Routing Essentials

Tecnologie e Servizi di Rete

Alessio Sacco, Guido Marchetto

*part of the slides adapted from material by Jim Kurose

Copyright Notice

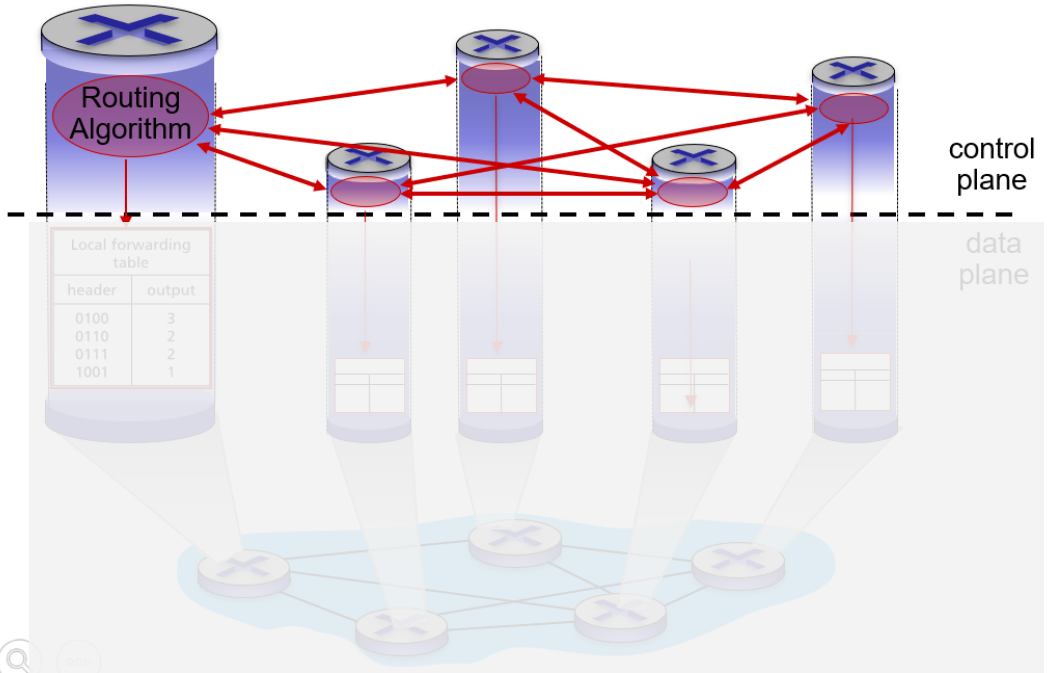
Outline

- Introduction and Taxonomy
  - Centralized routing
  - Distributed routing
    - Distance Vector algorithms
    - Link State algorithms
- Internet Routing Architecture
  - Intra-AS Routing protocols
  - Inter-AS Routing

Introduction and Taxonomy

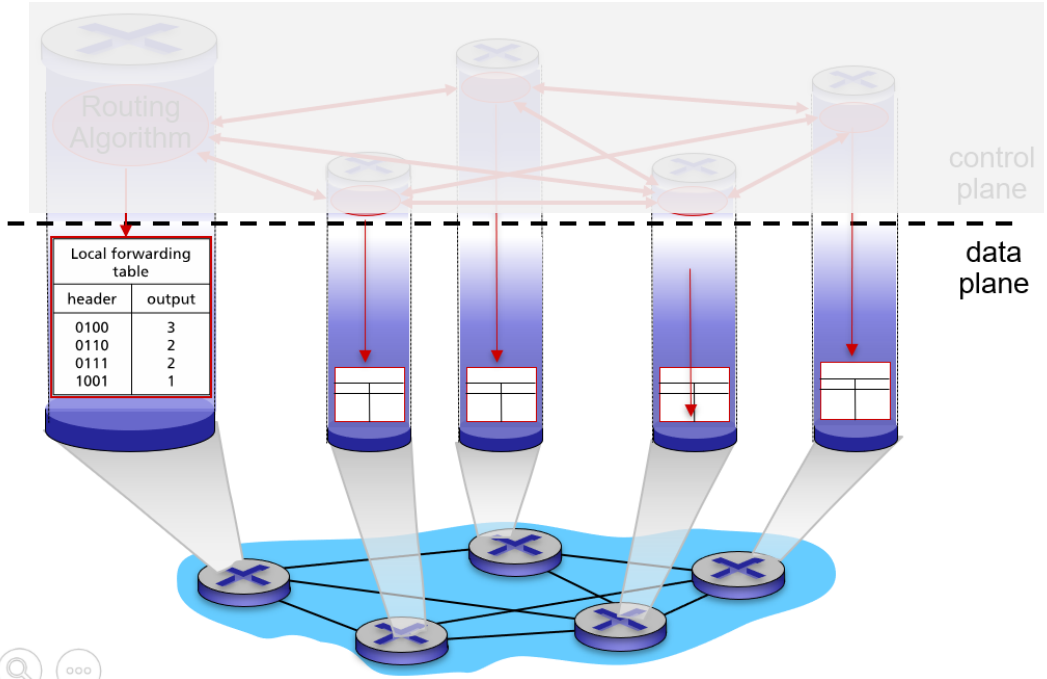
Routing

- Determination of (optimal) path from source to destination for packet transmission in a computer network
- Based on routing algorithms and protocols
- It is a *control plane* operation



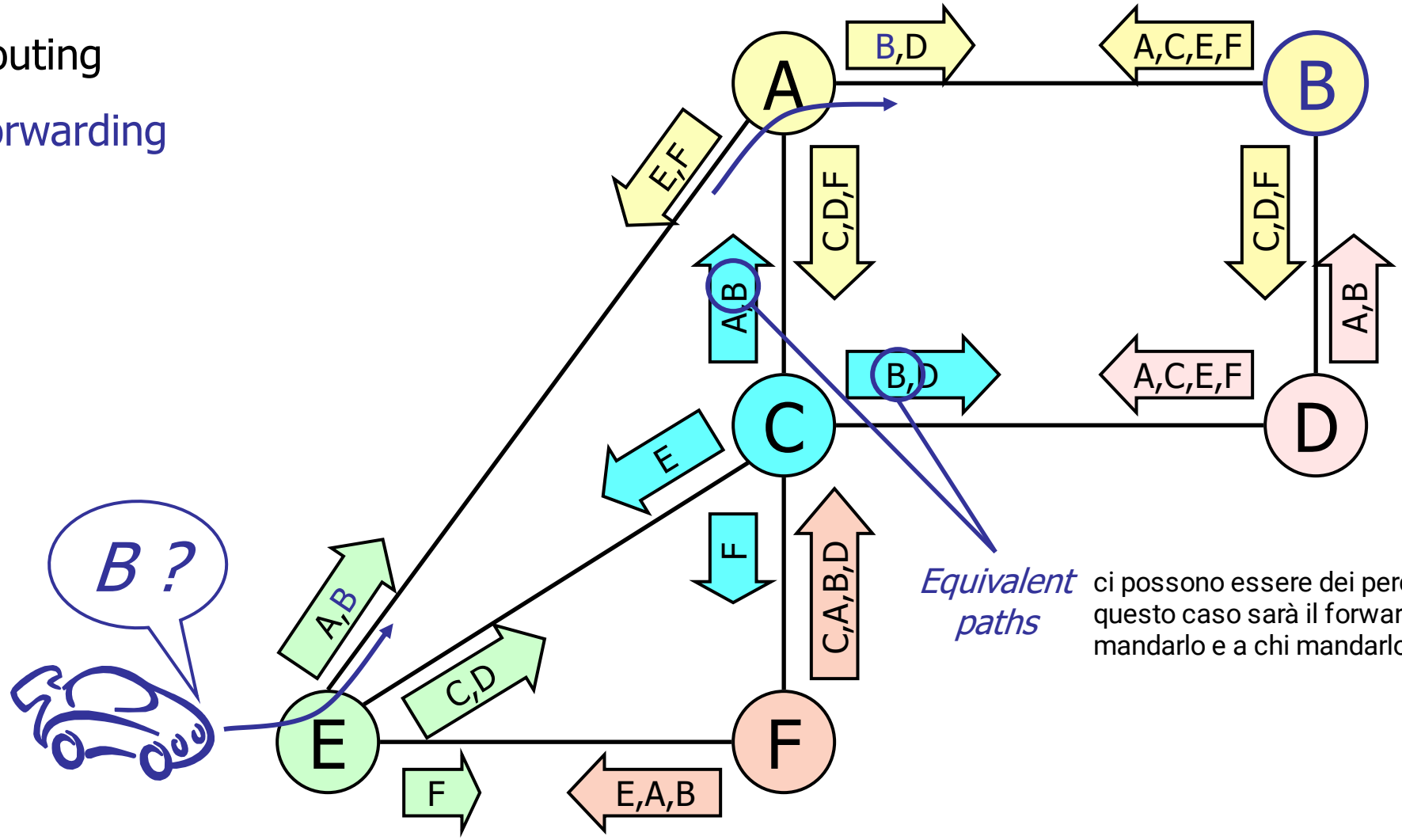
Forwarding

- Transmission of packets from one network device (router) to the next hop in their route
  - no real-time decision-making, just routing table look-up
  - next-hop choice predetermined by a routing algorithm and written in the routing table
- It is a *data plane* operation



Forwarding and Routing

1. Routing
2. Forwarding



Equivalent paths ci possono essere dei percorsi equivalenti, in questo caso sarà il forwarding a decidere se mandarlo e a chi mandarlo

Forwarding Procedures

- Forwarding by Network Address (most common)
- Label Swapping
  - e.g., MPLS (see later)
- Source Routing (rarely used)
  - The sender host writes the entire path in the packet
    - The sender must know the topology of the network
      - Often, source-routing technologies offer the possibility to calculate dynamically the route
  - End-users should interact with the users, instead of dealing with network internals

3 procedure principali per fare foewarding:
-tramite network address (quello che succede in IP)

chiamata anche routing by network address --> il motivo di sta stronzata è che all'inizio pensavano fossero simili, invece routing e forwarding sono ben diversi

-Label Swapping: cambia un po' la strategua--> abbiamo .ip destination e .nextHop (approfondiamoa dicembre)

--> Host che manda il pacchetto dice già che percorso vuole fare -->host carico di responsabilità e router molto semplice in quanto deve solo leggere il pacchetto e spedirlo nel luogo in cui ha chiesto l'host

NON COMUNE

- Note: Historically the difference between routing and forwarding was not clear, hence the confusion in the name of these algorithms

Forwarding Process

FONDAMENTALE, IL ROUTER DEVE ESSERE VELOCISSIMO A FARE FORWARDING, più è veloce a prendere e smistare, più tutto va veloce

- Routing Information Base (RIB): classical routing table, listing all the destinations present in the network
- Forwarding Information Base (FIB): same table, but rearranged in a way to speed up the processing of the packets (fast lookup)

costruttore può ottimizzare la struttura dati, ottimizzando le informazioni che ci scrive --> es: invece di scrivere next-hop, scrivo direttamente il nome dell'interfaccia in cui voglio mandare il pacchetto

- Forwarding Phases:
  - Next-hop/output port selection (using routing tables)
  - Switching: transfer to output port
  - Transmission

-decidere la porta in output
-porto il pacchetto in coda alla porta
-mando pacchetto

tutta la rete deve essere a conoscenza della mia posizione

Routing Algorithm Classification

- Non-adaptive algorithms (static)
  - Fixed Directory routing (static routing)
  - Random Walk, Flooding, Selective Flooding

- Adaptive algorithms (dynamic) noi ci concentriamo su routing dinamco
  - Central Routing
  - Isolated Routing
    - Hot Potato, Backward Learning
  - Distributed Routing
    - Distance Vector
    - Link State

Non-adaptive Algorithms

- Fixed Directory routing
  - Static routing
  - Manual configuration

equivalente dell'amministratore che scrive la tabella di routing a mano
-amministratore ha conoscenza totale della topologia e dice dove andare

- (Selective) flooding and similar
  - Random -> ricevo un pacchetto su una porta e scelgo una porta a caso su cui mandarlo, molto usata nelle rete IOT dove qualunque porta scelgo, posso arrivare alla mia destinazione (in quanto le reti IOT sono molto connesse)
    - Randomly select an output port
  - Flooding -> mando pacchetto su tutte le porte ad eccezione di quella da cui l'ho ricevuta
    - Forward on any output port (except the one on which the PDU was received)
  - Deflection or «hot potato» -> posso scegliere il percorso che ha la coda più scarica (simile a random ma magari un pelo più intelligente)
    - Used on regular topologies
    - If the correct port is available, route on it. If busy, select an available (non optimal) port

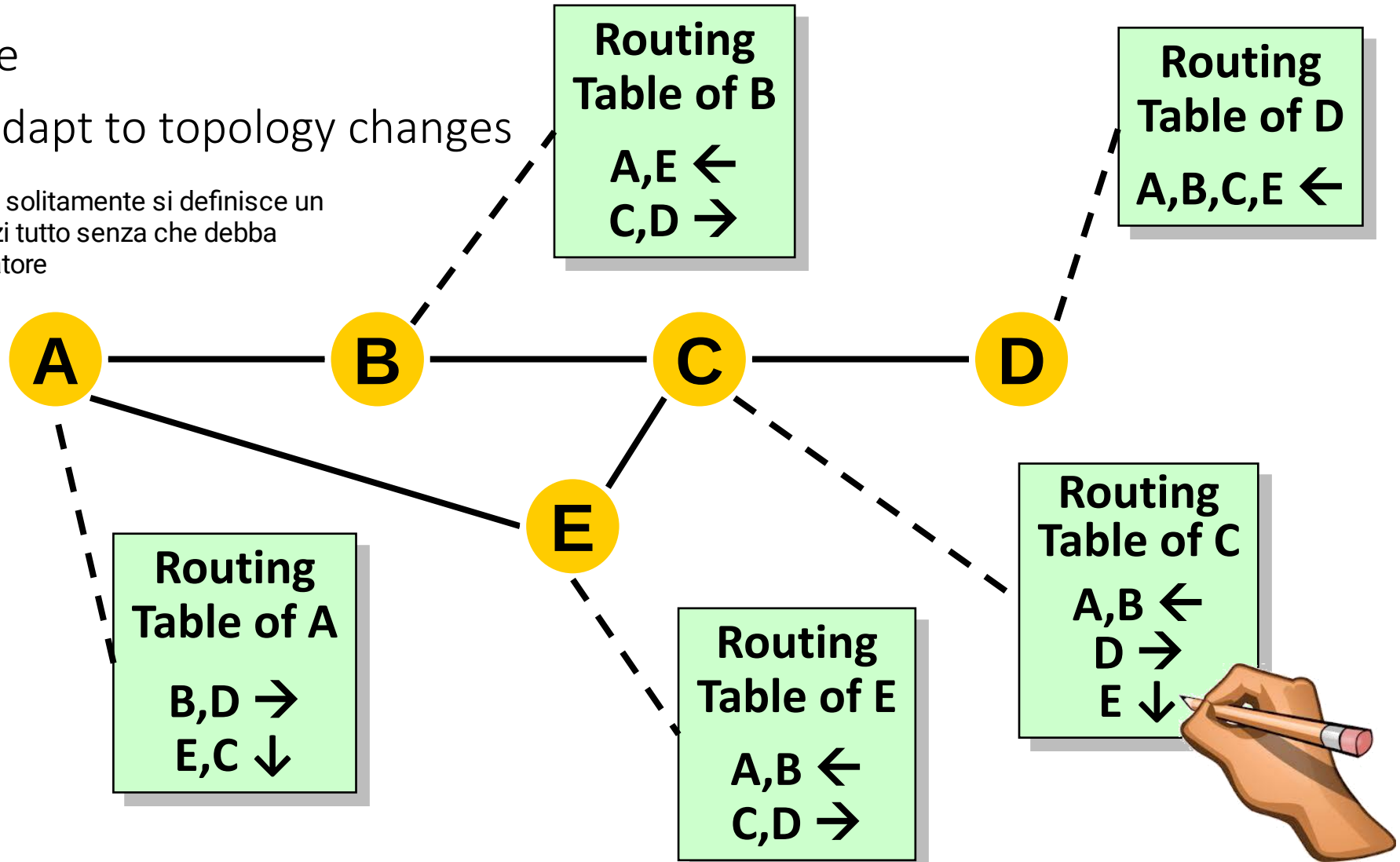
Fixed Directory Routing

- Admin has full control
- Error prone
- Does not adapt to topology changes

in alcuni casi serve ma solitamente si definisce un algoritmo che automizzi tutto senza che debba intervenire l'amministratore

