Routing Essentials

Tecnologie e Servizi di Rete

Computer Network Technologies and Services

Mario Baldi, Claudio Casetti, Guido Marchetto, Fulvio Valenza

*part of the slides adapted from material by Jim Kurose

Copyright Notice

- This set of transparencies, hereinafter referred to as slides, is protected by copyright laws and provisions of International Treaties. The title and copyright regarding the slides (including, but not limited to, each and every image, photography, animation, video, audio, music and text) are property of the authors specified on page 1.
- The slides may be reproduced and used freely by research institutes, schools and Universities for non-profit, institutional purposes. In such cases, no authorization is requested.
- Any total or partial use or reproduction (including, but not limited to, reproduction on magnetic media, computer networks, and printed reproduction) is forbidden, unless explicitly authorized by the authors by means of written license.
- Information included in these slides is deemed as accurate at the date of publication. Such information is supplied for merely educational purposes and may not be used in designing systems, products, networks, etc. In any case, these slides are subject to changes without any previous notice. The authors do not assume any responsibility for the contents of these slides (including, but not limited to, accuracy, completeness, enforceability, updated-ness of information hereinafter provided).
- In any case, accordance with information hereinafter included must not be declared.
- In any case, this copyright notice must never be removed and must be reported even in partial uses.

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

find a route from source to destination at network layer, Using the layer 3 header.

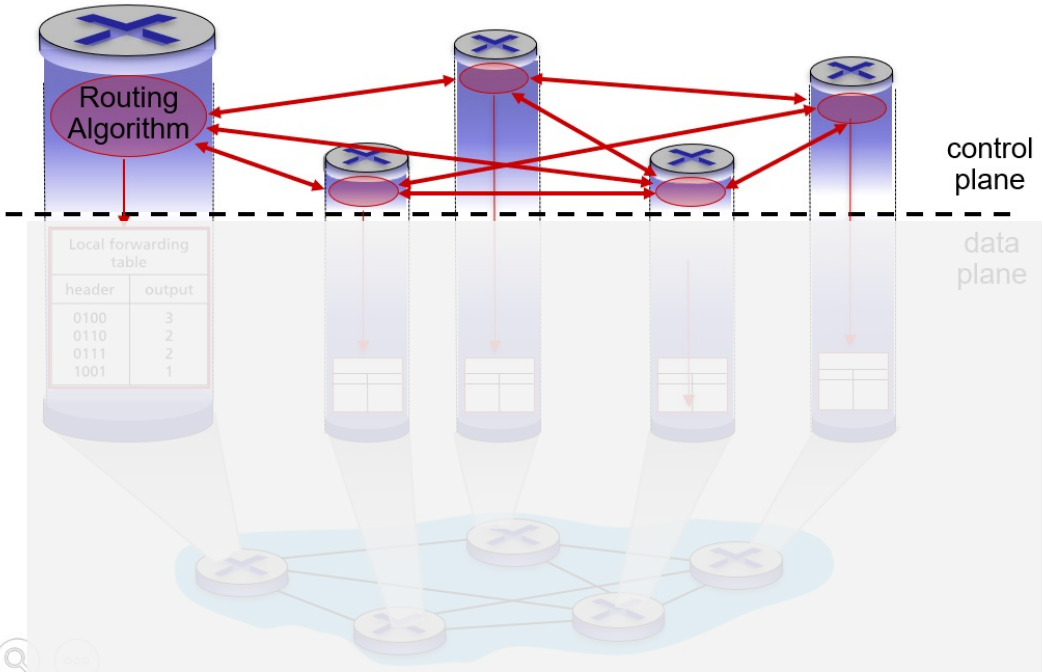
- Determination of (optimal) path from source to destination for packet transmission in a computer network

find a path on the topology based on some criteria

learn the weights that we need to run the routing algorithm

- Based on routing algorithms and protocols
- It is a *control plane* operation

they don't involve the active exchange of user data.



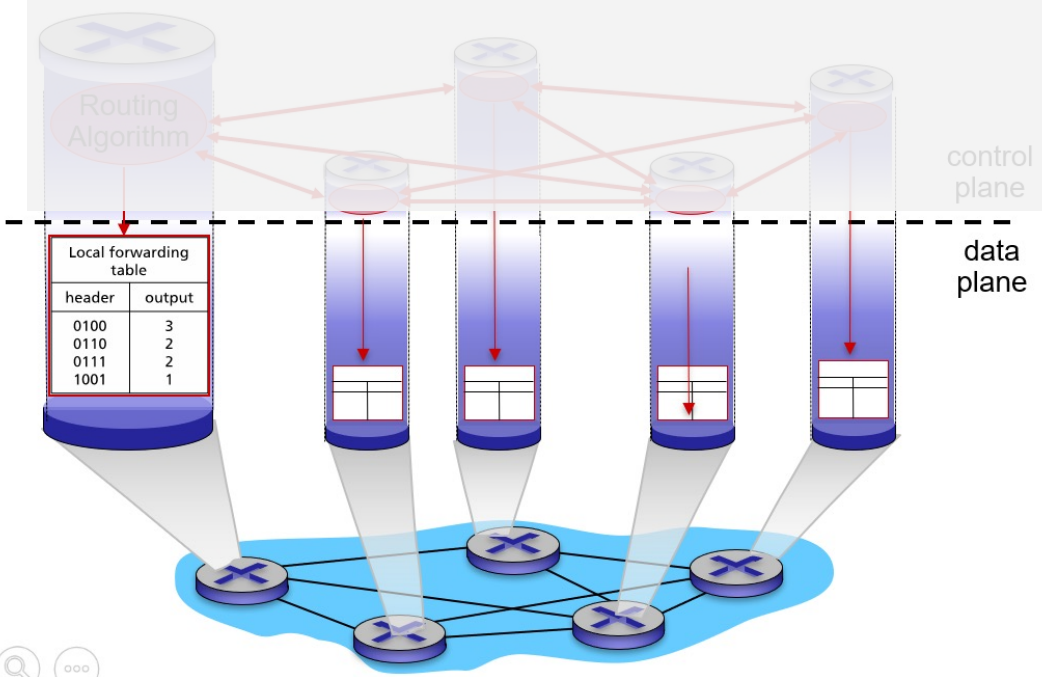
Forwarding

inoltro

receive a packet, looking at a possible destination among a list of possible destination+ matching finding the longest match. and then sending the packet to the port that have that address.

- 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

carry data.



Forwarding Procedures

there are a lot of forwarding procedures.

1) forwarding by network address, with the longest match.

- Forwarding by Network Address (most common)
- Label Swapping it speeds up the operation procedure.
  - e.g., MPLS (see later)
- Source Routing (rarely used) the source decide what path it want its packet to follow and it writes the path in the header.

Forwarding Phases

- Next-hop/output port selection (using routing tables)

  i look at the routing table, and first i select the next hop, in which output i have to choose.
- Switching: transfer to output port internal procedure of the router.
- Transmission last part the packet leaves the router.

Routing Algorithm Classification

- Non-adaptive algorithms (static) part of the control plane operation, the result of a routing protocol to compute the routes.
- Adaptive algorithms (dynamic)

