

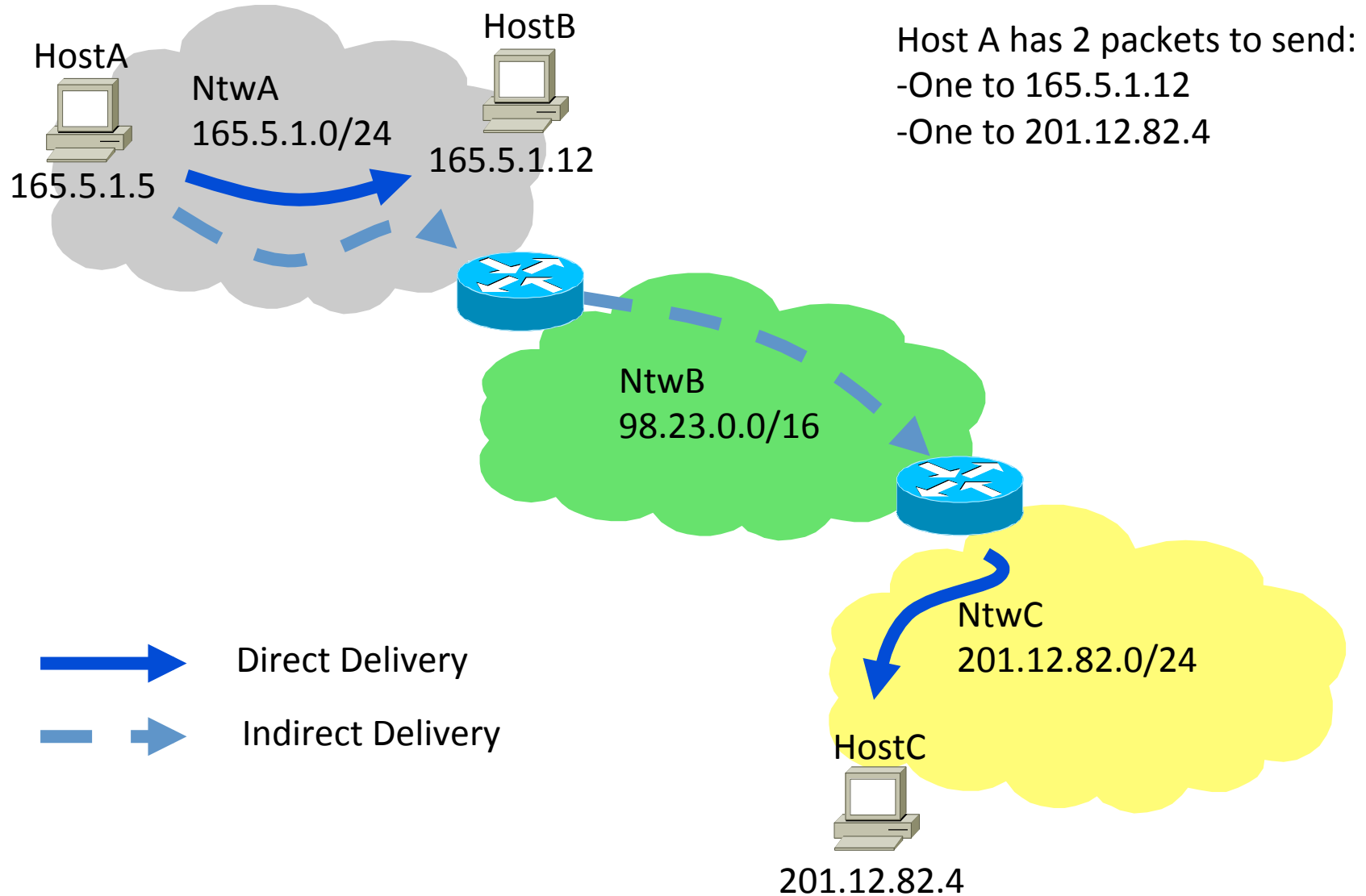
Internet Routing



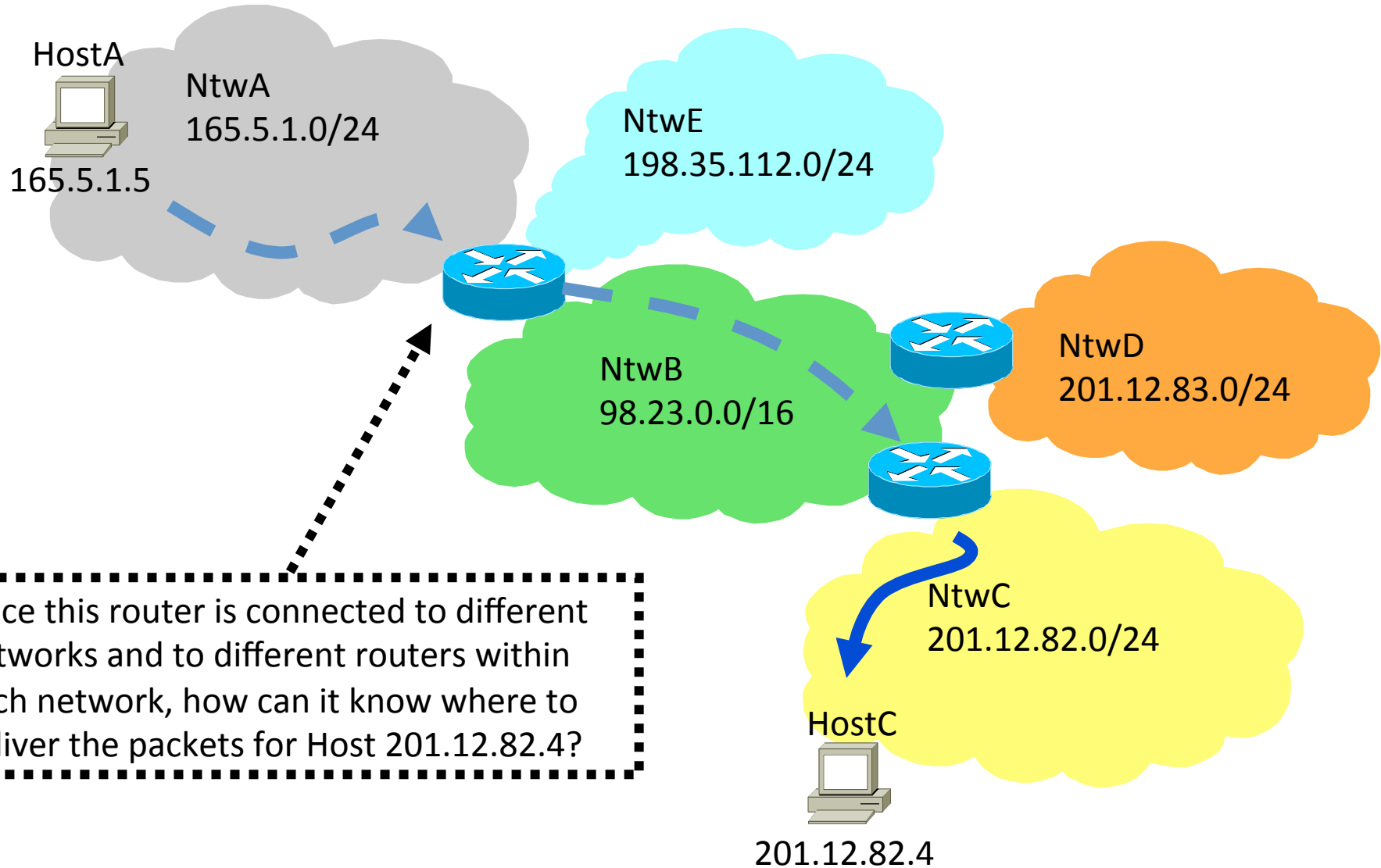
Direct / Indirect Delivery

- When a host wants to send a message to a host that belongs to the **same** network
 - ➔ direct delivery
 - The IP address belongs to the same network
 - The physical address is obtained with ARP
- When a host wants to send a message to a host that belongs to **another** network
 - ➔ indirect delivery
 - Give the message to the router that will take care of the delivery
 - The intermediate step is chosen thanks to the routing algorithms

Direct / Indirect Delivery



Direct / Indirect Delivery



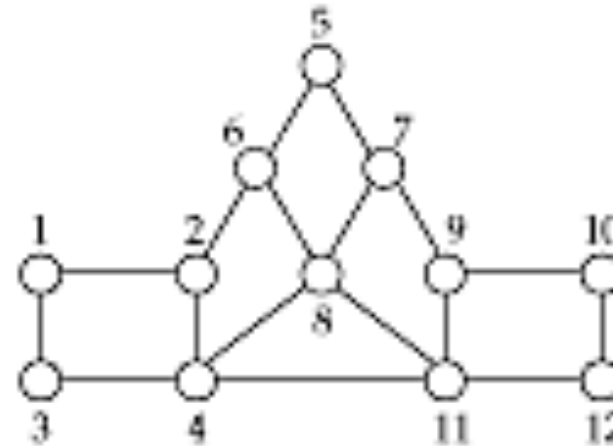
Since this router is connected to different networks and to different routers within each network, how can it know where to deliver the packets for Host 201.12.82.4?

Routing: What is it?

- Process of finding a path from a source to every destination in the network
- Suppose you want to connect to Antarctica from your desktop
 - what route should you take?
 - does a shorter route exist?
 - what if a link along the route goes down?
 - what if you are on a mobile wireless link?
- Routing deals with these types of issues

Basics

- A routing protocol sets up a **routing table** in routers
 - internal table that says, for each destination, which is the next output to take
- A node makes a local choice depending on global topology: this is a fundamental problem



ROUTING TABLE AT 1

Destination	Next hop	Destination	Next hop
1	—	7	2
2	2□	8□	2□
3	3□	9□	2□
4	3□	10□	2□
5	2□	11□	3□
6	2	12	3

Key problem

- How to make correct local decisions?
 - each router must know something about global state
- Global state
 - inherently large
 - dynamic
 - hard to collect
- A routing protocol must intelligently summarize relevant information

Requirements

- Minimize routing table space
 - fast to look up
 - less info to exchange
- Minimize number and frequency of control messages
- Robustness
- Stability: avoid
 - loops
 - oscillations
- Use optimal path

Routing Algorithm classification

Global or decentralized information?

Global:

- all routers have complete topology, link cost info
- “link state” algorithms

Decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- “distance vector” algorithms

Static or dynamic?