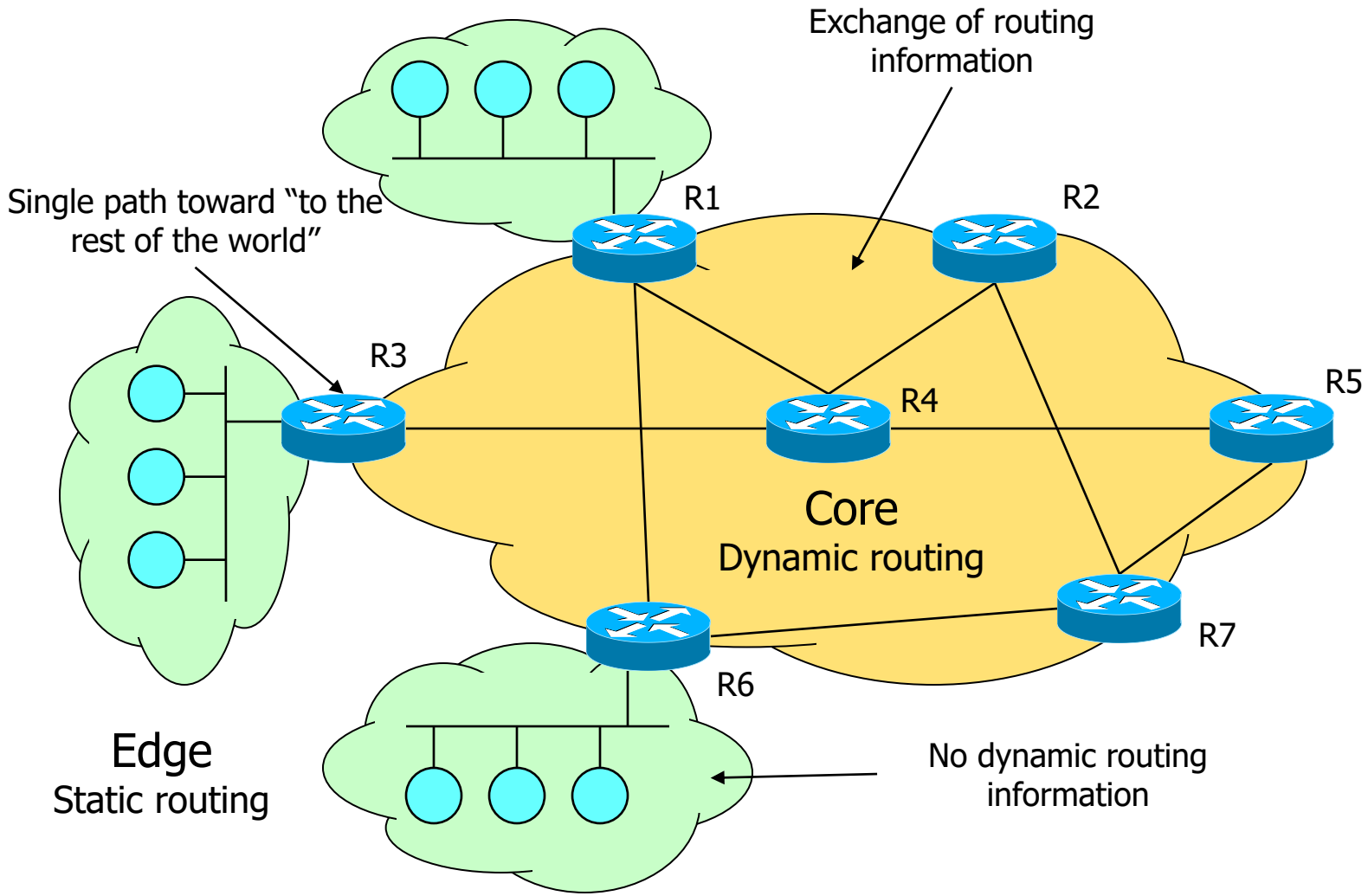
amministratore router per router, dice a quale porta andare per raggiungere chi

---- per reti con grandi nodi, difficile per l'amminisatore
---- NON tiene conto di cambiamenti sulla topologia
-se si rompe qualcosa che famo?
-se cambia qualcosa che famo? un nuovo host ?



usato magari all'edge (ai bordi della rete centrale - HomeGateway di casa). Potrei, staticamente, dire al mio router R3 di mandare tutto in default route a R4 (server di telecom) tutto ciò che non è della rete locale. Telecom non vuole che io mi fotta i dati di altri, quindi un minimo protegge --> i router parlano tra loro ma io non posso vedere come è fatta tutta la rete. Dunque i router di frontiera potrebbero essere settati staticamente

Static Routing find applications at the edge



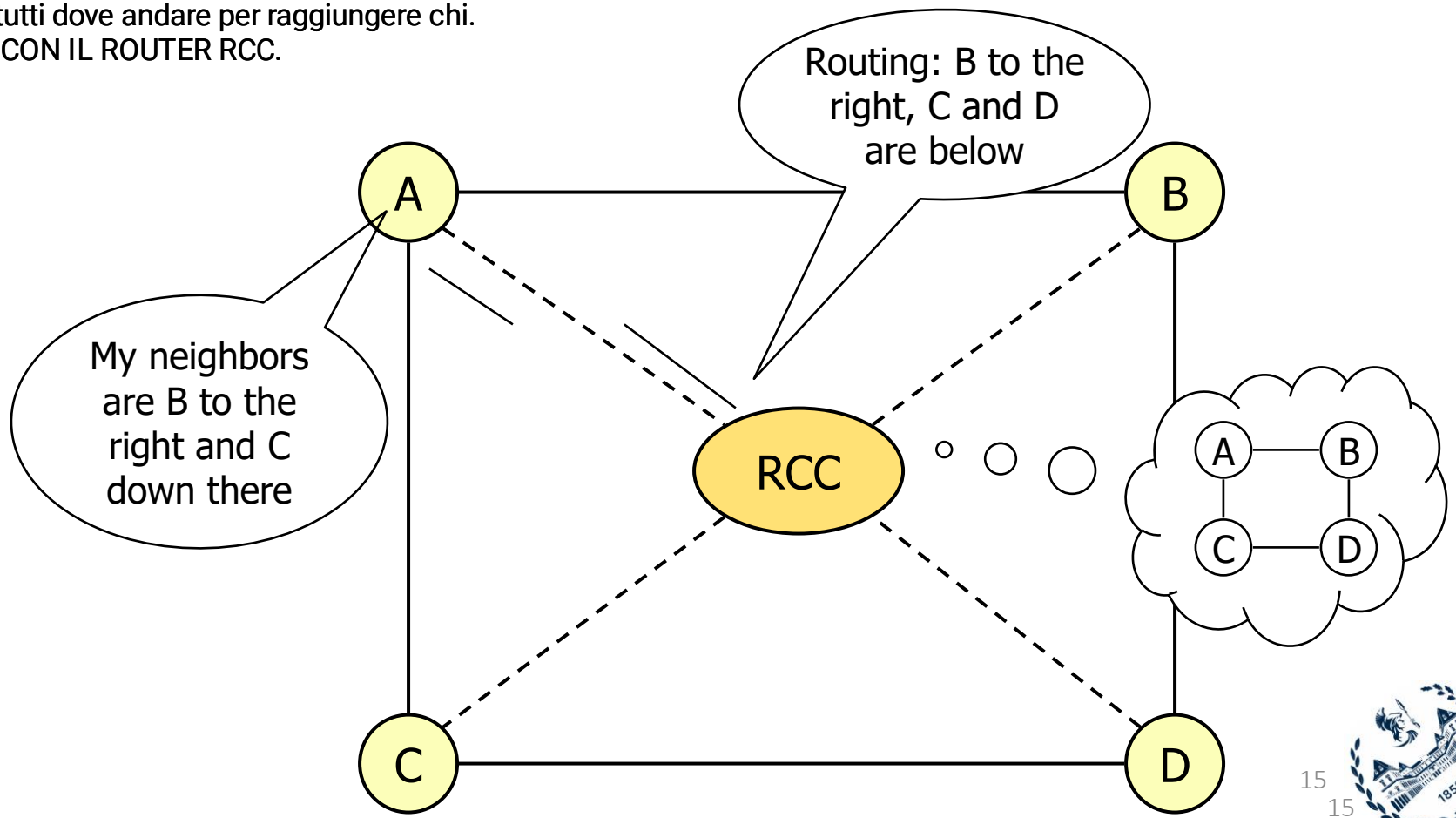
Adaptive Algorithms

- Centralized Routing
- Isolated Routing
- Distributed Routing
  - Distance Vector
  - Link State

Centralized Routing (SDN)

- One node (Routing Control Center, RCC) collects info from all other nodes
  RCC colleziona le informazioni dagli altri nodi e costruisce i percorsi con tutti gli altri. Ogni nodo dice a RCC con chi è collegato, dunque RCC conosce tutta la topologia e può dire a tutti dove andare per raggiungere chi.
  HO SOSTITUITO L'AMMINISTRATORE CON IL ROUTER RCC.
- Computes the path
- Distributes to all nodes

PROBLEMA: se c'è un guasto tra A e B, A avvisa RCC e RCC gli dice come altro può raggiungere --> il tutto è automatizzato, l'amministratore dorme tranquillo



Centralized Routing: PROs

- Routing Control Center (RCC)
  - Calculates and distributes routing tables
  - Needs information from all nodes
  - Optimizes performance
  - Simplifies troubleshooting

nelle reti è molto difficile debuggare, avendo un'entità centrale, io chiedo a RCC la topologia e lui me la fornisce. Se vedo che c'è qualche problema, posso vedere dove è l'errore e capire chi sta mandando informazioni sbagliate e correggere.

PROBLEMA: se RCC non funziona bene, sono fottuto, non funziona più niente porco zio --> Single Point Failure

Centralized Routing: CONs

- RCC is single point of failure --> ho un singolo punto di controllo, se questa entità non funziona più bene, non funziona più nulla
  ----- vorrei evitare di dipendere da un solo -->rischio di restare senza nulla
- RCC is bottleneck --> deve gestire tanto traffico, questo approccio non funziona nel momnento in cui la mia rete è particolarmente grande
  - Significant network load in proximity of RCC
- Not suitable for highly dynamic networks
  inoltre, più è dinamica la mia rete, più l'RCC ha difficoltà a stare dietro.

nota: RETE DINAMICA: rete che scambia molti messaggi con RCC, magari perchè varia molto

opposto rispetto al centralized --> qui ogni nodo è indipendente

Isolated Routing

- Each node decides independently ogni nodo indipendente, non c'è alcuno scambio di informazioni
- Opposite behavior compared to centralized routing
  - There is no RCC and each node chooses the path autonomously without exchanging information with the other routers
- No exchange of information
- E.g., Backward Learning (associava il mac alla porta vedendo su quale porta riceveva il pacchetto senza comunicare effettivamente con gli altri)
  - IEEE 8O2.1D switches

ci soffermiamo maggiormente

Distributed Routing

i router devono cooperare, devono parlare tra di loro e scambiare informazioni.
Alla fine di tutto, ogni router deciderà in maniera indipendente ma usando le informazioni che arrivano da gli altri.

- Routers co-operate in exchanging connectivity information
- Each router decides independently, but coherently ci interessa la COERENZA e non l'ottimo.
- Combines advantages of isolated and centralized routing --> parlo con più router
- Two big families: 2 approcci principali
  - Distributed Routing with Partial Information
    - Distance Vector algorithms--> a me basta decidere con informazioni parziali --> mi fido di quello che mi dicono in miei vicini
  - Distributed Routing with Global Information
    - Link State algorithms -> voglio informazioni globali, tutti comunicano --> più oneroso

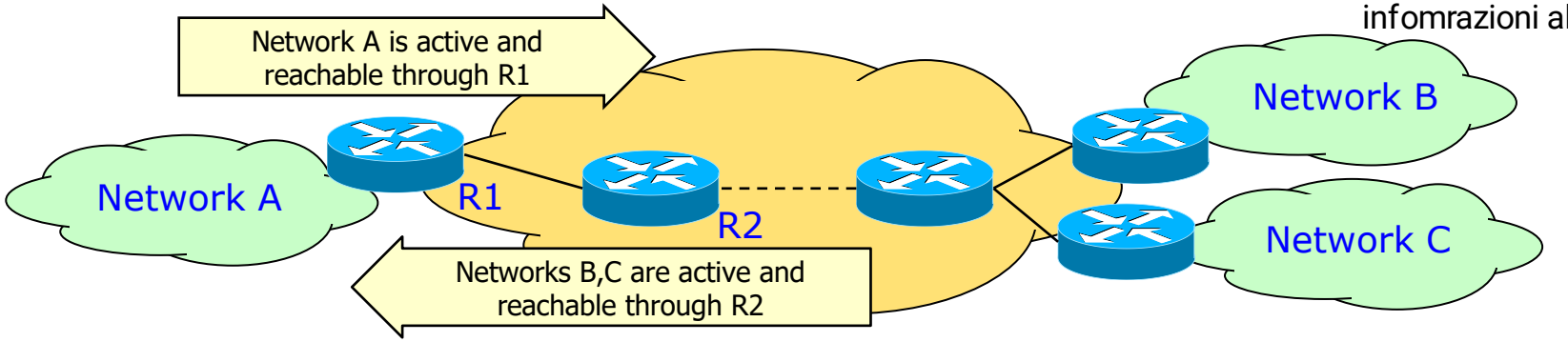
se la coerenza non è garantita, la nostra rete può funzionare male

What does a routing algorithm do?

-Ogni router annuncia le reti locali:
ROUTER R1: deve dirmi che lui ha la network A (130.192/16)
-Ogni router riceve le informazioni che gli altri hanno generato
ROUTER R1: saprà della presenza di network B e Network C, comunicazione avviene attraverso R2 che si fa portavoce di questa informazione propagandola, facendo da tramite.

- In the most common routing algorithms (e.g., Distance Vector, Link State), each router
  - 1) Generates information about the reachability of local networks
    - E.g., Router R1 tells the rest of the network that NetA exists and it is reachable through it
  - 2) Receives information about the reachability of remote networks
    - E.g., Router R1 is told that NetB and NetC exist and are reachable through R2
  - 3) Propagates information about the reachability of remote networks
    - E.g., Router R2 tells that NetB and NetC exist and are reachable through it
- Each router can enable only one direction among (1) and (2)

propagazione e ricezione sono indipendenti, io vorrei poter disabilitare ricezione o propagazione.
Amministratore non vuole che l'home gateway di qualcuno si metta ad annunciare la rete privata di qualcuno quindi potrei disabilitarla, alllo stesso modo magari gli voglio impedire di accedere alle infomrazioni altrui.



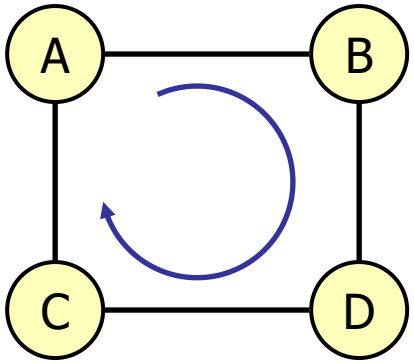
Routing algorithms: choosing criteria

- Objectives
  - Network coherence: avoid loops, “black holes”, etc.
  - Algorithm must operate automatically
- Optimality
  - Preferred way to achieve consistency
  - Metrics: criteria used to measure the optimality

-deve essere automatico, devono rendersi conto da soli di chi è collegato a loro

tutti i router devono agire allo stesso modo, se ho un router che fa lo stronzo e decide magari di mandare in senso anti-orario quando abbiamo deciso di trasmettere in senso orario --> rischio di loop --> algoritmo di routing non soddisfa il suo scopo ---- altra ipotesi: buco nero --> il pacchetto non raggiunge mai destinazione

tutti i router devono puntare allo stesso concetto di ottimo --> si definiscono delle metriche, dei criteri per cui io valuto se una sceltà è buona o no (es: throughput e latenza, scegliere un giusto bilanciamento tra le due e farlo valere per tutti i router)
--> quando decido i percorsi devo capire quale metrica voglio ottimizzare



Example of consistency: “send always the packet clockwise”

Metrics (1)

- Measure how good a path is by looking at its cost
  - When comparing two paths, the one with the lowest cost is preferred
  - Possible criteria: -minimizzare un costo, possibili esempi di costo:
    - Minimum distance (e.g., number of hops)
    - Minimum delay
    - More criteria mixed together with specific weights
    - There may be contrasting parameters (e.g., maximization of network usage and minimization of delays at the same time)

distanza: km? non un granchè in quanto ogni router dovrebbe capire la distanza tra lui e il prossimo ed etc.
–> usiamo come metrica il nr di HOP –> molto banale ma molto funnzionale in quanto in effetti i link sono molto affidabili mentre i router potrebbero non esserlo (es: buffer pieno, devo scartare quel pacchetto –> pacchetto perso) e introducono nuovo ritardo per fare il tempo di processing.

senti 3 minuti sta slide (circa minuti 75)

Metrics (2)

obbiettivo del router: trasportare pacchetti utente

- Metrics and Complexity
  - More criteria → The algorithm becomes more complex, and may require more resources at run-time
  - Please remember that routers are there to forward user traffic, not to calculate paths
- Metrics and Stability
  - Some metrics may cause instability problems
  - E.g., metric based on network load

se ho congestionato rete sopra, inizio a mandare sotto, poi congestiono sotto e inizio a mandare sopra , etc all'infiinito –> si fa un po' sopra e un po' sotto ma non si raggiunge mai stabilità –> na bella cacca

  - Route flapping may be an important issue
    - Route flapping: the frequent change of preferred routes
    - Route flapping → frequent transients → connectivity issues

A volte conviene avere una metrica stabile piuttosto che superottimizzata e non stabile

Distance Vector Algorithm

-ogni nodo informa gli altri riguardo le destinazioni che lui riesce a raggiungere, per ogni destinazione informa anche sul costo che c'è per raggiungerla
-informazione propagata solo al router a lui vicino
-si basa su vhiacchericcio/gossip --> R1 dice al router che ha vicino (R2) che può raggiungere A(costo1), B(costo2), C(costo10), R2 riceve questa informazioni e propaga questa informazione dicendo che può raggiungere A(Costo 1+costoPerRaggiugnereR1), B(2+costoPerRaggiugnereR1), C(10+costoPerRaggiugnereR1)

- Every node periodically exchanges with adjacent nodes a Distance Vector, containing
  - Reachable destinations
  - Current distance to each destination
    - E.g., number of hops on the path