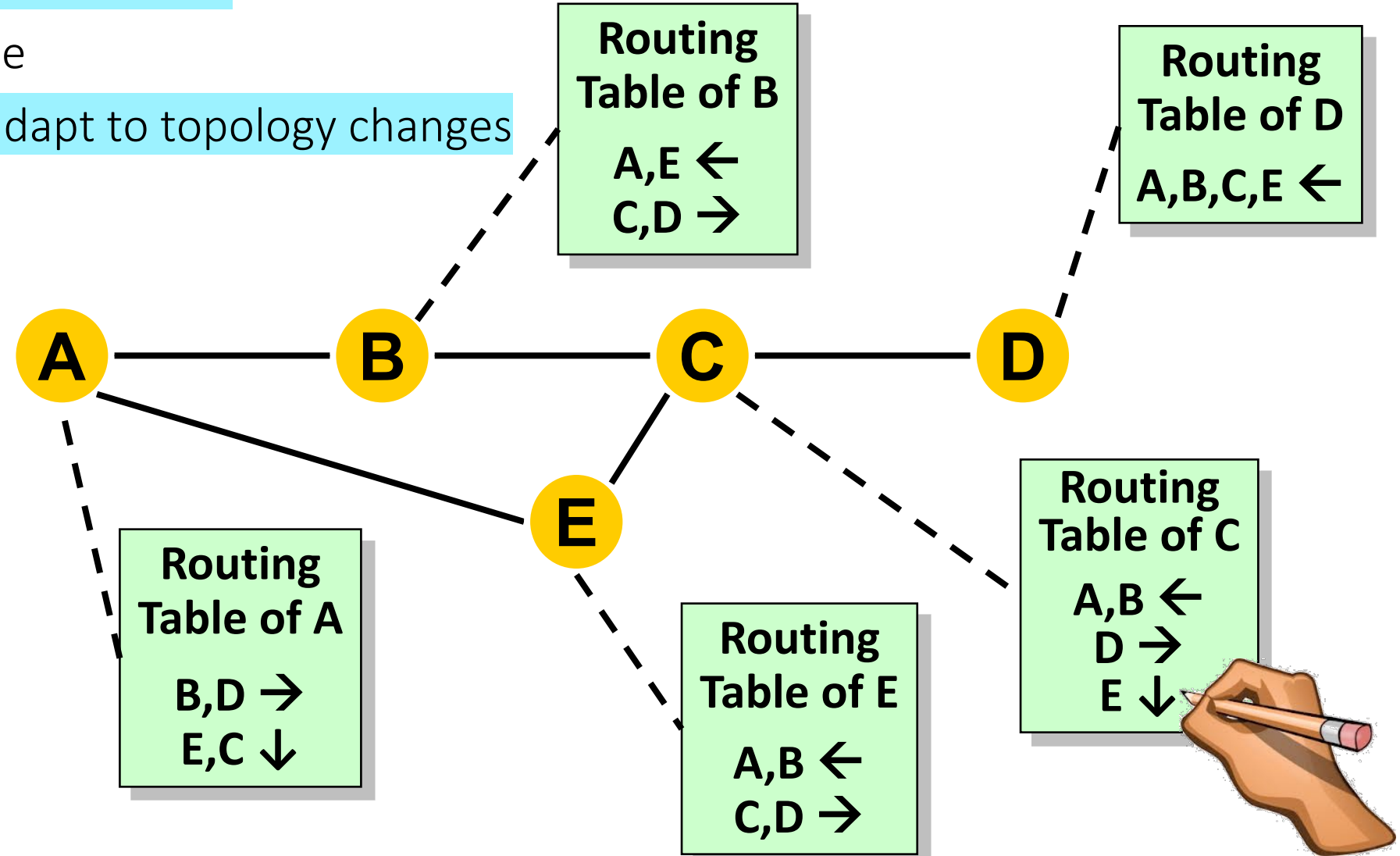
Non-adaptive Algorithms

usually the result of manual configuration, for home network.

- Fixed Directory routing
  - Static routing
  - Manual configuration
- (Selective) flooding and similar they not depend on an explicit state of the network
  - Random on a random output port forward the packet. Sooner or later like a ball in a flipper it will reach the destination
    - Randomly select an output port
  - Flooding i forward on any output port, used by advanced algorithms when u don't know yet the topology of the network. carefull to not create a broadcast storm (loop).
    - Forward on any output port (except the one on which the PDU was received)
  - Deflection or «hot potato» on the port that has the shortest queue, shortest waiting, but we risk to create loop between two ports.
    - 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



Adaptive Algorithms

react to changes to the network. changes like a new node being added and other things like that..

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

Centralized Routing

starting from the topology, we have a centralized routing control center, that is a node that receive information from all the routers, about topology, and has a global picture of the topology and computes the path, for everyone.so it creates a routing table for everyone.

- One node (Routing Control Center, RCC) collects info from all other nodes
- Computes the path
- Distributes to all nodes

13

Centralized Routing: PROs

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

Centralized Routing: CONs

- RCC is single point of failure
- RCC is bottleneck
  - Significant network load in proximity of RCC
- Not suitable for highly dynamic networks too much overhead

Isolated Routing

- Each node decides independently they learn the topology and decide where they can send the packet trough backward learning
- No exchange of information
- E.g., Backward Learning
  - IEEE 8O2.1D switches

Distributed Routing

used nowadays in the internet.

- Routers co-operate in exchanging connectivity information
- Each router decides independently, but coherently

if a packet is send to a node and the node gives use back, cause they have taken indipendet decision we create a loop.

- Combines advantages of isolated and centralized routing
- Two big families:
  - Distributed Routing with Partial Information
    - Distance Vector algorithms
  - Distributed Routing with Global Information
    - Link State algorithms

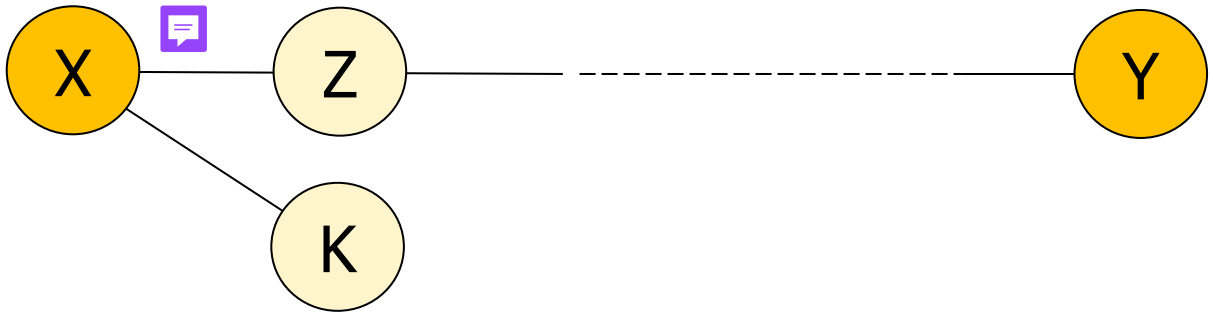
WE CREATE a TREE and all nodes use it . It combines advantages of isolated routing with the one of centralized routing.

Distance Vector Algorithm

- 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
- 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



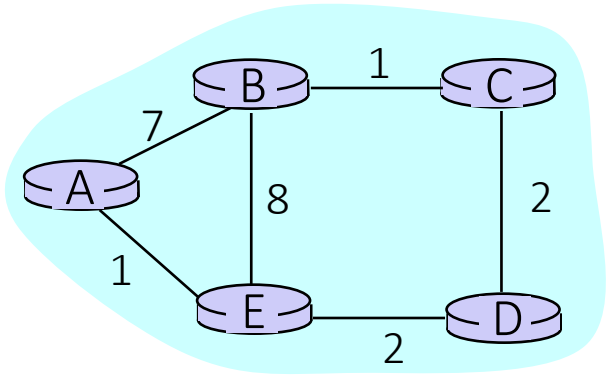
Source

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

Destination

Next hop

Distance Table: example



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

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

$$D^E(A,B) = c(E,B) + \min_w \{D^B(A,w)\}$$
$$= 8+6 = 14$$ loop!

Cost to .. Via ...

neighbour

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

destination

distance table of router E

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

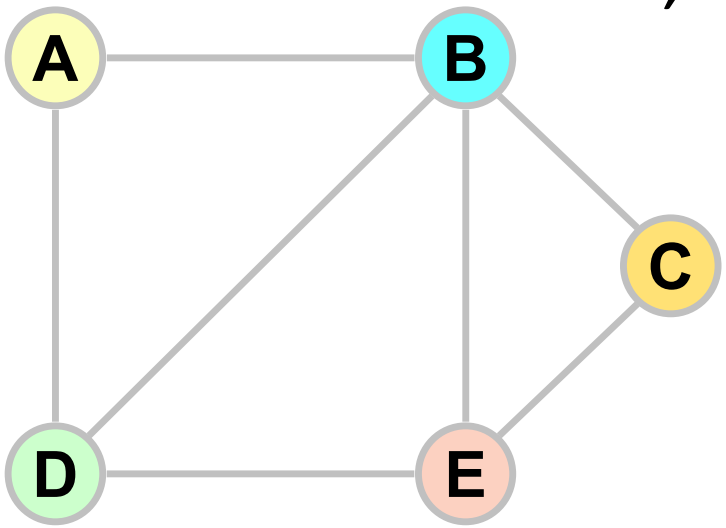
Next Hop, Cost

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

Routing Table

Sample Scenario

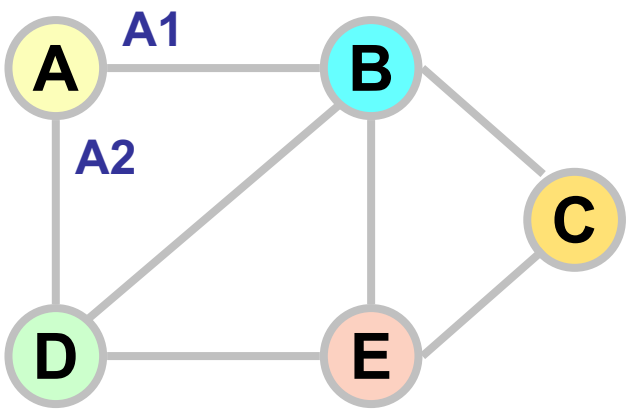
two main structure, distance table and routing table.



Routing information stored by A

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

Distance Vector Merging and Generation



Received from line A1

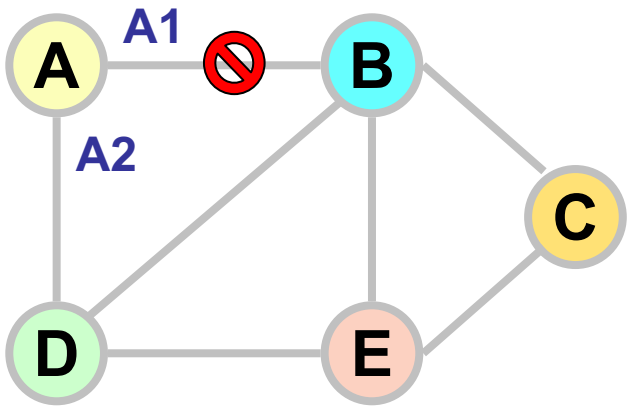
Received from line A2

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

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

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

Topology Change



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

Void!!

