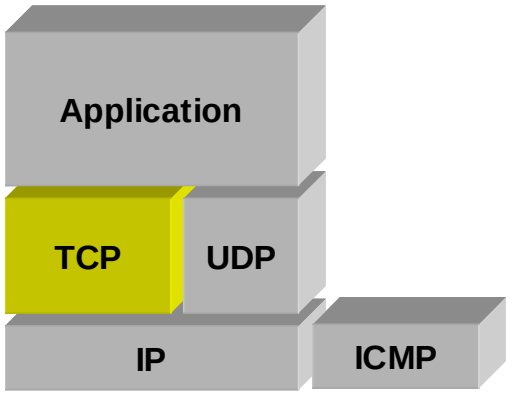
No.	Port
53	DNS
67	BootP server
68	BootP client
69	TFTP
111	Sun RPC
123	Network Time Prot.
161	SNMP mon.
162	SNMP trap
512	UNIX comstat
513	UNIX rwho
514	System log

Task UDP

- L'applicazione locale passa i dati process all'entità UDP
- UDP aggiunge il suo header ed un *temporary header*
- UDP aggiunge il valore della checksum all'header, scarta il *temporary header* e passa il datagramma all'entità IP
- L'entità UDP remota, ricevuto il datagramma, aggiunge il *temporary header* e calcola la checksum
- Se il valore calcolato coincide con quello riportato nell'header, il datagramma viene consegnato al processo applicativo identificato dal *port number*
- Altrimenti il datagramma è scartato



Transport layer: TCP



- Transmission Control Protocol (TCP)
- Di fatto costituisce il 90% del traffico Internet

- Protocollo Connection Oriented
- Comunicazione Full-duplex unicast
- Stato della connessione sincronizzato tra i due end-point

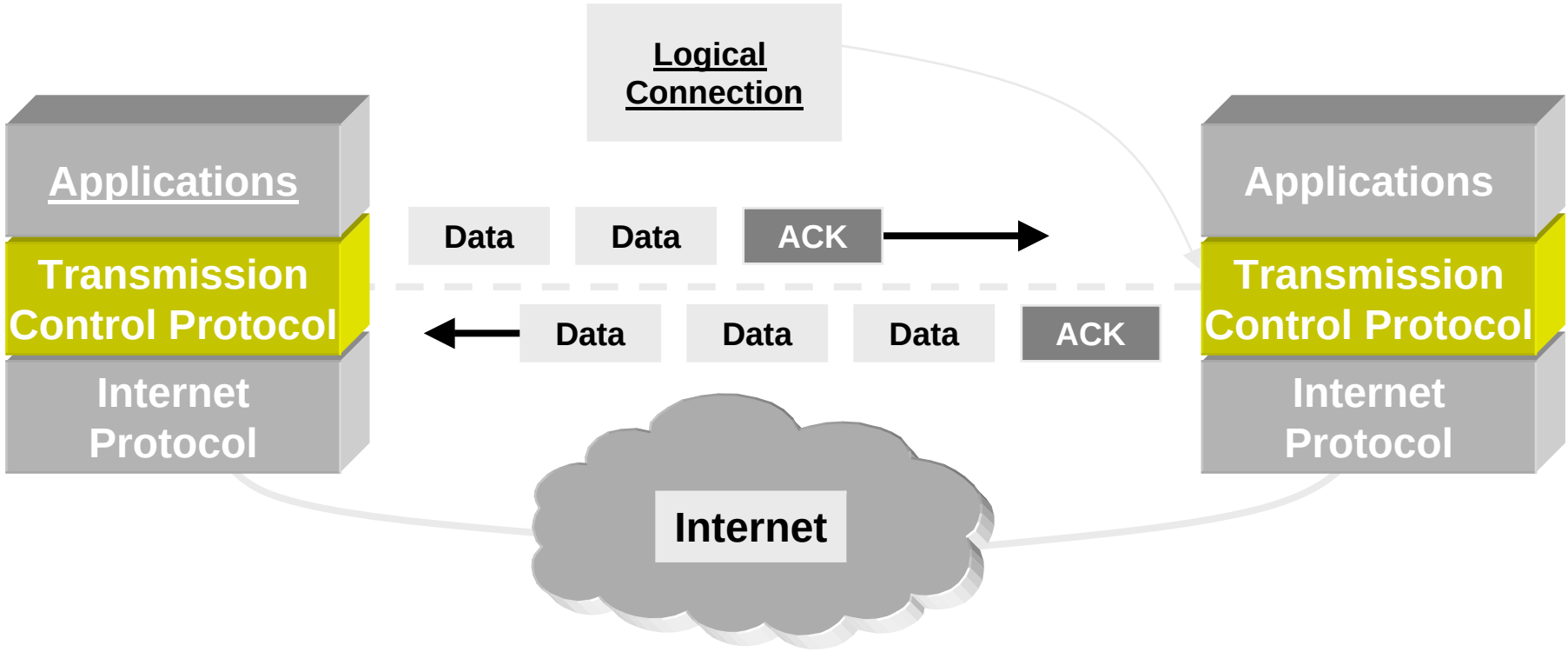
- TCP fornisce:
  - Affidabilità
  - Garanzia di sequenza
  - Controllo di flusso
  - Streaming
  - Adattamento della velocità di trasmissione tra gli end-point (controllo di congestione)

Assunzioni del TCP

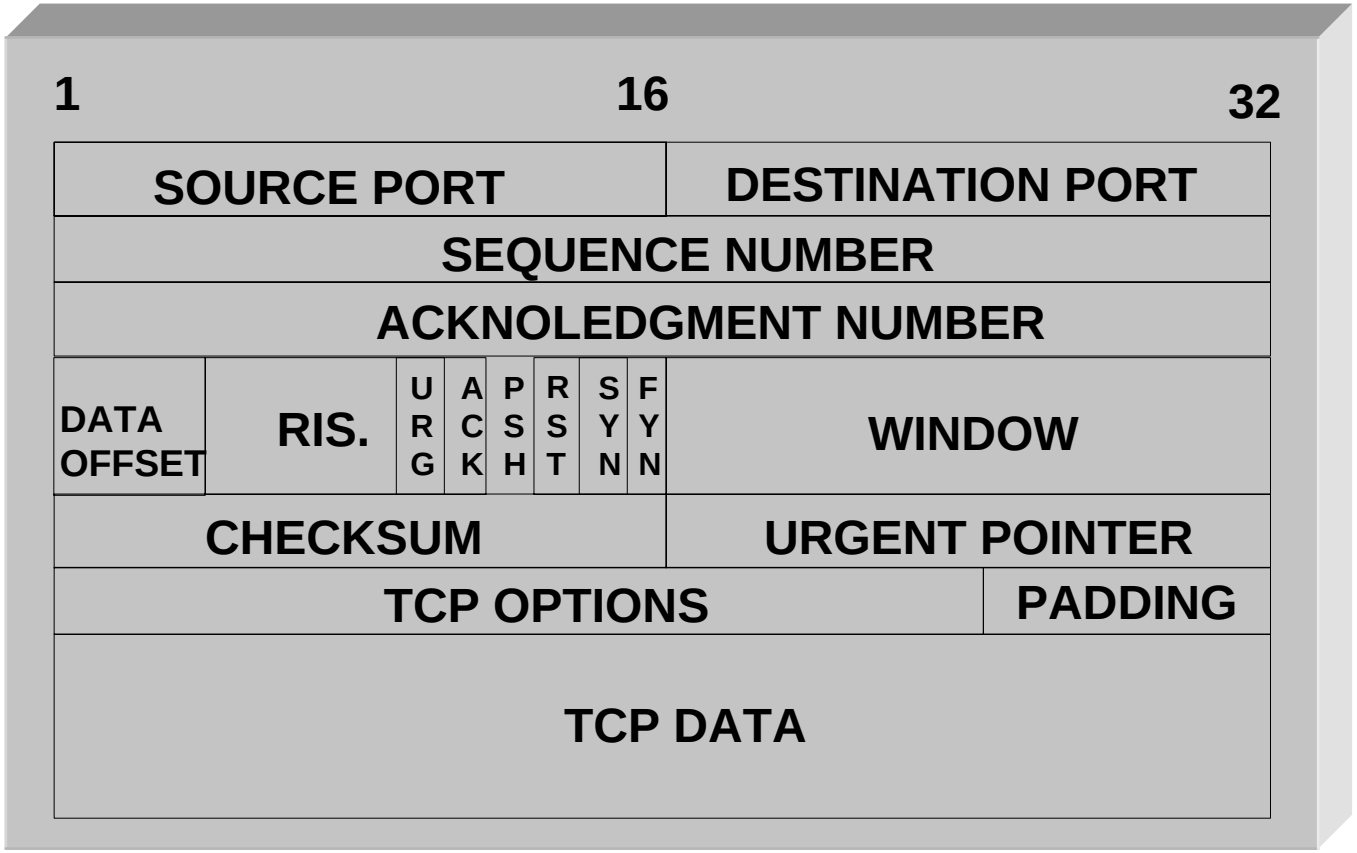
- Equità nell'utilizzo delle risorse di rete tra diversi flussi TCP
- Utilizzo efficiente delle risorse di rete a scapito della predicibilità del ritardo di trasmissione
- Comportamento adattativo:
  - Sincronizza sorgente e ricevitore per ottimizzare il flusso di trasporto dati ("*inject only what the receiver is expected to retrieve from the net*")
  - Capacità di adattare la trasmissione all'attuale bandwidth-delay-product: $Window\_Size \geq Bandwidth$ (byte/sec) x $Round\text{-}Trip\ Time$ (sec)
- La perdita di pacchetti è usualmente attribuita alla congestione dei link piuttosto che ad errori di trasmissione sul canale

Task TCP

- *Connection identifier* di 96 bit
  - 32 bit IP source address + 16 bit source port
  - 32 bit IP destination address + 16 bit destination port
- Port number possono essere riutilizzati più volte (sulla stessa macchina) per differenti connessioni



Formato del segmento TCP (1)



Formato del segmento TCP (2)

- SOURCE PORT (16 bits): *port number* dell'applicazione sorgente
- DEST. PORT (16 bits): *port number* dell'applicazione destinazione
- SEQUENCE NUMBER (32 bits): numero di sequenza assegnato al primo ottetto di dati o numero iniziale di sequenza da utilizzare per una connessione
- ACKNOWLEDGMENT NO. (32 bits): numero di sequenza del prossimo ottetto che il sender si aspetta di ricevere
- DATA OFFSET (4 bits): lunghezza dell'header in parole da 32-bit
- RESERVED (6 bits): per uso futuro
- FLAGS (6 bits): identifica funzioni speciali
- WINDOW (16 bits): dimensione della finestra di ricezione (numero di ottetti che il ricevitore è disposto ad accettare)
- CHECKSUM (16 bits): codice di ckecksum per il controllo d'errore
- URGENT POINTER (16 bits): posizione nel segmento dei dati "*urgent*"

TCP ports

No.	Port
0	Reserved
1	TCP mux
5	RJE
7	Echo
9	Discard
10	11 Active users
13	Daytime
15	Who is up?
17	Quote
53	DNS
67	BOOTP server
68	BOOTP client
69	TFTP
75	Private dialout
77	RJE
79	Finger
101	NIC server

No.	Port
19	Chargen
20	FTP (data)
21	FTP (Control)
23	TELNET
25	SMTP
37	Time
39	Resloc 1
42	Host NS
43	whois
105	CSNet NS
109	POP, Ver. 2
111	Sun RPC
113	Authentication service
115	Simple FTP
117	UUCP Data
119	News
102	ISO-TSAP

No.	Port
93	Dev Contrl
123	NTP
95	SUPDUP
103	X.400
104	X.400-SND
129	Passwgen
137	NETBIOS
161	SNMP agent
162	SNMP manager
160-223	reserved

Opzioni nel TCP

- Maximum-receive-segment-size (MSS)
  - Concordata all'apertura della connessione (nel *SYN packet*)
  - Deve essere preceduta da un'operazione di *MTU discovery* (nell'Ethernet MTU=1500 bytes)

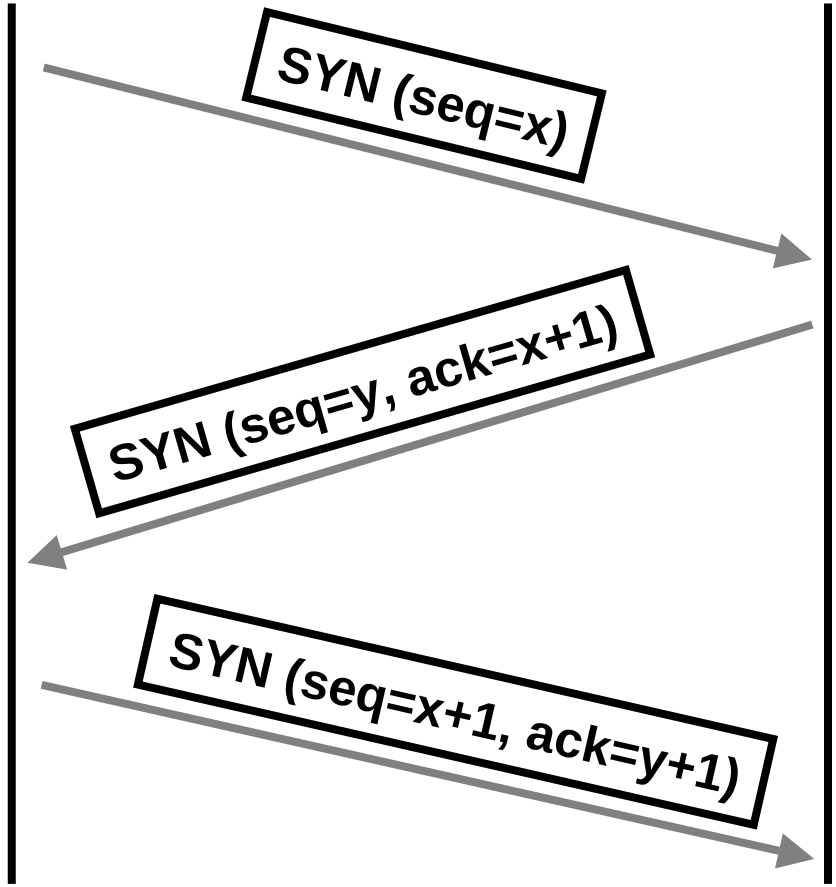
- Window-scale option
  - Aumenta la dimensione della finestra di ricezione (2^{30} bytes rispetto a 64 Kbytes)
  - Utile per reti caratterizzate da ritardi elevati (es. satellite)
  - Negoziato all'attivazione della connessione

- SACK-permitted option
  - *Selective acknowledgement*
  - implementa la ritrasmissione in modalità *selective repeat*
  - Negoziato all'attivazione della connessione

Instaurazione della connessione

- L’entità TCP comunica con:
  - Il processo applicativo sullo stesso host, al fine di determinare:
    - Il tipo di connessione da stabilire
    - I parametri di trasferimento dei dati
  - L’entità paritetica (TCP) sull’host remoto per:
    - Allocare le risorse di comunicazione
    - Verificare che la sessione di comunicazione sia mutuamente stabilita
    - Identificare i socket locale e remoto da utilizzare
    - Settare i parametri di trasferimento dati (es. numero di sequenza iniziale, finestra di ricezione, ...)