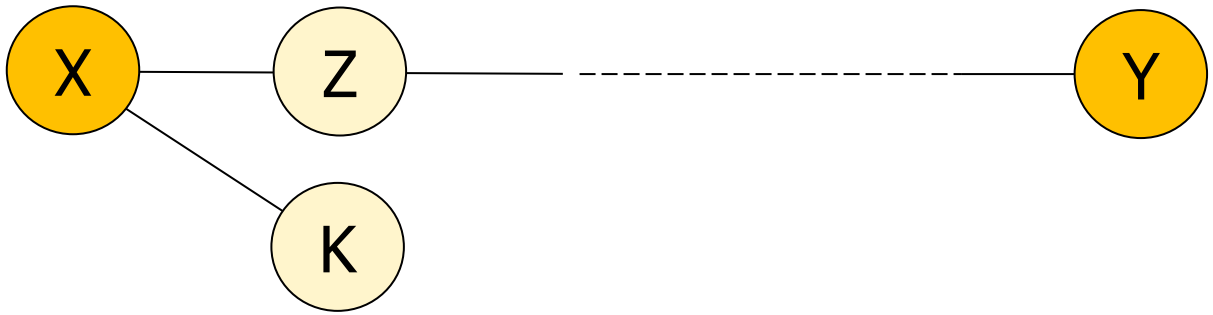
se R2 non conosce niente, prende quei valori. Se R2, già conosce A,B,C e ha già un modo per raggiungerli, cambia valori e nextHop solo se sono migliori. Altrimenti ignora.

dunque ogni router avrà òa distance tbale che contiene per ogni destinazione, il costo per raggiungerla e il next hop --> colui che mi ha annunciato quel costo

- Each node maintains two data structures:
  - Distance Table (DT): lists costs to any known destination through each neighbor (next hop)
  - Routing Table (RT): lists costs and next hops of selected routes to each destination
- A Distance Vector contains the same information as the RT, minus the next hop
- Nodes compare the received vector with the current DT and RT and modify them
  - Adding new destinations
  - Modifying routing if new paths are shorter
  - Modifying costs if needed
- Bellman–Ford algorithm

Distance Vector: implementation

- Structure of Distance Table in every node:
  - One row for each possible destination
  - One column for each adjacent node
- Example: Element of node X's DT, for destination Y through adjacent node Z
  - Cost of reaching Y from X, using Z as next hop qual è il costo per raggiunge Y da X, attraverso Z



Source → X

$$D^{Z}(Y,Z) = c(X,Z) + \min_{w} \{D^{Z}(Y,w)\}$$

Destination

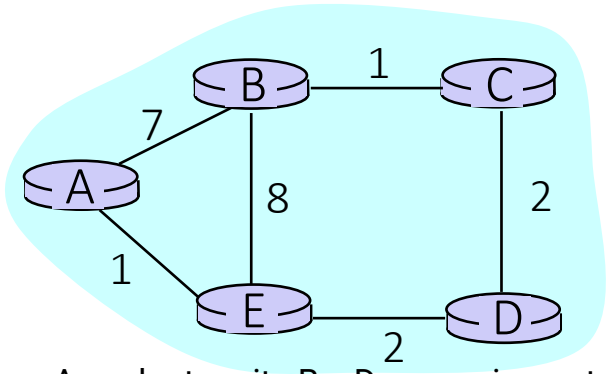
Next hop

Distance Table: example

ad un certo punto, dopo che i router si sono scambiati le varie informazioni, avrò la mia tabella piena

dunque all'inizio si ha una fase di transitorio in cui si ha instabilitàù
-finito il transitorio, quanOD HO COMPLETATO LA MIA TABELLA, SI HA STABILITA'
--> tabella rimane fissa finchè non ci sarà un cambiamento di topologia

COME RAGIONA IL ROUTER E?



E scopre che può raggiungere A anche tramite B e D ma ovviamente sceglie il diretto con A.

$$D^{E}(C,D) = c(E,D) + \min_w\{D^{D}(C,w)\}$$
$$= 2+2 = 4$$

$$D^{E}(A,D) = c(E,D) + \min_w\{D^{D}(A,w)\}$$
$$= 2+3 = 5 \quad \text{loop!}$$

$$D^{E}(A,B) = c(E,B) + \min_w\{D^{B}(A,w)\}$$
$$= 8+7 = 15 \quad \text{loop!}$$

Cost to .. Via ...

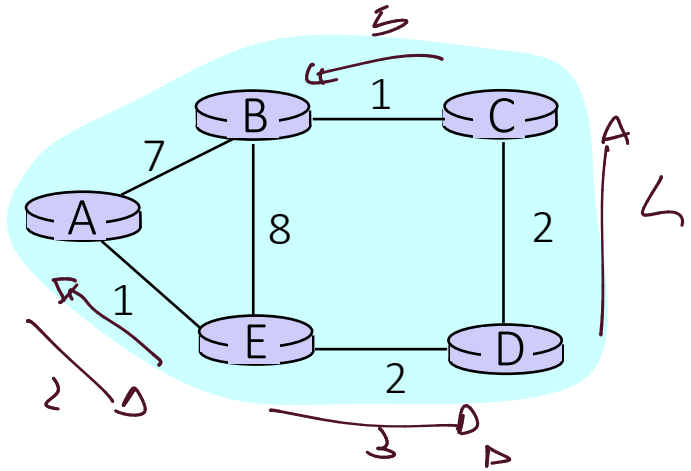
$D^{E}()$

destination	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

se E scegliesse di passare da D per raggiungere A, si creerebbe un loop pericoloso. Per come funziona il distance vectore, non me ne accorgerei

Distance Table: example

COME RAGIONA IL ROUTER E?



$$D^{E}(C,D) = c(E,D) + \min_{w}\{D^{D}(C,w)\}$$
$$= 2+2 = 4$$

$$D^{E}(A,D) = c(E,D) + \min_{w}\{D^{D}(A,w)\}$$
$$= 2+3 = 5$$ loop!

$$D^{E}(A,B) = c(E,B) + \min_{w}\{D^{B}(A,w)\}$$
$$= 8+7 = 15$$ loop!

Cost to .. Via ...

$D^{E}()$	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

destination

From DT to RT

Cost to .. Via …

$D^E()$	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

destination

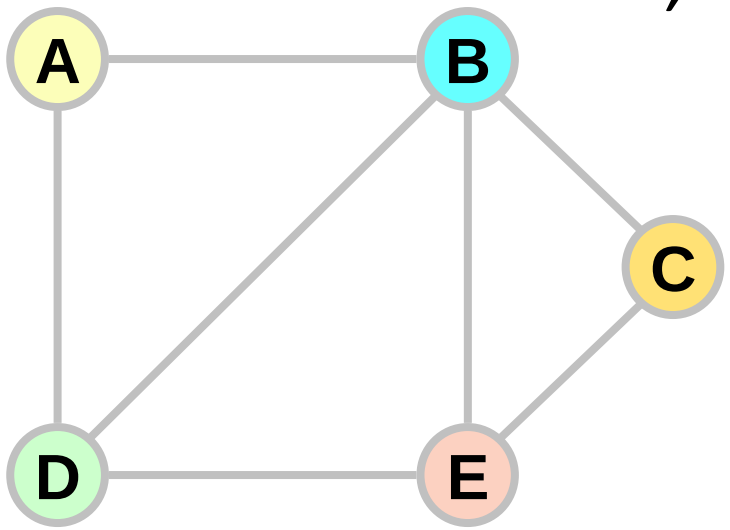
Destination Table

Next Hop, Cost

destination	
A	A,1
B	D,5
C	D,4
D	D,2

Routing Table

Sample Scenario



Routing information stored by A

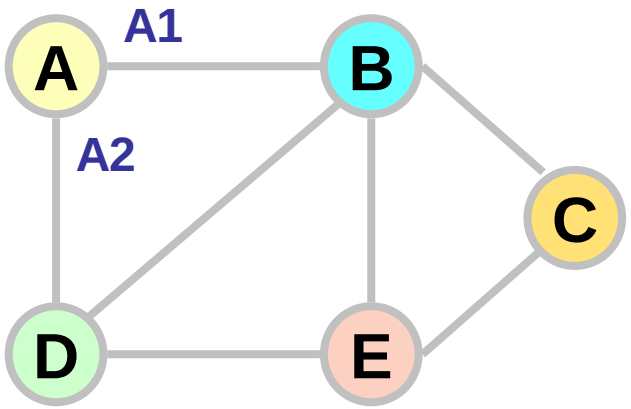
A riceve informazioni dai suoi vicini:

Loc (A)	DV (B)	DV (D)
A, 0	A, 1	A, 1
	B, 0	B, 1
	C, 1	C, 2
	D, 1	D, 0
	E, 1	E, 1

D non dice come raggiunge C, dice solo il costo
--> questo causa rischi di loop

in realtà il router non crea la distance table ma il distance vector che inoltra agli altri router e mi permetterà di creare la routing table

Distance Vector Merging and Generation



E è raggiunto da entrambi a costo 1 --> è indifferente quale scelgo

Received from line A1

Received from line A2

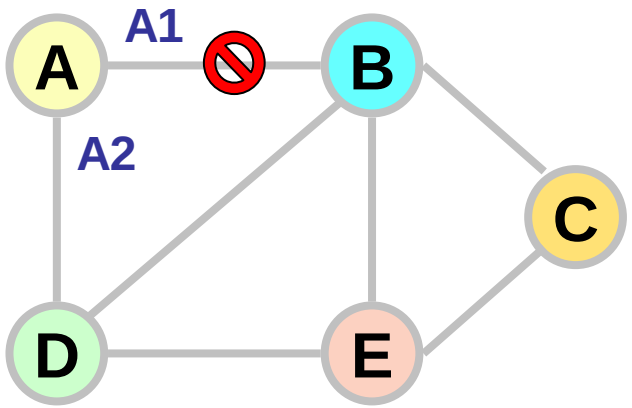
Loc (A)	DV (B)	DV (D)	ROUT. TABLE (A)			DV (A)
A, 0	~~A, 1~~	~~A, 1~~	A,	local,	0	A, 0
	B, 0	~~B, 1~~	B,	A1,	1	B, 1
	C, 1	~~C, 2~~	C,	A1,	2	C, 2
	~~D, 1~~	D, 0	D,	A2,	1	D, 1
	E, 1	E, 1	E,	A2,	2	E, 2

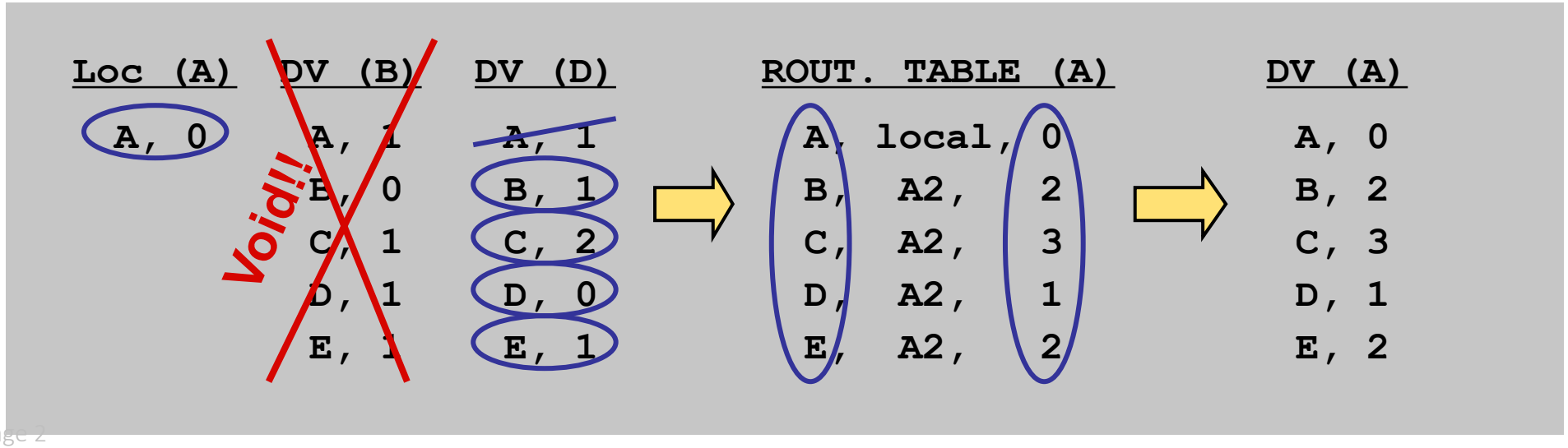
Topology Change

se ho un cambiamento tipologia, la routing table non è più valida, devo rifare i vari calcoli --> sad

i distance vector vengono inviati in maniera periodica quindi si A si rende conto che B non c'è più e lo ignora invalidando il suo distance vector. Si ricalcola routing table di A.

A genera il nuovo distance vector, il DV(A) sarà ricevuto da tutti gli altri che vedono un costo cambiato e capiscono che c'è stato un cambiamento





Example: Cold Start

RT (A)	RT (B)	RT (C)	RT (D)	RT (E)
A,loc,0	B,loc,0	C,loc,0	D,loc,0	E,loc,0

A sends its DV

RT (A)	RT (B)	RT (C)	RT (D)	RT (E)
A,loc,0	A,B1, 1	C,loc,0	A,D1, 1	E,loc,0
	B,loc,0		D,loc,0	

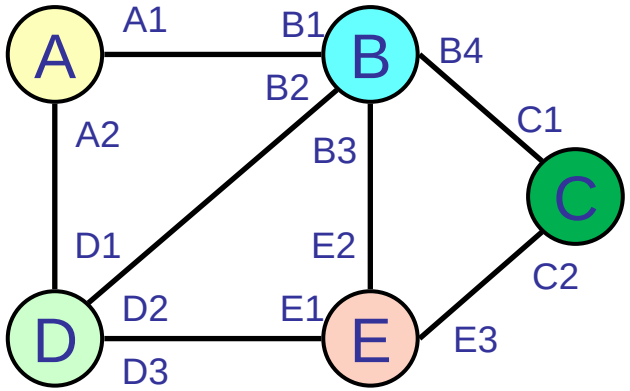
B and D send their DVs

RT (A)	RT (B)	RT (C)	RT (D)	RT (E)
A,loc,0	A,B1, 1	A,C1 ,2	A,D1, 1	A,E2, 2
B,A1, 1	B,loc,0	B,C1 ,1	B,D2, 1	B,E2, 1
D,A2, 1	D,B2, 1	C,loc,0	D,loc,0	D,E1, 1
				E,loc,0

All send their DVs

. . .

RT (A)	RT (B)	RT (C)	RT (D)	RT (E)
A,loc,0	A,B1, 1	A,C1 ,2	A,D1, 1	A,E2, 2
B,A1, 1	B,loc,0	B,C1 ,1	B,D2, 1	B,E2, 1
C,A1, 2	C,B4, 1	C,loc,0	C,D2, 2	C,E3, 1
D,A2, 1	D,B2, 1	D,C2, 2	D,loc,0	D,E1, 1
E,A2, 2	E,B3, 1	E,C2, 1	E,D3, 1	E,loc,0



Issues

serie di problemi di questo algoritmo.

- Black Hole

BUCO NERO: R1 manda a R2 un pacchetto destinato alla NetB, R2 non ha la più pallida idea di dove sia NetB e allora scarta il pacchetto

  - A router does no longer know where a destination is, even if it is still reachable through an alternate path→traffic for this destination is discarted
  - DoS attack by malicious node advertising impossibly short, non-existent routes to attract traffic, which is then discarded

-> molto probabile che succeda nel transitorio, o durante un guasto. --> distance vectore è lento a reagire
è difficile che ci sia un buco nero dopo il transitorio in assenza di guasti o cambiamenti

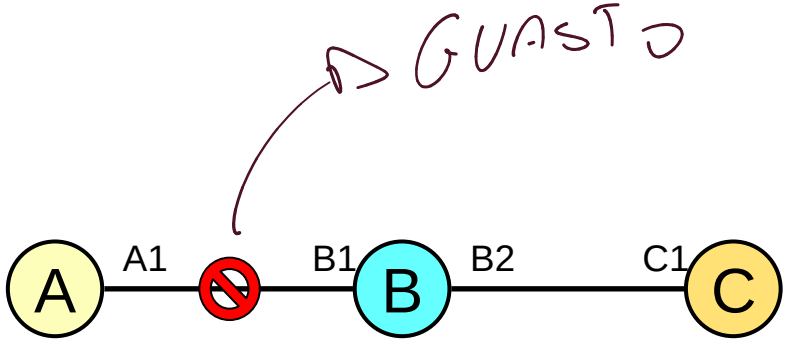
–> se si vuole bloccare il traffico nei confronti di un sito (es italia vuole non far usare google agli utenti –> Italia va da Telecom e dice di bloccare l'accesso)
ALLORA TELECOM dice che il router per raggiungere google.com è il router dell'aula R4, ma il router dell'aula R4 ovviamente non conosce come contattare google.com e quindi BUCO NERO, tutto viene scartato

- Count to infinity
  - Misalignment of Routing Tables due to slow convergence time of the Distance Vector algorithm and inconsistent information at different nodes

- Bouncing Effect
  - (Temporary) Loop caused by inconsistent Routing Tables following a link failure

Common Reason: lack of information on the global topology

Count to Infinity



B:

Loc (B)	~~DV (A)~~	DV (C)	RT (B)
B, 0	~~A, 0~~	A, 2	A,B2 ,3
	~~B, 1~~	B, 1	B,loc,0
	~~C, 2~~	C, 0	C,B2 ,1

C:

Loc (C)	DV (B)	RT (C)
C, 0	A, 1	A,C1 ,2
	B, 0	B,C1 ,1
	C, 1	C,loc,0

B ricalcola routing table quindi conosce solo se stesso e il DV di C,
B si rende conto che il suo link non va più ma quel cretino di C mi ha appena detto che può raggiungere A con costo 2 e quindi pensa di poterlo ancora raggiungere passando da C.
Il distance vector di B ora pensa di poter raggiungere A con 3

B sends DV

Loc (C)	DV (B)	RT (C)
C, 0	A, 3	A,C1 ,4
	B, 0	B,C1 ,1
	C, 1	C,loc,0

C quindi ora pensa di poter raggiungere A con 4 passando da B

C sends DV

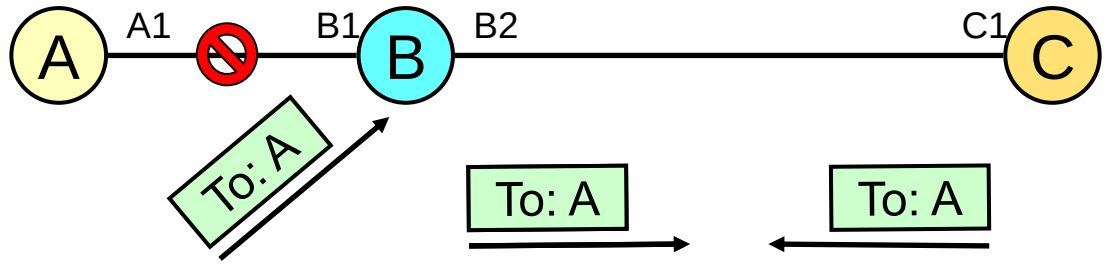
il tutto va avanti all'infinito--> instabilità --> traffico va saltando tra BeC e non riesco a contattare più A.
Questo traffico inutile rallenta anche il traffico che dovrebbe avvenire tra B e C.
unica salvezza è il TTL, distance vectore andrà ancora all'infinito ma i pacchetti che rimbalzano, a na certa crepano in quanto TTL supera soglia (8 bit)

B ora pensa di poter raggiungere A con 5 passando da C

Loc (B)	DV (C)	RT (B)
B, 0	A, 4	A,B2 ,5
	B, 1	B,loc,0
	C, 0	C,B2 ,1

Count to Infinity!