ROUT. TABLE (A)		
A,	local,	0
B,	A2,	2
C,	A2,	3
D,	A2,	1
E,	A2,	2

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

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	

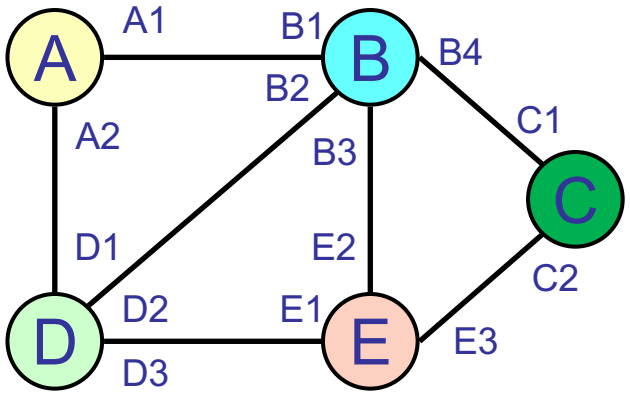
B and D send their DVs

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



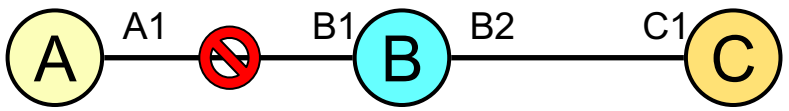
Issues

- Black Hole attack, all the nodes believes to the malitious nodes and the net is broken. unless we authenticate.
  - DoS attack by malicious node advertising impossibly short, non-existent routes to attract traffic, which is then discarded
- Count to infinity the counter is modified and this lead to a bouncing effect
  - 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

'c' doesnt know that 'a' is not reachable anymore. 'b' will believe thanks to 'c', that 'a' is reachable to interface 'c', so increase the counter to reach 'A', the same thing 'C', and so on and so far.



the worst thing is there is a huge loop, so if during this time a packet needs to be sent to a we would have a bouncing effect too.

B:

Loc (B)
B, 0

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

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

RT (B)
A,B2 ,3
B,loc,0
C,B2 ,1

C:

Loc (C)
C, 0

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

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

B sends DV

Loc (C)
C, 0

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

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

C sends DV

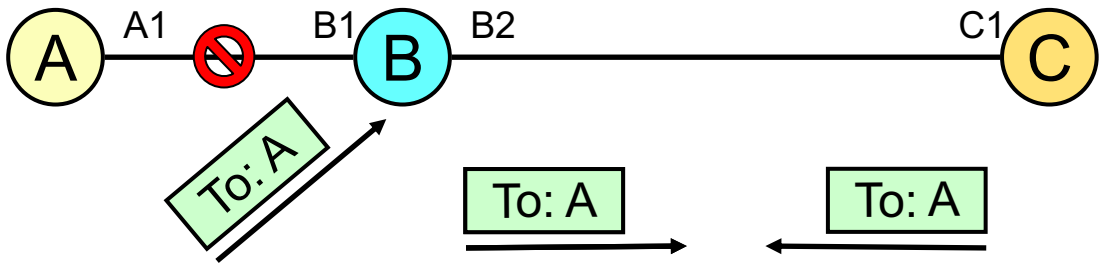
Loc (B)
B, 0

DV (C)
A, 4
B, 1
C, 0

RT (B)
A,B2 ,5
B,loc,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

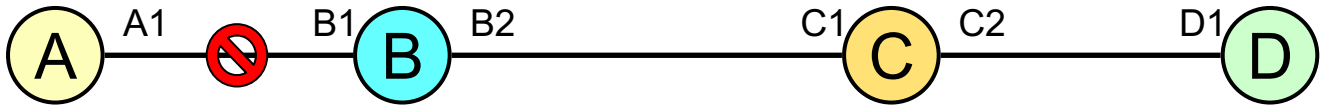
Loc (C)	DV (B)	RT (C)
C, 0	A, 3	A,C1 ,4
	B, 0	B,C1 ,1
	C, 1	C,loc,0

(Partial) solutions

- 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" B will not try to reach A trough C.



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

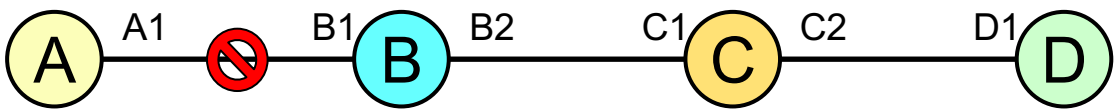
DV (C→C2)
A, 2
B, 1
C, 0

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)

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:

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

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

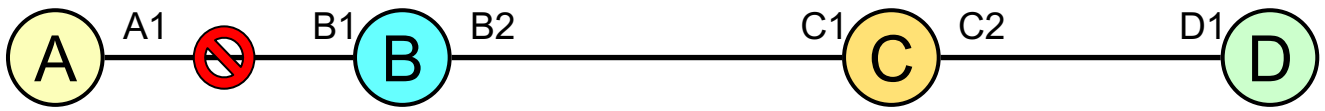
- The router that noted the fault may not participate to any loop at least until the timeout of Hold Down timer

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
- 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



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)

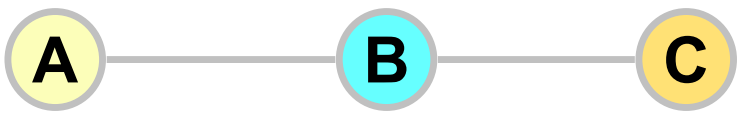
A,	inf
B,	inf
C,	0
D,	1

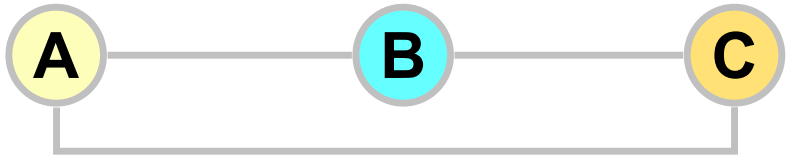
RT (B)

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

Bottom Line

- PROs:
  - Simple to implement and deploy
  - Very little configuration
- CONs:
  - Exponential worst-case complexity and convergence time $O(n^2)$
  - Routers do not know the network topology loops and infinity counter.
  - e.g., B cannot distinguish between:



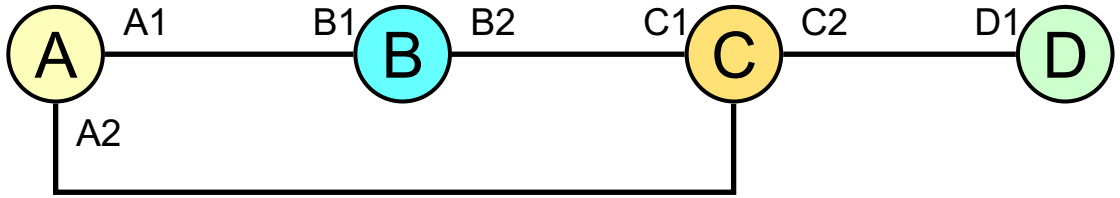


  - Convergence limited by slower links and routers set pace
  - Complex tuning and troubleshooting
  - Large routing traffic (and storage)
    - Not suitable for large complex networks

Path Vector

save also the path trough we arrive to the destination.

- Nodes exchange dynamically-updated path information
- Easy to detect loops in the vector no count to infinity.
- Increased overhead



A:

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]

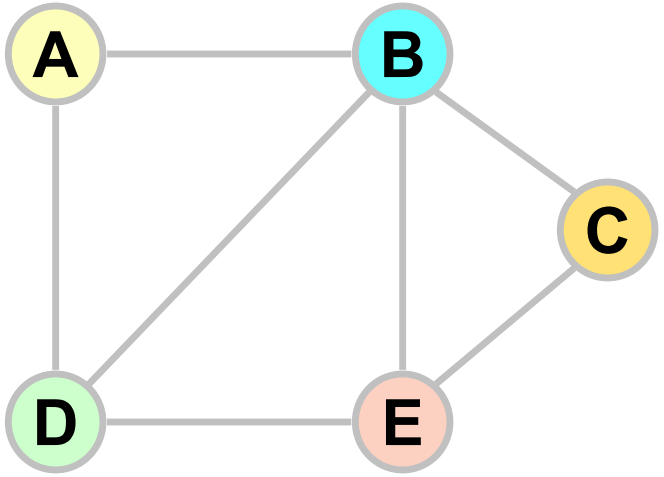
Link State algorithm

dynamic routing algorithm who react to changes of the topology, trying to find alternative paths.

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

Link State Database

thanks to the Flooding algorithm. writes down the id of the packet he just forwarded, so that if he receive again this packet nows there is a loop.