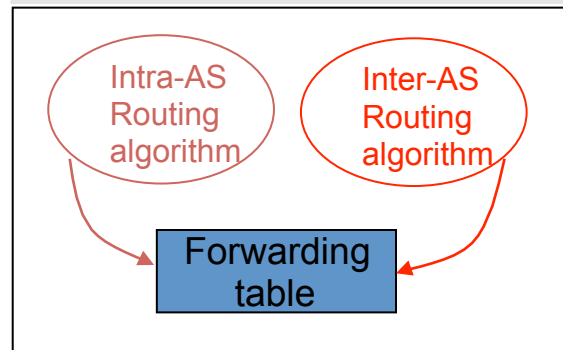
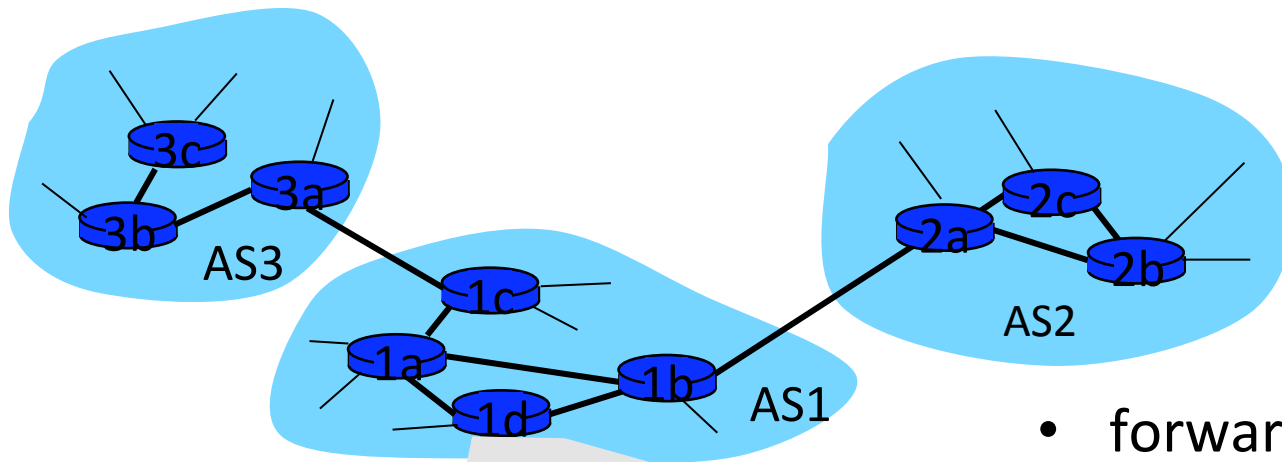
Static:

- routes change slowly over time

Dynamic:

- routes change more quickly
 - periodic update
 - in response to link cost changes

Interconnected ASes



- forwarding table configured by both intra- and inter-AS routing algorithm
 - intra-AS sets entries for internal dests
 - inter-AS & intra-As sets entries for external dests

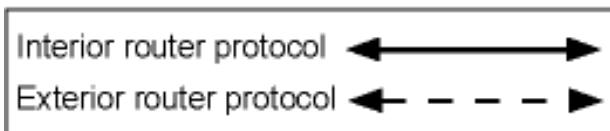
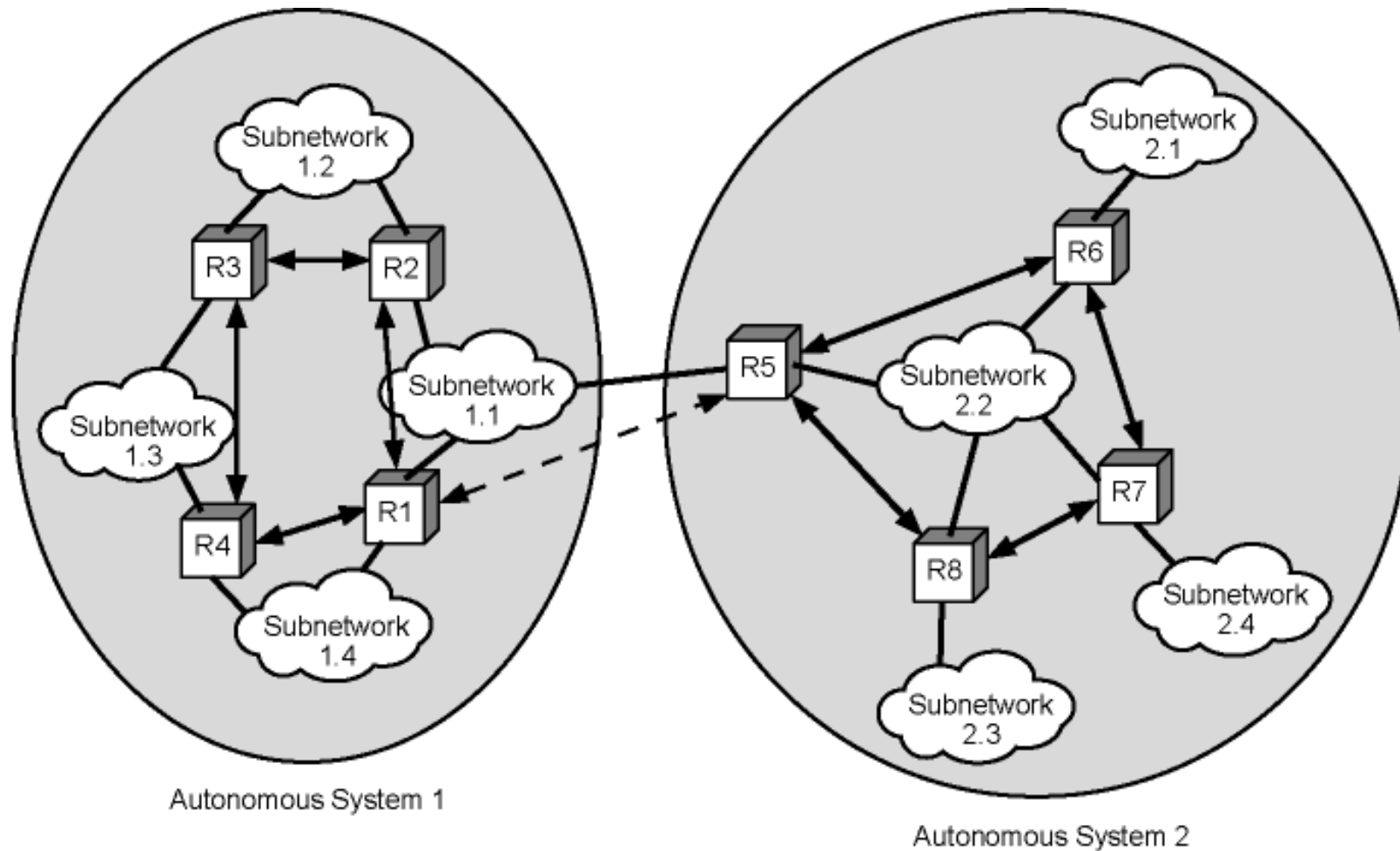
Interior Routing Protocol (IRP)