Bouncing Effect



B:

Loc (B)	DV (A)	DV (C)	RT (B)
B, 0	~~A, 0~~	A, 2	A,B2 ,3
	~~B, 1~~	B, 1	B,loc,0
	~~C, 2~~	C, 0	C,B2 ,1

C:

Loc (C)	DV (B)	RT (C)
C, 0	A, 1	A,C1 ,2
	B, 0	B,C1 ,1
	C, 1	C,loc,0

B sends its DV

Loc (C)	DV (B)	RT (C)
C, 0	A, 3	A,C1 ,4
	B, 0	B,C1 ,1
	C, 1	C,loc,0

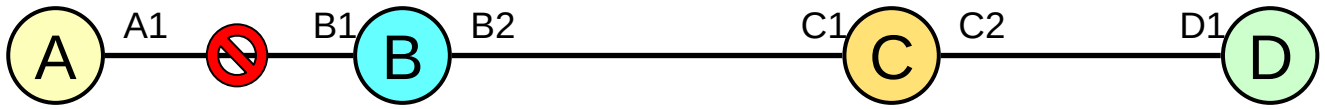
(Partial) solutions non risolvono a 100%

- Split Horizon
- Path Hold Down
- Route Poisoning

Split Horizon

"If C reaches destination A through B, it is useless for B trying to reach A through C"

il costo del protocollo è che ora C non deve più generare un DV per tutti i suoi vicini ma deve generare un DV diverso in base a chi lo manda



B dice che non riesce più a raggiungere A, C in questo caso non comunica a B la rotta A, quindi non ho il counting infinity e i loop

B:

Loc (B)
B, 0

~~DV (A)~~
~~A, 0~~

DV (C)
C, 0
D, 1

RT (B)
B,loc,0
C,B2 ,1
D,B2, 2

DV (C→C1)
C, 0
D, 1

DV (C→C2)
A, 2
B, 1
C, 0

Split Horizon

- Prevents loops between two nodes
- Speeds up convergence
- "Personalized" DVs to neighbors
  - DV of C to B does not contain destinations reached through B
- In actual implementations, route has to expire (e.g., A in DV from D)

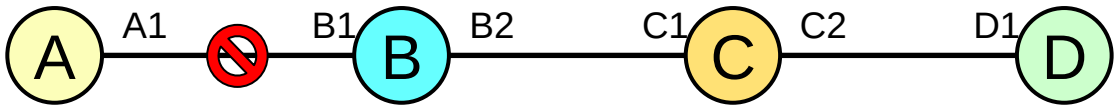
NON FUNZIONA SEMPRE

Path Hold Down

QUANDO UN LINK HA UN GUASTO, ASSUMO che tutte le destinazioni raggiungibili in quel modo, non sono più raggiungibili --> destinazione in quaranetna

If link L fails, all destinations reachable through link L are considered unreachable for a certain period of time
(i.e., no routes to them are computed)

B mette A in quarantena --> non accetta per un po' di tempo DV o pacchetti per A.
C non si accorge del guasto quindi continua ad annunciare la destinazione A --> count infinity su C-D



B:

Loc (B)
B, 0

DV (A)
A, 0
B, 1
C, 2
D, 3

Quarantine!

DV (C)
A, 2
B, 1
C, 0
D, 1

RT (B)
B,loc,0
C,B2 ,1
D,B2, 2

C:

Loc (C)
C, 0

DV (B)
B, 0
C, 1
D, 2

DV (D)
A, 3
B, 2
C, 1
D, 0

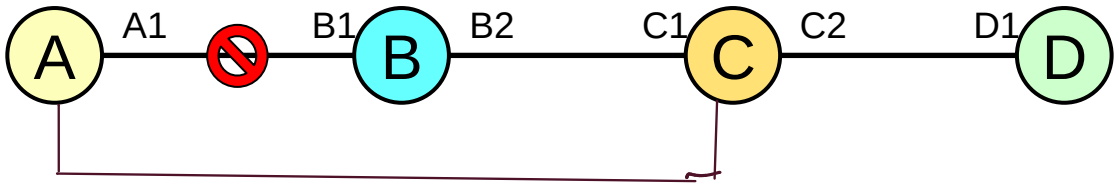
still Count to Infinity!

RT (C)
A,C2, 4
B,C1, 1
C,loc,0
D,C2, 1

Path Hold Down

If link L fails, all destinations reachable through link L are considered unreachable for a certain period of time (i.e., no routes to them are computed)

B mette A in quarantena --> non accetta per un po' di tempo DV o pacchetti per A.
C però in questo caso può continuare a parlare con A, stessa cosa farà D ma B ha messo in quarantena A quindi per un po' non lo vedrà



B:

Quarantine!

Loc (B)
B, 0

~~DV (A)~~
~~A, 0~~
~~B, 1~~
~~C, 2~~
~~D, 3~~

~~DV (C)~~
~~A, 2~~
B, 1
C, 0
D, 1

RT (B)
B,loc,0
C,B2 ,1
D,B2, 2

C:

still Count to Infinity!

Loc (C)
C, 0

DV (B)
B, 0
C, 1
D, 2

DV (D)
A, 3
B, 2
C, 1
D, 0

RT (C)
A,C2, 4
B,C1, 1
C,loc,0
D,C2, 1

Path Hold Down

- High convergence time for the examined node (even with an alternate path)

- The router that noted the fault may not participate to any loop at least until the timeout of Hold Down timer

How to react to Cost Increase?

- When routes have increasing costs, routing loops may happen

- Cost-increasing routes in DVs are not used
  - e.g., two back-to-back advertisements show a cost increase

    Se ricevo due volte un icnremento di costo sul DV --> quarantena -->problema è che può non essere un count infinity ma in realtà un costo che sta aumentando normalmente e potrei invalidare un percorso funzionante
- May be combined with Path Hold Down

- CON: Might block routes with legitimate cost increase

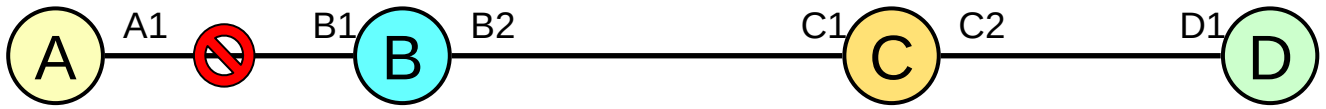
Split Horizon with Poisonous Reverse

- An invalid route is advertised at infinite distance/cost instead of just omitting it
  - It can substitute or complement Path Hold Down
- More aggressive
- Only advantage: faster convergence
- No need to wait for route expiration

se non presente nel distance vectore, l'annuncio comunque ma a costo infinity

avveleno certe rotte mettendole a costo infinito

inviare tutte le destinazioni conviene in quanto così il router capisce un minimo cosa è successo



DV (C→C1)

A, inf
B, inf
C, 0
D, 1

DV (C→C2)

A, 2
B, 1
C, 0
D, inf

B:

Loc (B)

B, 0

~~DV (A)~~

~~A, 0~~
~~B, inf~~
~~C, inf~~
~~D, inf~~

DV (C)

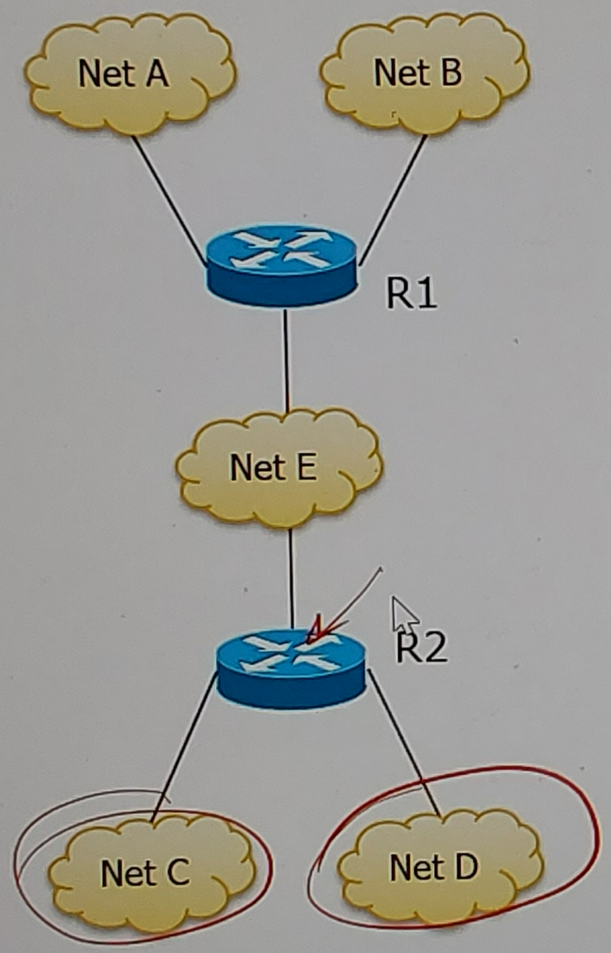
A, inf
B, inf
C, 0
D, 1

RT (B)

B,loc,0
C,B2 ,1
D,B2, 2

nella pratica bisogna fare alcuni accorgimenti

Distance Vector in real networks: example



Without Split Horizon

DV (R1)
Net A,1
Net B,1
Net C,2
Net D,2
Net E,1

With Split Horizon

DV (R1)
Net A,1
Net B,1
Net E,1

DV (R2)
Net A,2
Net B,2
Net C,1
Net D,1
Net E,1

DV (R2)
Net C,1
Net D,1
Net E,1

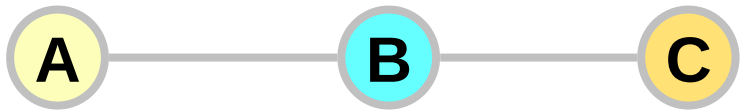
Bottom Line

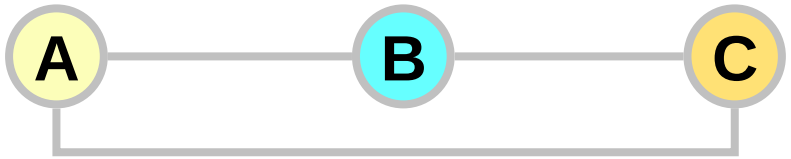
il DV non riesce a distinguere topologia

- PROs:
  - Simple to implement and deploy
  - Very little configuration
- CONs:

c non sa se conosce A in maniera diretta o se perchè lo ha detto B, quindi C non riesce a distinguere

  - Exponential worst-case complexity and convergence time $O(n^2)$
  - Routers do not know the network topology --> router non conoscono la topologia della rete
  - e.g., B cannot distinguish between:





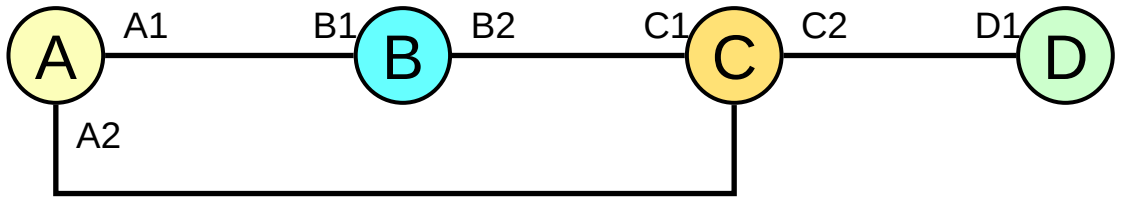
  - Convergence limited by slower links and routers set pace
  - Complex tuning and troubleshooting --> per debuggare vado sui router, ogni router ha la collezione dei DV ricevuti, difficile da debuggare
  - Large routing traffic (and storage)
    - Not suitable for large complex networks

->parte da Distance Vector ma con qualch emodifica

Path Vector

--> i nodi inviano il loro Path Vector:
-destinazione
-costo
-nodi coinvolti

- Nodes exchange dynamically-updated path information
- Easy to detect loops in the vector
- Increased overhead



A:

non usato da IP

Loc (A)
A, 0

PV (B)
A, 1, [B]
B, 0, [-]
C, 1, [B]
D, 2, [B,C]

PV (C)
A, 1 [C]
B, 1, [B]
C, 0, [-]
D, 1, [D]

RT (A)
A, loc, 0
B, A1, 1
C, A2, 1
D, A2, 2

PV (A)
A, 0, [-]
B, 1, [A]
C, 1, [A]
D, 2, [A,C]

non usato da IP

Link State algorithm

--> ogni nodo invia informazioni su costi e stato di ogni suo link a tutti gli altri nodi
-->dunque sto inviando a tutti i nodi un'informazione locale
-facendo così ogni nodo può costruire la topologia della rete

- Every node sends cost info (status) of each link in broadcast to all other nodes
  - Implemented using multicast among routers

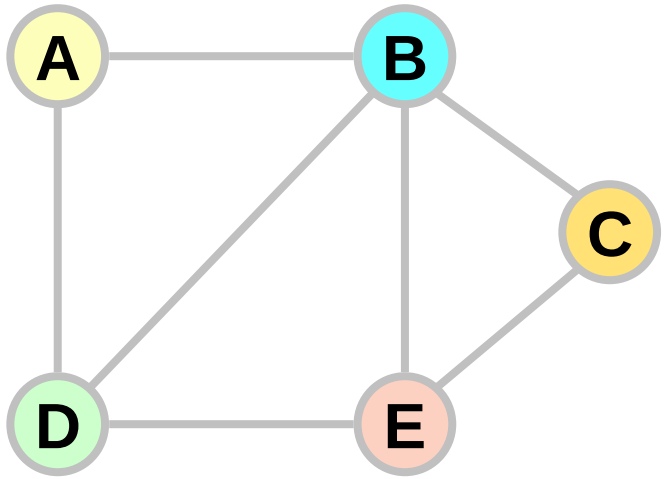
SELECTIVE FLOODING -> FLOODING SPECIALE

- Every nodes builds the network topology, knowing all link costs
- Every node computes minimum cost path (paths) toward every other node
  - info stored in the routing table
- Dijkstra algorithm is used to compute the optimal path
  - Iterative algorithm: after *k* iterations the minimum cost path toward *k* destinations are obtained
  - Only works for positive costs

OGNI NODO:
-costruisco mappa/topologia
-calcolo percorsi minimi

Link State Database

ogni router riceve tutta la lista di informazioni e quindi può costruirsi l'intera mappa della rete



Stored by each Node

LS (A)	LS (B)	LS (C)	LS (D)	LS (E)
B, 1	A, 1	B, 1	A, 1	B, 1
D, 1	C, 1	E, 1	B, 1	C, 1
	D, 1		E, 1	D, 1
	E, 1			

svantaggio : 1) il pacchetto va inviato a tutti
2) lento in costruzione routing table -->DJIKSTRA

Dijkstra Algorithm

- Low complexity: L * log (N)
  - L: number of links
  - N: number of nodes
- Shortest Path First
  - The next node “nearest” to the root is identified
  - Its next hop is inserted into the routing table

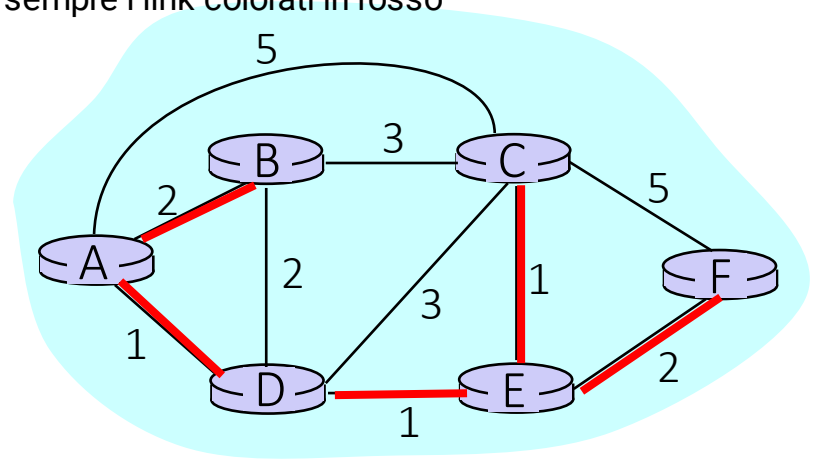
- Bellman-Ford: N * L
  - Dijkstra’s algorithm is more scalable than Bellman-Ford

Dijkstra: example

A ha 3 vicini. Qual è quello che raggiunge a costo minimo? D --> A aggiunge D al suo albero
AD ha 3 vicini. Qual è quello che raggiungo a costo minimo? E --> AD aggiungono E all'albero
ADE ha etc