Three-way handshake

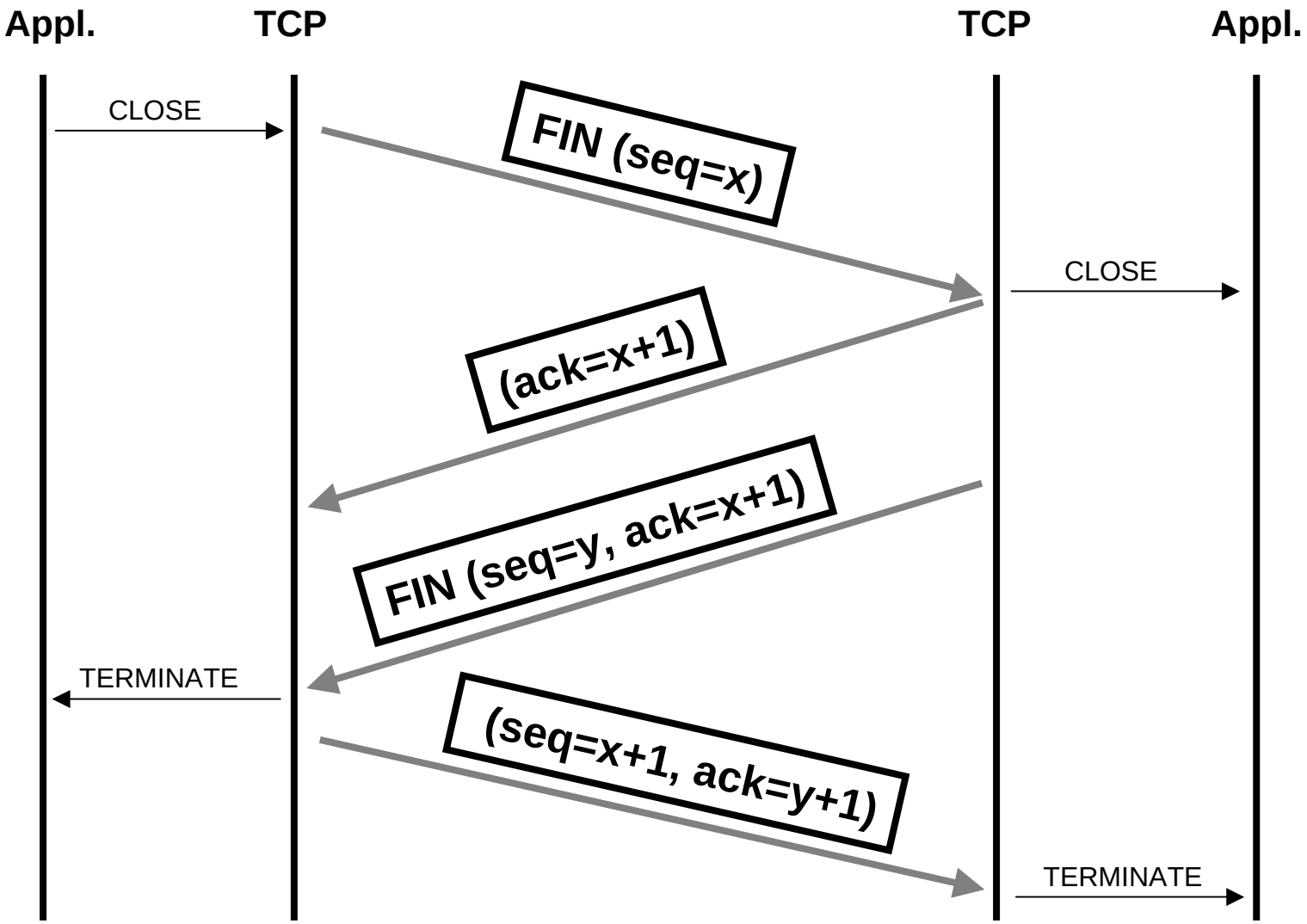


- Previene false connessioni
- Ignora ack di richieste di connessione in ritardo o duplicati
- Permette di includere dati nel messaggio di richiesta di connessione
- Se il tentativo di instaurazione fallisce:
  - si ritenta dopo 3s,
  - poi dopo altri 6s
  - e ancora dopo ulteriori 12s

Rilascio della connessione

- Una delle due entità applicative invia una primitiva CLOSE all'entità TCP
- L'entità TCP informa la corrispondente remota (inviando un segmento FIN) e le trasferisce tutti i dati che si trovano al momento nel proprio buffer
- L'entità TCP remota informa l'applicazione della richiesta di rilascio e trasmette tutt i dati bufferizzati al peer TCP
- L'entità remota riscontra la richiesta (segmento FIN ACK) quando ha terminato di trasmettere tutti i dati in coda
- Quando l'entità TCP locale riceve l'ack, comunica al processo applicativo la primitiva TERMINATE, invia all'entità TCP remota un ACK e rilascia la connessione
- L'entità remota riceve l'ack e passa una primitiva TERMINATE al processo applicativo

Rilascio della connessione



Trasferimento dati

- Le due entità TCP trasferiscono dati:
  - Trasmissione full duplex
  - PUSH function (PSH): forza i limiti del controllo di flusso
  - URGENT function (URG): trasmette dati urgenti fuori banda (identificabili nel segmento mediante URGENT POINTER)

- Servizi di Connection management:
  - Numerazione di sequenza
  - acknowledgment positivo e cumulativo
  - sliding window con numerazione in bytes
  - ritrasmissione

Numerazione di sequenza

- Ogni ottetto trasmesso è identificato da un numero di sequenza
- La duplicazione del numero di sequenza è evitata grazie al contatore a 32 bit

- Sequence numbers support
  - Riordino in sequenza dei segmenti ricevuti
  - acknowledging
  - Controllo di flusso

- *Il valore dell'ack specifica il numero di sequenza del prossimo ottetto che il trasmettitore si aspetta di ricevere*

Error detection

- Permette di verificare se il segmento ricevuto è corrotto
  - La checksum copre l'header, la parte dati ed un pseudoheader
  - Le informazioni per completare lo pseudoheader sono passate dall'IP (indirizzi IP sorgente e destinazione, valore del campo *Protocol*)
- Verifica se sussistono condizioni di errore dovute al processo applicativo
  - incorrect service request
  - internal processing error

Controllo di flusso

- All'instaurazione della connessione, ogni peer specifica la dimensione della finestra di ricezione (*receive window*) disponibile per la connessione
  - La *receive window* rappresenta lo spazio disponibile nel buffer di ricezione (***R.WND***)
- Ad ogni ACK, il ricevitore comunica il valore attuale del buffer di ricezione (*Window Advertising*)
  - Il numero di bytes che il ricevitore è pronto a memorizzare aggiorna il valore della finestra di trasmissione del sender (***S.WND***)
  - Il valore ***S.WND*** definisce l'ammontare di spazio disponibile nel buffer di trasmissione

Sliding window

- Controlla il flusso di dati in transito tra *peer entity*
  - Utilizza la numerazione di sequenza associato agli ACK
  - I *Credits* rappresentano il numero di ottetti che possono essere trasmessi
- Ritrasmissione
  - Il TCP sender associa un timer ad ogni segmento ed allo scadere di un *timeout* effettua la ritrasmissione
  - Dati non riscontrati devono essere memorizzati dalla sorgente per eventuali ritrasmissioni
- La scelta della dimensione della finestra ha un impatto significativo sulle prestazioni del TCP
  - Default window size: 4.096 bytes
  - Maximum window size: 65.536 bytes

Controllo di Congestione

- Il protocollo TCP, a causa dell'assenza di meccanismi di controllo del rate di trasmissione nelle reti IP, può trasmettere dati ad elevata velocità (limitato solo dalla dimensioen della finestra di ricezione) causando congestione nei gateway

- Deve quindi essere implementato a livello TCP un meccanismo di controllo della congestione (integrato con il controllo di flusso)
  - Si utilizza un terzo tipo di finestra: *(sender) congestion window* (***C.WND***)
  - ***S.WND = min(R.WND;C.WND)***

- Ad una connessione unidirezionale vengono associati due stati:
  - ***Slow start***: ***C.WND*** cresce esponenzialmente a partire da un MSS
  - ***Congestion avoidance***: ***C.WND*** cresce linearmente e torna allo slow start nel momento in cui si verifica congestione
  - Una soglia tempo variante (***SSTHRES***) definisce la transizione tra i due stati

Slow start

- Inizializzazione:
  - $SSTHRES = S.WND_{initial}/2$
  - $C.WND = 1*MSS$

- La $C.WND$ cresce esponenzialmente fino alla soglia $SSTHRES$
  - La $C.WND$ aumenta di $1*MSS$ per ogni ack ricevuto entro il timeout
  - $C.WND = 1*MSS, 2*MSS, 4*MSS, 8*MSS, 16*MSS, ...$

- Se un datagramma viene perso, il TCP attribuisce l'evento ad uno stato di congestione (e non al errori sul canale) ed allo scadere del timeout di trasmissione:
  - Resetta la $C.WND = 1*MSS$
  - Riprende nell stato slow start

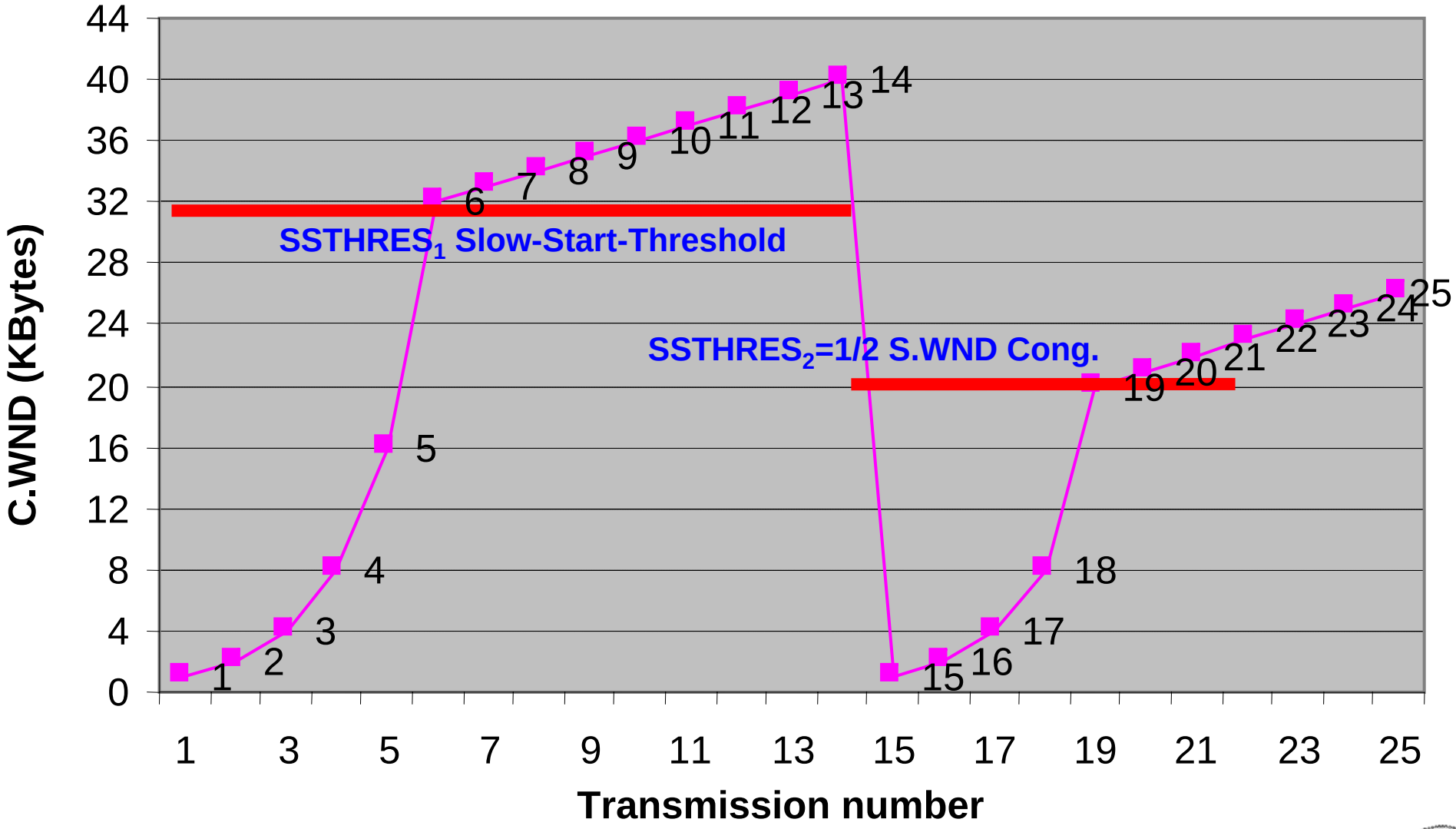
- Quando la $C.WND$ supera il valore della soglia $SSTHRES$, il trasmettitore passa nello stato di *congestion avoidance*