Stored by each Node

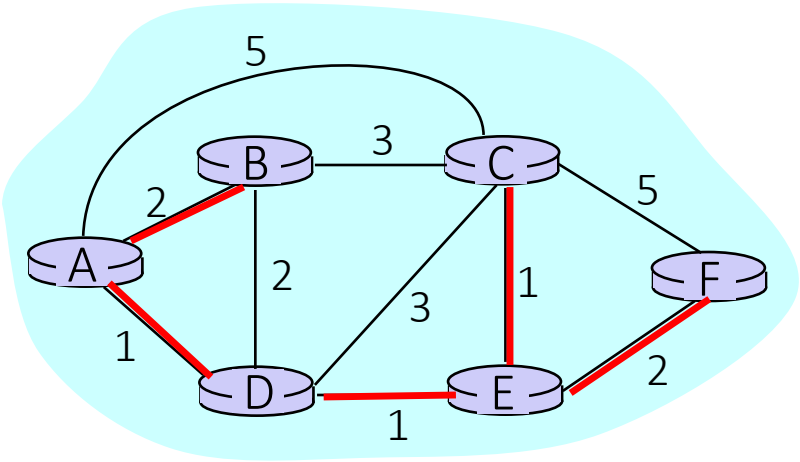
LS (A)	LS (B)	LS (C)	LS (D)	LS (E)
B, 1	A, 1	B, 1	A, 1	B, 1
D, 1	C, 1	E, 1	B, 1	C, 1
	D, 1		E, 1	D, 1
	E, 1			

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

Dijkstra: example

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					



Next hop, cost

destination	
B	B,2
C	D,3
D	D,1
E	D,2
F	D,4

PROs

- Rapid Convergence
  - LSs spread quickly
  - No intermediate processing
- Limited Routing Traffic and Storage
  - Link states are small
  - Fast and efficient neighbor greeting
- It rarely generates loops
- Simple to understand and troubleshoot
  - All nodes have identical databases

CONs

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

Link State Generation

- In principle: when there is a topology change in theory.
- Actual protocols: LS are generated periodically in reality
  - Increased reliability

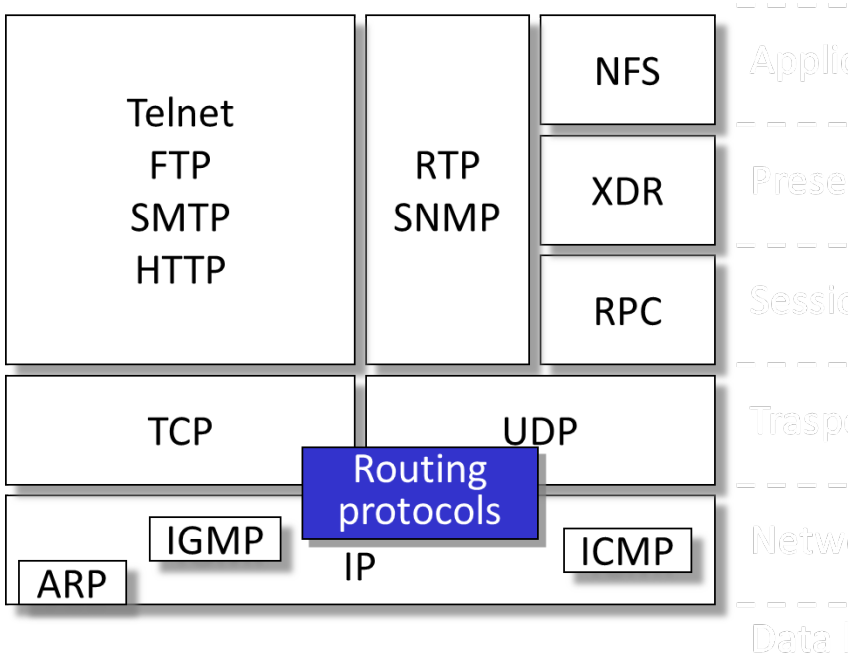
Internet Routing Architecture

Routing Protocols

which protocols we use in the internet.

- 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?

  how far the protocol can reach ?
  - A subnet?
  - An operator's network?
  - A country?
  - The Internet at large? NOT THIS.



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

we need to making routing scalable.

administrative autonomy

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

Autonomous System = ISP, 90 % is true, has its own internat management,

- A set of subnets grouped based on the same
  - Topology
  - Organizational criteria
- E.g., the subnets of a like large internet service provider
- Administration
  - Autonomous internal routing choices
  - Negotiated external routing choices island in an arcipelagus.
- Scalability
  - Not all information propagated everywhere
- Identification: AS number
  - Two bytes
  - Assigned by IANA

Internet approach to scalable routing

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

inside the AS

- intra-AS routing
  - 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

  Border routers.

OUTSIDE the AS

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

tipo il BGP, connects border routers

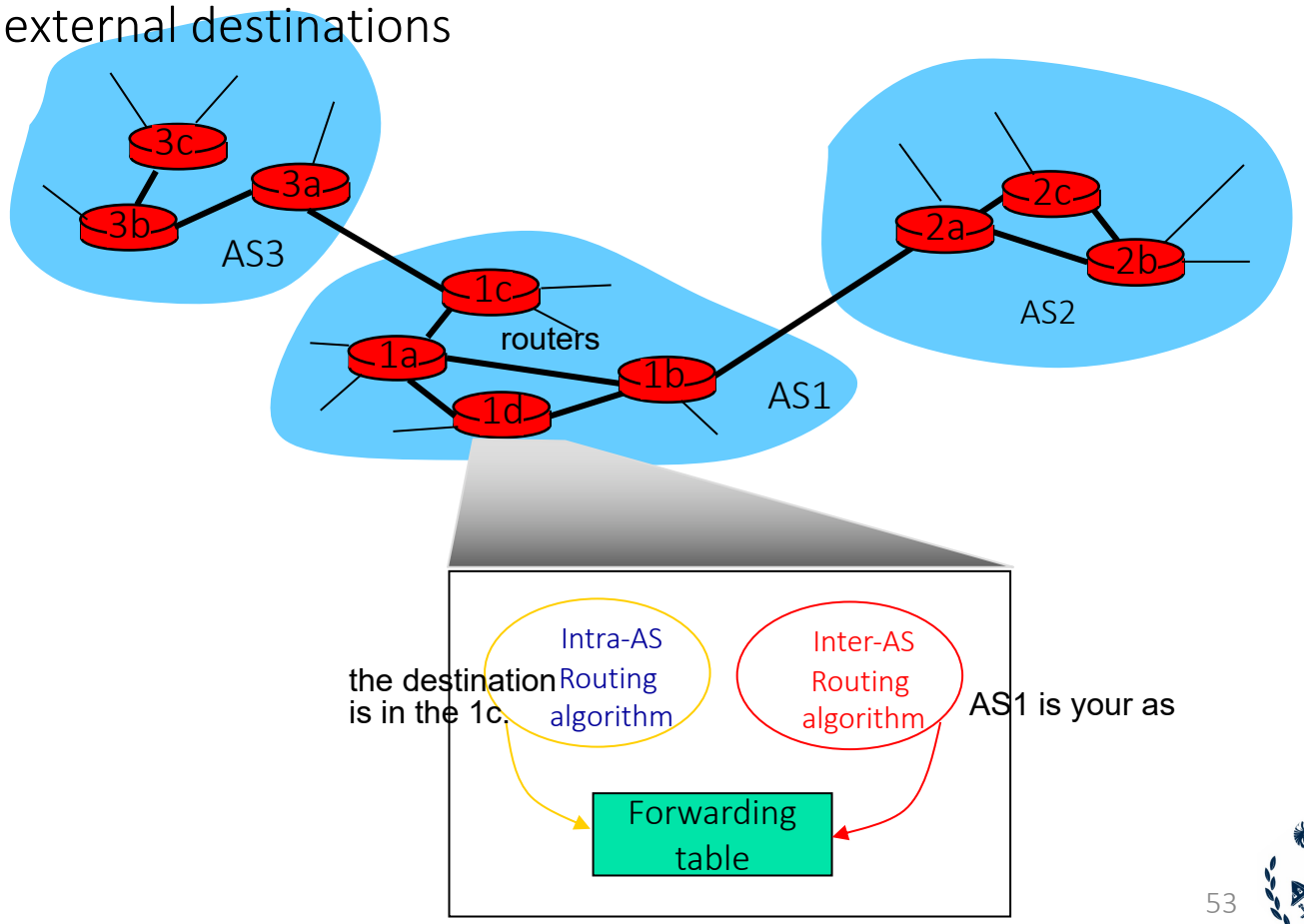
Internet Routing Architecture

Intra-AS Routing

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

how to reach the other routers and their subnets, that are part of the AS, or not. there is a collaboration between this two algorithm