- Passes routing information between routers within AS
- Does not need to be implemented outside AS
 - Allows IRP to be tailored
- May be different algorithms and routing information in different connected AS
- Need minimum information from other connected AS
 - At least one router in each AS must talk
 - Use Exterior Routing Protocol (ERP)

Exterior Routing Protocol (ERP)

- Pass less information than IRP
- Router in first system determines route to target AS
- Routers in target AS then co-operate to deliver datagram
- ERP does not deal with details within target AS

Application of Exterior and Interior Routing Protocols



Approaches to Routing – Distance-vector

- Each node (router or host) exchange information with neighboring nodes
 - Neighbors are both directly connected to same network
- First generation routing algorithm for ARPANET
- Node maintains vector of link costs for each directly attached network and distance and next-hop vectors for each destination
- Used by Routing Information Protocol (RIP)
- Requires transmission of lots of information by each router
 - Distance vector to all neighbors
 - Contains estimated path cost to all networks in configuration
 - Changes take long time to propagate

Approaches to Routing – Link-state

- Designed to overcome drawbacks of distance-vector
- When router initialized, it determines link cost on each interface
- Advertises set of link costs to all other routers in topology
 - Not just neighboring routers
- From then on, monitor link costs
 - If significant change, router advertises new set of link costs
- Each router can construct topology of entire configuration
 - Can calculate shortest path to each destination network
- Router constructs routing table, listing first hop to each destination
- Router does not use distributed routing algorithm
 - Use any routing algorithm to determine shortest paths
 - In practice, Dijkstra's algorithm
- Open shortest path first (OSPF) protocol uses link-state routing.
- Also second generation routing algorithm for ARPANET

Link-state Routing

Each router must:

- Learn about the neighbours
 - Send HELLO packets
- Measure line cost:
 - e.g. by ECHO packets
- Build link state packets
 - Include seq. # and age
- Distribute link state packets
 - Flooding
- Compute shortest paths
 - Run Dijkstra's algorithm

Memory required for input: $k \cdot n$

(for subnet with n routers with k neighbors each)

OSPF

Open Shortest Path First



Open Shortest Path First

- Nel 1988 IETF ha avviato la standardizzazione di un nuovo protocollo di routing
- IETF ha elencato in fase di avvio della standardizzazione un insieme di requisiti che il nuovo protocollo avrebbe dovuto rispettare:
 - soluzione NON proprietaria – aperta
 - parametri di distanza multipli
 - algoritmo dinamico
 - routing basato su *Type of Service*
 - *load balancing*
 - supporto di sistemi gerarchici
 - funzionalità di sicurezza
- Open Shortest Path First (1990, RFC 1247)

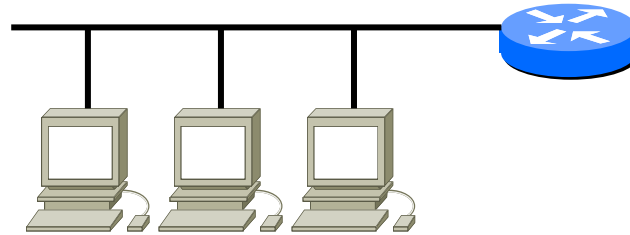
Criteri di progettazione

- I tre principali criteri di progettazione del protocollo OSPF sono:
 - distinzione tra host e router
 - reti broadcast
 - suddivisione delle reti di grandi dimensioni

Distinzione host/router (1)

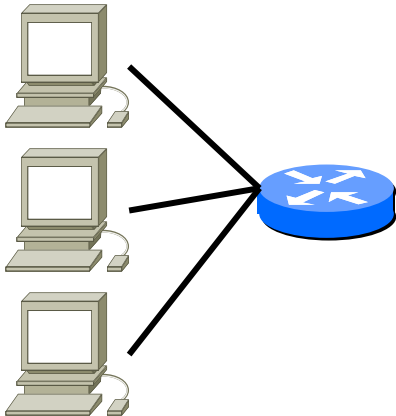
- Nelle reti IP generalmente gli host sono collocati nelle aree periferiche della rete a sottoreti locali connesse alla Big Internet attraverso router
- Il modello link state prevede che il database *link state* includa una entry per ogni link tra host e router
- OSPF introduce il concetto di link ad una *stub network*
 - il link viene identificato dall'indirizzo della sottorete

Distinzione host/router (2)