Congestion avoidance

- ***C.WND*** cresce linearmente
  - La ***C.WND*** aumenta di ***1*MSS*** per ogni intera finestra correttamente ricevuta (entro il timeout)
  - ***C.WND*** = ***x*MSS*** , ***(x+1)*MSS, (x+2)*MSS, (x+3)*MSS, (x+4)*MSS, . . .***

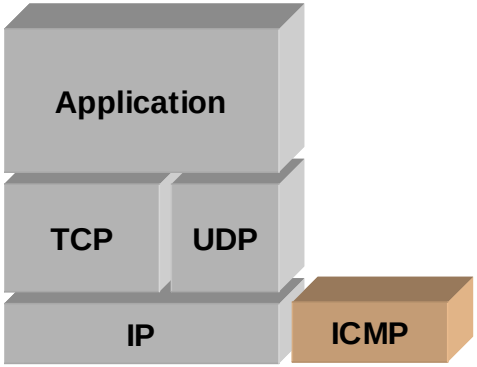
- Alla scadenza di un timeout di ritrasmissione:
  - ***SSTHRES*** è ridotta alla metà dell'attuale ***S.WND*** ***(SSTHRES=S.WND$_{current}$/2)***
  - ***C.WND =1*MSS***
  - *Il trasmettitore torna allo stato di slow start*

Controllo di Congestione: esempio



Altri protocolli: ICMP

- *Internet Control Message Protocol*
- Notifica errori all'host sorgente
- Fornisce informazioni su eventi inattesi
- È incapsulato in datagrammi IP
- Definisce 9 tipologie di messaggi
- Strumento di network management



Altri protocolli: PPP

- *Point-to-Point Protocol*
- Fornisce una modalità standard per il trasporto multiprotocollo su link punto-punto
  - Gestisce link sincroni o asincroni
  - Generalmente usato per il trasporto di IP su link
- Tre componenti fondamentali:
  - formato di frame basato su HDLC
  - LCP (*Link Control Protocol*)
    - Configura ed effettua test sulla connessione data-link
  - NCP (*Network Control Protocols*)
    - Configura i protocolli supportati (esempio, configurazione dell'indirizzo IP nelle chiamate dial-up)

Altri protocolli: DHCP

- *Dynamic Host Configuration Protocol*
- Estensione del protocollo *BOOTP*
- Assegna indirizzi IP on-demand in modo dinamico (*IP address lease*)

- DHCP consente inoltre di inviare all'host ulteriori dati di configurazione:
  - tabella di instradamento completa
  - Valori di timeout
  - Valori di default di parametri di rete (es. TTL)

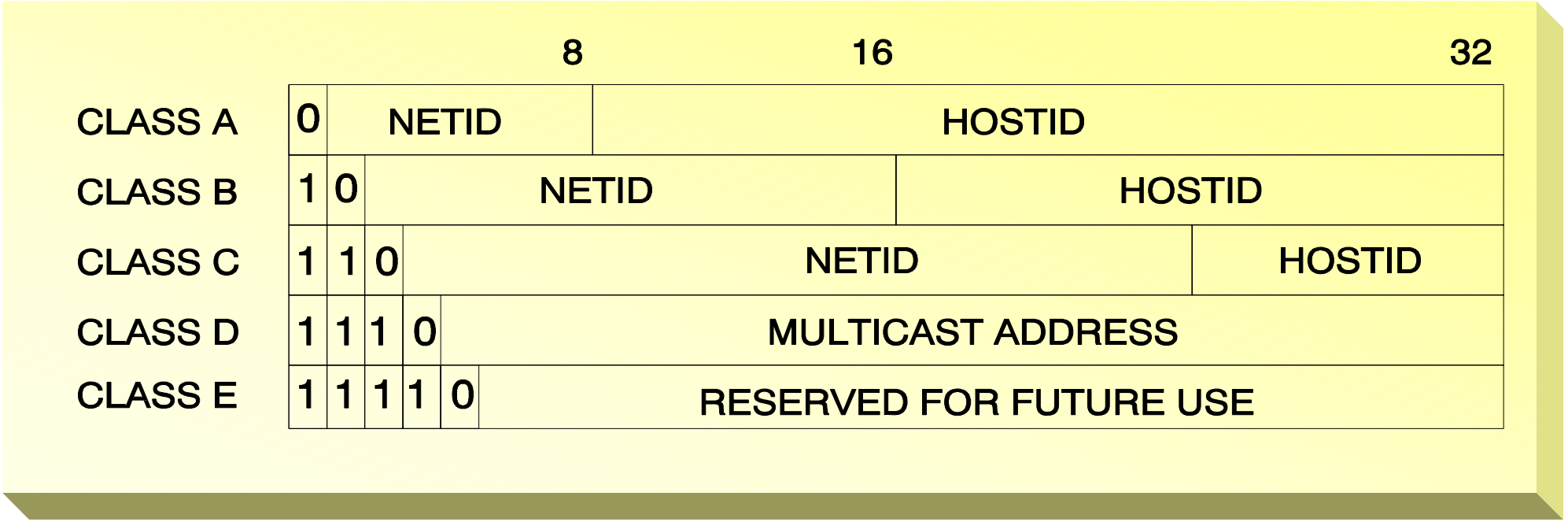
Indice

- Introduzione
- Lo stack protocollare
- Indirizzamento e instradamento
- Architettura di Internet

  - Indirizzamento IP
  - Protocolli di Routing
    - Interior Gateway Protocols
    - Exterior Gateway Protocols
  - Address Resolution
  - Tools

Classi di reti

- Originariamente furono definiti cinque classi di indirizzi
  - Tre classi di indirizzi unicast furono destinate a diverse tipologie di organizzazioni
  - Una classe per indirizzare gruppi multicast
  - Una classe per usi futuri



Spazio di indirizzamento

- Class A:
  - da 1.H.H.H a 126.H.H.H
  - 7 bit per il *net-id*: max 126 reti (0 e 127 riservati)
  - 24 bit per l'*host-id*: max 16.777.214 hosts per ogni rete
- Class B:
  - Da 128.0.H.H a 191.254.H.H
  - 14 bit per il *net-id*: max 16.382 reti
  - 16 bit per l'*host-id*: max 65.534 hosts per rete
- Class C:
  - Da 192.0.1.H a 223.255.255.H
  - 21 bit per il *net-id* : max 2.097.150 reti
  - 8 bit per l'*host-id*: max 254 hosts per rete
- Class D:
  - Da 223.0.0.0 a 239.255.255.255
  - Riservato ai gruppi multicast

Indirizzi ad uso speciale

- NETID=0, interpretato come *"THIS" network*
- NETID=127, interpretato come *loopback*
- NETID=255, interpretato come *"ALL" the networks*
- Broadcast
  - Directed broadcast: tutti gli host di una specificata rete ricevono copia del datagramma
  - Limited o local network broadcast: tutti gli host di tutte le reti ricevono copia del datagramma
- Esempi:

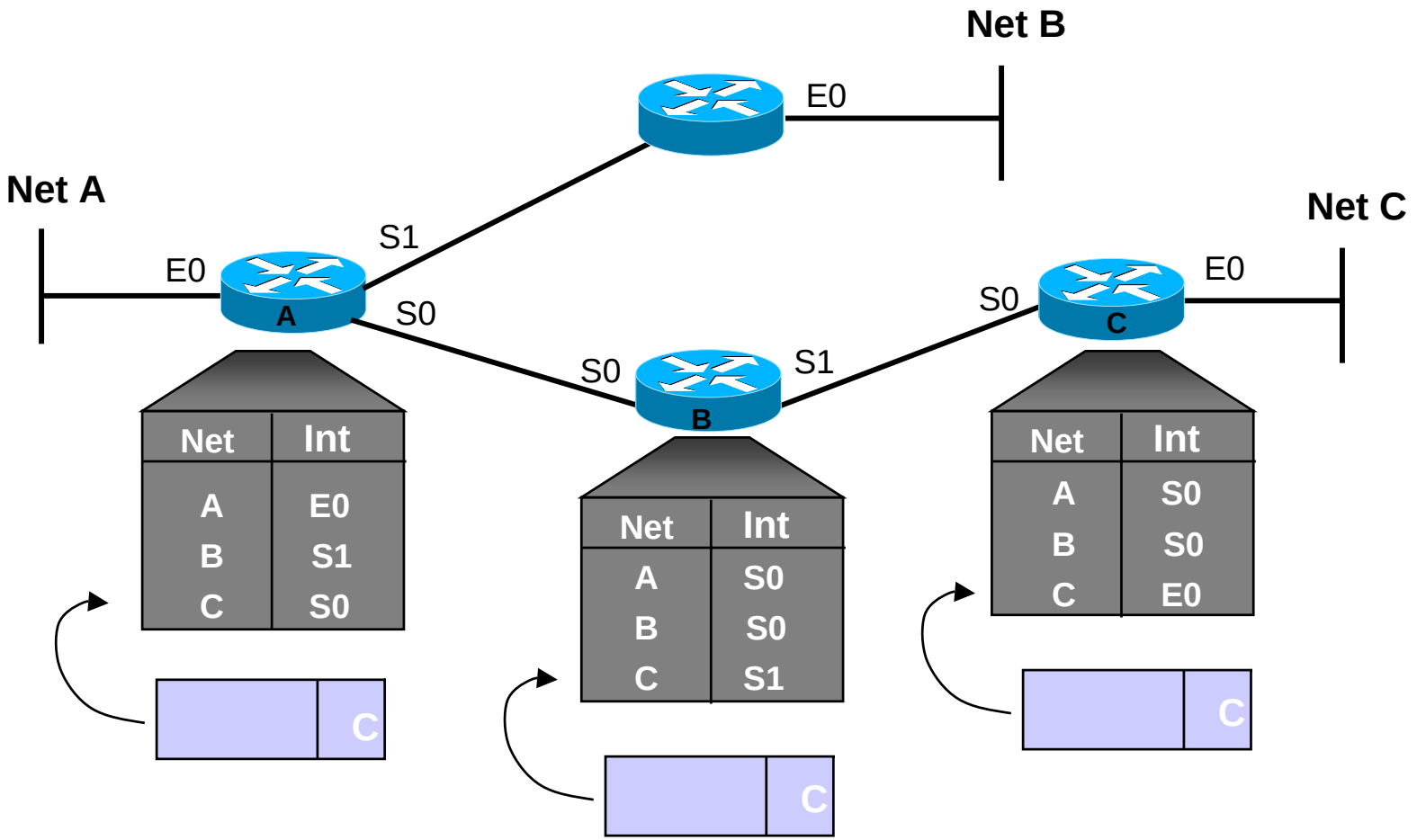
127.0.0.0	*this host*
192.168.0.0	*this network*
134.221.255.255	*broadcast* sulla rete 134.221.0.0
255.255.255.255	Limited broadcast sulle reti

Indice

- Introduzione
- Lo stack protocollare
- Indirizzamento e instradamento
- Architettura di Internet

  - Indirizzamento IP
  - Protocolli di Routing
    - Interior Gateway Protocols
    - Exterior Gateway Protocols
  - Address Resolution
  - Tools

Instradamento e attraversamento



Routing

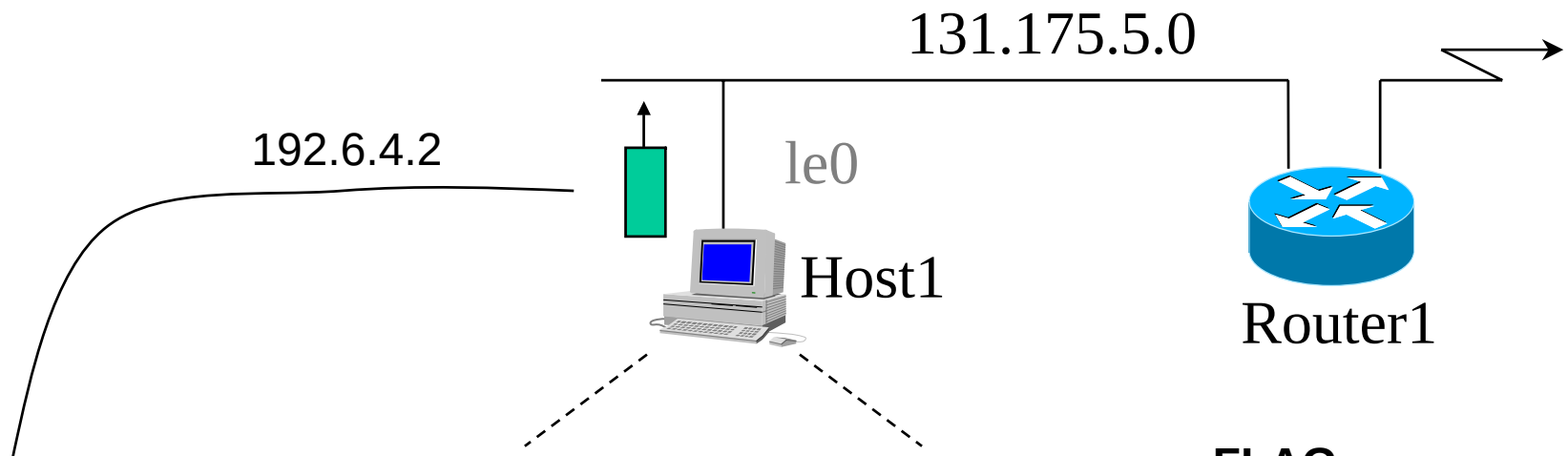
- Direct routing
  - NETID di destinazione e NETID dell'host sorgente sono uguali tra loro
  - Si estrae l'HOSTID dall'indirizzo IP
  - L'inoltro avviene mediante i meccanismi tipici del network access (es. Ethernet)

- Indirect routing
  - NETID di destinazione e NETID dell'host sorgente sono diversi
  - Sulla base dell'indirizzo IP di destinazione l'host sorgente determina quale gateway utilizzare per l'inoltro del datagramma

Tabella di routing

- Destinazione
  - IP address di uno specifico host
  - NETID di una specifica rete
  - *Default* per indicare tutti gli host o le reti non elencate singolarmente nelle altre entry
- IP address del gateway cui inoltrare il datagramma per raggiungere la destinazione
- Flag
- Metrica

Tabella di routing in un host

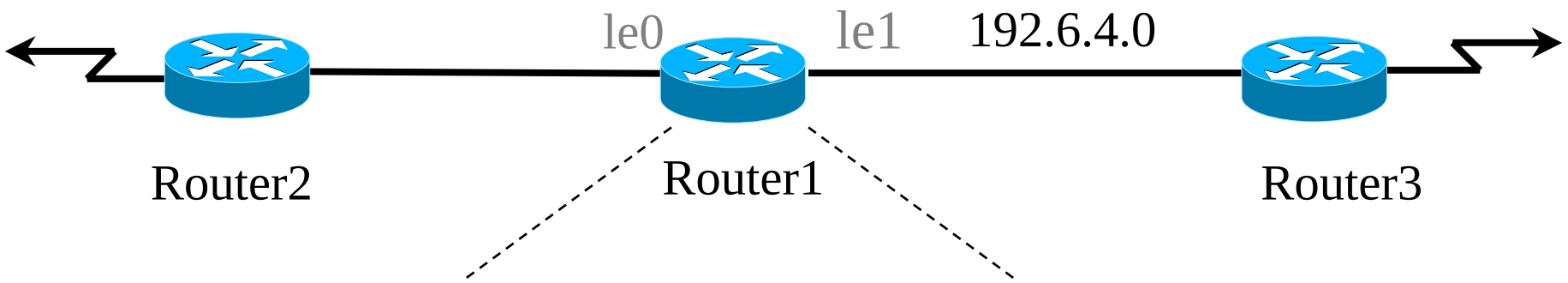


Destination	Gateway	Flags	Interface
host1	host1	UH	lo0
131.175.5.0	host1	U	le0
default	router1	UG	le0

FLAG

- **U**= link up (attivo)
- **D**= entry generata da un messaggio ICMP redirect
- **H**= host specific
- **G**= transito per un gateway

Tabella di routing in gateway



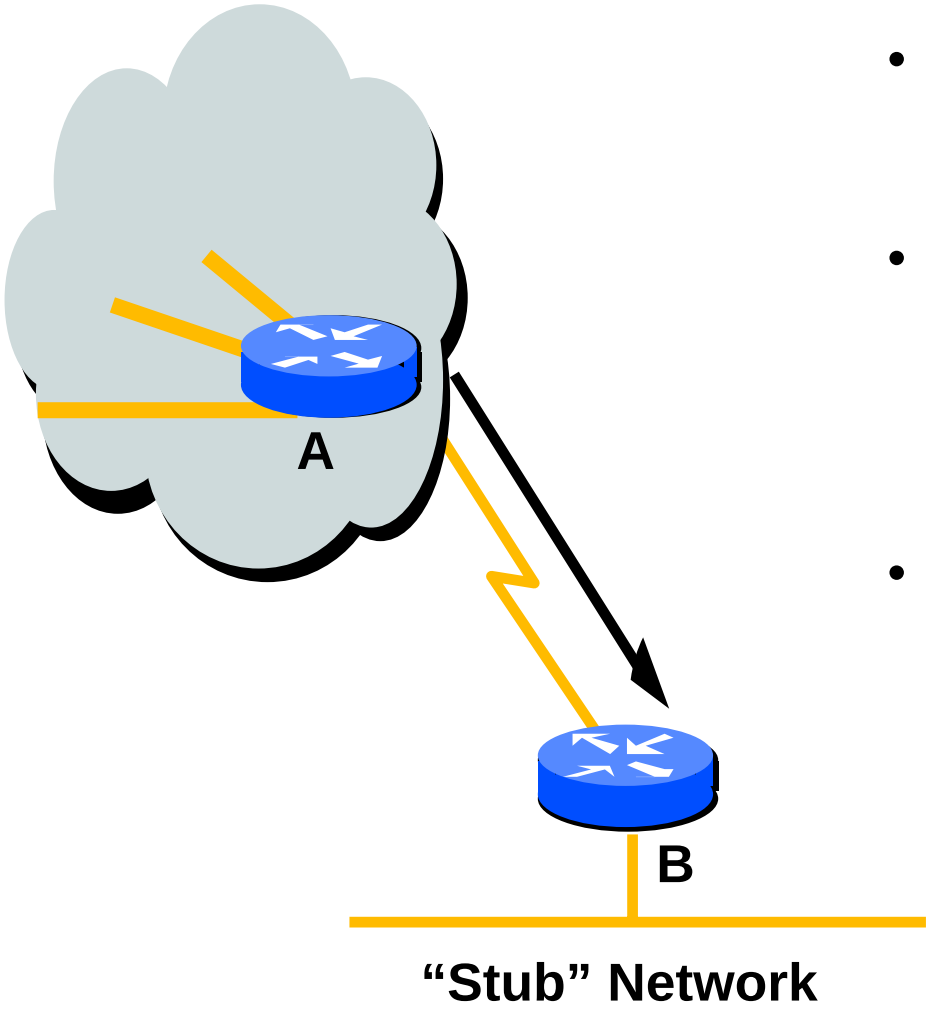
Destination	Gateway	Flags	Interface
router1	router1	UH	lo0
host2	router2	UHGD	le0
192.6.4.0	router1	U	le1
192.33.0.0	router3	UG	le1
default	router3	UG	le1

Algoritmo di inoltro del datagramma

Il nodo analizza l'indirizzo IP di destinazione dall'header del datagramma

- **`if`** <il *netid* di destinazione corrisponde a quello di qualsiasi rete direttamente connessa>
  `then` <invia il datagramma a destinazione mediante i meccanismi previsti dal livello sottostante (es. MAC Ethernet)>
- **`else if`** <trovi l'indirizzo completo della destinazione tra le entry host specific della tabella di routing>
  `then` <instrada il datagramma come specificato>
- **`else if`** <trovi il *netid* di destinazione tra le entry della tabella di routing>
  `then` <instrada il datagramma come specificato>
- **`else if`** <è indicata una *default route* nella tabella di routing>
  `then` <instrada il datagramma al *default gateway*>

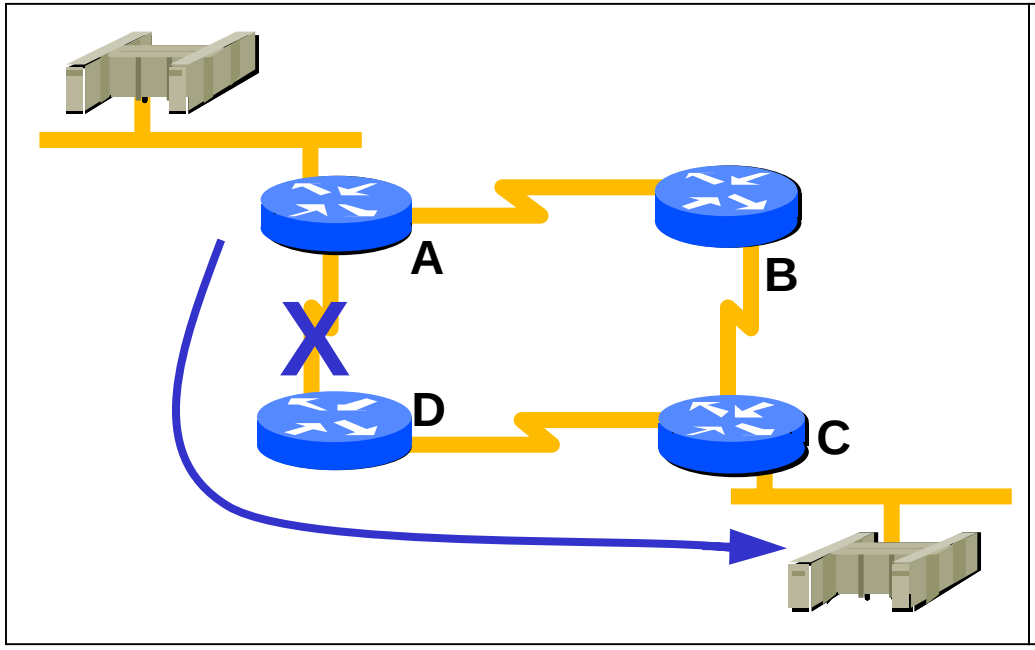
Routing statico



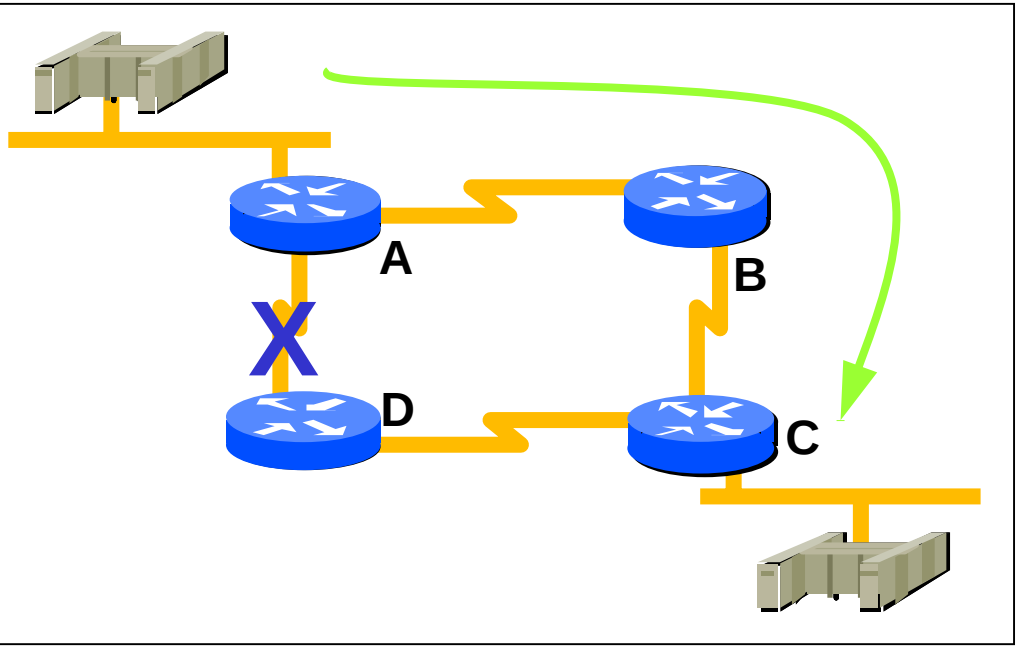
- Tabella di routing aggiornata manualmente dall’amministratore della rete
- Vantaggi
  - Adattabile alla volontà dell’amministratore di rete
  - Non coinvolto nel ciclo di aggiornamenti con altri routing
  - Elimina l’overhead del routing dinamico
- Stub network
  - Quando un nodo è raggiungibile mediante una sola route, il routing statico è sufficiente
  - Situazione tipica dei circuiti punto-punto o dei link a commutazione di circuito

Routing dinamico

Tipicamente l'internetworking richiede un routing dinamico



Una rete sperimenta un guasto sul percorso predefinito...



…e dinamicamente viene stabilito un instradamento alternativo.

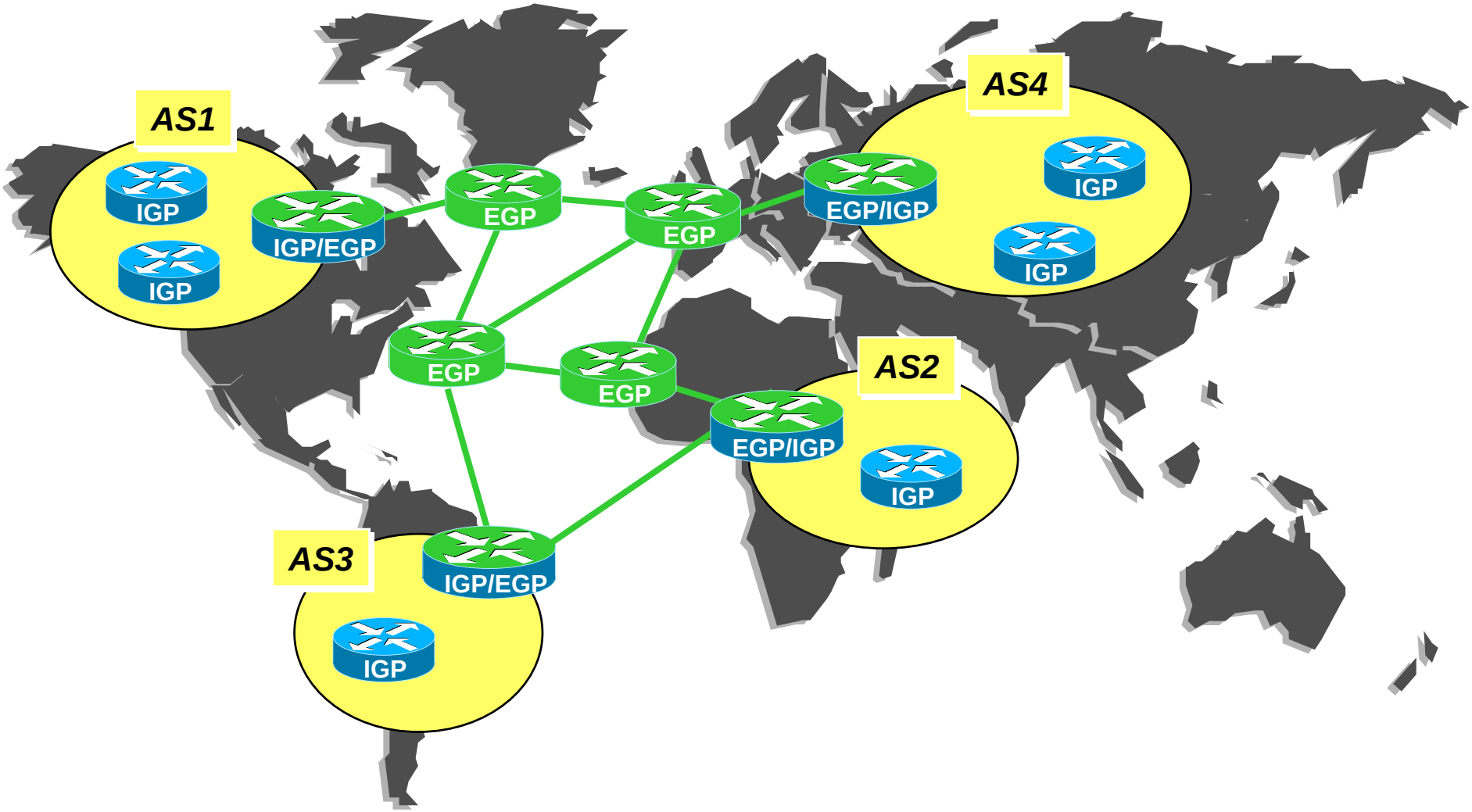
Protocolli di Routing

- Un protocollo di routing definisce:
  - L'algoritmo di routing, implementato per il calcolo del percorso
  - Le metriche ed i pesi da utilizzare
  - La dimensione, i contenuti, la frequenza e la modalità di scambio di informazioni tra nodi interessati
- Algoritmi di routing:
  - Link state vs distance vector
- Metriche
  - Hop count: lunghezza del percorso in termini di gateway attraversati
  - Reliability: dipendente dal bit error rate dei link attraversati
  - Delay: dipende da banda, lunghezza delle code, congestione e distanza fisica
  - Communication cost: costo del link
  - Banda e carico sul link

Strategia di routing in Internet

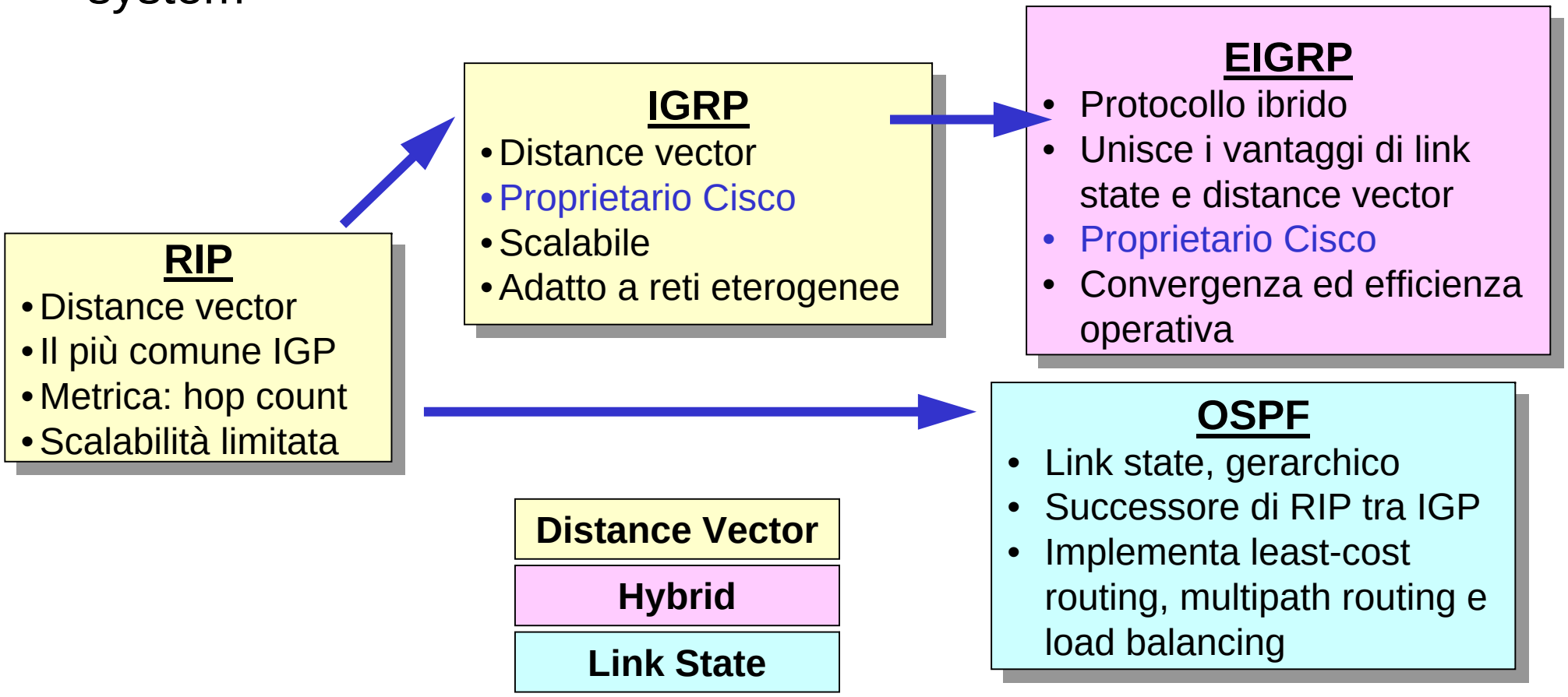
- Routing gerarchico su due livelli
  - Si basa sul concetto di Autonomous System
- Due tipi di gateway:
  - interior gateway
  - exterior gateway
- Almeno un interior gateway nella rete deve supportare:
  - Un Interior Gateway Protocol (IGP), per mantenere informazioni di routing per il proprio Autonomous System
  - Un Exterior Gateway Protocol (EGP) che scambia informazioni di raggiungibilità con altri autonomous system

Gateway protocols



Interior Gateway Protocols

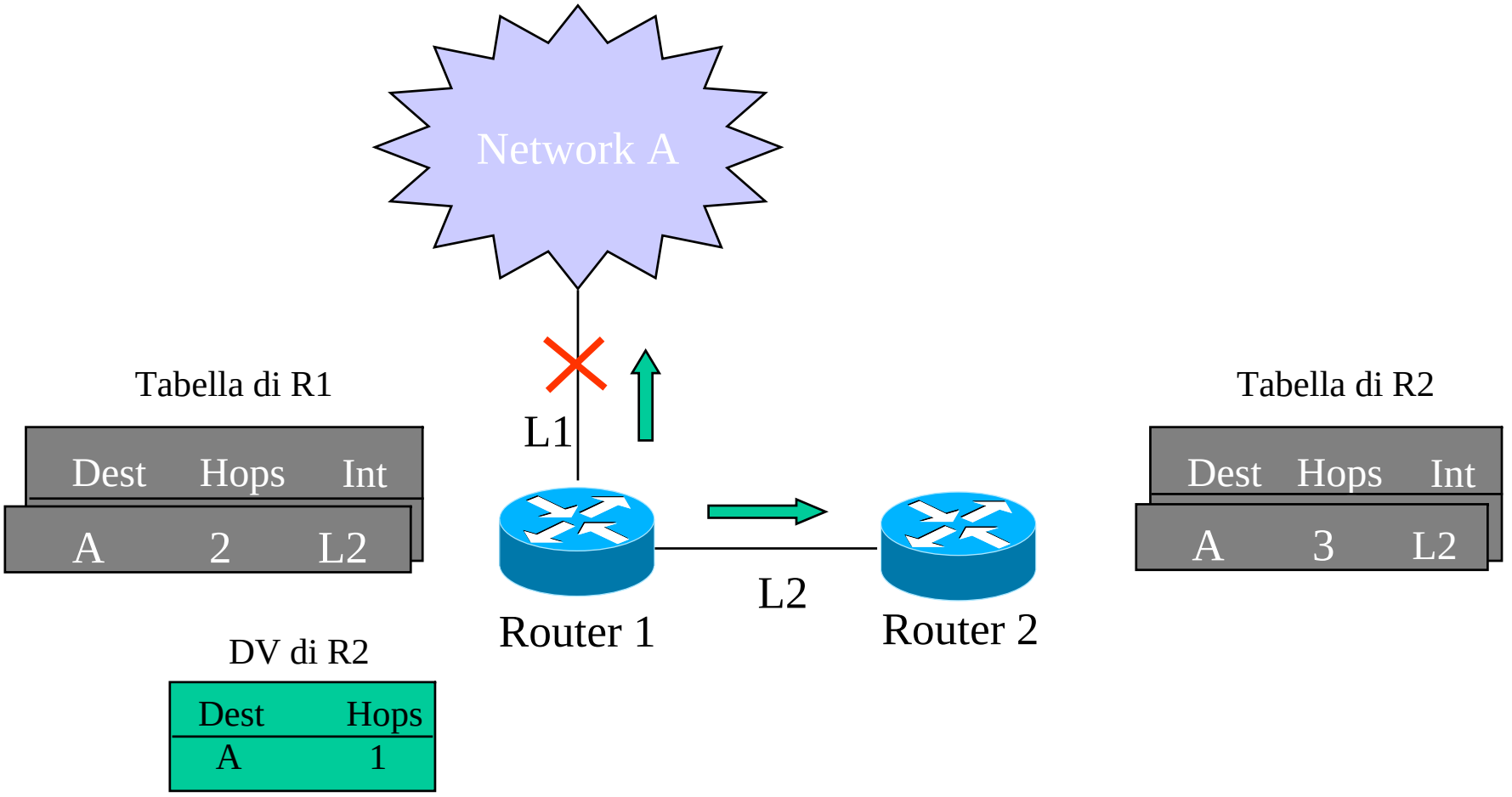
- Protocollo di routing utilizzato all'interno di un autonomous system



RIP

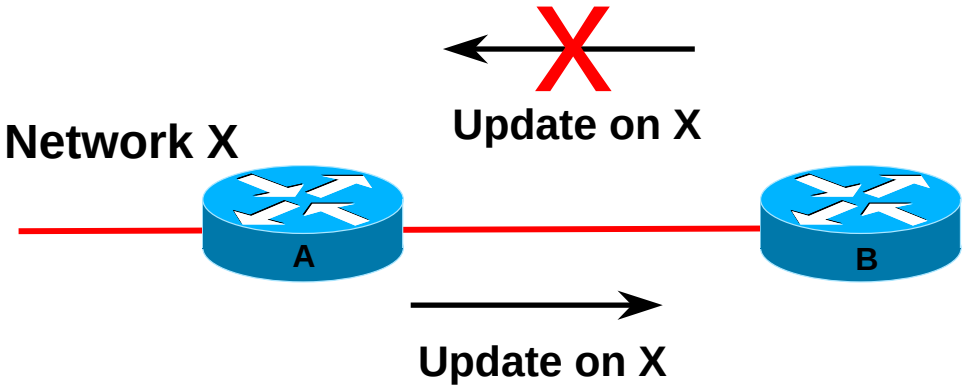
- Definito alla Berkeley (‘82)
- Algoritmo distance vector
  - Usa l’algoritmo di Bellman-Ford per il calcolo del cammino minimo
- La metrica adottata è l’hop count
  - Il limite superiore è di 15 hops (16 = irraggiungibile)
- Ogni nodo invia il proprio vettore distanza ai vicini ogni 30 secondi (utilizza la port 520 su UDP)
- Scarta informazioni di routing non confermate per lungo tempo (180 secondi)
- Limitazioni
  - Convergenza lenta
  - Calcolo di una sola route
  - Nessuna autenticazione tra nodi
  - Insensibile a variazioni di carico nella rete
  - Routing loops

Problemi di RIP: *count to infinity*



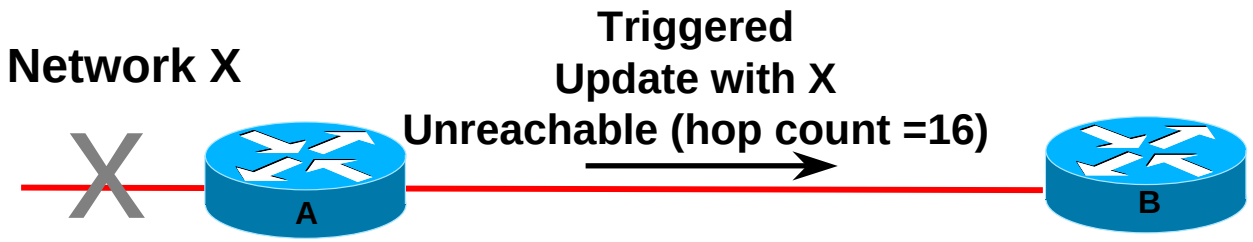
Patch per RIP (1)

- Alcune patches consentono di risolvere il problema del Count to Infinity
  - Split horizon
  - Poison reverse with triggered updates
  - Hold down
- Split horizon
  - È inutile (anzi dannoso) mandare un'informazione di routing nella direzione da cui essa proviene

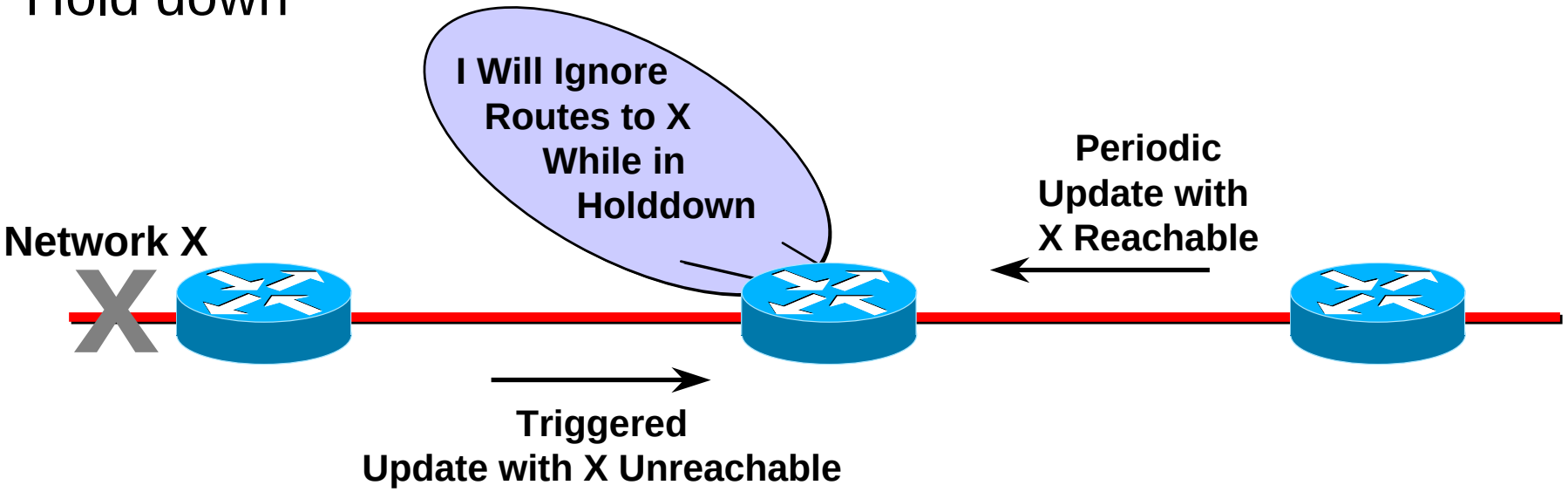


Patch per RIP (2)

- Poison reverse



- Hold down



Open Shortest Path First (OSPF)

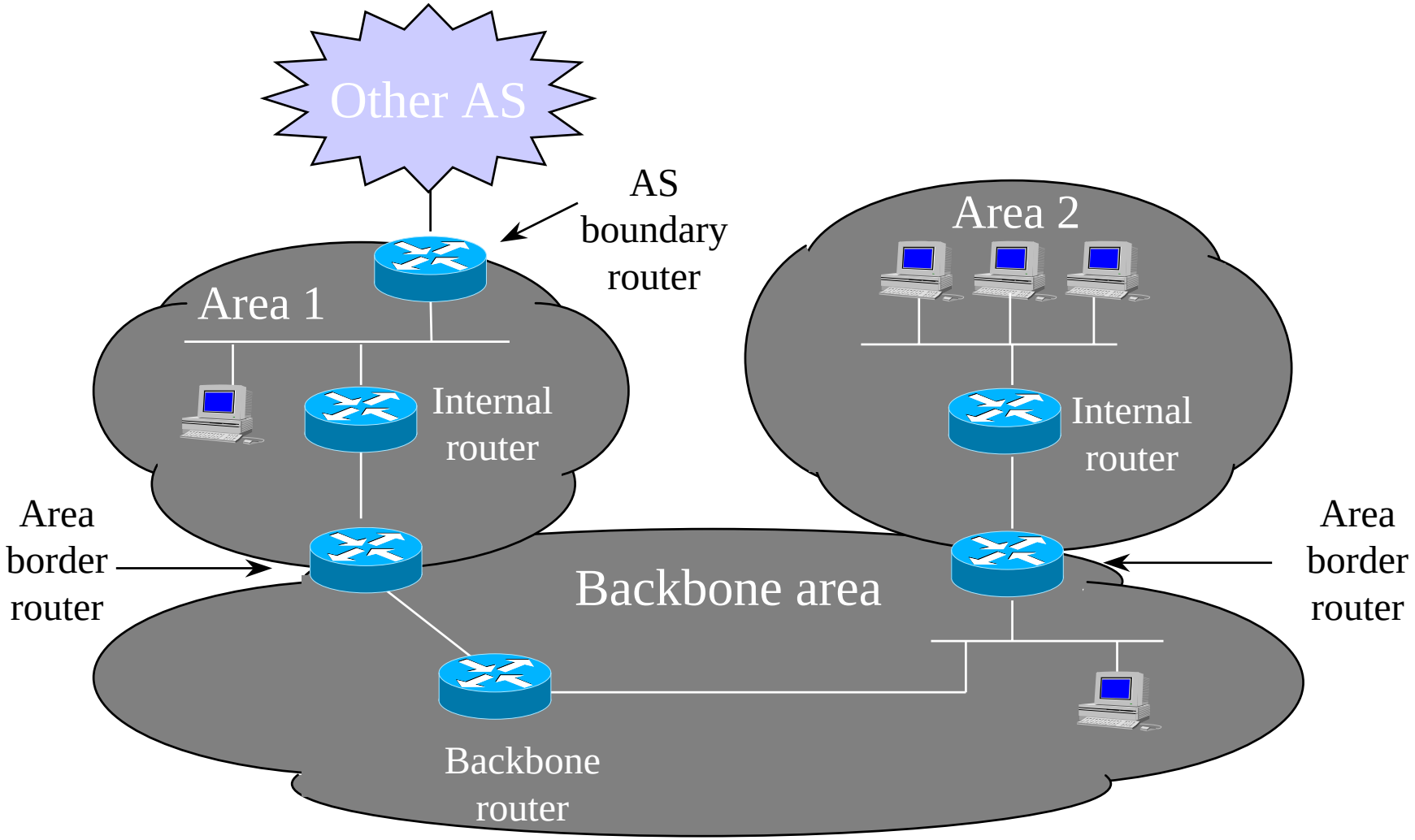
- Standard IETF (RFC 2178)

- OSPF v1 (‘88), OSPF v2 (‘94)
- Routing gerarchico
- Algoritmo link state
- Autenticazione
- Type of service (TOS) routing
- Load balancing

Routing gerarchico in OSPF: le aree

- Supporta routing gerarchico su due livelli in un singolo autonomous system
  - La rete si articola in un'area di backbone che interconnette altre aree
- I router OSPF di una stessa area si scambiano informazioni complete sulla topologia dell'area
- Nell'area di backbone, i router OSPF si scambiano informazioni complete sulla topologia del backbone e informazioni di raggiungibilità riguardo a tutte le altre aree

Routing gerarchico in OSPF: le aree



Metriche di OSPF

- OSPF calcola il costo di un percorso sommando la metrica di ciascun hop componente il percorso
- Due possibili metriche:
  - Di *default*, la metrica assegnata da OSPF è l'inverso della banda disponibile sul link, normalizzata alla banda dell'FDDI (10^8/Banda in bps)
  - In alternativa, è possibile configurare per ciascun link un *administrative cost* (compreso tra zero e 65535)

Tipi di messaggi in OSPF

- Scambio di messaggi periodici di ***hello***
- Scambio di informazioni complete sulla topologia dell'area all'avvio del router e periodicamente (***database description***)
  - Broadcast dell'intera tabella di routing ogni 30 minuti
- Inoltro asincrono di informazioni sulla variazione di stato di un link all'interno dell'area (***link state update***)
  - Ogni router dell'area invia un riscontro (**link state ack**)
- Scambio di informazioni su variazioni delle tabelle di routing tra aree
- Le informazioni sulla variazione di route vengono propagate attraverso l'area

Exterior Gateway Protocols

- Utilizzati per il routing inter-AS
- Ad ogni AS è assegnato un numero identificativo (da 1 a 65.535)
- Protocolli più utilizzati:
  - Exterior Gateway Protocol (EGP)
  - Border Gateway Protocol (BGP)

Exterior Gateway Protocol (EGP)

- EGP (‘84) fu originariamente implementato per comunicare la raggiungibilità dei router del backbone di ARPANET
- Implementa un algoritmo distance vector
- I messaggi di routing forniscono informazioni di raggiungibilità di reti
- Funzioni fondamentali di EGP
  - Neighbor acquisition
  - Neighbor reachability
  - Network reachability
- Limiti di EGP
  - Pesante scambio di messaggi di update
  - Non scalabile
  - Fornisce un unico percorso per ogni rete di destinazione
  - Non supporta il *load sharing*

Border Gateway Protocol (BGP)

- Standardizzato nel 1988 allo scopo di superare le limitazioni di EGP
- Versione corrente: BGP-4 (‘95)
- Sebbene pensato per il routing inter-AS, può essere implementato anche come IGP
- Usa un algoritmo distance vector
- BGP scambia informazioni di routing che contengono il percorso completo tra AS
- Consente di configurare il routing in accordo con le politiche definite tra i gestori delle reti interconnesse

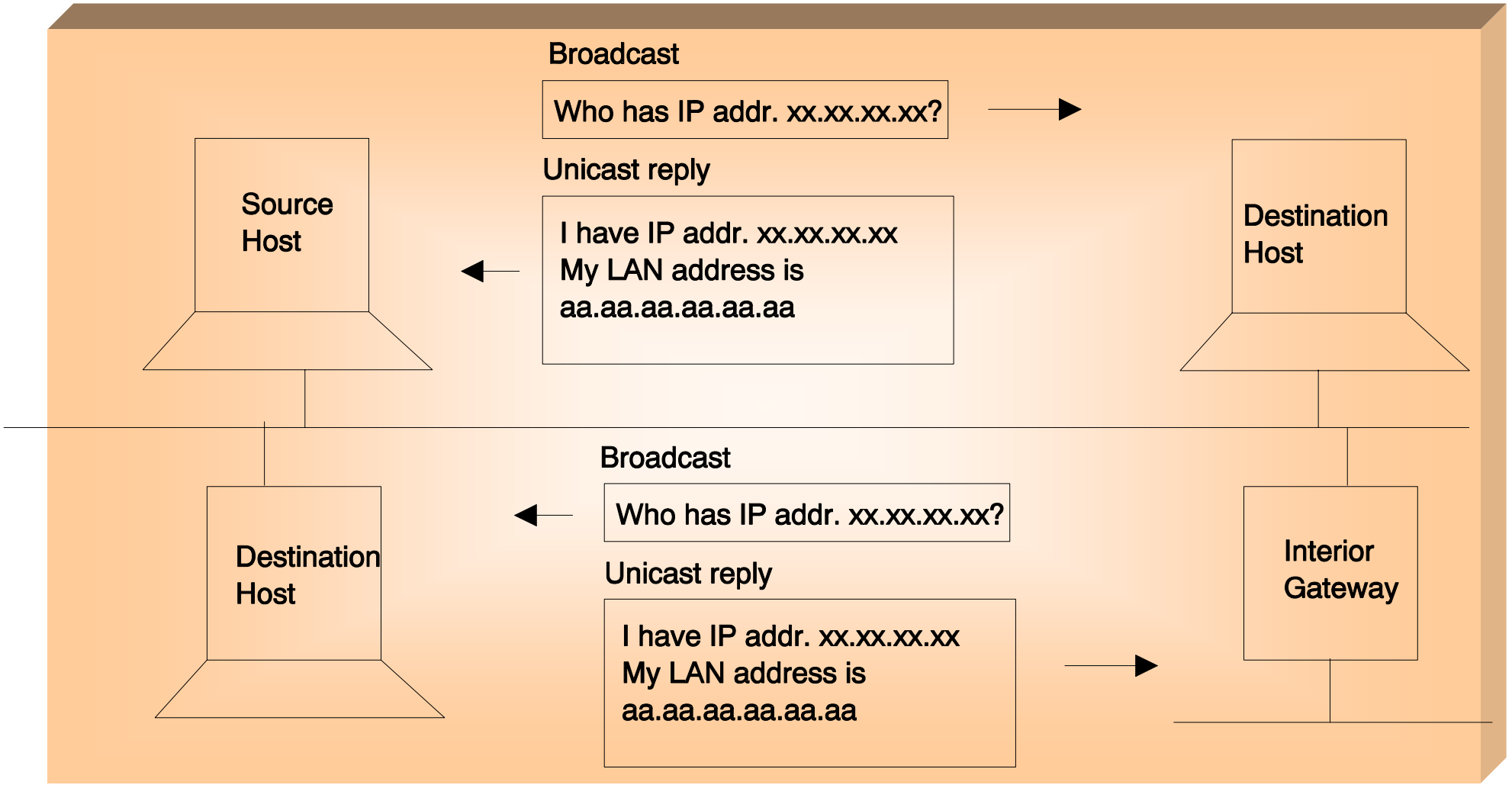
Confronto tra protocolli di routing

	Dist Vector o Link State	Interior o exterior	Metrica	Scalabilità	Convergenza	Standard
RIP	DV	interior	Hop count	15 hops	lenta	Si
IGRP	DV	interior	Hop count	255 hops	lenta	No
OSPF	LS	interior	Cost	50 routers per area, 100 areas	rapida	Si
EGP	DV-like	exterior	N/A	1000s routers	lenta	Si
BGP	DV-like	exterior	parametri configurabili	1000s routers	lenta	Si

Address resolution

- Tabelle statiche se il network access...
  - Ha un indirizzo di dimensione inferiore a 32 bit
  - È assegnato dall'amministratore della rete
  - Non è permanentemente associato all'host
- Meccanismo dinamico se il livello inferiore...
  - Ha un indirizzo di lunghezza superiore a 32 bit
  - Non è assegnato dall'amministratore della rete
  - È permanentemente associato all'host

Address Resolution Protocol nelle LAN



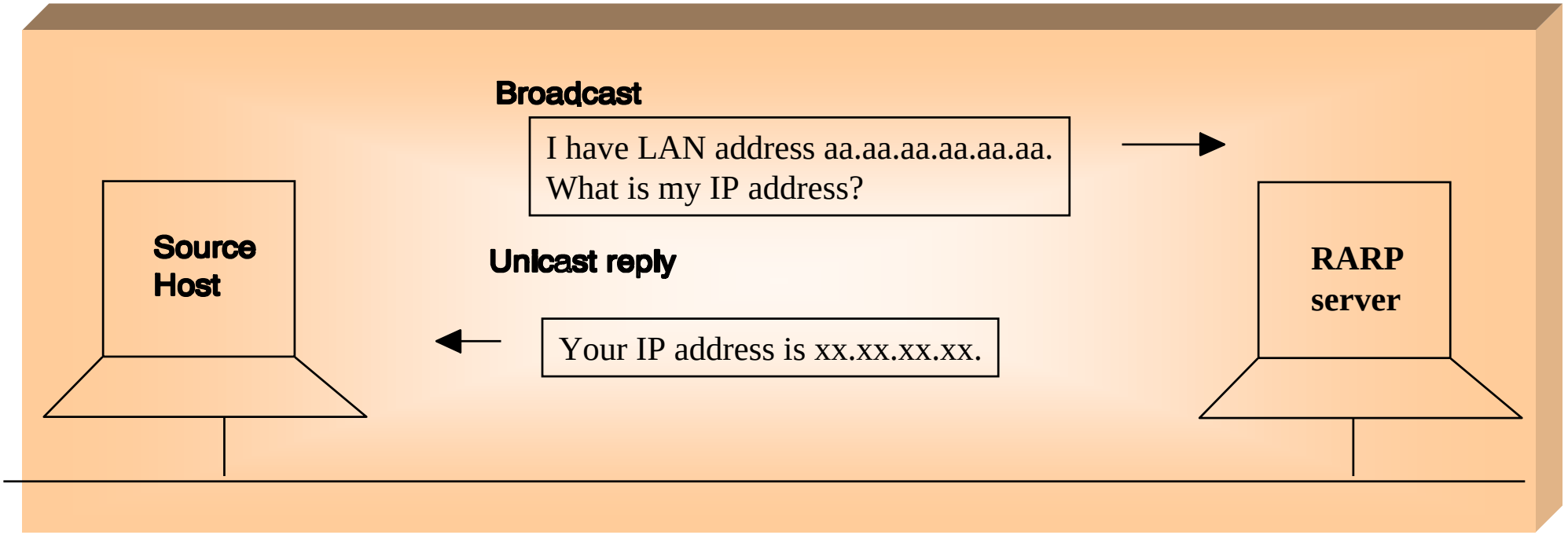
Algoritmo ARP

- Tutti gli host mantengono una cache (*ARP cache*) delle associazioni tra indirizzo IP e MAC recentemente acquisite
- Quando viene generato un datagramma destinato ad un indirizzo IP di cui non si conosce l'indirizzo MAC, l'host sorgente invia in broadcast a tutte le stazioni sulla LAN un messaggio di *ARP request* con la richiesta di binding per l'indirizzo IP di destinazione
- L'host destinazione riconosce il proprio indirizzo IP nel messaggio di *ARP request* e risponde con un *ARP reply* in modalità unicast all'host sorgente che riporta il binding indirizzo IP/indirizzo MAC

Reverse Address Resolution Protocol

- Un host può determinare il proprio indirizzo IP basandosi sull'indirizzo MAC
- Il primary RARP servers fornisce il binding precedentemente configurato dall'amministratore della LAN
- Un secondary RARP servers interviene in caso di indisponibilità del primario

RARP


Broadcast
I have LAN address aa.aa.aa.aa.aa.aa.
What is my IP address?
Source Host
Unicast reply
Your IP address is xx.xx.xx.xx.
RARP server

Tool di analisi della configurazione (1)

- **IPCONFIG (ipconfig /all)**

```
Windows NT IP Configuration
   Host Name . . . . . . . . . . :  host.company.it
   DNS Servers . . . . . . . . . :  194.20.8.1          (primary for Company domain)
                                    194.20.8.4          (secondary for Company domain)
                                    193.205.245.8       (higher DNS in it -dns2.nic.it)

   Node Type . . . . . . . . . . :  Broadcast
   NetBIOS Scope ID. . . . . . . :
   IP Routing Enabled. . . . . . :  No
   WINS Proxy Enabled. . . . . . :  No
   NetBIOS Resolution Uses DNS :    No

Ethernet adapter Elpc6561: (primary adapter)
   Description . . . . . . . . . :  FEM656B Ethernet Adapter
   Physical Address. . . . . . . :  00-50-04-92-8C-02
   DHCP Enabled. . . . . . . . . :  Yes
   IP Address. . . . . . . . . . :  192.168.0.137
   Subnet Mask . . . . . . . . . :  255.255.255.0
   Default Gateway . . . . . . . :  192.168.0.1
   DHCP Server . . . . . . . . . :  192.168.0.18
   Lease Obtained. . . . . . . . :  Thursday, September 28, 2003 9:26:1
```


Tool di analisi della configurazione (2)

La tabella di routing di un host è ottenibile mediante il comando **route print** (Windows) da una shell DOS

Network Address	Netmask	Gateway Address	Interface	Metric
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.137	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.255.0	192.168.0.137	192.168.0.137	1
192.168.0.137	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.0.255	255.255.255.255	192.168.0.137	192.168.0.137	1
224.0.0.0	224.0.0.0	192.168.0.137	192.168.0.137	1
255.255.255.255	255.255.255.255	192.168.0.137	192.168.0.137	1

Local Ethernet on LANs = 192.168.0.137
Default Gateway = 192.168.0.1

Tool di analisi della configurazione (3)

- **Netstat**
  - Displays protocol ***statistics*** and current TCP/IP network connections
  - `netstat [-a] [-e] [-n] [-s] [-p protocol] [-r] [interval]`
  - Important: `netstat -r` for the ROUTING TABLE

- **Route**
  - Manipulates network routing tables
  - `route [-f] [-p] [command [destination] [mask subnetmask]`
Command	*Purpose*
`print`	Prints a route
`add`	Adds a route
`delete`	Deletes a route
`change`	Modifies an existing route

Tool di analisi della configurazione (4)

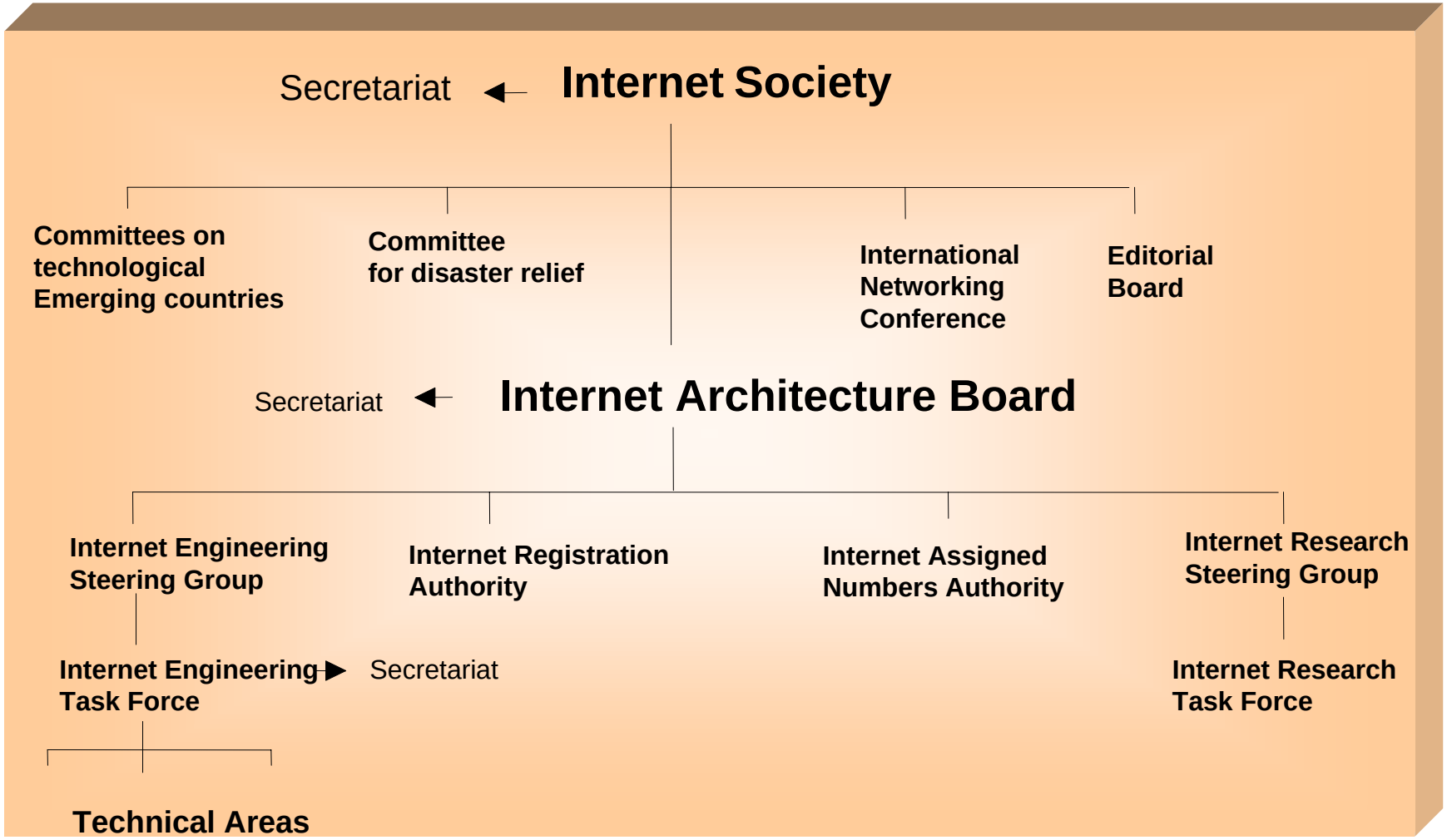
- **Tracert**
  - This diagnostic utility determines the route taken to a destination by sending Internet Control Message Protocol (ICMP) echo packets with varying Time-To-Live (TTL) values to the destination .
  - `tracert [-d] [-h maximum_hops] [-j computer-list] [-w timeout] target_name`

- **Ping**
  - verifies connections to a remote computer or computers
  - `ping [-t] [-a] [-n count] [-l length] [-f] [-i ttl] [-v tos] [-r count] [-s count] [[-j computer-list] | [-k computer-list]] [-w timeout] destination-list`

Indice

- Introduzione
- Lo stack protocollare
- Indirizzamento e instradamento
- Architettura di Internet
  - Organizzazione di Internet
  - Architettura di Internet
    - Backbone
    - NAP
    - ISP
  - Naming e DNS