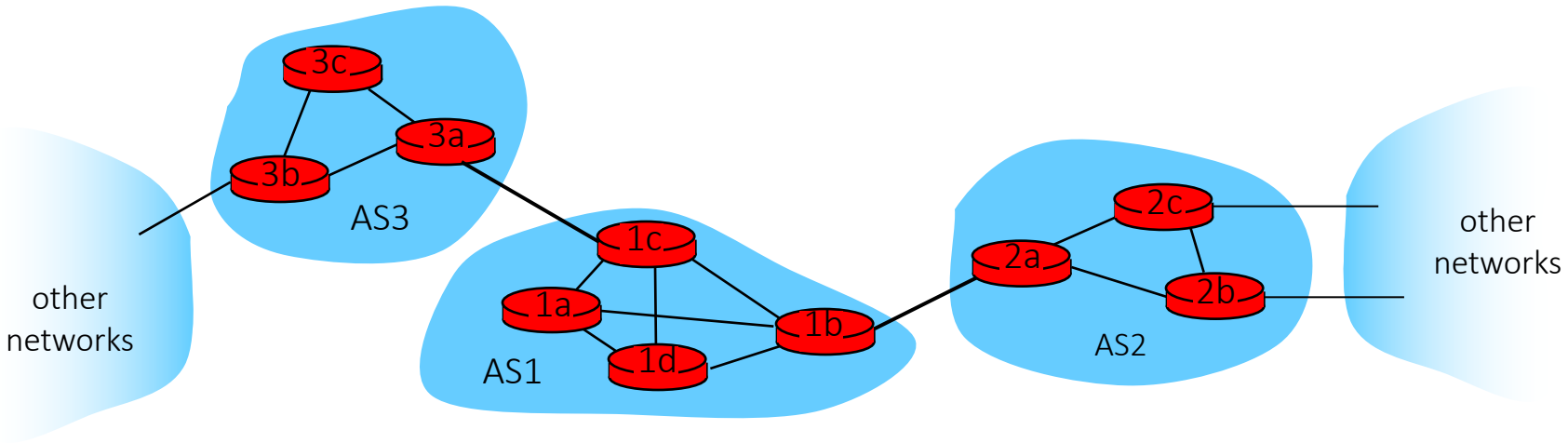
Inter-AS tasks

- suppose router in AS1 receives datagram destined outside of AS1:
  - router should forward packet to gateway router, but which one? 1c o 1b?

AS1 must:

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

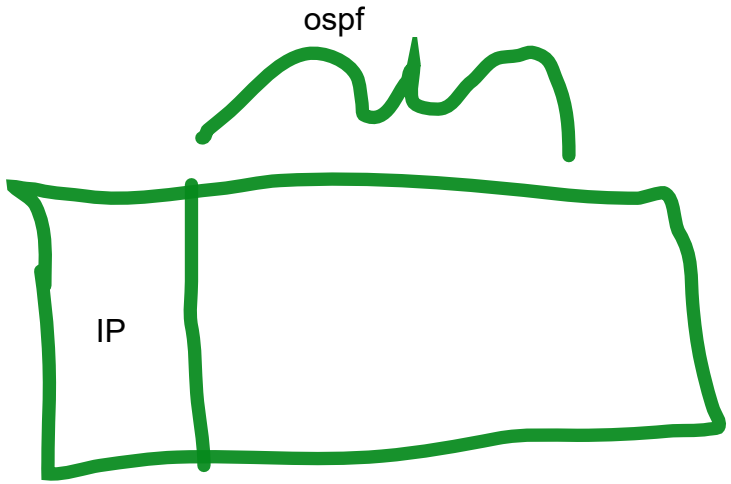
job of inter-AS routing!



Intra-AS Routing

- also known as *interior gateway protocols (IGP)*
- most common intra-AS routing protocols:
  - RIP: Routing Information Protocol (obsolete)home network, but not onbig networks, based on dv algorithm.
  - OSPF: Open Shortest Path First (IS-IS protocol essentially same as OSPF)based on link-state algorithm.
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary for decades, until 2016)

OSPF (Open Shortest Path First)

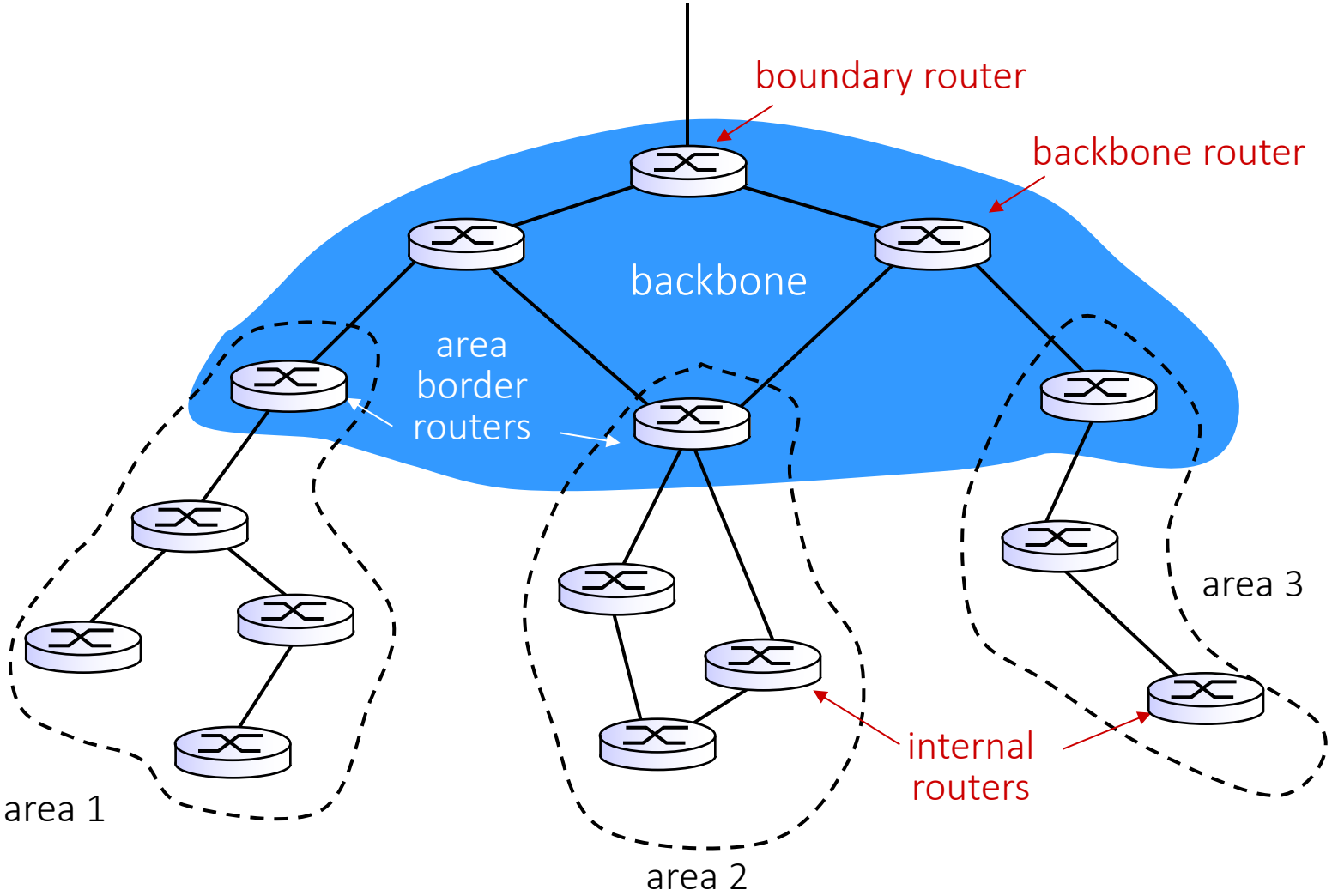


- "open": publicly available
- uses link-state algorithm
  - link state packet dissemination
  - topology map at each node
  - route computation using Dijkstra's algorithm
- router floods OSPF link-state advertisements to all other routers in entire AS
  - carried in OSPF messages directly over IP (rather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