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	infinity	infinity
1	AD	2,A	4,D		2,D	infinity
2	ADE	2,A	3,E			4,E
3	ADEB		3,E			4,E
4	ADEBC					4,E
5	ADEBCF					

dunque A, userà sempre i link colorati in rosso



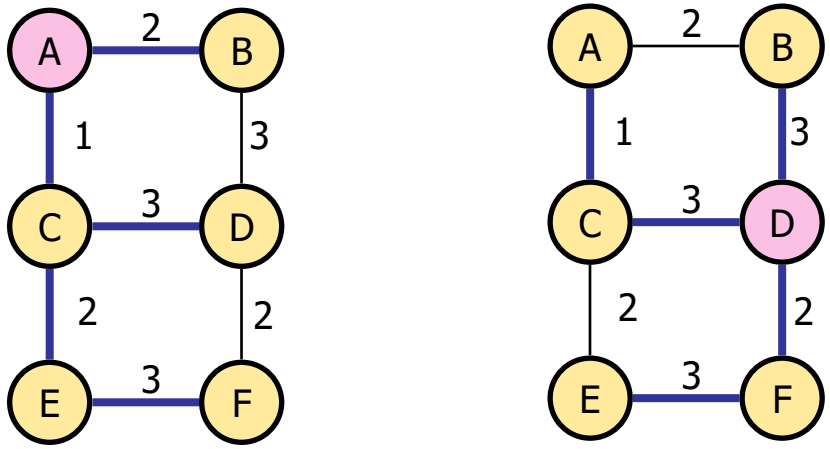
destination	Next hop, cost
B	B,2
C	D,3
D	D,1
E	D,2
F	D,4

Routing tree

ogni nodo ha gli stessi valori di partenza ma andrà a fare i propri calcoli quindi ogni router costruisce il proprio albero (non si ha il problema che si creava a livello 2 con lo spanning tree dove avevamo tutti lo stesso percorso)

- Each node has the same LS database
  - However, each node has a different routing tree to the destinations
    - Spanning Tree Protocol (L2 networks): all the nodes share the same routing tree
    - Better distribution of the traffic (reasonably, no unused links)
  - Obviously, the routing tree is consistent between different nodes

--> USO EFFICIENTE DELLA INFRASTRUTTURA DI RETE



PROs

- Rapid Convergence
  - LSs spread quickly
  - No intermediate processing

--> converge velocemente
-LS inviati velocemente in flooding --> prpagazione veloce anche perchè il pacchetto non va processato e modificato come nel DV ma va solo inoltrato a tutti e poi

- Limited Routing Traffic and Storage
  - Link states are small
  - Fast and efficient neighbor greeting

i pacchetti sono più piccoli in quanto ogni nodo manda informazioni solo sui vicini, non sulla rete (anche vero che però tutti inviano a tutti quindi DIPENDE)

- It rarely generates loops
- Simple to understand and troubleshoot
  - All nodes have identical databases

NEIGHBOR GREETING: nodi si scambiano dei messaggi di Hello all'accensione e in maniera periodica in modo tale che i router vicini si scoporono a vicenda

CONs

- High implementation complexity
  - Selective flooding
  - First implementation took several years
- Protocols with complex configuration

configurazione non molto facile in quanto

RIP implementava l'altro

OSPF protocollo che implementa LinkState

Link State Generation

- In principle: when there is a topology change
  - Allows a better utilization of the network
    - Does not consume network bandwidth

--> non si invia ogni tot di tempo, i pacchetti si generano solo in caso di cambiamenti nella rete

- Actual protocols: LS are generated periodically (with very low frequency)
  - Increased reliability
    - What about if a LS, for some reasons, is lost?

ogni 30 minuti si manda per far fronte ad eventuali perdite

per farci un'idea (RIP ogni 5 minuti, OSPF ogni 30 minuti)

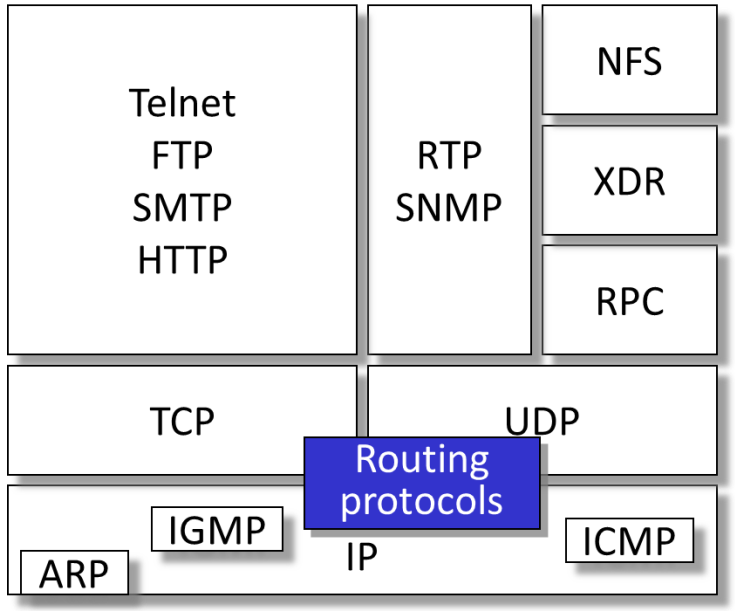
Internet Routing Architecture

Routing Protocols

tra lv3 e lv4 --> usano IP per mandare pacchetti ma non sono proprio protocolli di livello trasporto

- Protocols that allow routers to exchange information on the network and determine the best route to each destination
  - Uses routing algorithms
- Define metric(s) and their encoding in packets
- Specific timing
- Problem: what is the operational domain of the protocol?
  - A subnet?
  - An operator's network?
  - A country?
  - The Internet at large?

si definisco dei domini amministrativi che godono di una certa autonomia



Making routing scalable

Scale: with billions of destinations:

- can't store all destinations in routing tables!
- routing table exchange would swamp links!
- our routing description thus far:
  - idealized
  - all routers identical
  - network "flat"
  - … not true in practice

administrative autonomy

- internet = network of networks
- each network admin may want to control routing in its own network

Autonomous System

- A set of subnets grouped based on
  - Topology
  - Organizational criteria
- E.g., the subnets of a large internet service provider
- Administration
  - Autonomous internal routing choices
  - Negotiated external routing choices
- Scalability
  - Not all information propagated everywhere

io voglio che ci sia scalabilità --> devo valutare quando grande sono capace di gestirla

- Identification: AS number
  - Two bytes
  - Assigned by IANA

ogni autonomous System è identificato da 2 byte dallo IANA

Internet approach to scalable routing

internet: insieme di tanti Autonomous system che parlano tra loro (magari uno con RIP e l'altro con OSPF)

Aggregate routers into "autonomous systems" (AS) (a.k.a. "domains")

avrò due processi:

- Intra-AS routing routing interno al mio dominio
  - routing among hosts, routers in same AS ("network")
  - all routers in AS must run same intra-domain protocol
- routers in different AS can run different intra-domain routing protocol
- gateway router: at "edge" of its own AS, has link(s) to router(s) in other AS'es

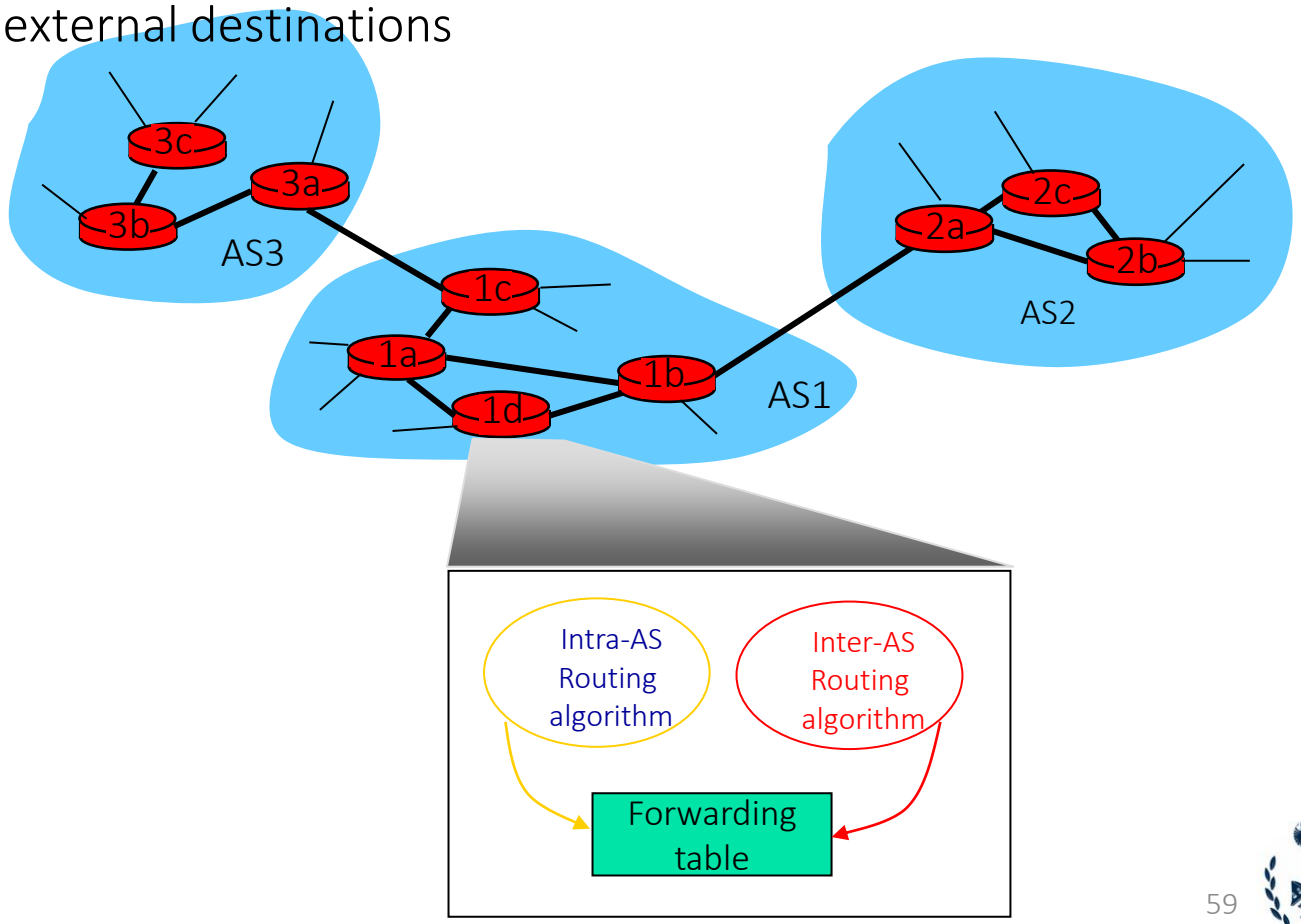
- Inter-AS routing routing tra domini
  - routing among AS'es
  - gateways perform inter-domain routing (as well as intra-domain routing)

mi serve un link che comunica tra i vari domini

Interconnected ASes

- Forwarding table configured by both intra- and inter-AS routing algorithm
  - intra-AS routing determine entries for destinations within AS
  - inter-AS & intra-AS determine entries for external destinations

con inter as routing -->scopre rotte interne
con intra --> scopre rotte esterne



Internet Routing Architecture

Intra-AS Routing

Inter-AS tasks

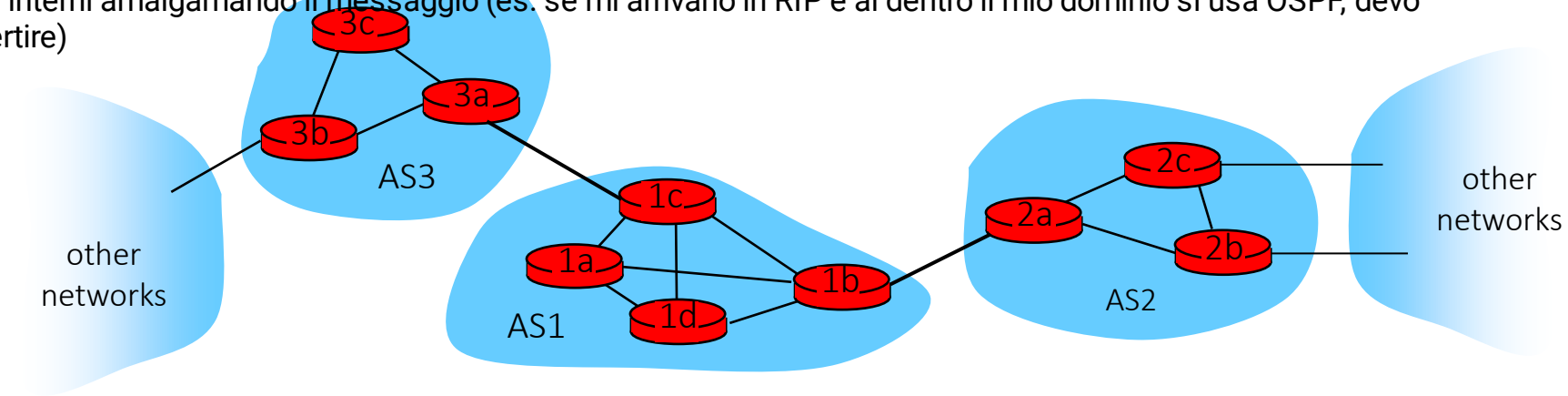
- suppose router in AS1 receives datagram destined outside of AS1:
  - router should forward packet to gateway router, but which one?

AS1 must:

1. learn which destinations are reachable through AS2, which through AS3
2. propagate this reachability info to all routers in AS1

job of inter-AS routing!

una volta che le informazioni sono state immesse, l'informazioni viene propagata
--> i router di frontiera ricevono per primi l'informazione oltrre-confine e poi fanno una redistribuzione verso i router interni amalgamando il messaggio (es: se mi arrivano in RIP e al dentro il mio dominio si usa OSPF, devo convertire)



Intra-AS Routing

- also known as *interior gateway protocols (IGP)*
- most common intra-AS routing protocols:
  - RIP: Routing Information Protocol (obsolete)
  - OSPF: Open Shortest Path First (IS-IS protocol essentially same as OSPF)
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary for decades, until 2016)

OSPF (Open Shortest Path First)

- "open": publicly available --> protocollo aperto
- uses link-state algorithm
  - link state packet dissemination
  - topology map at each node
  - route computation using Dijkstra's algorithm
- router floods OSPF link-state advertisements to all other routers in entire AS
  - carried in OSPF messages directly over IP (rather than TCP or UDP)
  - link state: for each attached link
- IS-IS routing protocol: nearly identical to OSPF
  link-state

OSPF “advanced” features

implementa algoritmo link state

- security: all OSPF messages authenticated (to prevent malicious intrusion)
- multiple same-cost paths allowed (only one path in RIP) più percorsi che raggiungono la stessa destinazione
- for each link, multiple cost metrics for different TOS (e.g., satellite link cost set low for best effort ToS; high for real-time ToS) permette di avere varie metriche di costi: -nrHop -Delay -BandWidth
- integrated uni- and multi-cast support:
  - Multicast OSPF (MOSPF) uses same topology database as OSPF
- Hierarchical OSPF in large domains -->ogni router ha la mappa completa della topologia --> OSPF usa gerarchia (next slide)

3) in base a cosa sto facendo mi potrebbe interessare una cosa o un'altra e quindi grazie a OSPF posso usare la metrica più idonea --> posso usare magari costo 1 per best effort, costo 100 per real time. OSPF capisce che ci sono i due costi su quel link. Quando arriva il pacchetto come posso fare il forwarding? --> RoutingTable ha destinazione | NextHop | Costo --> per avere questi costi diversi devo avere 2 RoutingTable (una per BestEfford, una per RwalTime).
-->POLICY ROUTING: adattiamo il routing in base a delle policy

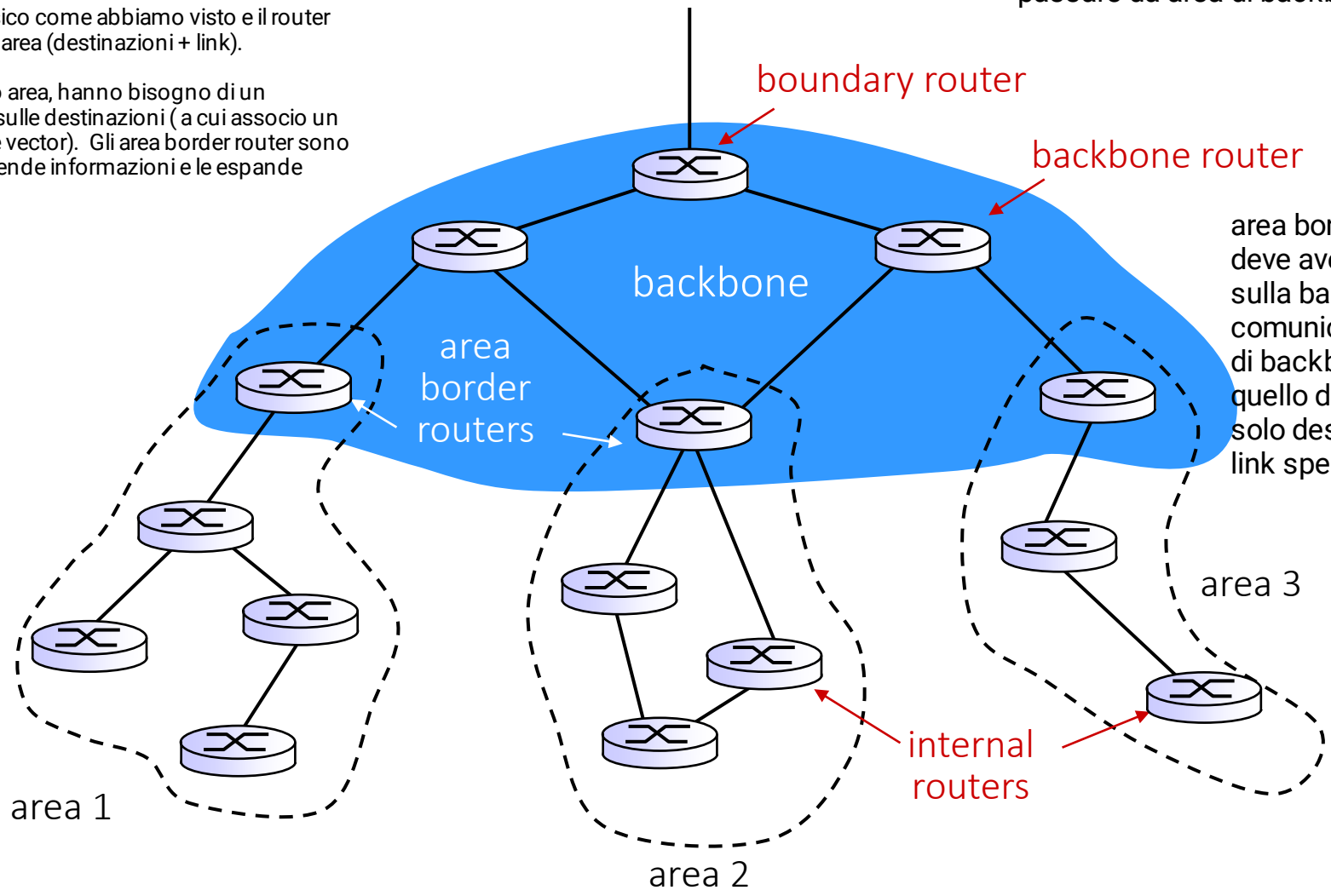
Hierarchical OSPF

divide il dominio totale in 3 aree:
-area di bacbone
-area border routers
-

area di backbone: tutta il traffico tra aree deve passare tramite l'area di blacbone (il traffico tra area1 e area2 deve a prescindere da tutto passare da area di backbone.

dentro ogni area abbiamo un ospf classico come abbiamo visto e il router conosce bene la topologia della propria area (destinazioni + link).