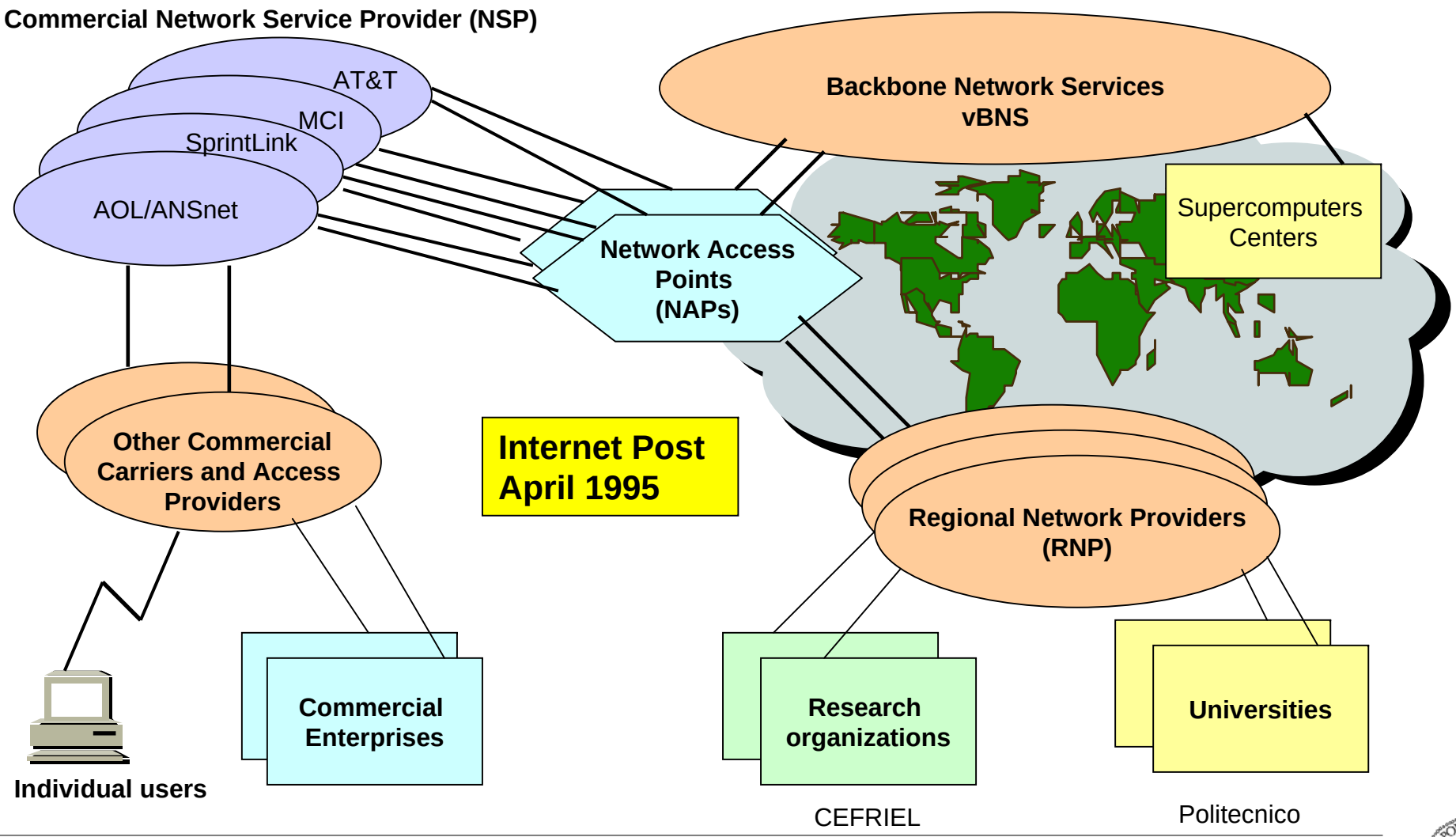
Organizzazione di Internet



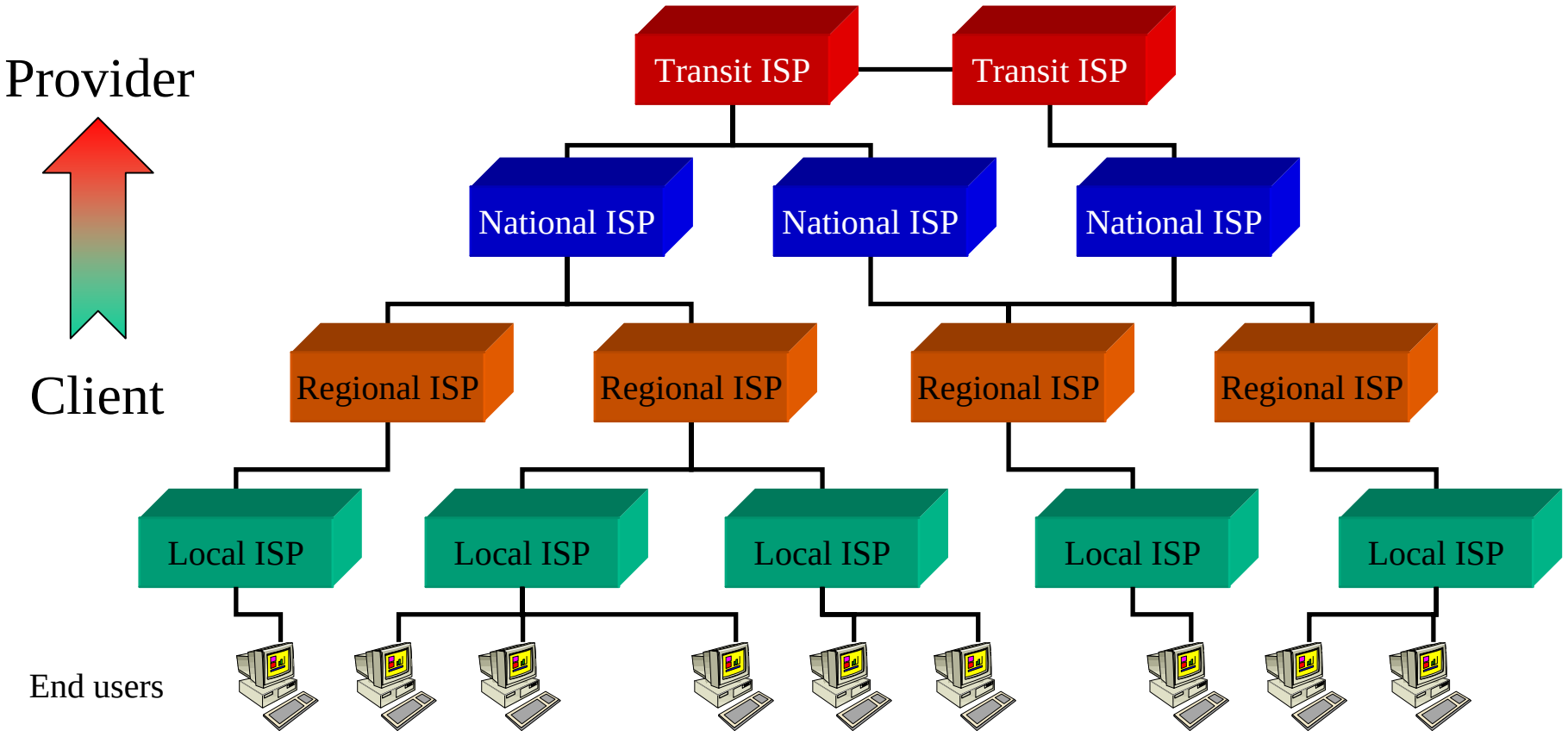
Il processo di standardizzazione

- Quattro stadi di sviluppo:
  - Basic development: conduce alla formalizzazione della specifica
  - Proposed standard
  - Draft Standard: la specifica deve rimanere allo stato di Proposed Standard e devono esserci almeno due implementazioni indipendenti capaci di interoperare
  - Internet Standard: dopo 4 ulteriori mesi di sperimentazione
    - La specifica deve essere stabile e ben compresa
    - “Technically competent”
    - Devono esserci molteplici implementazioni indipendenti ed interoperabili
    - Si è creato un consenso diffuso attorno alla specifica circa la sua utilità nell’applicazione in Internet

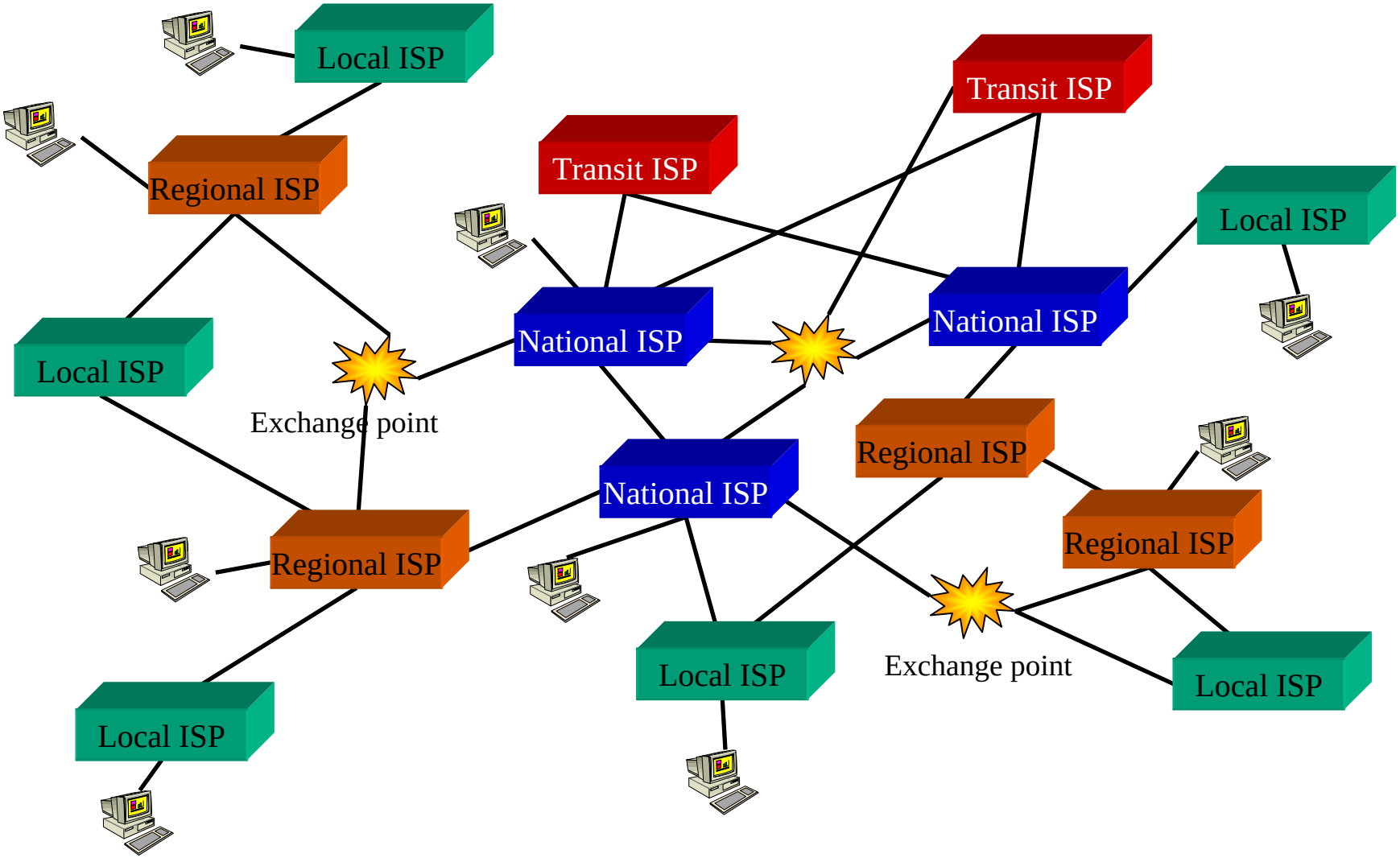
Internetworking architecture



Tier-ing in Internet



Architettura reale di Internet



I NAP italiani

- MIX (Milan Internet eXchange)
  - Situato a Milano
  - Supervisionato e gestito dall'AIIP (Associazione Italiana Internet Providers, http://www.aiip.it)
  - Include tutti i maggiori provider italiani
- NAUTILUS
  - Situato a Roma
  - Ospitato da CASPUR (Consortium for the Applications of Supercomputation for University and Research), dell'Università di Roma "La Sapienza"
  - Vi partecipano i maggiori provider

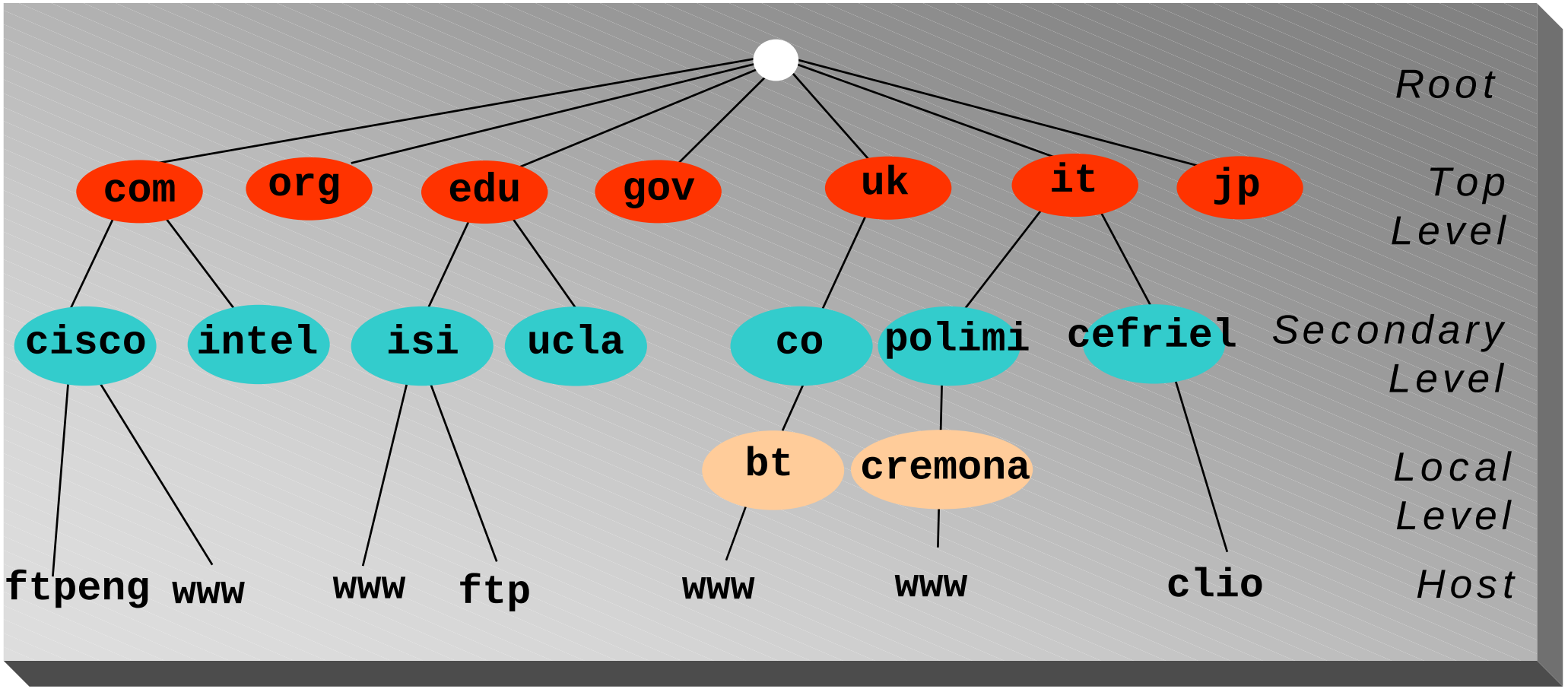
Domain Name System (DNS)

- Associa il nome simbolico di un host (nome di dominio) al suo IP address
- Struttura di naming articolata ad albero
- *Naming resolution system*:
  - *resolvers* (host programs): richiede ai *name servers* il mapping tra nome ed indirizzo di un altro host
  - *Domain name servers (DNS):* rispondomno alle richieste di risoluzione del nome di dominio
- Gerarchico
  - Viene richiesto al server di livello gerarchico superiore la risoluzione di un nome di dominio non noto
- Il DNS contiene i mapping che ricadono nel proprio dominio di competenza
- I DNS sono indipendenti e cooperanti
- I DNS conoscono il nome di almeno un root server (NS al più alto livello gerarchico)
  - Italia: DNS2.NIC.IT (http://www.nic.it)

Formato di un Domain Name

- Suddiviso in label separate da un punto
- La label più a sinistra rappresenta il dominio locale
- La label più a destra rappresenta il top-level domain
- Top Level Domain
  - Generic (gTLD): .com, .net, .edu, .gov, .mil, .org, .int
    - New: .biz, .info, .name, .pro, .museum, .aero .and .coop
  - Country Code (ccTLD): US, CA, IT, UK, JO, FR...
    - New: .eu

Domain name tree



Algoritmo di domain resolution

- L'applicazione sull'host richiede il mapping
  - Deve conoscere l'indirizzo del DNS primario (ed eventualmente di quello secondario)
  - L'indirizzo del DNS può essere ottenuto via DHCP
- Il client (resolver) accetta la richiesta, crea un messaggio di *query* e lo spedisce al DNS locale
- Il DNS locale cerca nella sua cache il mapping
- Se trova l'associazione la restituisce al resolver richiedente
- Altrimenti, può essere configurata una delle seguenti opzioni:
  - Il DNS locale richiede il mapping ad un DNS gerarchicamente superiore
  - Redirige il client ad un altro DNS