OSPF "advanced" features

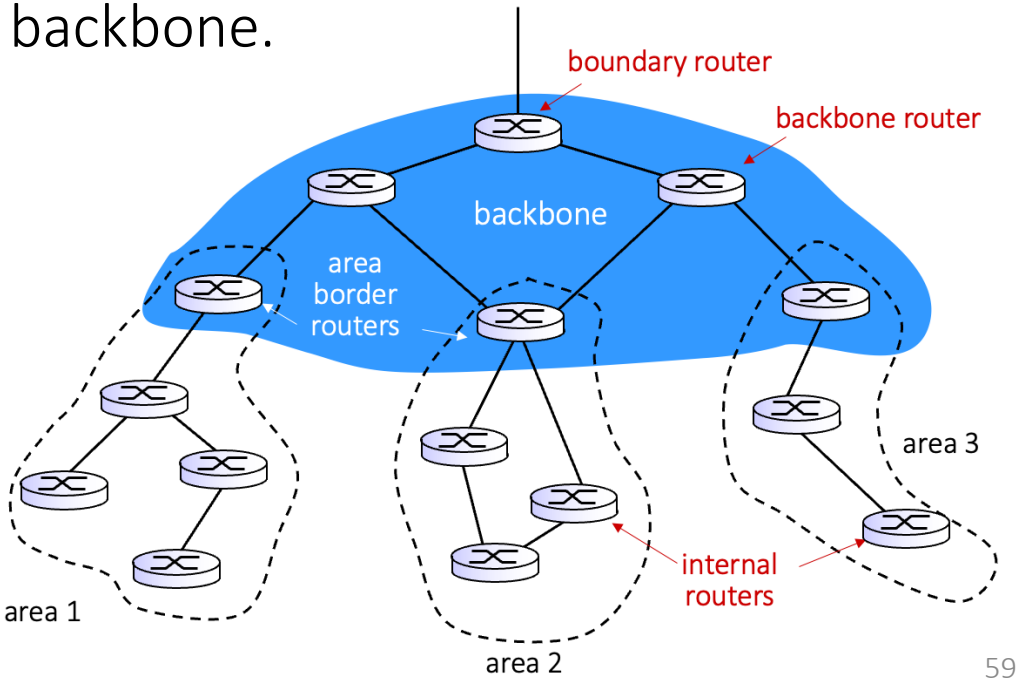
- security: all OSPF messages authenticated (to prevent malicious intrusion)
- multiple same-cost paths allowed (only one path in RIP)
- for each link, multiple cost metrics for different TOS (e.g., satellite link cost set low for best effort ToS; high for real-time ToS) type of service.
- integrated uni- and multi-cast support:
  - Multicast OSPF (MOSPF) uses same topology database as OSPF
- hierarchical OSPF in large domains see next slide.

Hierarchical OSPF



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



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

scale:

- hierarchical routing saves table size, reduced update traffic

performance:

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

Internet inter-AS routing: BGP

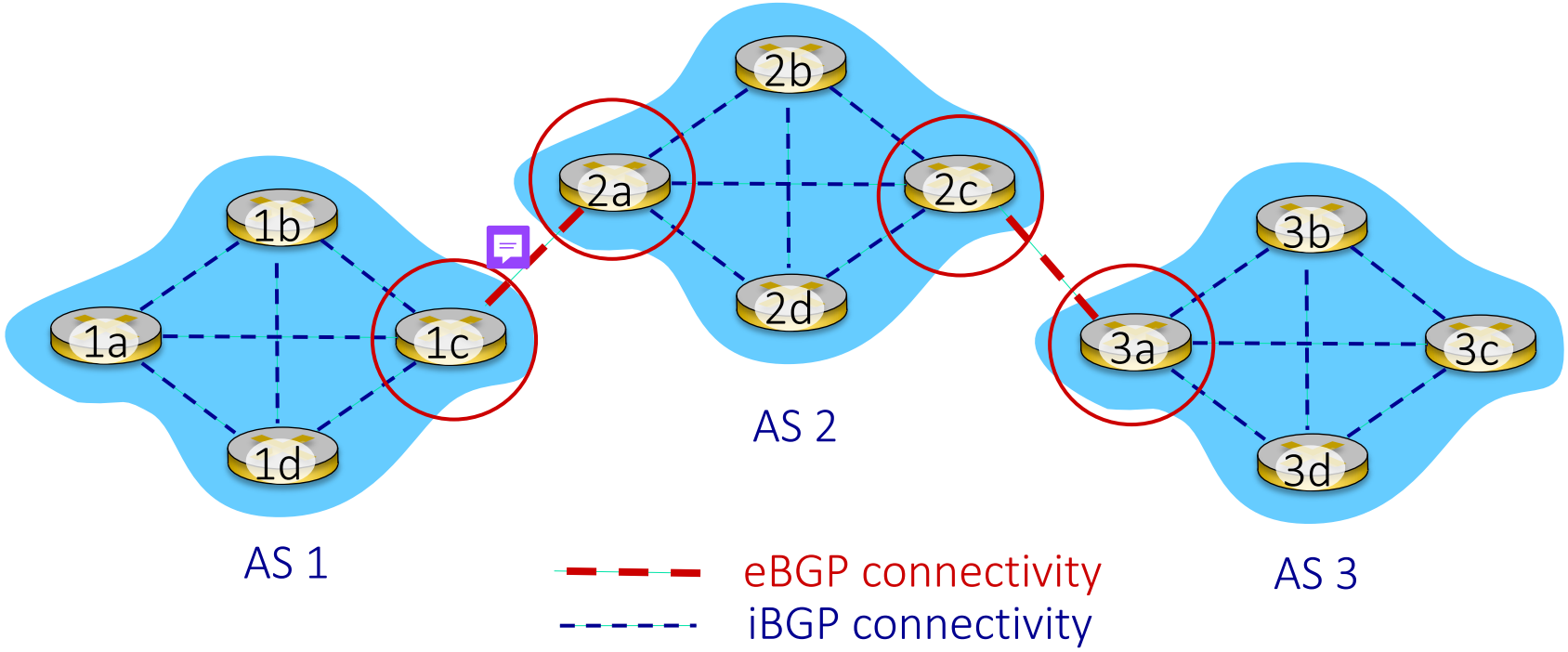
THE GLUE OF THE INTERNET

- BGP (Border Gateway Protocol): the de facto inter-domain routing protocol
  - "glue that holds the Internet together"
- BGP provides each AS a means to:
  - obtain subnet reachability information from neighboring ASes (eBGP)
  - propagate reachability information to all AS-internal routers (iBGP)
  - determine "good" routes to other networks based on reachability information and policy

MONEYYY

- 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

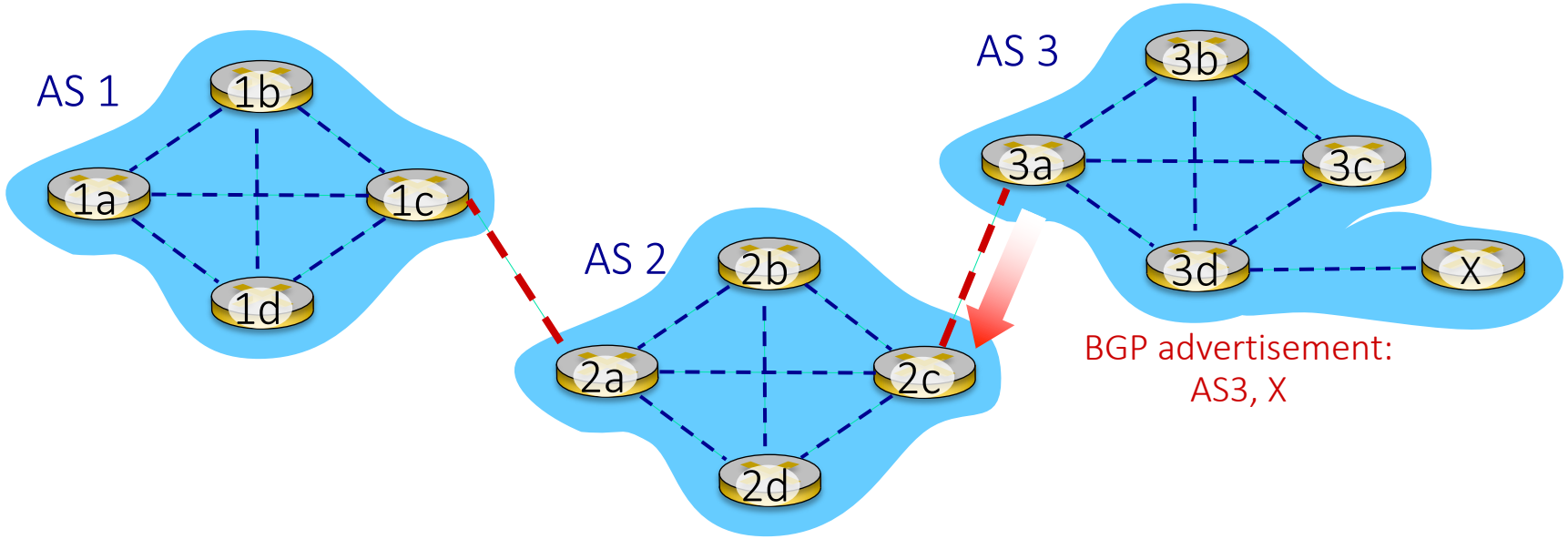
eBGP, iBGP connections



gateway routers run both eBGP and iBGP protocols

BGP basics

- 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:
  - AS-PATH: list of ASes through which prefix advertisement has passed
  - NEXT-HOP: indicates specific internal-AS router to next-hop AS
- Policy-based routing:

maybe can happen, but it goes against their interest too.