Per tutto quello che è al di fuori della loro area, hanno bisogno di un riassunto --> il router ha le informazioni sulle destinazioni (a cui associo un costo) ma non dei link (simile a distance vector). Gli area border router sono router presenti in entrambe le aree --> prende informazioni e le espande all'interno dell altre aree.

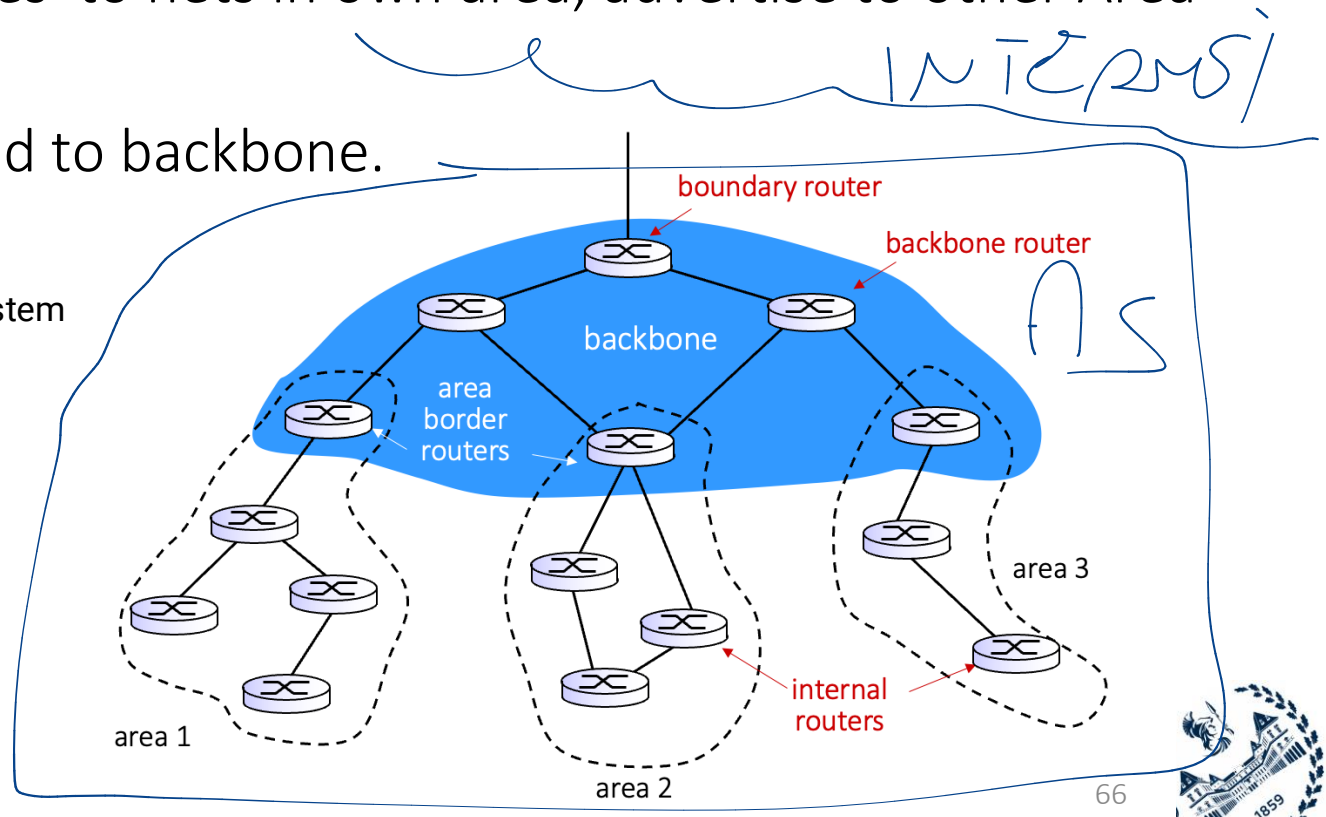


area border router: ogni area deve avere un'area border router sulla backbone.
comunica ad area 1 tutto quello di backbone e a backbone tutto quello di area 1 --> comunica solo destinazioni e costo senza link specifici e topologia

Hierarchical OSPF

- two-level hierarchy: local area, backbone.
  - link-state advertisements only in area
  - each nodes has detailed area topology; only know direction (shortest path) to nets in other areas.
- area border routers: "summarize" distances to nets in own area, advertise to other Area Border routers.
- backbone routers: run OSPF routing limited to backbone.
- boundary routers: connect to other AS'es

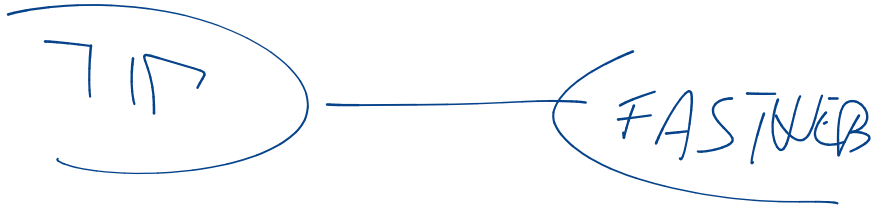
boundary routers: permettono di far comunicare diversi autonomous system



Internet Routing Architecture

Inter-AS Routing

Why different Intra-, Inter-AS routing ?



policy:

- inter-AS: admin wants control over how its traffic routed, who routes through its net.
- intra-AS: single admin, so no policy decisions needed

dato che a parlare sono dei competitors, c'è bisogno di policy
–> ogni Auton. System vuole avere il controllo dic osa si annuncia, come avviene il controllo e come si propaga nella propria rete
–> quando si parla tra diversi A.S., non vogliono far sapere tutto agli altri (Tim non vuole far sapere tutto a fastWeb, non voglio comunicarti dove e come ho dei guasti)
–> un amministratore vuole quindi controllare quello che si dice e quello che gli viene detto

scale:

- hierarchical routing saves table size, reduced update traffic

ok gerarchico ma se facciamo OSPF su tutta l'area, non scala più --> c'è bisogno di sommarizzazione

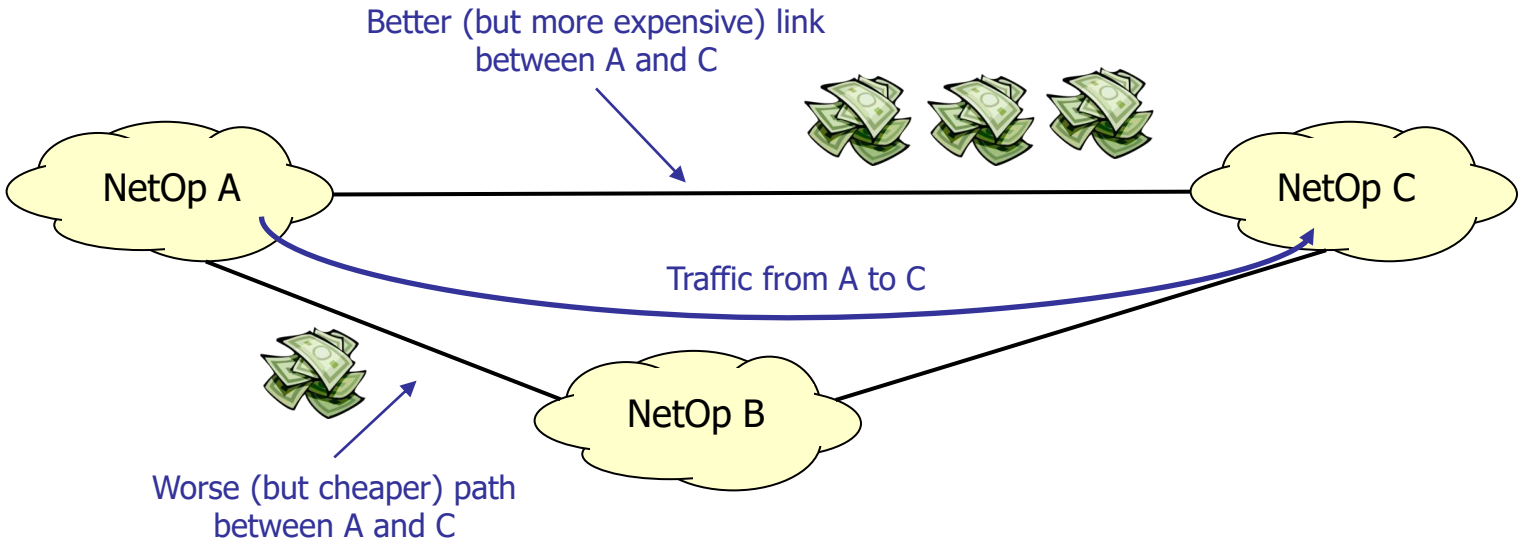
performance:

- intra-AS: can focus on performance
- inter-AS: policy may dominate over performance

--> la performance può non essere la cosa che più mi interessa, potrei dar priorità ad aspetti economici, sicurezza...
(magari il collegamento diretto mi costa 30dollari, un collegamento più lungo mi costa 1 dollaro) -->cazzomene, prendo il più economico
(es su slide seguente)

Example: economic reasons

- A network operator may prefer a longer path because of economic reasons



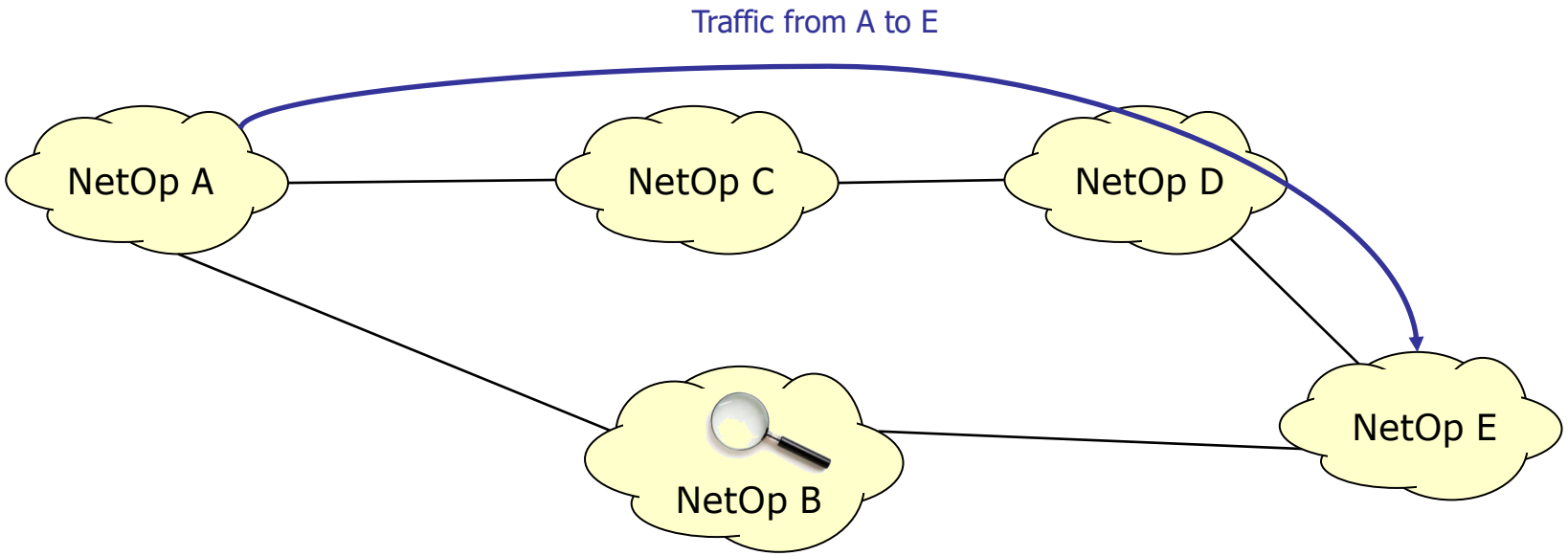
Sometimes we may prefer longer (but **cheaper**!) paths to best paths

in OSPF si parlava solo di banda, non potevi considerare denaro

Example: security reasons

non mi fido di B, si fa troppo i fatti miei, passo da C che è più affidabile
--> questa policy va settata dall'amministatore

- A network operator can represent a security threat for the traffic of another network operator
  - Network operator A would like to avoid its traffic toward E to go through B, as that network operator is not trusted



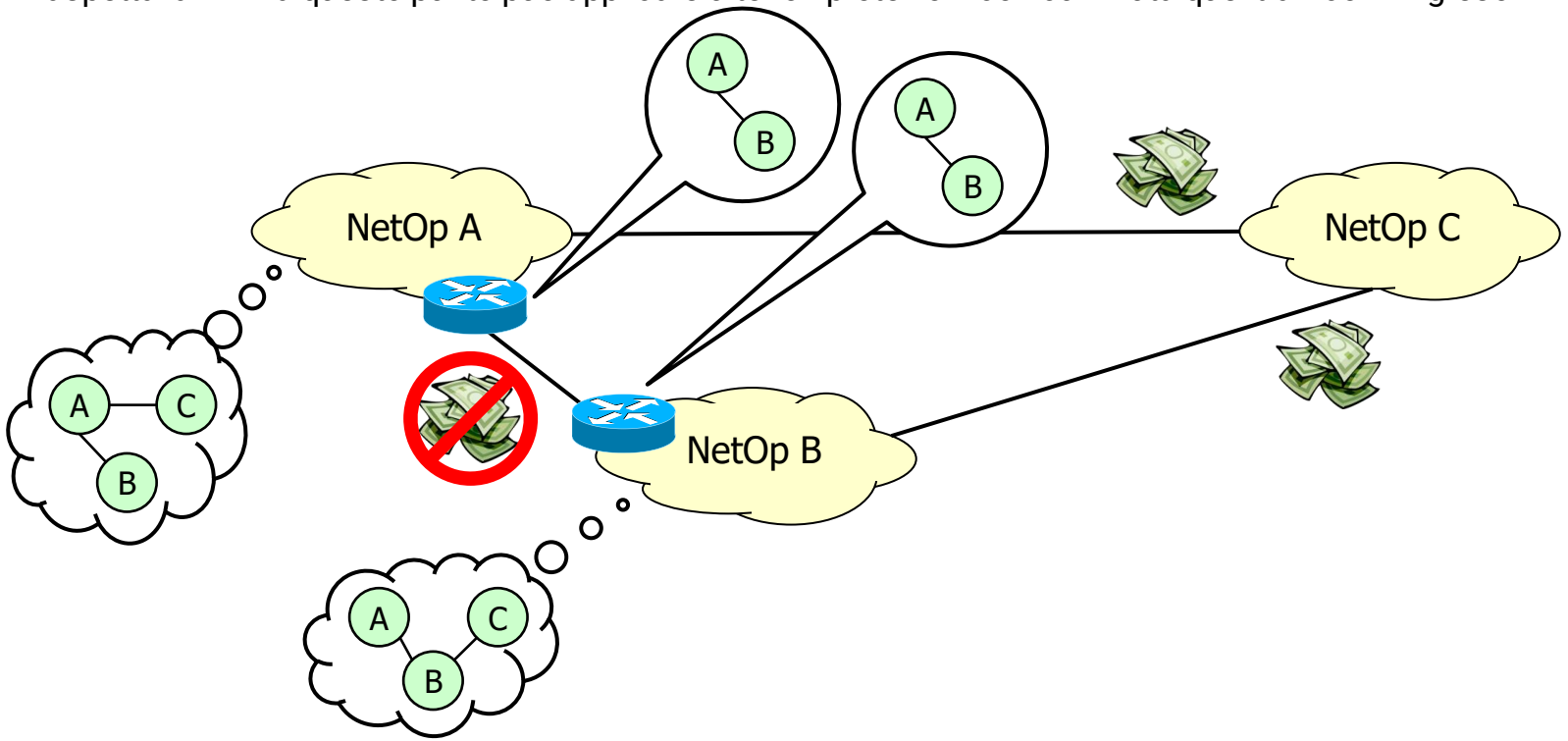
Sometimes we may prefer longer (but **safer**!) paths to best paths

Example: route hiding

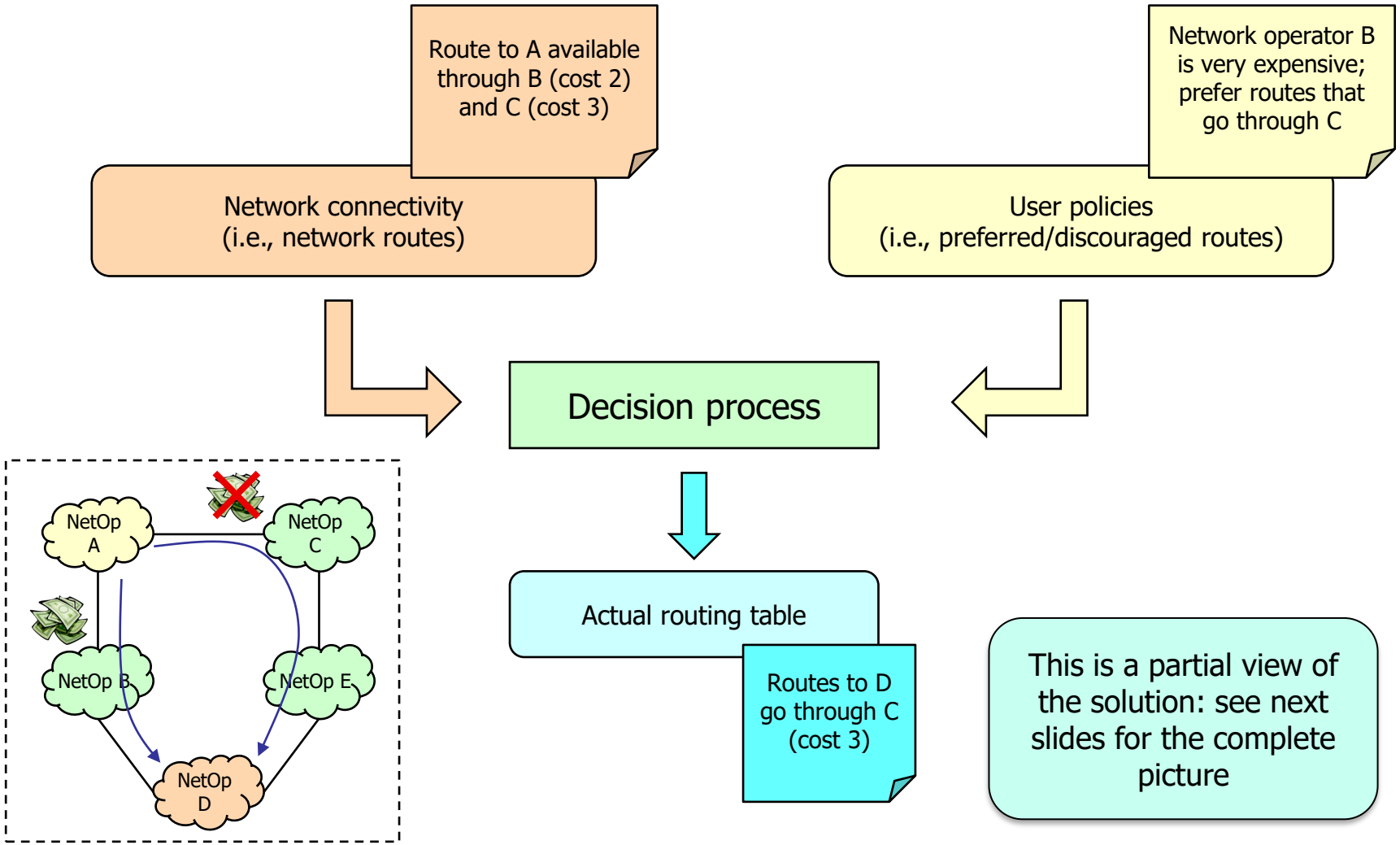
io, operatore B, ho un bel link con operatore C ma non voglio comunicarlo ad A perchè non voglio trasportare il suo traffico
--> ad A comunica che è collegato solo con lui e non con C

- Inter-AS routing protocol can propagate “partial” connectivity
  - Some paths are not propagated to the other party
- In any case, routers must be protected from actual traffic through appropriate ACLs

A però è un grande stronzo, manda comunque il traffico a B per parlare con C. B si trova a dover trasportare del traffico che non si aspettava --> B a questo punto può applicare ulteriori protezioni con cui rifiuta quel traffico in ingreso.



Solution: a new class of routing protocols

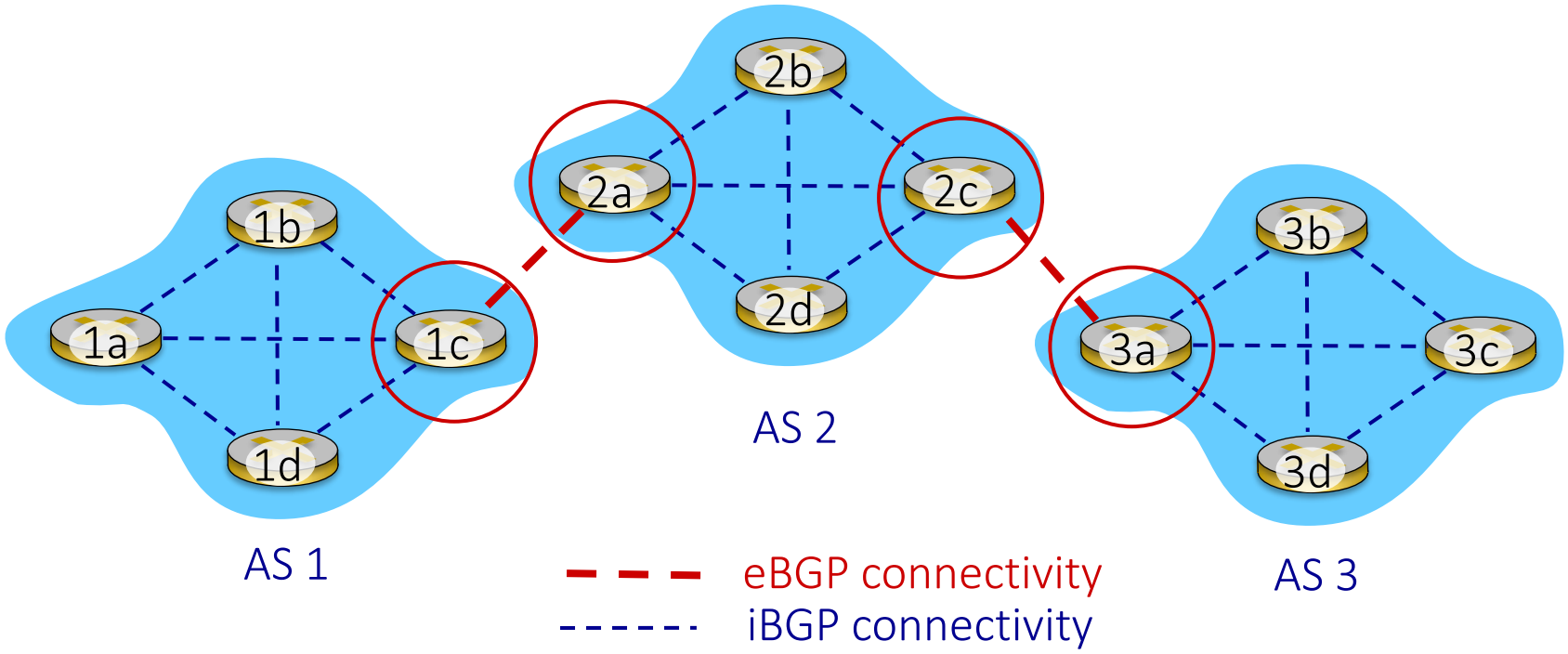


Internet inter-AS routing: BGP

protocollo di frontiera che tiene internet unita in quanto permette di fare connettività tra tutti.
diventato lo standard in quanto usato da tutti

- BGP (Border Gateway Protocol): the de facto inter-domain routing protocol
  - "glue that holds the Internet together"
- BGP provides each AS a means to: permette ad ogni A.S. di ottenere informazioni sui peopri vicini.
  - obtain subnet reachability information from neighboring ASes (eBGP) e: exterior --> due router di due A.S. diversi che si scambiano informazioni
  - propagate reachability information to all AS-internal routers (iBGP) i: interior –> tra i router all'interno della rete che parlano BGP (router che sono collegati ad altri A.S. NON TRA TUTTI I ROUTER ALL'INTERNO DELLA RETE –>alcuni router parlano solo OSPF)
  - determine "good" routes to other networks based on reachability information and policy
- Not necessarily shorter path
  - Choice based on policies
  - Reflect agreements among ASs
- allows subnet to advertise its existence to rest of Internet: "I am here"
- destinations can be aggregated
  - 195.1.2.O/24 and 195.1.3.O/24 can be announced as 195.1.2.O/23

eBGP, iBGP connections



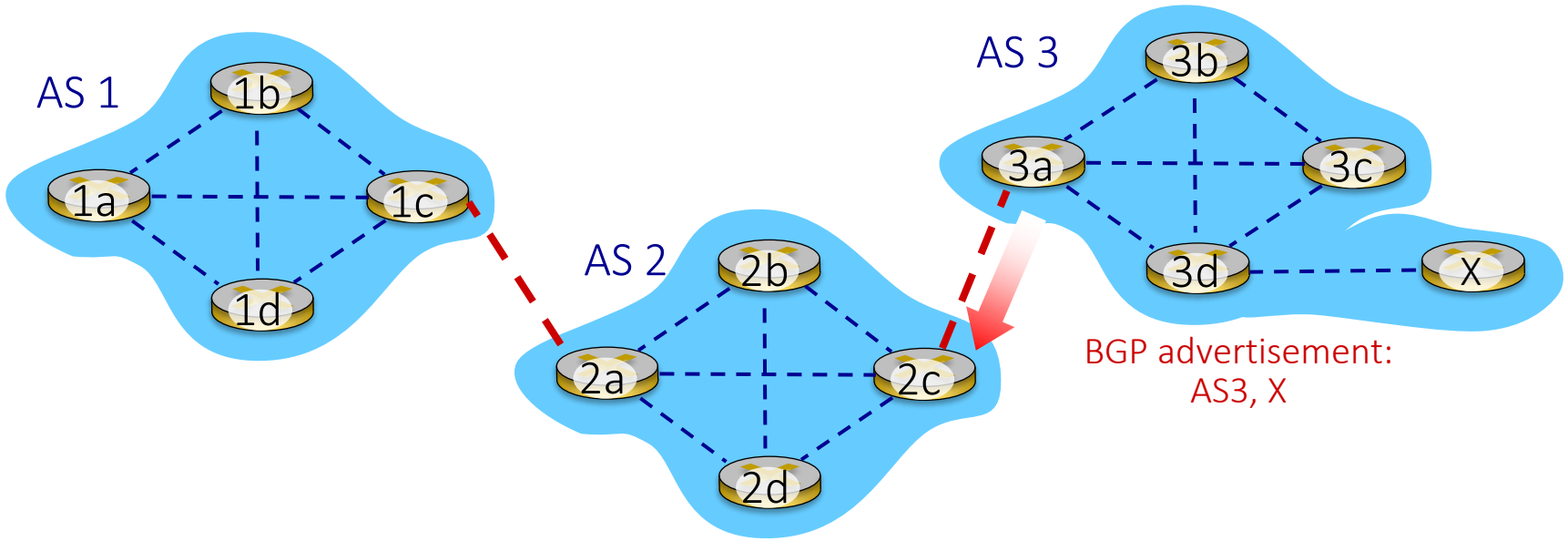
gateway routers run both eBGP and iBGP protocols

BGP basics

peers: router che parlano BGP --> lo scambio di BGP avviene su connessione TCP

si usa il PATH vector (dest | cost | HOP) dove gli HOP sono la lista degli A.S.

- BGP session: two BGP routers ("peers") exchange BGP messages over semi-permanent TCP connection:
  - advertising paths to different destination network prefixes (BGP is a "path vector" protocol)
- when AS3 gateway router 3a advertises path AS3,X to AS2 gateway router 2c:
  - AS3 promises to AS2 it will forward datagrams towards X



Path attributes and BGP routes

- advertised prefix includes BGP attributes
  - prefix + attributes = “route”
- two important attributes: ogni pacchetto BGP ha due campi all'interno del protocollo:
  - AS-PATH: list of ASes through which prefix advertisement has passed lista degli AS attraverso cui è passato
  - NEXT-HOP: indicates specific internal-AS router to next-hop AS router per parlare con il prossimo AS
- Policy-based routing:
  - gateway receiving route advertisement uses import policy to accept/decline path (e.g., never route through AS Y).
  - AS policy also determines whether to advertise path to other other neighboring ASes

BGP messages

si scambiano alcuni pacchetti:
-UPDATE: ogni AS fa update in caso di nuova destinszione
-KEEPALIVE: per vedere se il peer è sempre presente

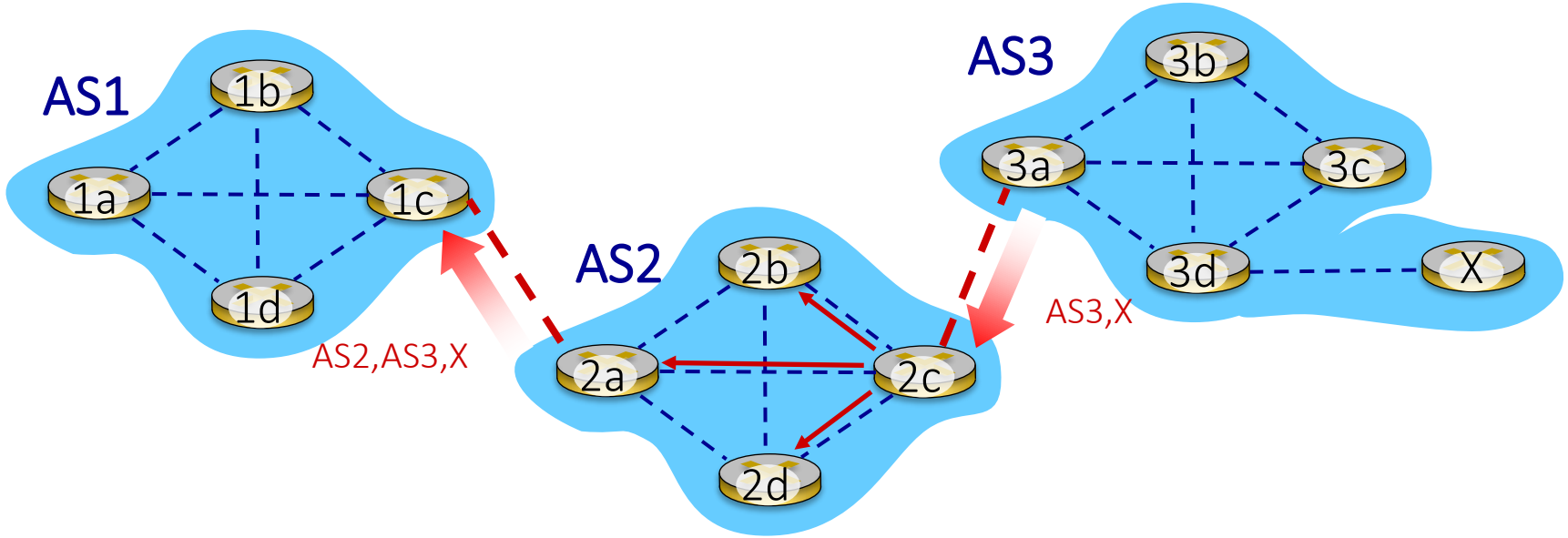
- BGP messages exchanged between peers over TCP connection
- BGP messages:
  - OPEN: opens TCP connection to remote BGP peer and authenticates sending BGP peer
  - UPDATE: advertises new path (or withdraws old)
  - KEEPALIVE: keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - NOTIFICATION: reports errors in previous msg; also used to close connection

nota: amministratore setta a mano i peer,(non in automatico)

vista male

BGP path advertisement

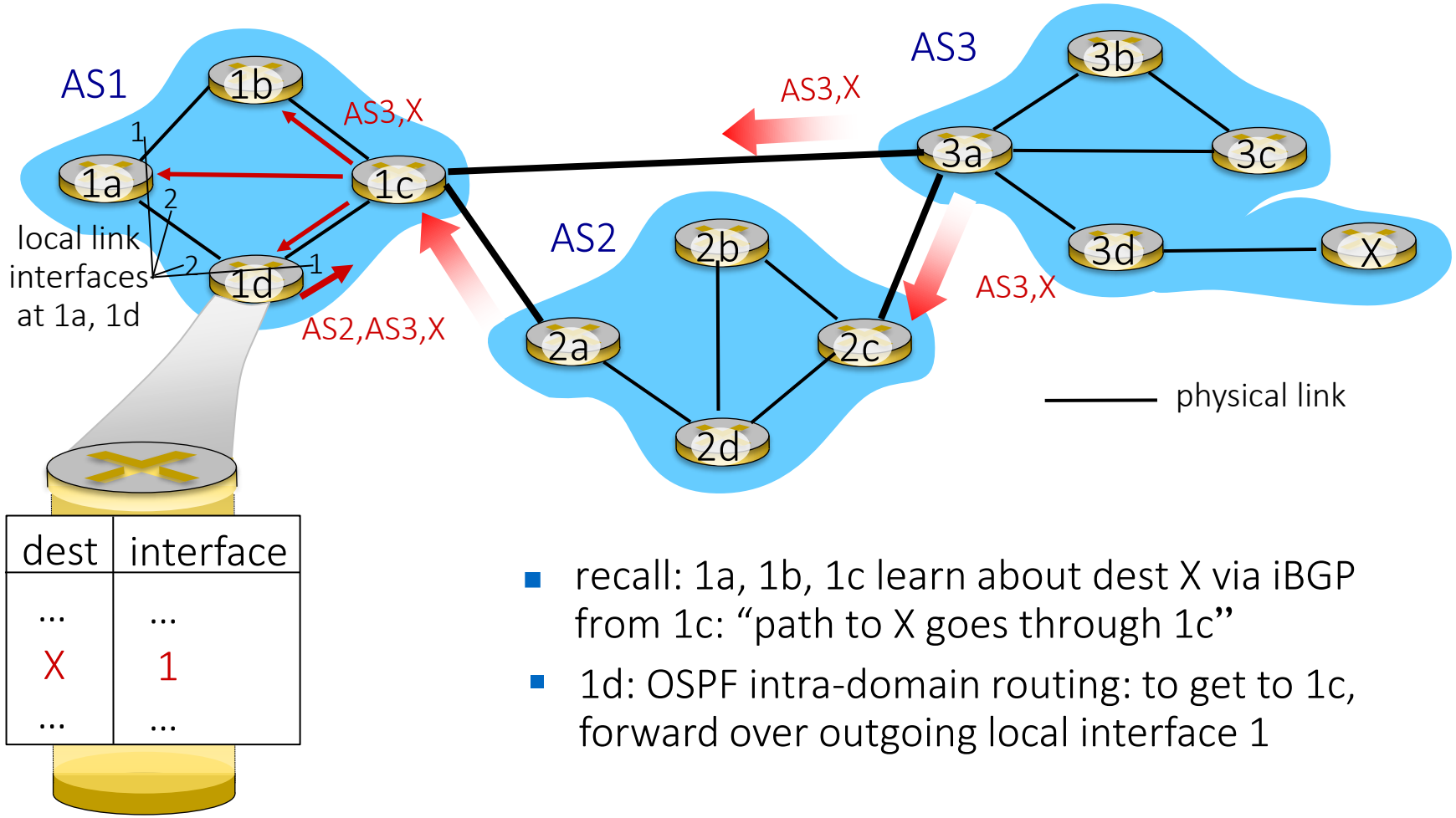
- AS2 router 2c receives path advertisement AS3,X (via eBGP) from AS3 router 3a
- Based on AS2 policy, AS2 router 2c accepts path AS3,X, propagates (via iBGP) to all AS2 routers
- Based on AS2 policy, AS2 router 2a advertises (via eBGP) path AS2, AS3, X to AS1 router 1c



BGP, OSPF, forwarding table entries

in questo caso 1c, comunica la nuova informazione : posso raggiungere X autonomamente senza passare da AS2

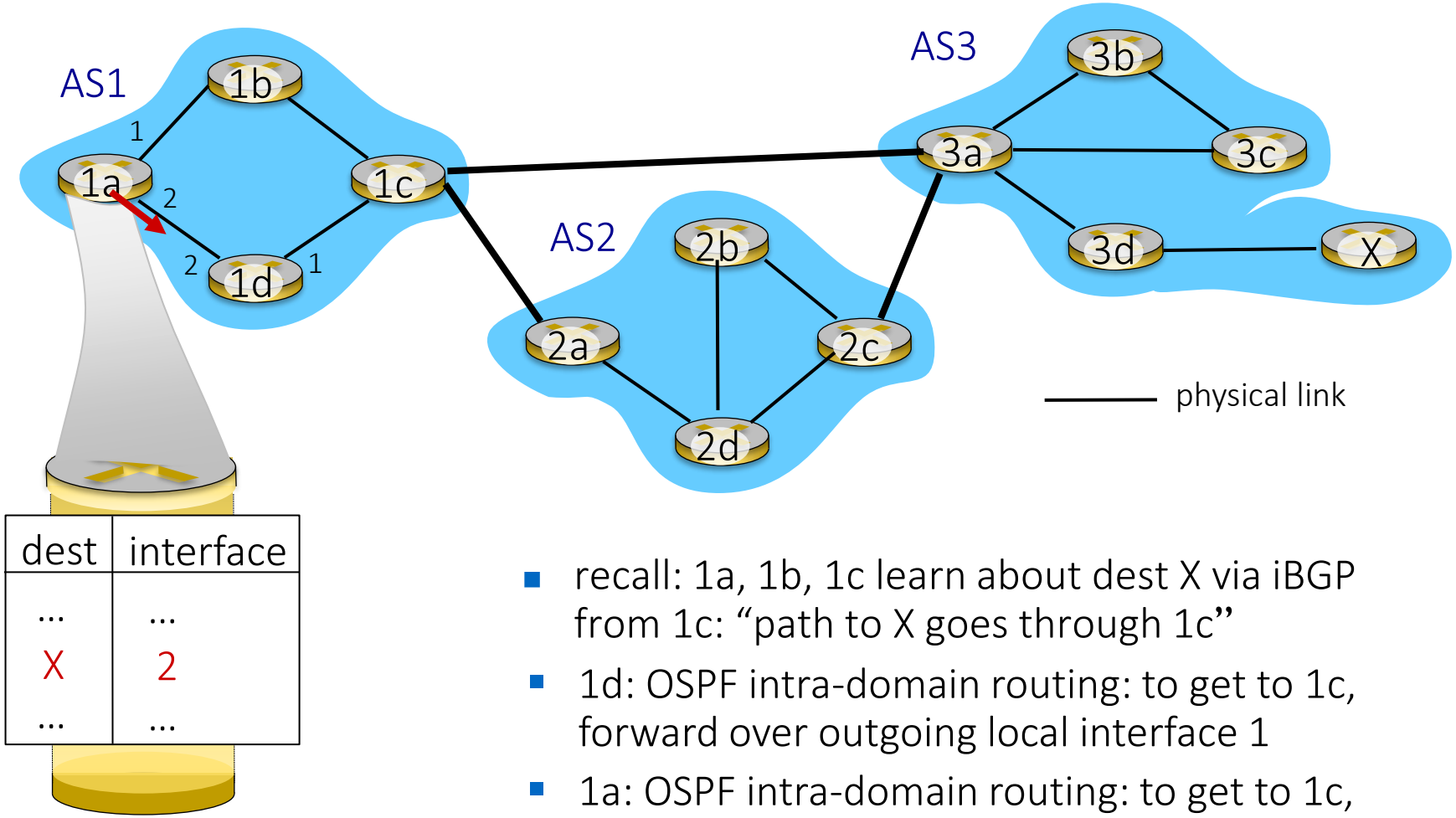
Q: how does router set forwarding table entry to distant prefix?



- recall: 1a, 1b, 1c learn about dest X via iBGP from 1c: "path to X goes through 1c"
- 1d: OSPF intra-domain routing: to get to 1c, forward over outgoing local interface 1

BGP, OSPF, forwarding table entries

Q: how does router set forwarding table entry to distant prefix?

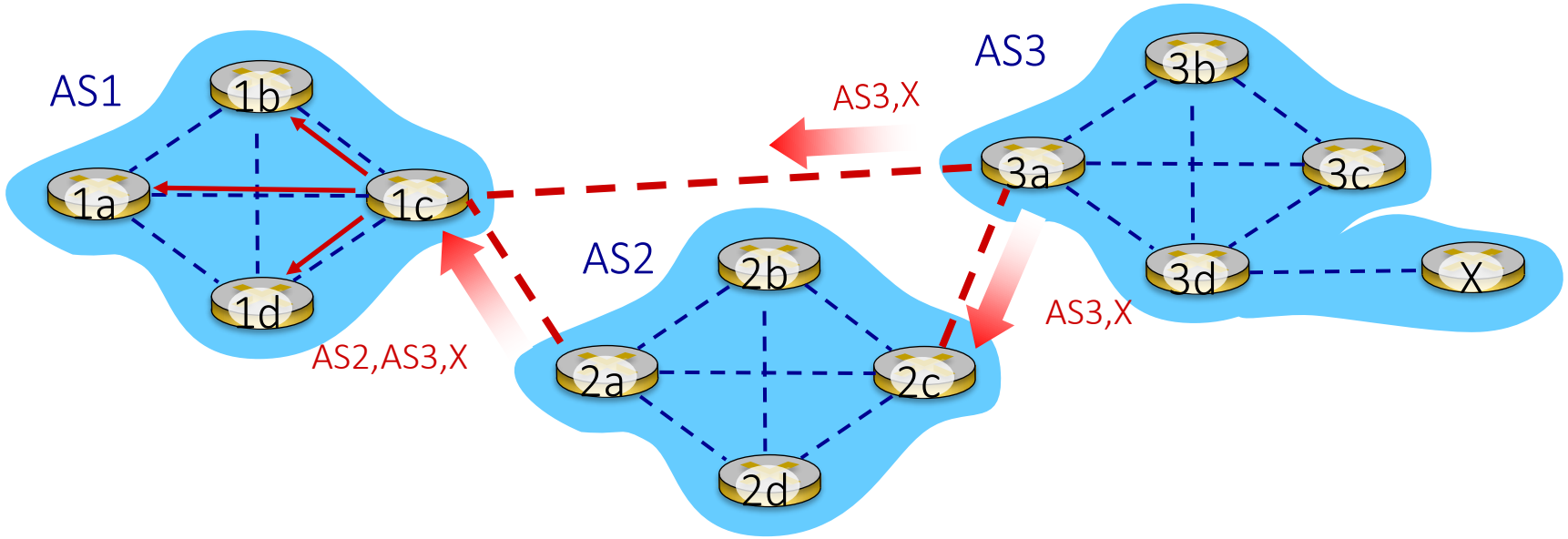


- recall: 1a, 1b, 1c learn about dest X via iBGP from 1c: "path to X goes through 1c"
- 1d: OSPF intra-domain routing: to get to 1c, forward over outgoing local interface 1
- 1a: OSPF intra-domain routing: to get to 1c, forward over outgoing local interface 2

BGP path advertisement

1c riceve l'informazione di X sia da AS3 che da AS2:
-se ci sono policy settate, sceglie in base alle policy
altrimenti:
-scelgo percorso che ha meno AS -->slide dopo

- Gateway router may learn about multiple paths to destination:
  - AS1 gateway router 1c learns path AS2,AS3,X from 2a
  - AS1 gateway router 1c learns path AS3,X from 3a
  - Based on policy, AS1 gateway router 1c chooses path AS3,X, and advertises path within AS1 via iBGP



BGP route selection

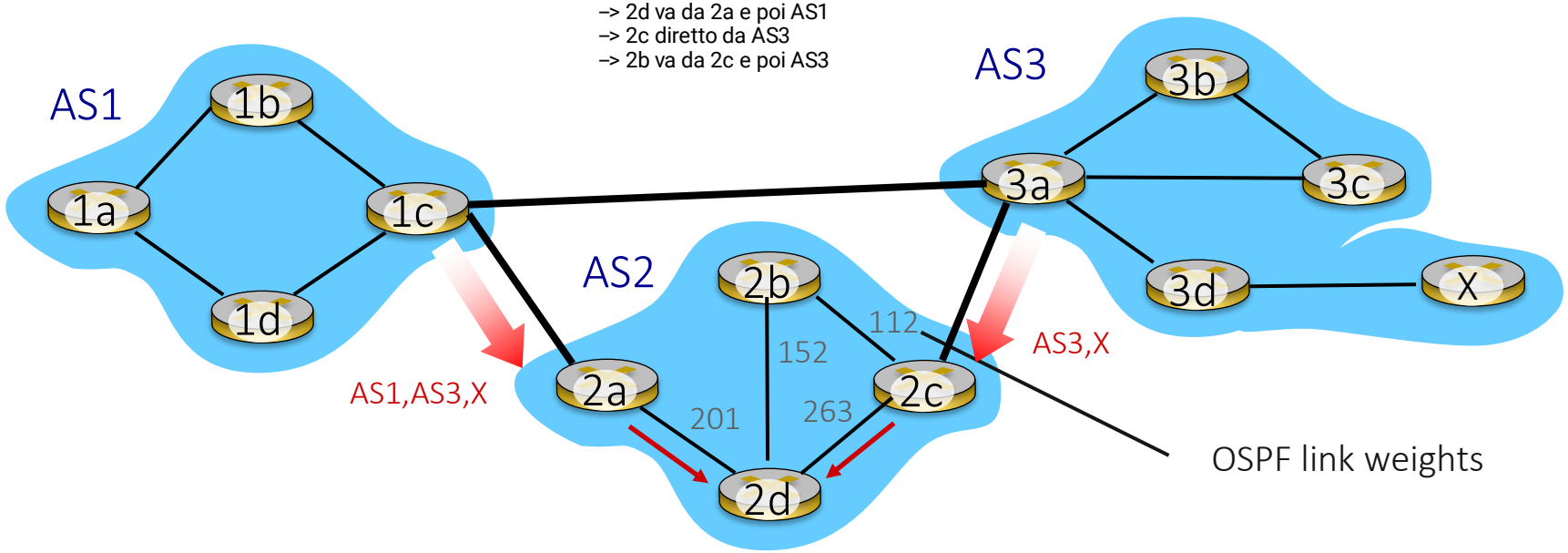
- Router may learn about more than one route to destination AS, selects route based on:
  1. local preference value attribute: policy decision
  2. shortest AS-PATH
  3. closest NEXT-HOP router: hot potato routing
  4. additional criteria : invento io

Hot Potato Routing

ogni AS ha dei costi interni settati da OSPF

AS2 riceve due informazioni su x
–> 2a lo riceve da AS1 che dice che lo raggiunge con AS1-AS3
–>2c lo riceve da AS3 che dice che lo raggiunge attraverso AS3

HOT POTATO: io, router, prendo l'exit point più vicino
–>2a esce da AS1
–> 2d va da 2a e poi AS1
–> 2c diretto da AS3
–> 2b va da 2c e poi AS3



- 2d learns (via iBGP) it can route to X via 2a or 2c
- *hot potato routing:* choose local gateway that has least intra-domain cost (e.g., 2d chooses 2a, even though more AS hops to *X*): don't worry about inter-domain cost!
- BGP would however select (AS3, X) based on criterium 2

BGP Connection Types

connessione:
-tipo transito: ---------- (operatore per trasportare il traffico, solitamente, si fa pagare)
-tipo peering: ---------- inoltre solitamente gratutità (non si scambiano soldi per il trasporto di traffico)

- there are two types of interconnections:

- Transit: an ISP can provide reachability to the entire Internet for another endpoint (e.g., enterprise, content or application provider, residential broadband provider, etc.).
  - The endpoint entity pays the ISP to carry traffic to and from the Internet.
- Peering: The networks interconnect to exchange only traffic that originates or terminates within their networks (including the networks of their customers, in the case of carriers and Tier 1 networks)

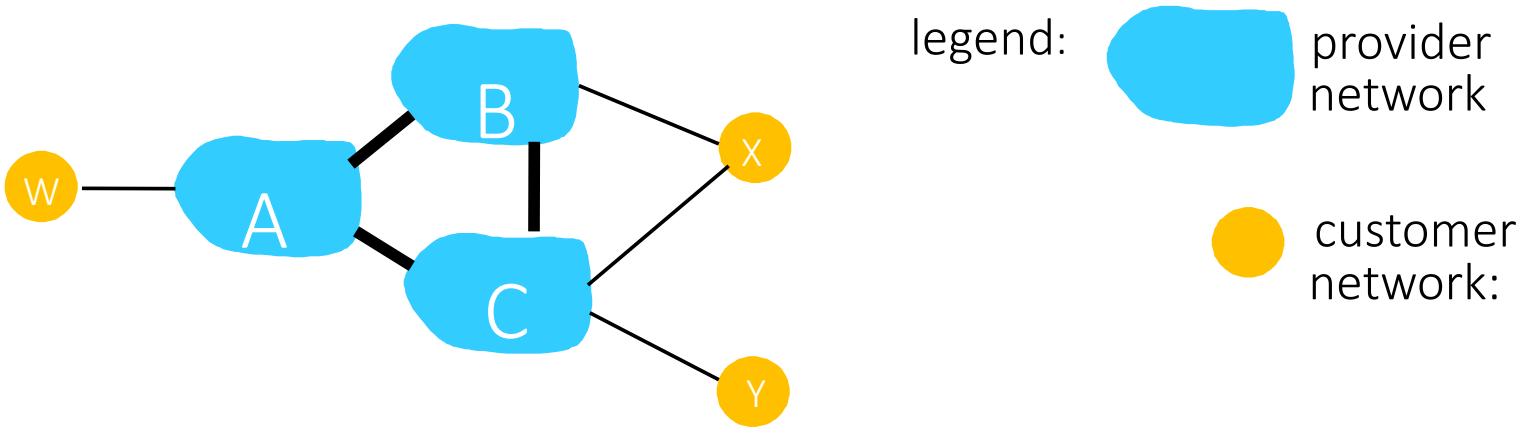
solo per traffico che ha origine o termina in quella specifica rete

  - Public peering through an internet exchange point (IXP) -> Peering with *multiple* networks
  - Private peering: networks interconnect to exchange only traffic that originates or terminates within their networks

ognuno può arrivare agli IXP

BGP: achieving policy via advertisements

- Let us look at a case of *Private Peering*
  - A,B,C are provider networks
  - X,W,Y are customer (of provider networks)
  - X is dual-homed: attached to two provider networks
  - Policy to enforce: X does not want to route from B to C via X -> Private peering
  - .. so X will not advertise to B a route to C



BGP: achieving policy via advertisements

- Let us now look at a case of *Peering Agreements* among ISPs
  - A advertises path Aw to B and to C
  - B chooses not to advertise BAw to C:
  - B gets no "revenue" for routing CBAw, since none of C, A, w are B's customers
  - C does not learn about CBAw path
  - C will route CAw (not using B) to get to w
- Rule of thumb: traffic flowing across an ISP's backbone network must have either a source or a destination (or both) in a network that is a customer of that ISP;