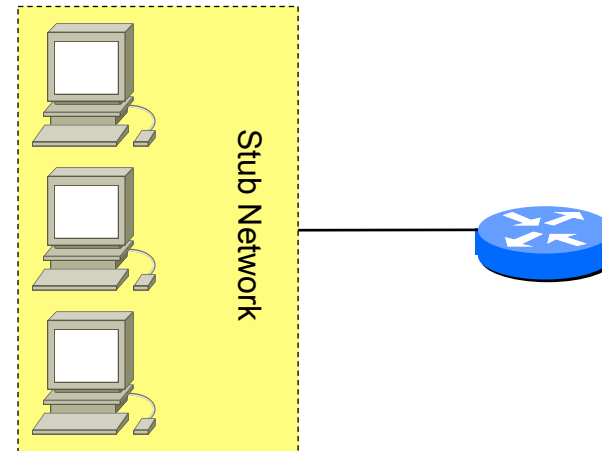


*Configurazione
fisica*

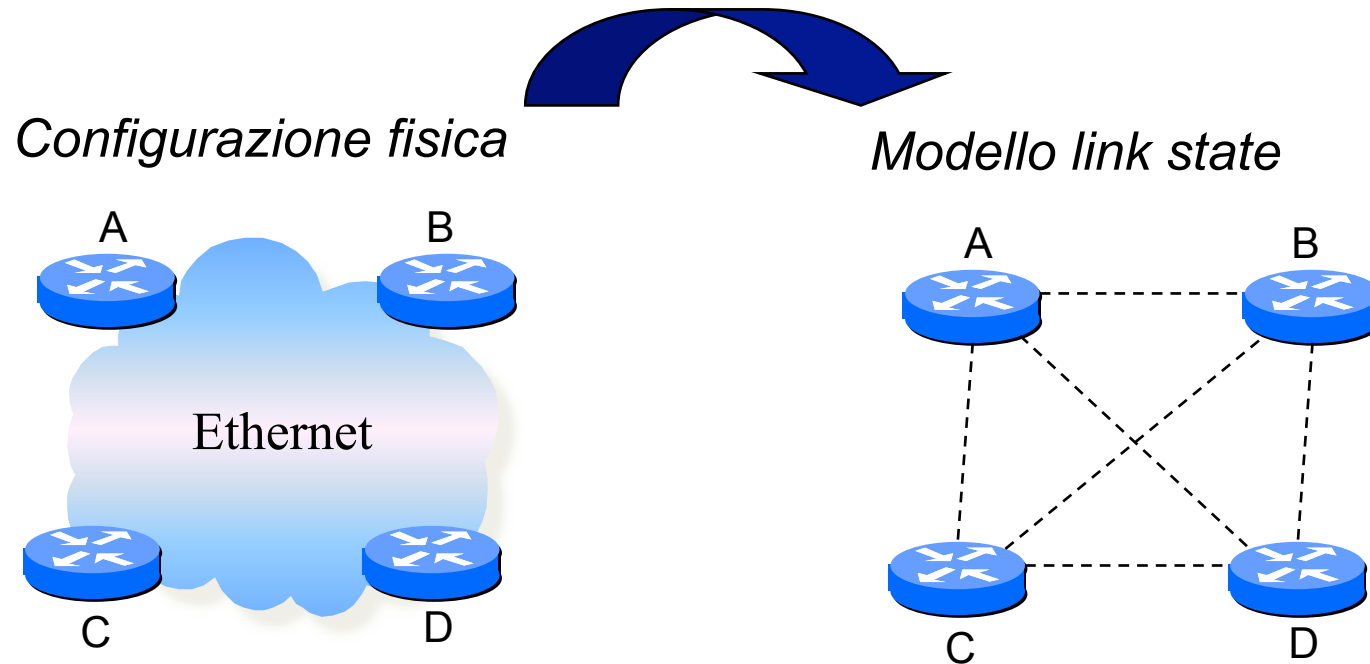
Modello link state classico



Modello OSPF



Reti broadcast (1)

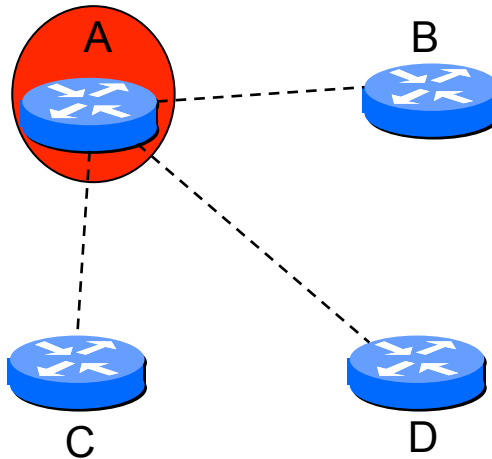


$$\frac{N \cdot (N - 1)}{2}$$

adiacenze

Reti broadcast (2)

- Elezione di un nodo della rete broadcast a *designated router*
- L'aggiornamento delle adiacenze viene fatto da tutti gli altri router solamente verso il designated router

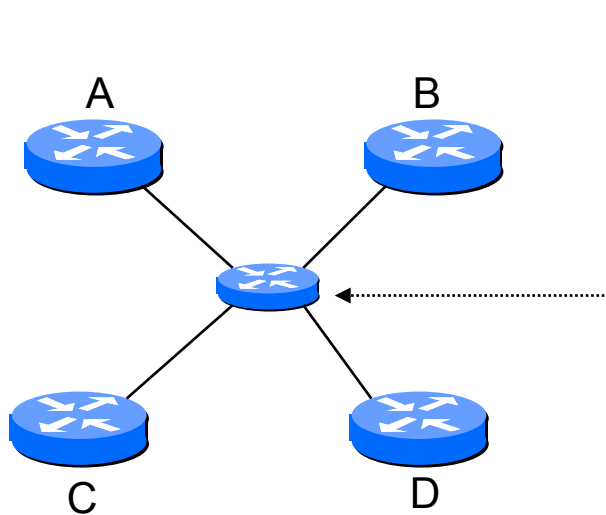


Reti broadcast (3)

- Ogni aggiornamento di link viene notificato solamente al designated router
 - indirizzo multicast “all designated router”: 224.0.0.6
- Se l’aggiornamento modifica il database link state del designated router, quest’ultimo lo propaga in flooding a tutti gli altri nodi
 - indirizzo multicast “all OSPF router”: 224.0.0.5
- Affidabilità ottenuta attraverso un *backup designated router* operante in modo “silenzioso”

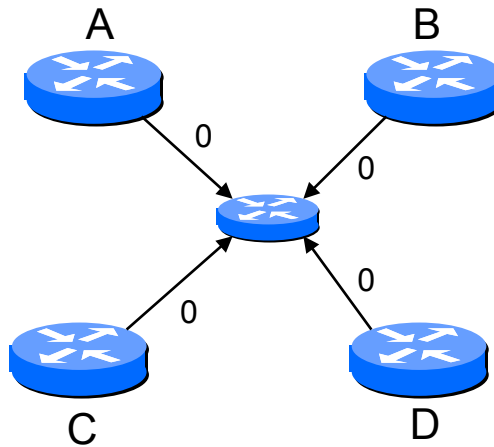
Reti broadcast (4)

- La rete viene rappresentata come un **nodo virtuale**
- Due link per ogni router
 - dal nodo virtuale al router
 - dal router al nodo virtuale (**network link**)



Reti broadcast (5)

- La distanza tra due nodi “raddoppia”
- Soluzione: i network link hanno peso nullo

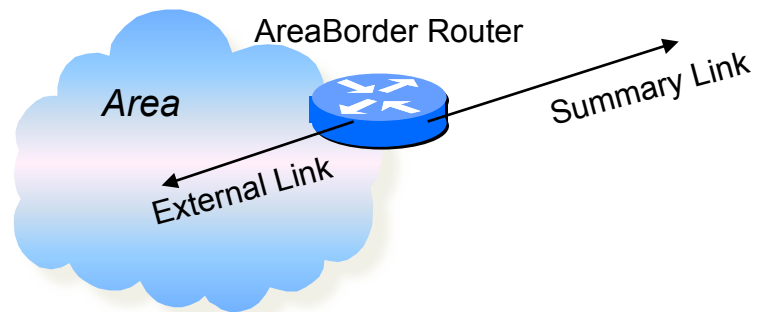


Aree multiple (1)

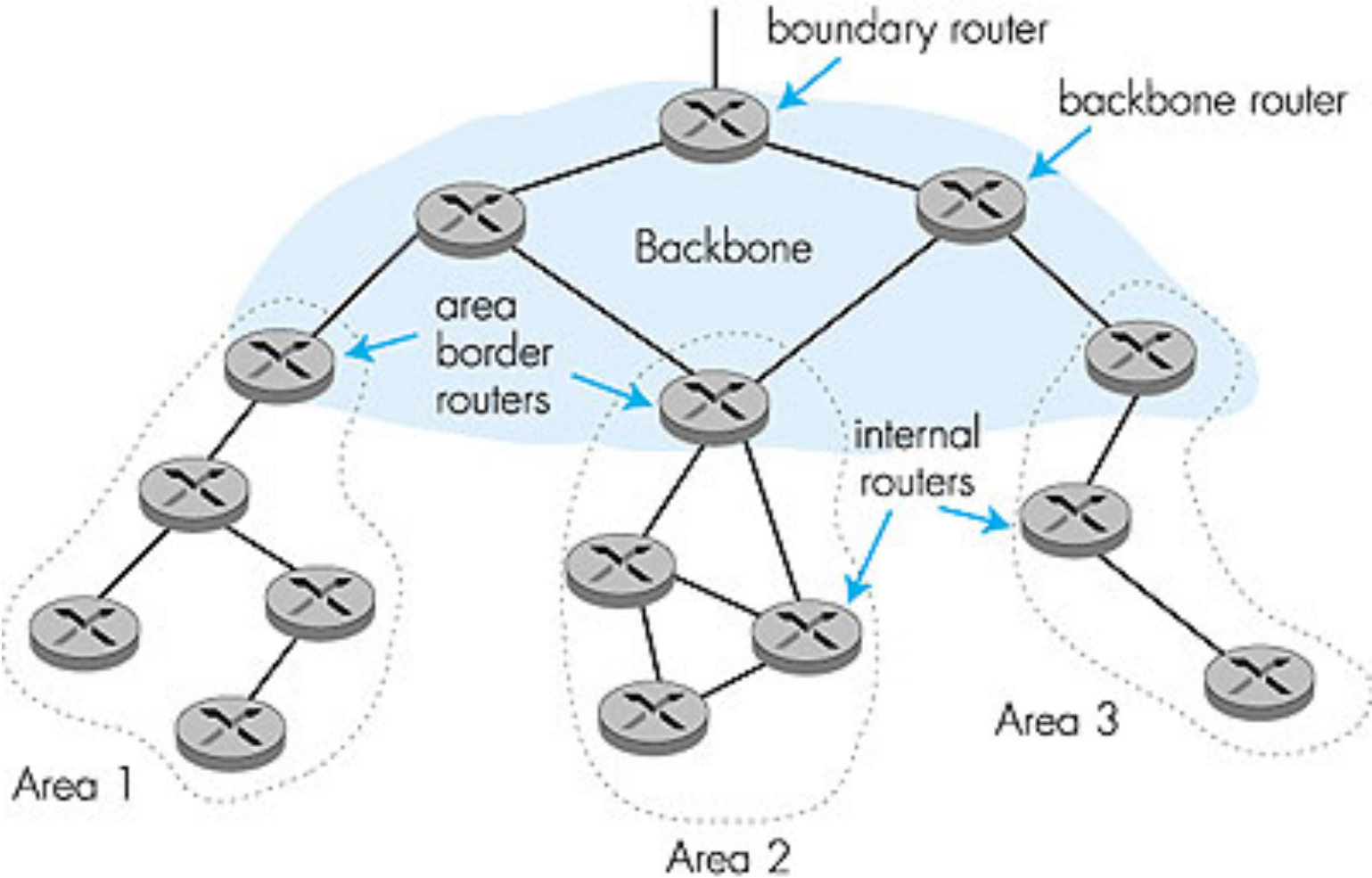
- Il numero di nodi della rete influenza direttamente:
 - dimensione del database di link state di ogni nodo
 - tempo di calcolo dei percorsi ottimi nella rete
 - quantità dei messaggi di routing distribuiti
- OSPF prevede di “spezzare” l’intera rete in un insieme di sezioni indipendenti chiamate **aree**.
- Sono locali ad ogni area
 - i record del database di link state
 - il flooding dei messaggi di routing
 - il calcolo dei percorsi ottimi di instradamento
- **Backbone Area**: area di livello gerarchico superiore

Areae multiple (2)

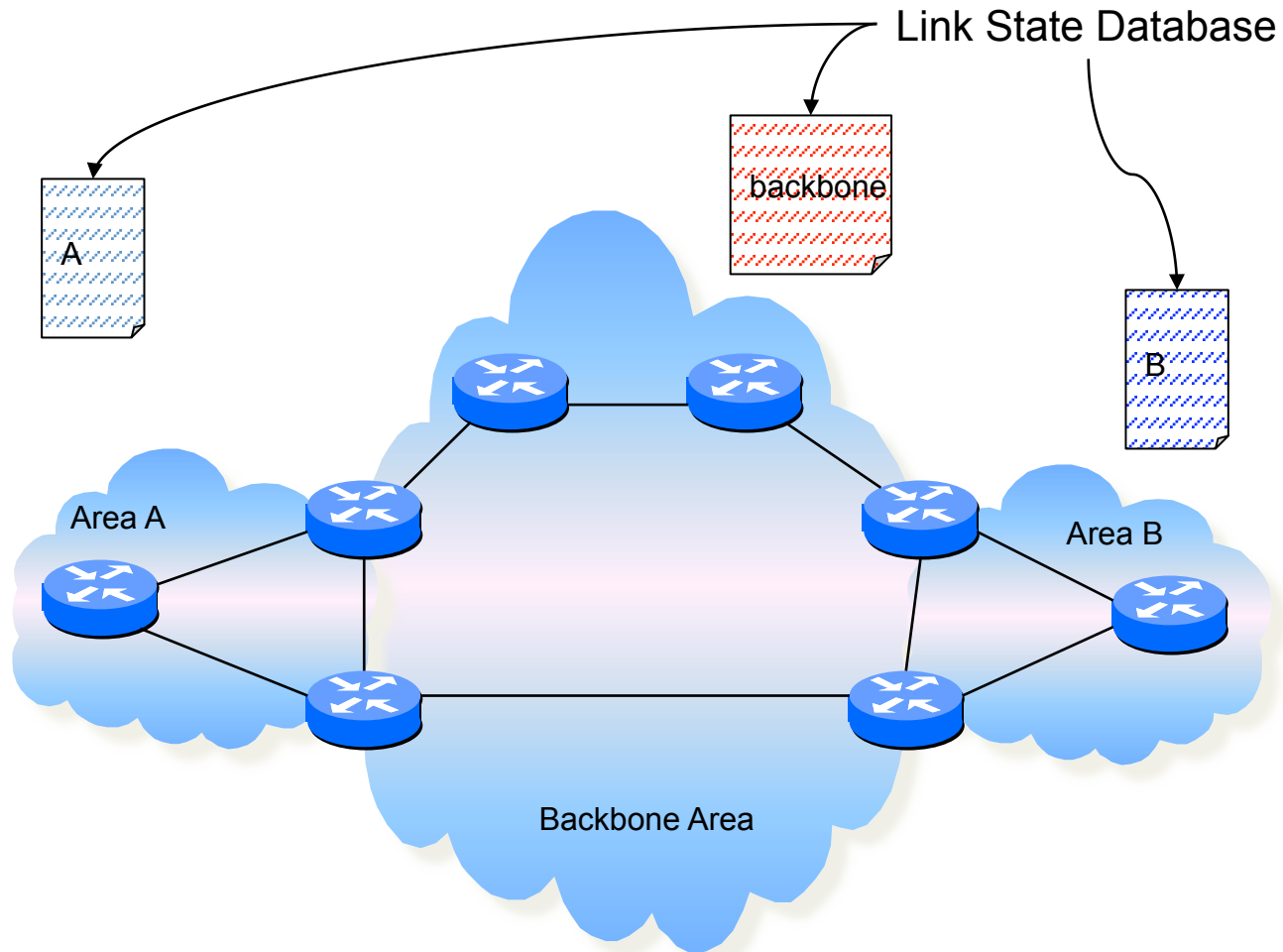
- **Area-Border Router**
 - router sono configurati come appartenenti a più aree in modo da garantire l'instradamento inter-area
- Gli Area-Border Router distribuiscono
 - external link state record
 - informano i nodi di un area relativamente ai percorsi uscenti
 - summary link state record
 - informano i nodi della backbone area dei percorsi entranti



Hierarchical OSPF

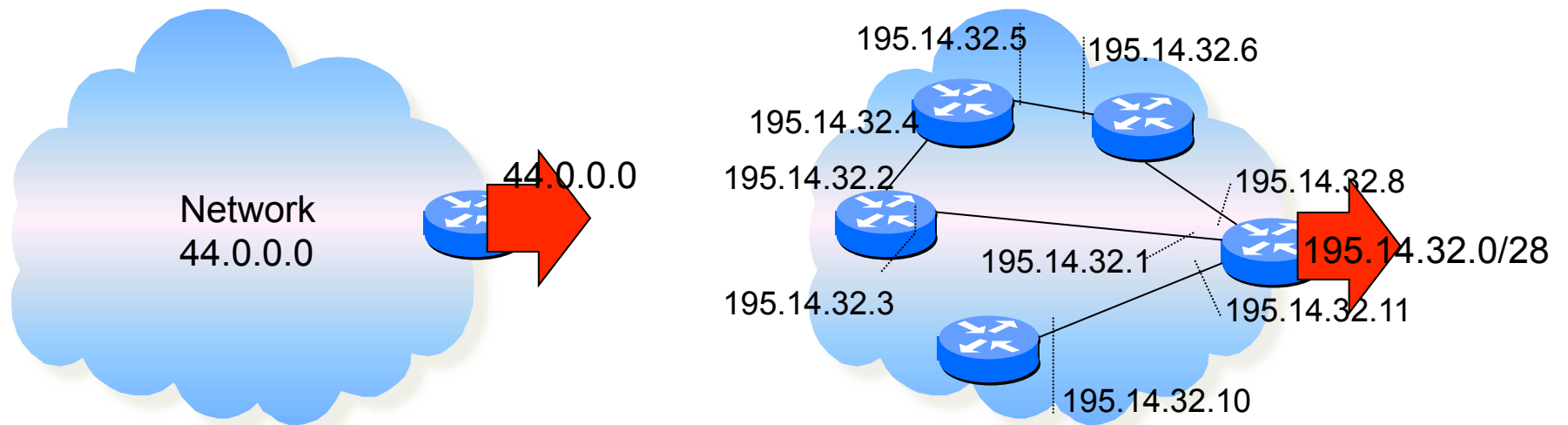


Areae multiple (3)



Are multiple (4)

- Ogni border router “sommарizza” le informazioni di instradamento relative alla propria area



Type of Service

- Per ogni link nel database link state possono essere memorizzate più metriche
 - Type of Service Metrics
- Al momento di aggiornamento delle tabelle di routing vengono distribuite tutte le metriche presenti per ogni link
- Il calcolo del percorso ottimo viene fatto
 - sempre per quanto riguarda la metrica di default (ToS 0)
 - opzionalmente per le altre metriche
- I pacchetti IP vengono quindi instradati sulla base del valore contenuto nel campo ToS del loro header

Il protocollo OSPF

- Il protocollo OSPF utilizza a sua volta 3 protocolli per svolgere le proprie funzionalità
 - Hello Protocol
 - Exchange Protocol
 - Flooding Protocol

Messaggi OSPF (1)

- I messaggi OSPF sono trasportati direttamente all'interno dei pacchetti IP
 - non viene utilizzato il livello di trasporto
- Tutti i messaggi OSPF condividono lo stesso header di 24 byte

<i>Version #</i>	<i>Type</i>	<i>Packet length</i>
<i>Router ID</i>		
<i>Area ID</i>		
<i>Checksum</i>	<i>Auth Type</i>	
<i>Authentication</i>		
<i>Authentication</i>		

Messaggi OSPF (2)

- Version # = 2
- Type: indica il tipo di messaggio
- Packet Length: numero di byte del messaggio
- Router ID: indirizzo IP del router di riferimento

<i>Version #</i>	<i>Type</i>	<i>Packet length</i>
<i>Router ID</i>		
<i>Area ID</i>		
<i>Checksum</i>	<i>Auth Type</i>	
<i>Authentication</i>		
<i>Authentication</i>		

Messaggi OSPF (3)

- Area ID: identificativo dell' area
 - 0 per la Backbone area
- Auth Type: tipo di autenticazione
 - 0 no autenticazione, 1 autenticazione con passwd
- Authentication: password

<i>Version #</i>	<i>Type</i>	<i>Packet length</i>
<i>Router ID</i>		
<i>Area ID</i>		
<i>Checksum</i>		<i>Auth Type</i>
<i>Authentication</i>		
<i>Authentication</i>		

Il protocollo Hello

- Funzioni:
 - verificare l'operatività dei link
 - elezione del *designated router* (e relativo elemento di backup)
- Messaggi:
 - Hello

<i>Common header (type = 1, hello)</i>		
<i>Network mask</i>		
<i>Hello interval</i>	<i>Options</i>	<i>Priority</i>
<i>Dead interval</i>		
<i>Designated router</i>		
<i>Backup Designated router</i>		
<i>Neighbor</i>		

Hello Protocol: formato pacchetto (1)

- Network mask: maschera della sottorete cui appartiene l'interfaccia
- Hello interval: intervallo temporale di separazione tra due messaggi di Hello

<i>Common header (type = 1, hello)</i>		
<i>Network mask</i>		
<i>Hello interval</i>	<i>Options</i>	<i>Priority</i>
<i>Dead interval</i>		
<i>Designated router</i>		
<i>Backup Designated router</i>		
<i>Neighbor</i>		

Hello Protocol: formato pacchetto (2)

- Designated router: indirizzo IP del designated router
 - 0 se non è stato ancora eletto
- Backup designated router: indirizzo IP del backup designated router

<i>Common header (type = 1, hello)</i>		
<i>Network mask</i>		
<i>Hello interval</i>	<i>Options</i>	<i>Priority</i>
<i>Dead interval</i>		
<i>Designated router</i>		
<i>Backup Designated router</i>		
<i>Neighbor</i>		

Hello Protocol: formato pacchetto (3)

- Neighbor: lista di nodi adiacenti da cui ha ricevuto un messaggio di Hello negli ultimi **dead interval** secondi

<i>Common header (type = 1, hello)</i>		
<i>Network mask</i>		
<i>Hello interval</i>	<i>Options</i>	<i>Priority</i>
<i>Dead interval</i>		
<i>Designated router</i>		
<i>Backup Designated router</i>		
<i>Neighbor</i>		

Hello protocol: procedure

- Regole di elezione del designated router
 - viene utilizzato il campo **priority** del pacchetto Hello
 - ogni router viene configurato staticamente con un valore di priority
 - [0,255]
 - viene selezionato il router con il più alto valore
 - i router con priorità 0 non possono essere eletti

Il protocollo Exchange

- Funzioni:
 - sincronizzazione dei database link state (bring up adjacencies) tra due router che hanno appena verificato l'operatività bidirezionale del link che li connette
 - protocollo client-server
 - messaggi:
 - Database Description Packets
 - Link State Request
 - Link State Update
 - N.B. il messaggio Link State Update viene distribuito secondo le politiche del protocollo di Flooding

Exchange Protocol: messaggi (1)

- Database Description

<i>Common header (type = 2, db description)</i>			
<i>0</i>	<i>0</i>	<i>Options</i>	<i>0</i>
<i>DD sequence number</i>			
<i>Link State Type</i>			
<i>Link State ID</i>			
<i>Advertising router</i>			
<i>Link State Sequence Number</i>			
<i>Link State Checksum</i>		<i>Link State Age</i>	

Exchange Protocol: messaggi (2)

- Link State Request

<i>Common header (type = 3, link state request)</i>
<i>Link State Type</i>
<i>Link State ID</i>
<i>Advertising router</i>

- Link state Update

<i>Common header (type = 4, link state update)</i>
<i>Number of link state advertisement</i>
<i>Link state advertisement #1</i>
<i>Link state advertisement #2</i>

Il protocollo di Flooding

- Funzioni:
 - aggiornare il database link state dell'autonomous system a seguito del cambiamento di stato di un link
- Messaggi:
 - Link State Update

<i>Common header (type = 4, link state update)</i>
<i>Number of link state advertisement</i>
<i>Link state advertisement #1</i>
<i>Link state advertisement #2</i>