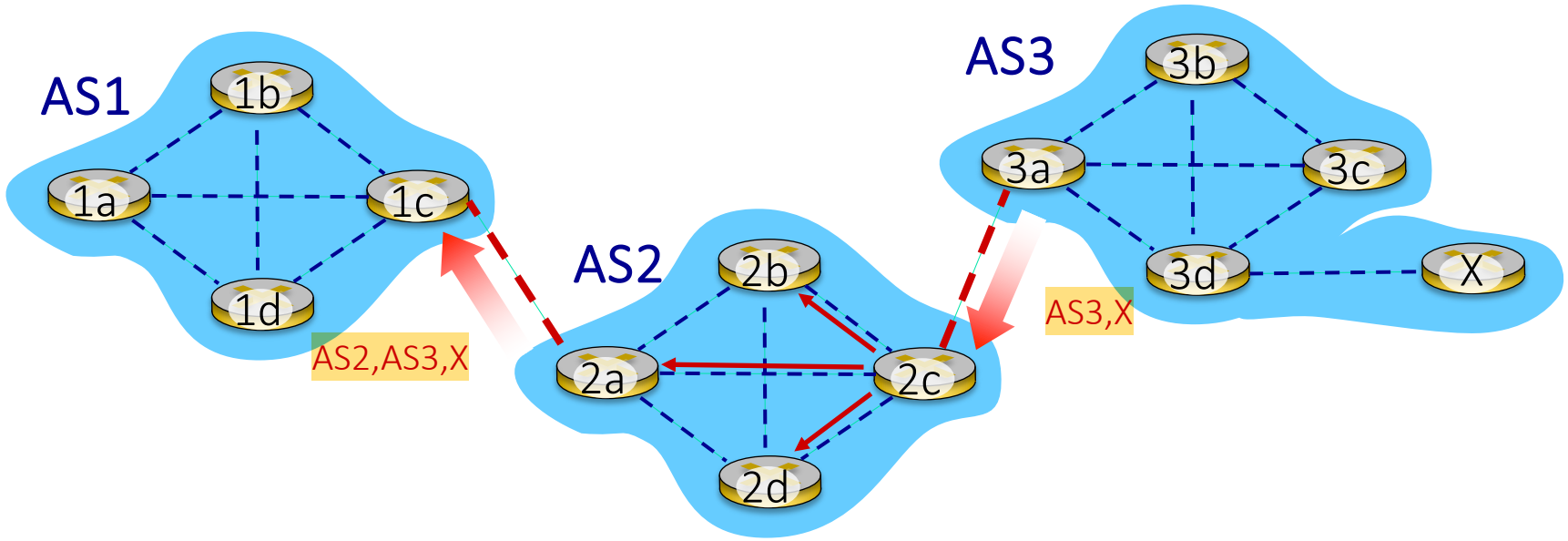
  - gateway receiving route advertisement uses import policy to accept/decline path (e.g., never route through AS Y).
  - AS policy also determines whether to advertise path to other other neighboring ASes

BGP messages

- 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

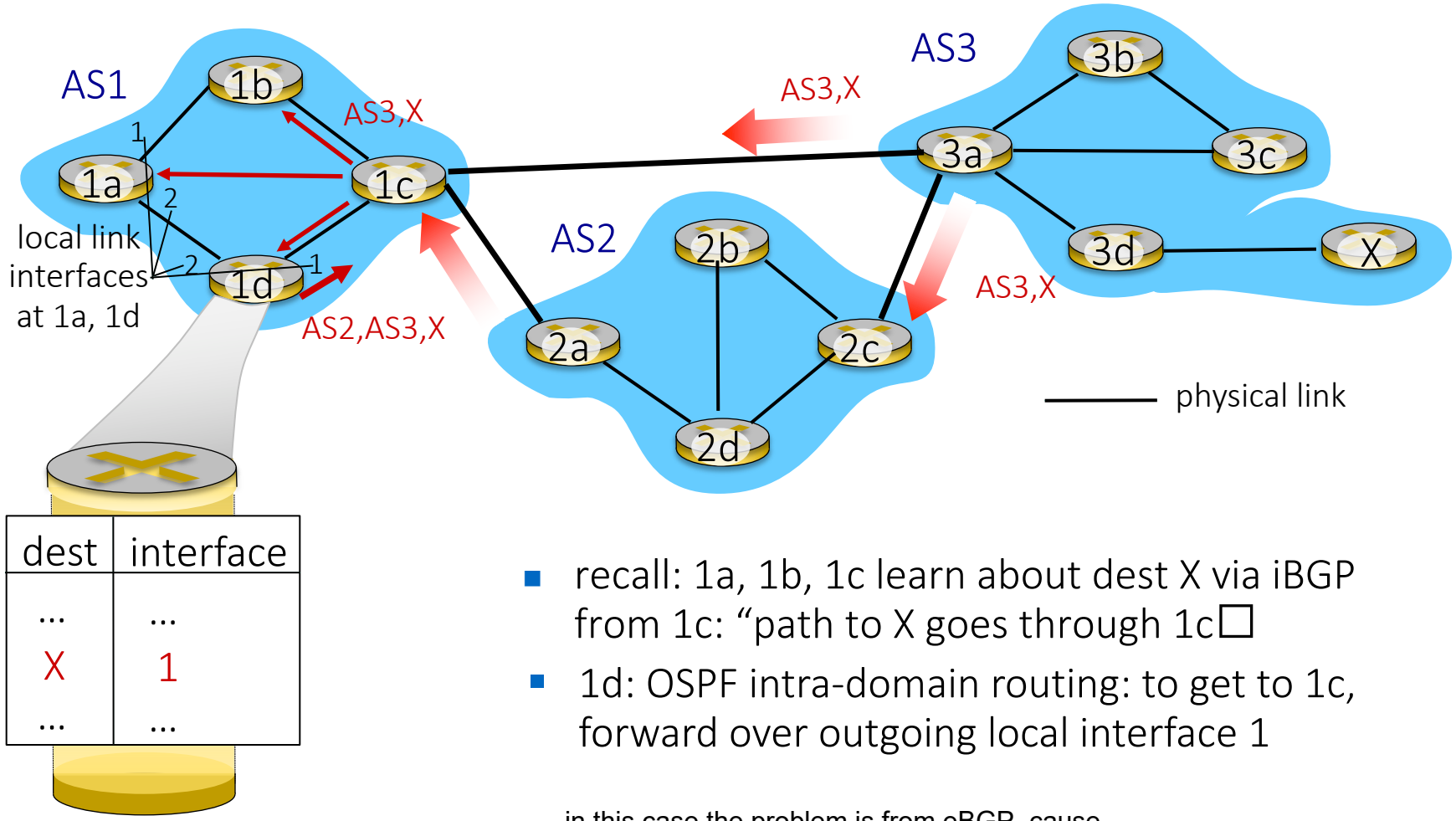
BGP path advertisement

- 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

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



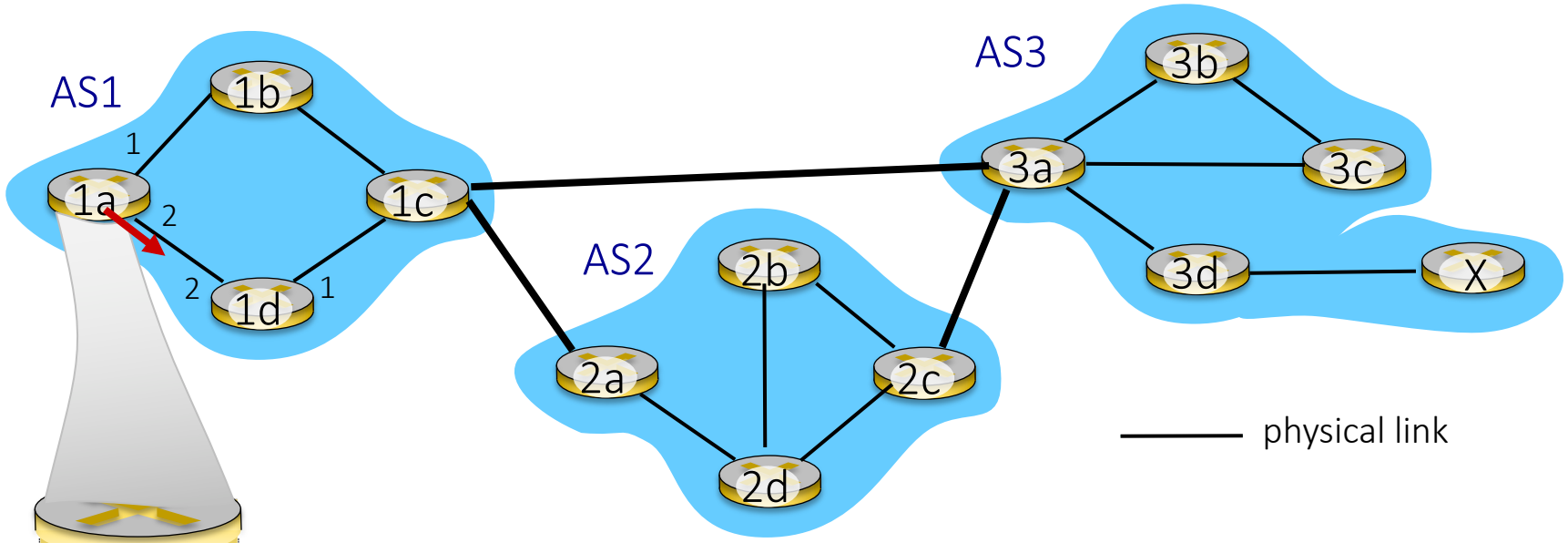
dest	interface
…	…
X	1
…	…

- 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

in this case the problem is from eBGP, cause we have multiple paths to arrive at a destination and we need to choose a criteria to route. (next)

BGP, OSPF, forwarding table entries

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

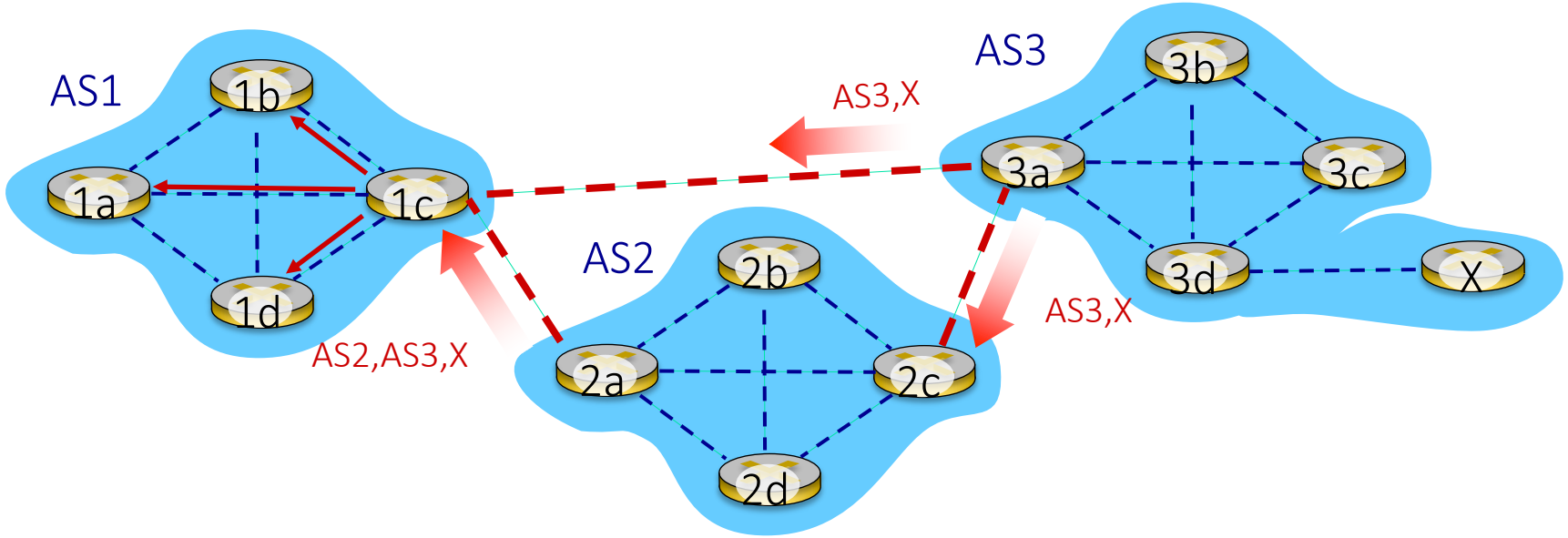


dest	interface
…	…
X	2
…	…

- 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

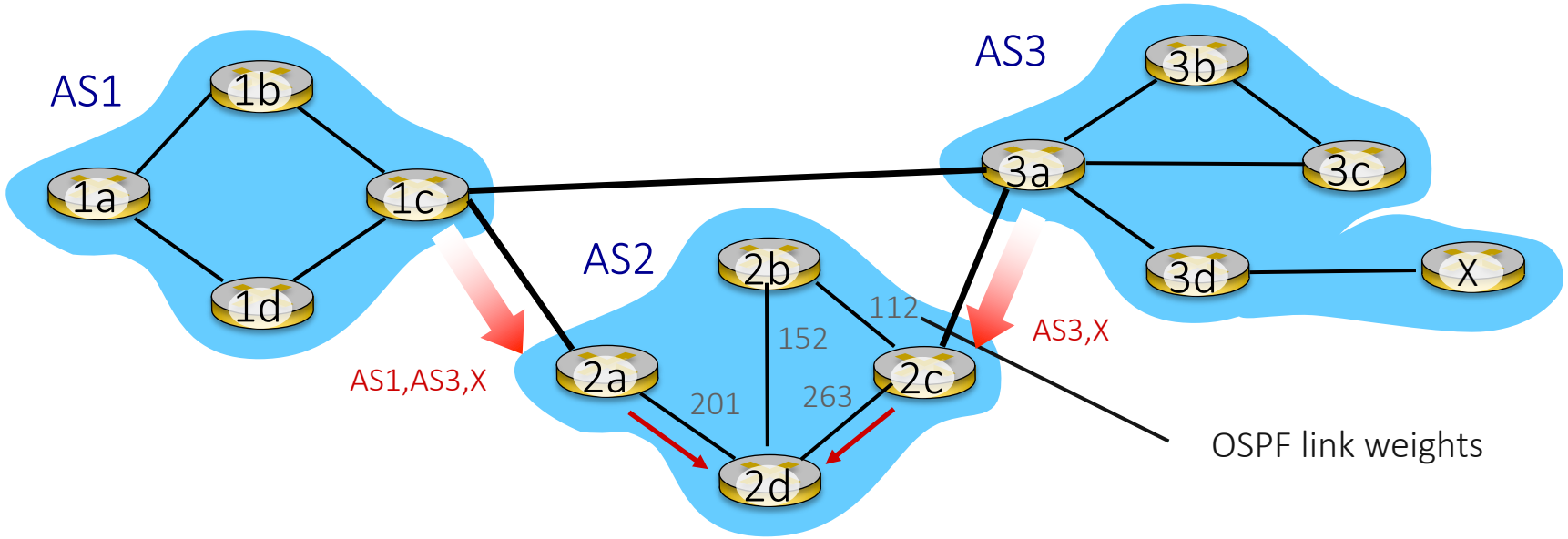
- gateway router may learn about multiple paths to destination:
  - AS1 gateway router 1c learns path AS2,AS3,X from 2a
  - AS1 gateway router 1c learns path AS3,X from 3a
  - Based on policy, AS1 gateway router 1c chooses path AS3,X, and advertises path within AS1 via iBGP



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 MONEYYYY
  2. shortest AS-PATH if money are the same, less distance
  3. closest NEXT-HOP router: hot potato routing finding the next hop that goes trough a link with the lowest end-to-end delay.
  4. additional criteria

Hot Potato Routing



- 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

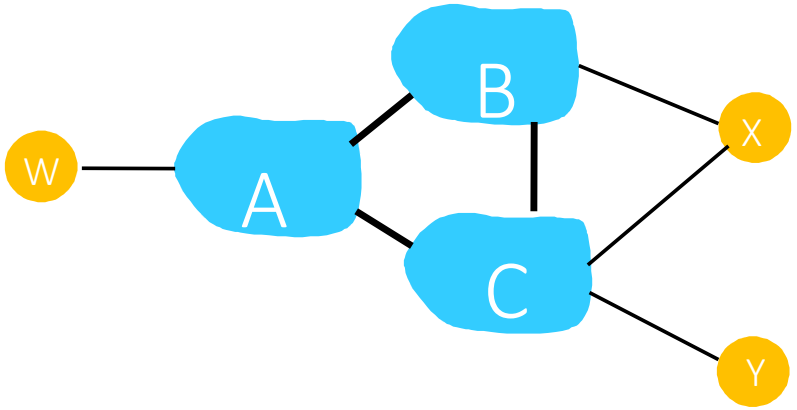
- there are two types of interconnections:

- 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)
  - Public peering through an internet exchange point (IXP) -> Peering with *multiple* networks

    not an AS
  - Private peering: networks interconnect to exchange only traffic that originates or terminates within their networks

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



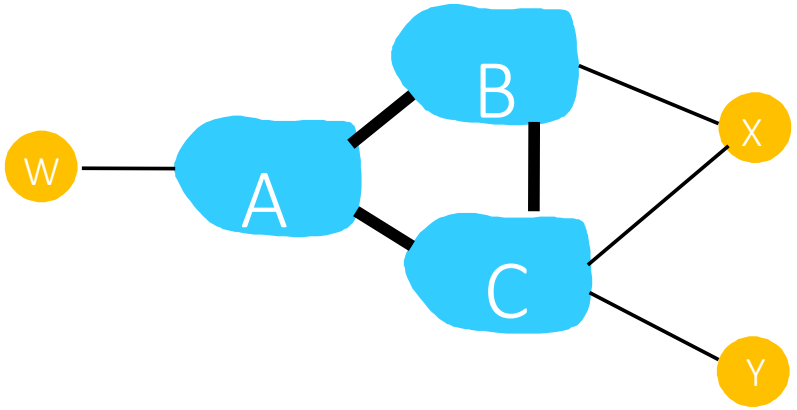
legend:

provider network

customer network:

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
- <u>Rule of thumb</u>: 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;



legend:

provider network

customer network: