Lecture 12.

Introduction to IP Routing

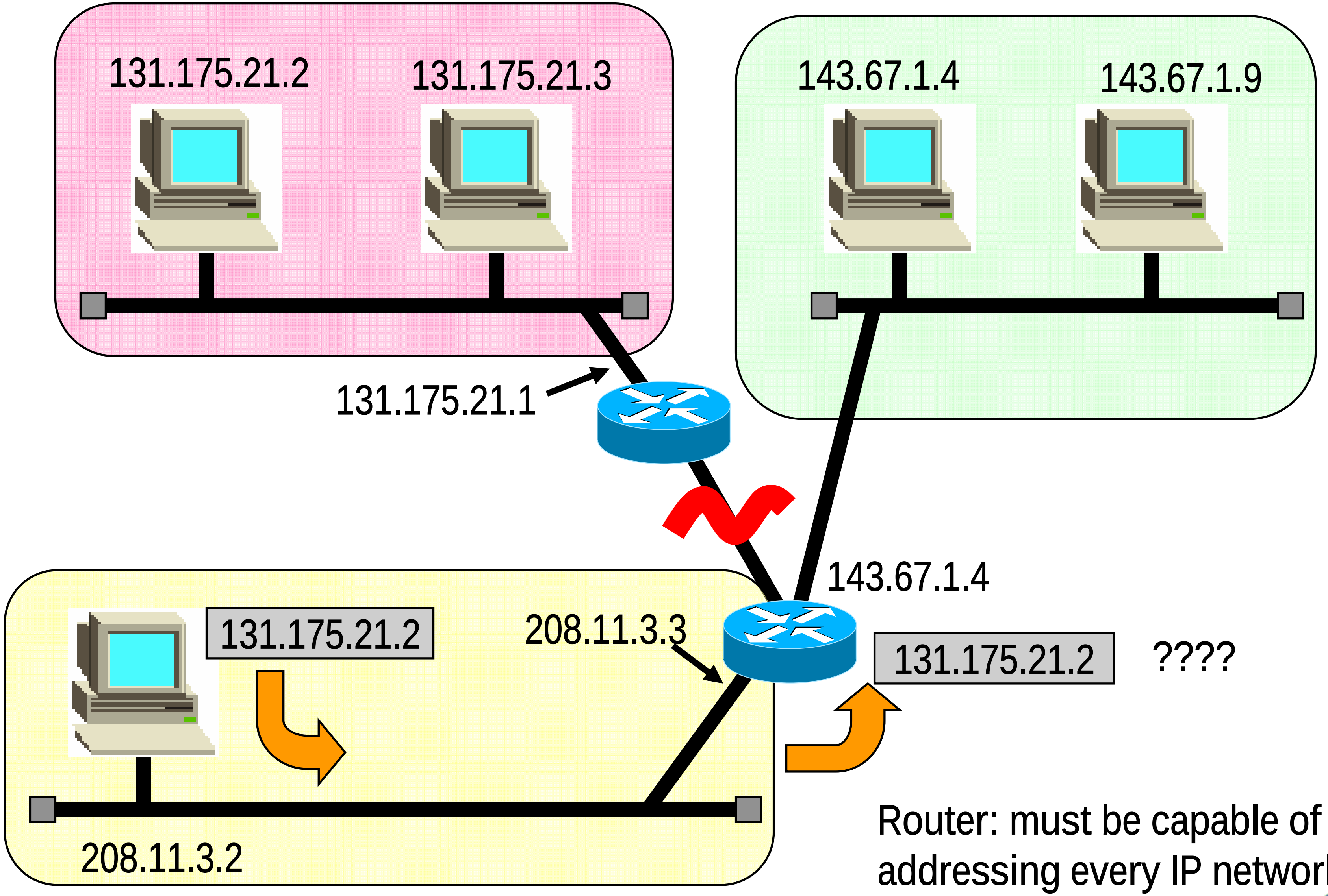
G.Bianchi, G.Neglia, V.Mancuso

Università degli Studi di Palermo

Why introduction?

- ➔ **Routing: <u>very complex</u> issue**
  - ⇨ need in-depth study
  - ⇨ entire books on routing
- ➔ **our scope:**
  - ⇨ give a flavour of basic routing structure and messaging
  - ⇨ give an high-level overview of IP routing protocols

Routing

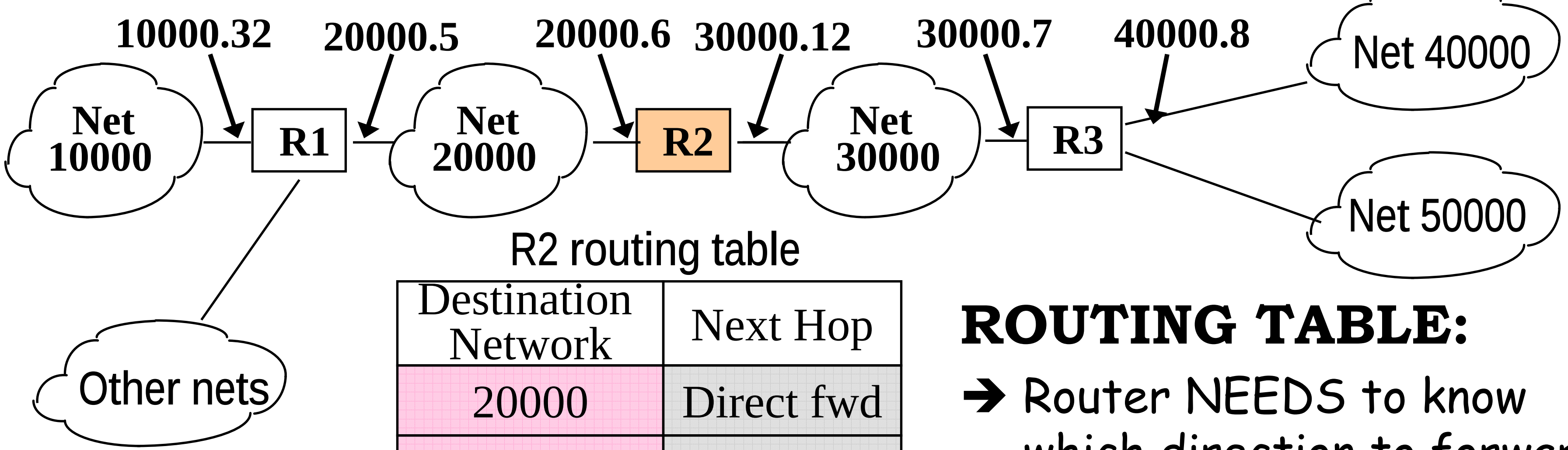


Router: must be capable of addressing every IP network

Routing Table

Destination network	Next router
... ...	... ...
131.175.0.0	144.21.32.4
... ...	... ...

Routing table



R2 routing table

Destination Network	Next Hop
20000	Direct fwd
30000	Direct fwd
10000	20000.5
40000	30000.7
50000	30000.7
default	20000.5

ROUTING TABLE:

➔ Router NEEDS to know which direction to forward the datagram
  - ⇨ to let it reach the final destination

➔ But DOES NOT NEED to know the detailed path!
  - ⇨ It stores only the NEXT HOP router.

Three cases:
1) direct forwarding
2) Indirect forwarding (explicit)
3) Indirect forwarding via default router (when available)

Default route

- ➔ **Frequent in small and medium size networks**
  - ⇨ generally administrator sends to router of higher network hierarchy
  - ⇨ e.g. our 131.175.15.254 (tlc) router defaults to elet router, which defaults to polimi, which defaults to Cilea router
- ➔ **Large networks (class B sized) should default only when strictly necessary**
  - ⇨ to avoid traffic increase and suboptimal router
- ➔ **TOP LEVEL ROUTING DOMAINS**
  - ⇨ maintain routing information to most Internet sites, and do not use <u>any</u> default route
  - ⇨ 5 in 1993: NFSNET, CIX, NSI, SprintLink, EBONE

Routing operation

assume router with IP address X

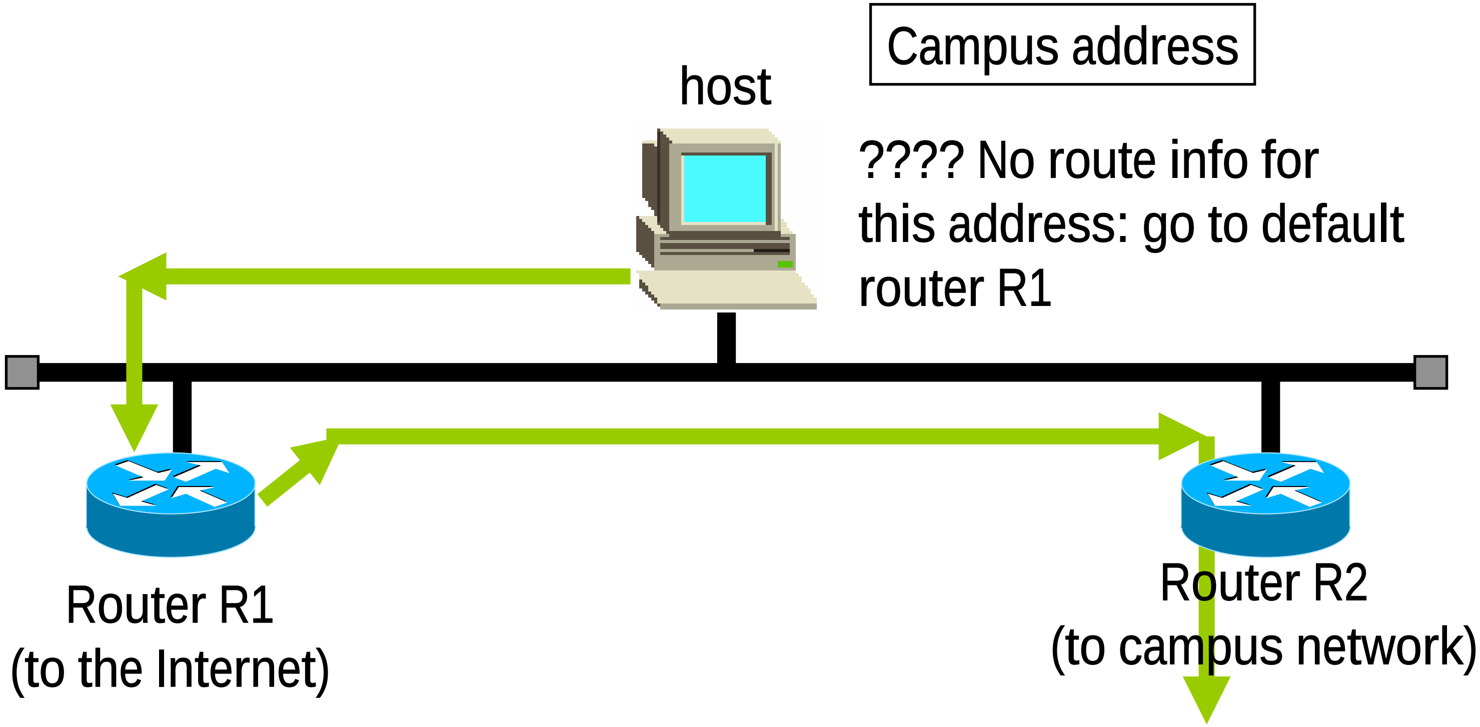
1) extract destination IP (**Y**) from datagram
2) if **Source Route Option**, forward accordingly
3) if **Y==X**, deliver packet to specified protocol
4) **decrease TTL**; if TTL=0 throw away datagram and send ICMP “time expired” message
5) if ***X.and.Netmask==Y.and.Netmask***, direct forwarding of datagram (use ARP)
6) extract next hop router from routing table, and forward packet to next router
7) If no next hop, forward to default router
8) if no default route, declare route error and send notification via ICMP

ICMP host and Network unreachable errors

ICMP type 3 errors, codes 0 (network) and 1 (host)

- **Host unreachable**
  - network found, but packet could not be delivered to host
- **Network unreachable**
  - route error (network not found in routing table)

Typical redirection case



Clearly, host should have used R2 immediately…

redirect

- ➔ **R1 operation**
  - ⇨ looks up routing table, and determine that R2 is the proper path
  - ⇨ in the mean time, it realizes that <u>packet comes from same interface</u> on R2 network
  - ⇨ this makes R1 understand that redirection is possible
  - ⇨ thus sends a ICMP redirect error message
- ➔ **Host:**
  - ⇨ when receiving a redirect message, it updates its routing table
  - ⇨ basically, host *LEARNS* from redirects (easier task for admin that does not need to correctly configure all hosts)!

ICMP redirect

0 ... 7	8 ... 15	16 ... 31
Type (5)	Code (0-3)	Checksum
Router IP address that should be used		
IP Header + first 8 bytes IP of original datagram data		

REDIRECT CODES

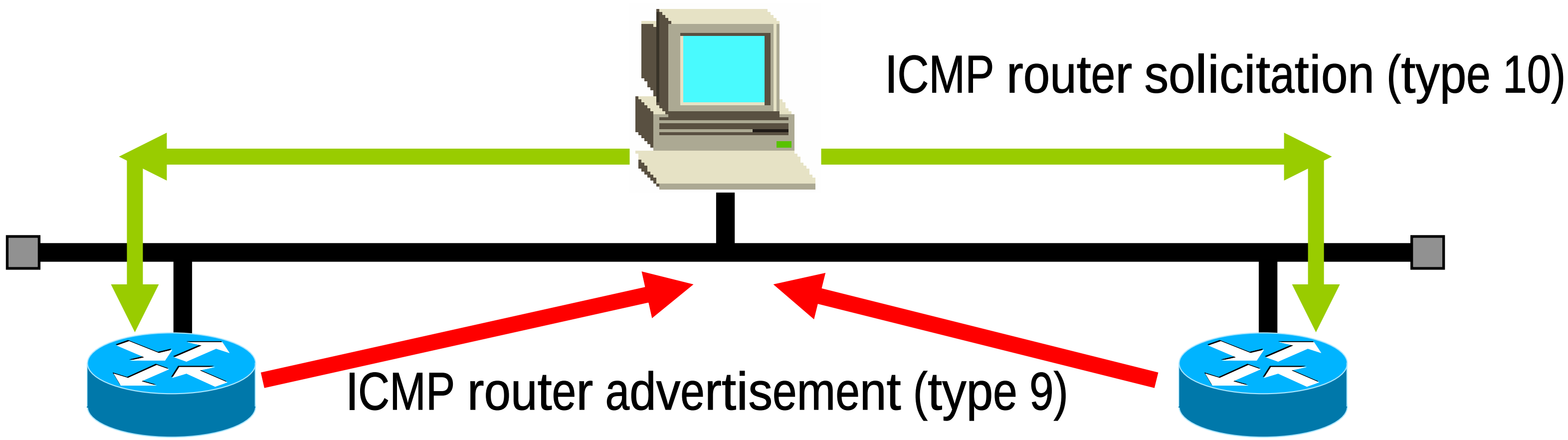
0	Redirect for network
1	Redirect for host
2	Redirect for TOS and network
3	Redirect for TOS and host

The only one used in practice: 1 (Redirect for host)

- Only routers may use redirect (other routers are assumed to be informed by full-fledged routing protocol, and not by occasional redirects!!
- redirect must be addressed to hosts (not routers)
- *network redirection hard to be used (without netmask info!)*

Host routing table creation

- ➔ **Manual creation**
- ➔ **via router solicitation ICMP message**



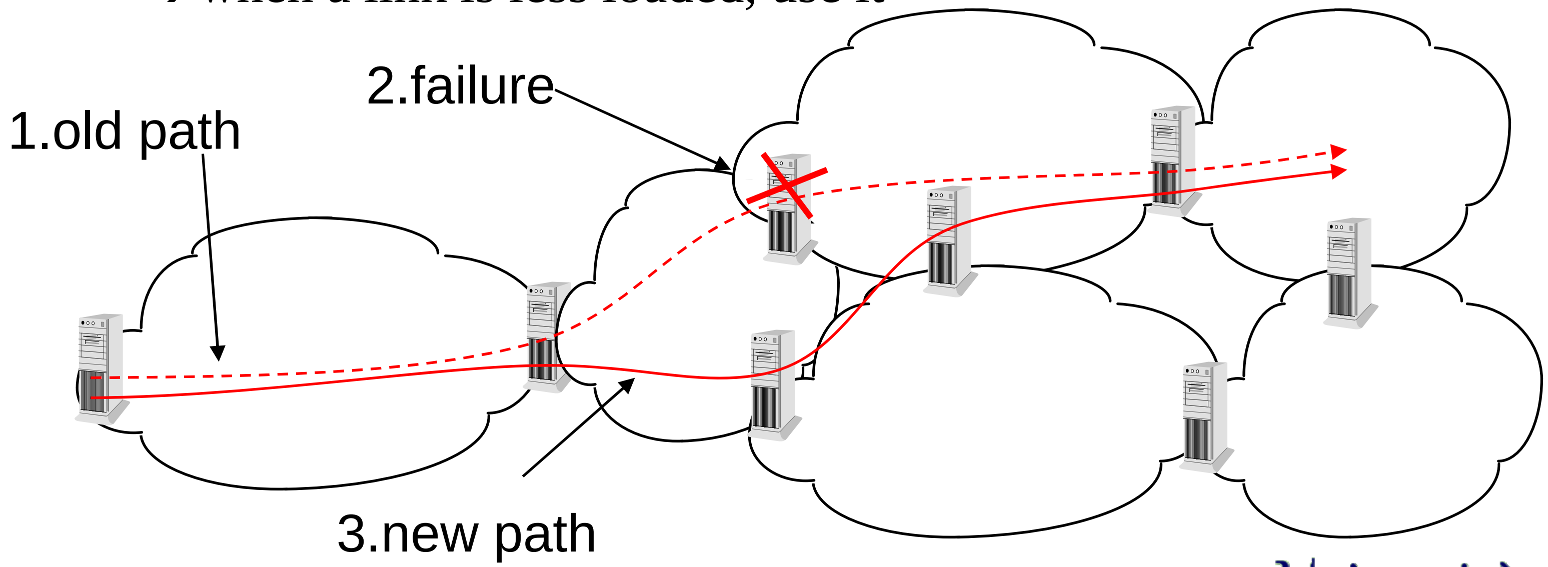
Router solicitation: asks who are the routers connected
Router advertisement: return router list and preference
preference: when multiple routers are connected to the same network
preference values configured by administrator

Static versus dynamic routing

- **Static routing**
  - based on static routing table entries
    - entered manually
    - changed via ICMP redirects
- **Fine when**
  - network small
  - single connection point to other networks
  - no alternative paths toward destinations
- **Not fine when <u>one of above conditions</u> fails**

Dynamic (adaptive) routing
All IP routing protocols are dynamic

➔ **Routing table entries change in time, depending on**
- ⇨ link failures
  - →when a link is down, you need to avoid it!
- ⇨ network topology changes
  - →when a new backbone added, use it!
- ⇨ Traffic load and congestion
  - →when a link is less loaded, use it



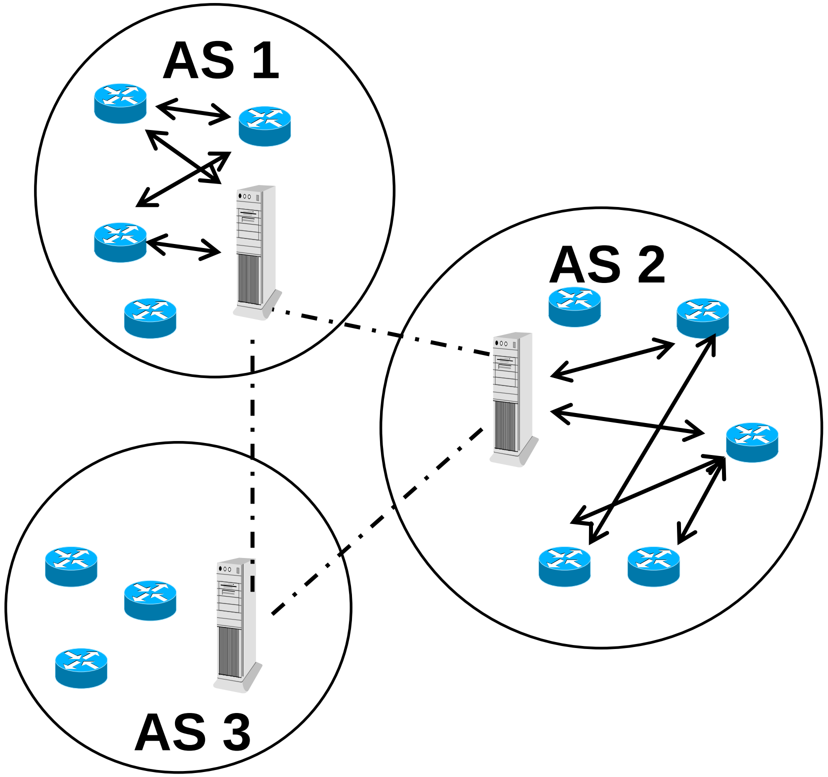
Dynamic routing

- ➔ **Requirement:**
  - ⇨ Information exchange among routers is required, to dynamically update routing table
    - → extra load
    - → need for inter-routing message formats
- ➔ **Risks**
  - ⇨ oscillation
    - → too fast adaptation procedures
  - ⇨ inefficiency
    - → too slow adaptation to changed situation
  - ⇨ loops

Autonomous Systems

a key concept for The Internet

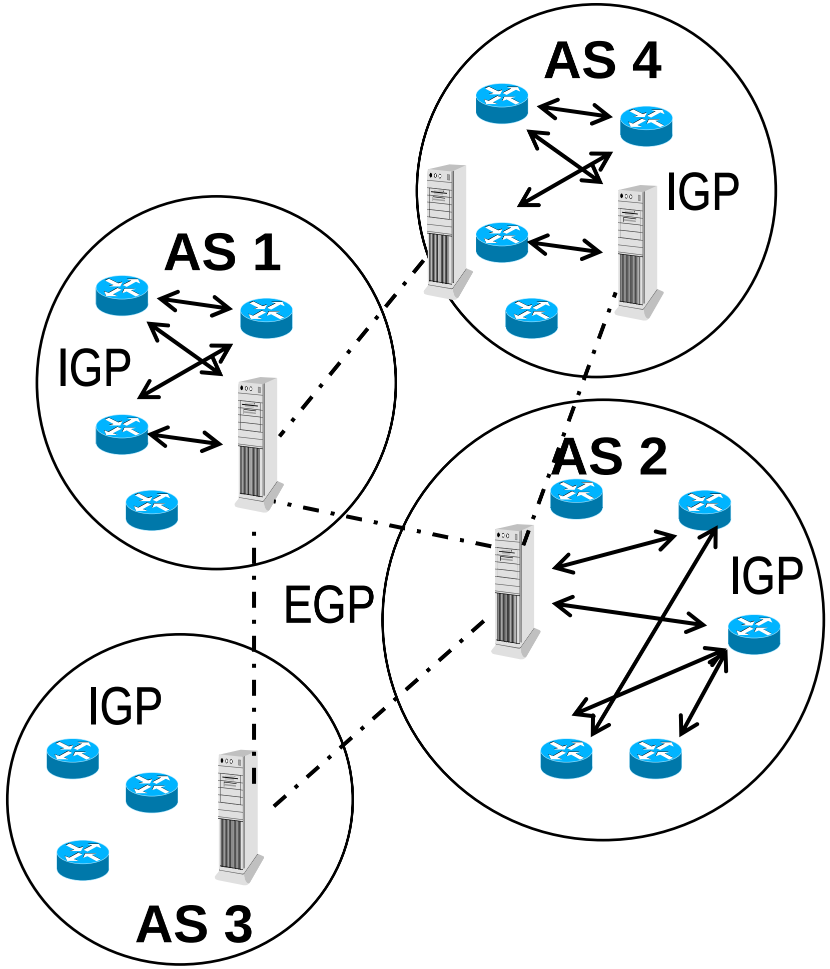
- **Internet organized as a collection of Autonomous Systems (ASs)**
- **each AS normally administered by a single entity**
- **each AS selects its own routing protocol to allow inter-router communication within the AS**



- **Interior Gateway Protocol (IGP)**
  - Intra-Domain routing protocol
  - within an AS
- **Exterior Gateway Protocol (EGP)**
  - Inter-Domain routing protocol
  - among different ASs

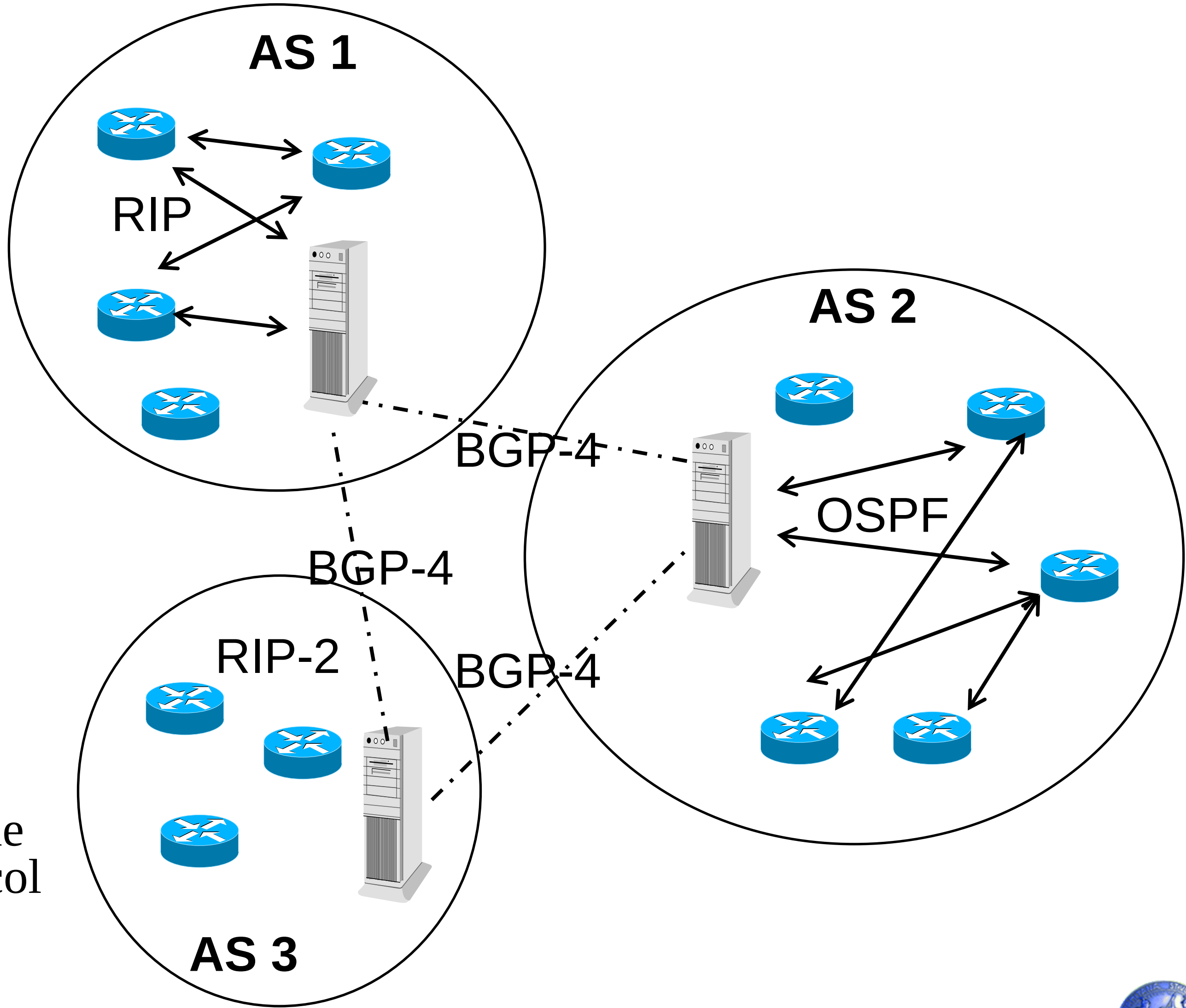
Border routers

- ➔ Routing within AS is arbitrary chosen by AS administrator
- ➔ but there must be one or more ***border routers*** in charge of communicating to the external world its internal routing information (data collected by the IGP used)
- ➔ Border routers are the only entitled to exchange EGP information



IP routing protocols

- **IGPs**
  - HELLO
  - *RIP1*
  - RIP2
  - *OSPF* (1 & 2)
  - IS-IS
  - IGRP, EIGRP
    proprietary
    (CISCO)
  - ...
- **EGPs**
  - EGP
    - yes: Same name of entire protocol class!
  - *BGP-4*



G.Bianchi, G.Neglia, V.Mancuso

RIP
Routing Information Protocol

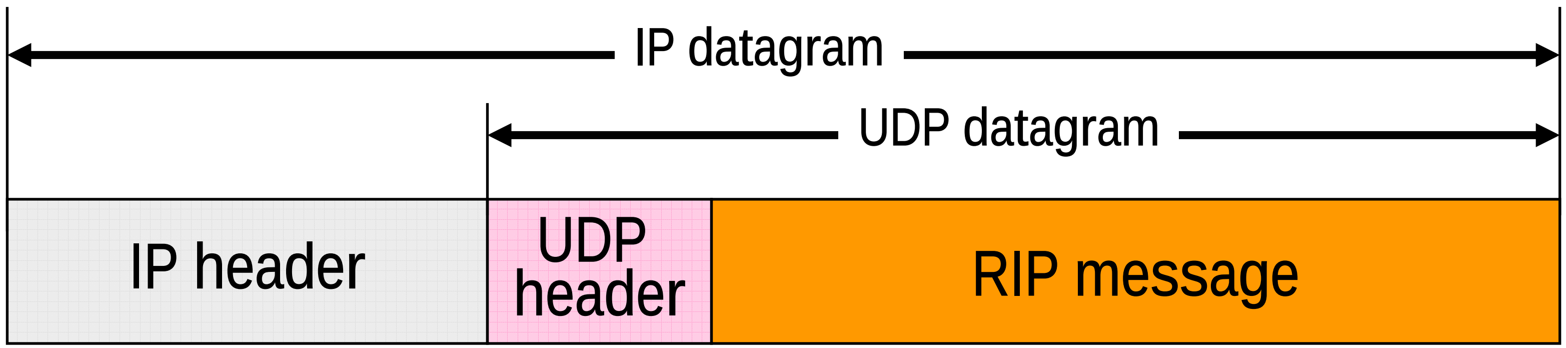
and distance vector protocols in general

G.Bianchi, G.Neglia, V.Mancuso

Università degli Studi di Palermo

Routing Information Protocol

- ➔ **Most widely used**
  - ⇨ and most criticized…
- ➔ **Official specification: RFC 1058 (1988)**
  - ⇨ but used from several years before
- ➔ **Uses UDP to exchange messages**
  - ⇨ well known UDP port = 520



RIP message

0 – 7	8 – 15	16 – 31	
Command (1-6)	Version (1)	0	
Address family (2)		0	20 bytes
IP address			
0			
0			
Metric			

● ● ● ● Up to 24 more routes with same 20 bytes format ● ● ● ●

<u>Command</u>: 1=request; 2=reply (3-6 obsolete or non documented)
<u>Address family</u>: 2=IP addresses
<u>metric</u>: distance of *emitting router* from the specified IP address in *number of hops (valid from 1 to 15; 16=infinite)*

Message size

- ⇨8 UDP header
- ⇨4 bytes RIP header
- ⇨20 bytes x up to 25 entries

➔**total: maximum of 512 bytes UDP datagram**

➔**25 entries: too little to transfer an entire routing table**

- ⇨more than 1 UDP datagram generally needed

Initialization

- ➔ **When routing daemon started, send special RIP request on every interface**
  - ⇨ command = 1 (request)
  - ⇨ address family = 0 (instead of 2)
  - ⇨ metric set to 16 (infinite)
- ➔ **This asks for complete routing table from all connected routers**
  - ⇨ allows to discover adjacent routers!

Operation after initialization

- **Request:**
  - asks for response relative to specific IP addresses listed in the request message
- **Response:**
  - return list of IP addresses with associated metric
  - if router does not have a route to the specified destination, returns 16
- **Regular update:**
  - routers send part (or all) of their table every 30s to adjacent routers
  - a router deletes (set metric to 16) an entry from its routing table if not refreshed within 6 cycles (180s)
    - deletion after additional 60s to ensure propagation of entry invalidation
- **triggered update:**
  - upon change of metric for a route (transmits only entries changed)

RIP 2

- ➔ **Does not change the protocol operation**
- ➔ **simply adds information in the all 0s fields of the RIP message**
- ➔ **It is designed to maintain full compatibility with RIP routers**
  - ⇨ al least if they don't get confused from the non 0 entries

RIP 2 message format

0 – 7	8 – 15	16 – 31
Command (1-6)	Version (2)	routing domain
Address family (2)		Route tag
IP address		
Subnet mask		
Next hop IP address		
Metric		

(Address family through Metric: 20 bytes)

● ● ● ● Up to 24 more routes with same 20 bytes format ● ● ● ●

Most important modification: **subnet mask** (allows use with VLSM and CIDR)

Next hop address: specifies where packet should be sent when addressed to Ipaddr

details in RFC 1388

RIP logic

➔ ***Distance Vector routing protocol***

- ⇨Bellman-Ford algorithm
  - →METRIC = distance
  - →STATE INFO = vector

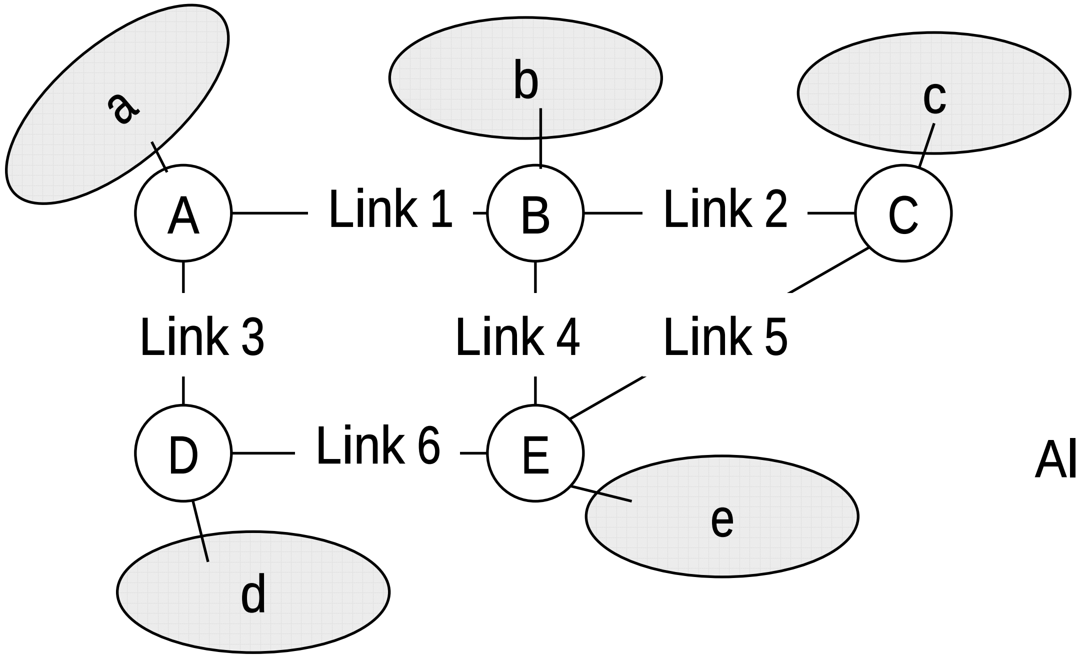
➔ **each router maintains a table with:**

- ⇨best known distance (in hop count) to each destination
- ⇨link to use to reach the destination

➔ **fully distributed protocol**

- ⇨vector (table) updates via communication with neighbors

RIP operation example (1)



All routers start from empty tables

Router A		
dst	hop	lnk
a	0	loc

Router B		
dst	hop	lnk
b	0	loc

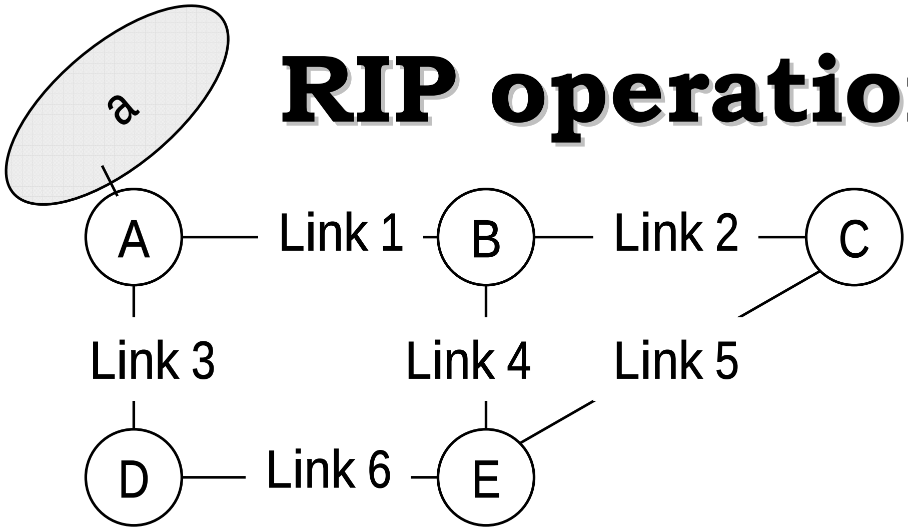
Router C		
dst	hop	lnk
c	0	loc

Router D		
dst	hop	lnk
d	0	loc

Router E		
dst	hop	lnk
e	0	loc

Direct forwarding

RIP operation example (2)



All routers start from empty tables

Router A		
dst	hop	lnk
a	0	loc

Router B		
dst	hop	lnk
b	0	loc

Router C		
dst	hop	lnk
c	0	loc

Router D		
dst	hop	lnk
d	0	loc

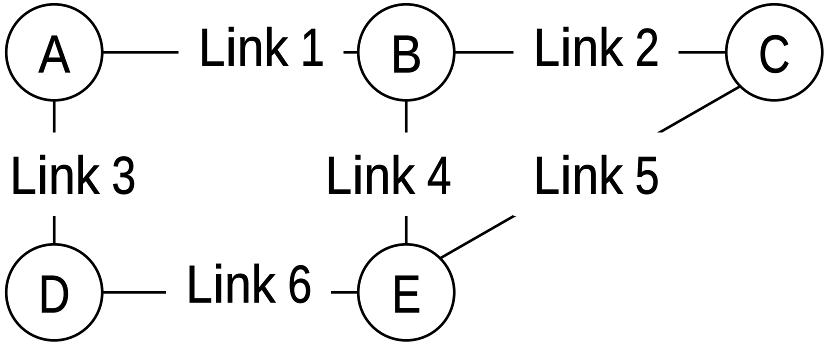
Router E		
dst	hop	lnk
e	0	loc

Router A emits message (A,0) to adjacent routers (B,D), which update table as:

Router B		
dst	hop	lnk
b	0	loc
a	1	1

Router D		
dst	hop	lnk
d	0	loc
a	1	3

RIP operation example (3)



New step: B propagates its updated routing table to neighbohrs A, C, E

Router B		
dst	hop	lnk
b	0	loc
a	1	1

Router A		
dst	hop	lnk
a	0	loc
b	1	1

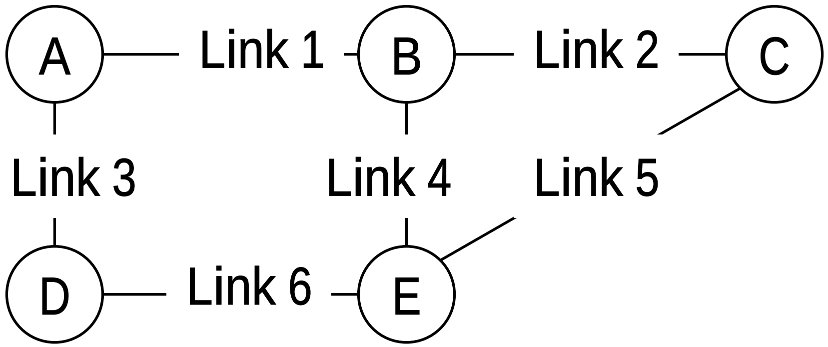
Router B		
dst	hop	lnk
b	0	loc
a	1	1

Router C		
dst	hop	lnk
c	0	loc
b	1	2
a	2	2

Router D		
dst	hop	lnk
d	0	loc
a	1	3

Router E		
dst	hop	lnk
e	0	loc
b	1	4
a	2	4

RIP operation example (4)



Step 3: D propagates its routing table to A, E

Router D		
dst	hop	lnk
d	0	loc
a	1	3

Router A		
dst	hop	lnk
a	0	loc
b	1	1
d	1	3

Router B		
dst	hop	lnk
b	0	loc
a	1	1

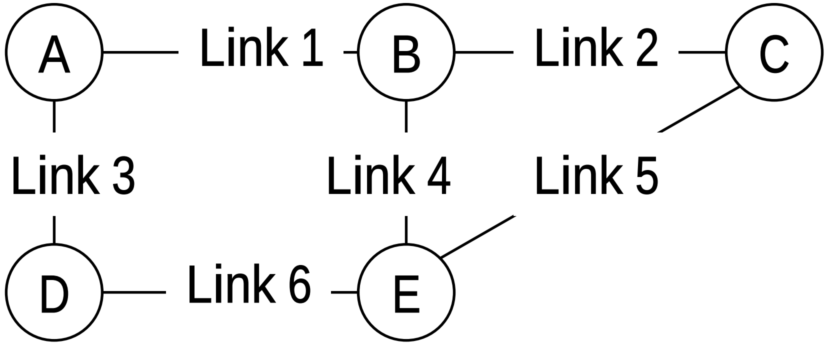
Router C		
dst	hop	lnk
c	0	loc
b	1	2
a	2	2

Router D		
dst	hop	lnk
d	0	loc
a	1	3

Router E		
dst	hop	lnk
e	0	loc
b	1	4
a	2	4
d	1	6

Already updated!

RIP operation example (5)



Step 4: A propagates its routing table to B,D

Router A		
dst	hop	lnk
a	0	loc
b	1	1
d	1	3

Router A		
dst	hop	lnk
a	0	loc
b	1	1
d	1	3

Router B		
dst	hop	lnk
b	0	loc
a	1	1
d	2	1

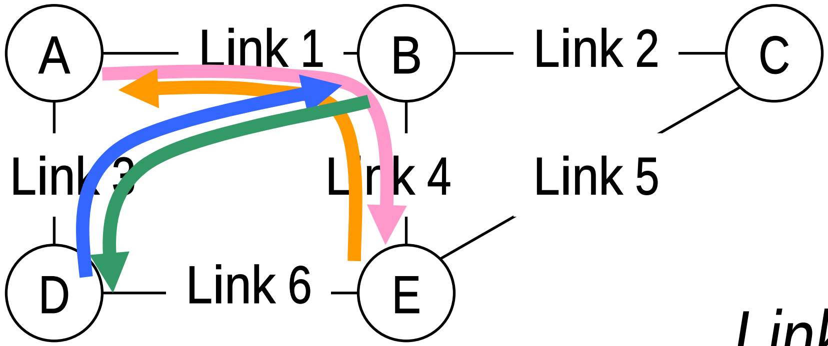
Router C		
dst	hop	lnk
c	0	loc
b	1	2
a	2	2

Router D		
dst	hop	lnk
d	0	loc
a	1	3
b	2	3

Router E		
dst	hop	lnk
e	0	loc
b	1	4
a	2	4
d	1	6

...ETC ETC ETC...

RIP operation example
final routing tables



Step 5: C -> B, E
Step 6: E-> B, C, D
Step 7: B-> A, C, E

Link 6 under-utilized!!

Router A		
dst	hop	lnk
a	0	loc
b	1	1
c	2	1
d	1	3
e	2	1

Router B		
dst	hop	lnk
a	1	1
b	0	loc
c	1	2
d	2	1
e	1	4

Router C		
dst	hop	lnk
a	2	2
b	1	2
c	0	loc
d	2	5
e	1	5

Router D		
dst	hop	lnk
a	1	3
b	2	3
c	2	6
d	0	loc
e	1	6

Router E		
dst	hop	lnk
a	2	4
b	1	4
c	1	5
d	1	6
e	0	loc

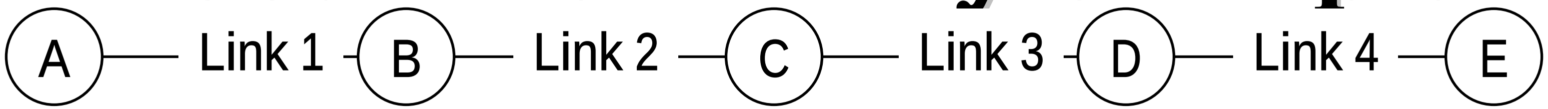
Apparent limits of RIP

- **Hop count is a too simple metric!**
  - But Bellman-ford algorithm does not require to operate with hop count! Can be trivially extended to different distance metric: the core of the algorithm does not change!
    - queue length on considered hop
    - time delay in ms
    - packet loss ratio measured
    - etc…
- **Slow convergence**
  - routers distant N hops need N steps to update their tables
- **Limited to small network sizes**
  - as infinite=16, nodes cannot be more than 15 hops far away
    - but just raise infinite to 32...

Real limit of RIP
"count to infinity" problem

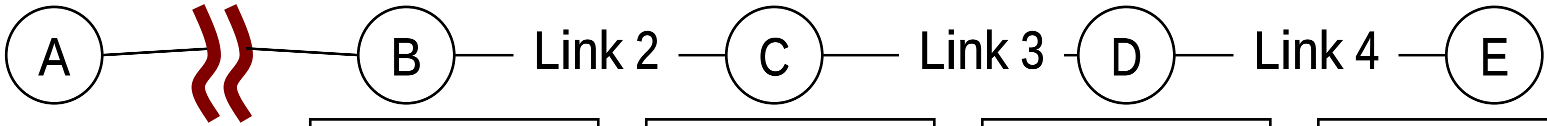
- ➔**Insane transient reaction to node/link failures!**
  - ⇨Convergence still remains, but very slow
  - ⇨Loops may occur while routing tables stabilize
  - ⇨the slower, the higher value infinite is chosen!!
    - ➔Values higher than 16 are terrible
  - ⇨An intrinsic and unavoidable drawback for all Distance Vector schemes

Count to infinity example



Router A			Router B			Router C			Router D			Router E		
dst	hop	lnk	dst	hop	lnk	dst	hop	lnk	dst	hop	lnk	dst	hop	lnk
a	0	loc	a	1	1	a	2	2	a	3	3	a	4	4

LINK 1 breaks



B does not get refreshes from a. Then uses refreshes from C, that tell a=3!

Router B			Router C			Router D			Router E		
dst	hop	lnk	dst	hop	lnk	dst	hop	lnk	dst	hop	lnk
a	3	2	a	2	2	a	3	3	a	4	4

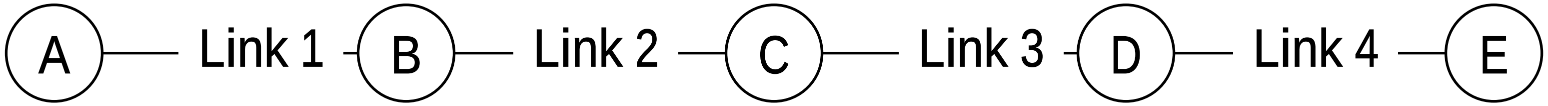
Next steps:

a	3	2	a	4	2	a	3	3	a	4	4
a	5	2	a	4	2	a	5	3	a	4	4
a	5	2	a	6	2	a	5	3	a	6	4
a	7	2	a	6	2	a	7	3	a	6	4

The count to infinity problem

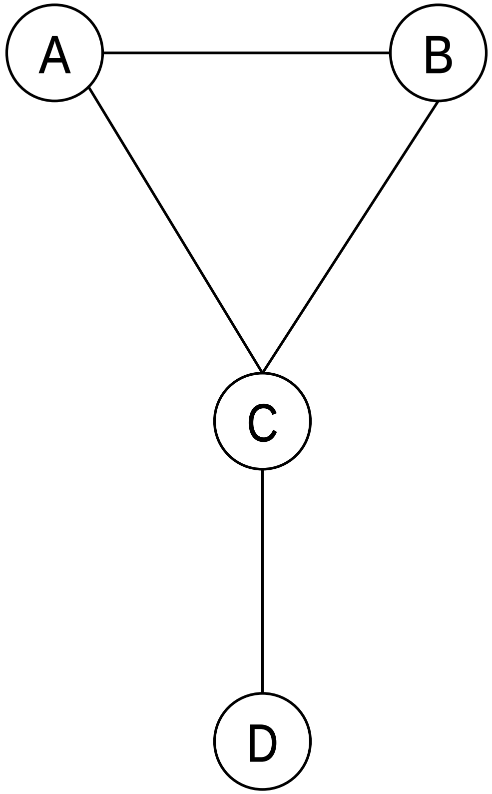
- ➔The problem is that NO ROUTERS have a value more than 1 + the minimum of adjacent routers
- ➔situation stabilizes only when count gets to infinity
- ➔it is more critical the higher infinity is set!

Split horizon "solution"



- ➔ **The distance is NOT reported on the line from which information comes**
  - ⇨ C tells correct distance to D, but lies (says infinity) to B
- ➔ **discovers link failure in 1 hop**

Split horizon failure



Line CD goes down…

1) because of split horizon rule,
   A and B tell C that dist(D)=inf
2) C concludes that D is unreachable
   and reports this to A and B
3) but A knows from B that dist(D)=2, and
   sets its dist=3
4) similarly, B knows from A distance from D
… etc until distance = infinite

Regardless the hack used, there is always a network topology that makes the trick fail!

OSPF
Open Shortest Path First

and Link state protocols in general

G.Bianchi, G.Neglia, V.Mancuso

Università degli Studi di Palermo

Link state routing protocol

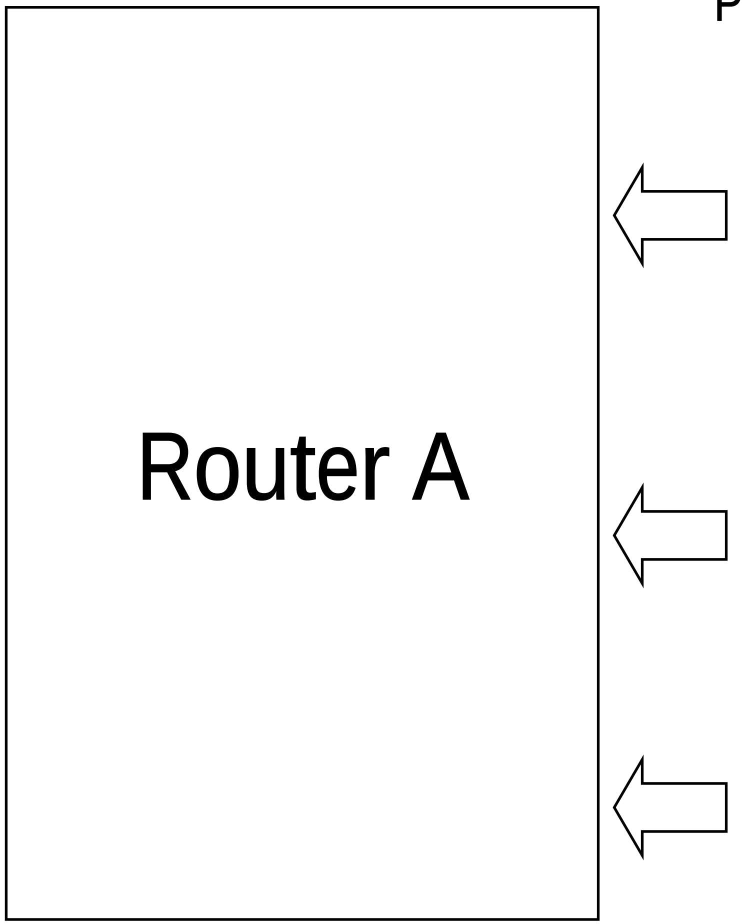
- ➔ **Each router must:**
  - ⇨ discover its neighbors
  - ⇨ measure a "cost" of the line connecting to the neighbor
    - → generally delay, e.g. via ICMP echo
    - → but may be link bandwidth, etc
  - ⇨ construct a packet containing the information about all the connected links
  - ⇨ send the packet to all the other routers

Flooding approach

When a router receives a packets:

- ➔ **Checks if the packet is new or a copy**
  - ⇨ a packet is new either if
    - →if it is first addressed to the node
    - →or if it was already received before, but THIS packet contains updated information
- ➔ **if old, destroys the packet;**
    - →to avoid duplicates and unuseful network load
- ➔ **if new, forwards the packets on all links except the one it came from,**
- ➔ **by evaluating each packets, the router dynamically reconstructs the network topology**

Example (1)



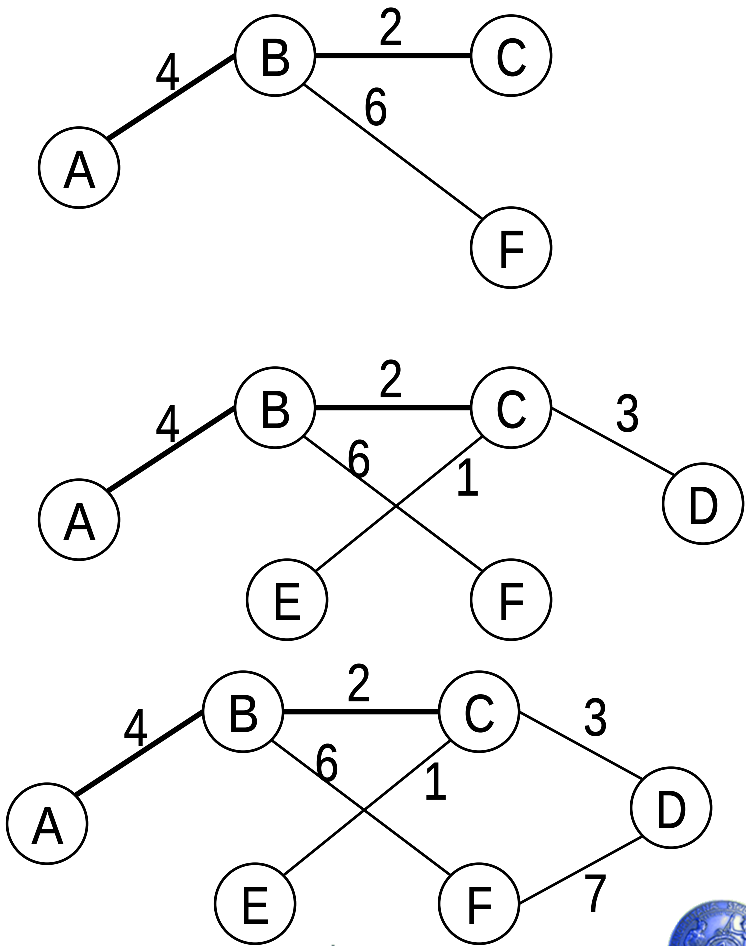
Packets received:

B	
a	4
c	2
f	6

C	
b	2
d	3
e	1

D	
c	3
f	7

Network discovery



Example (2)

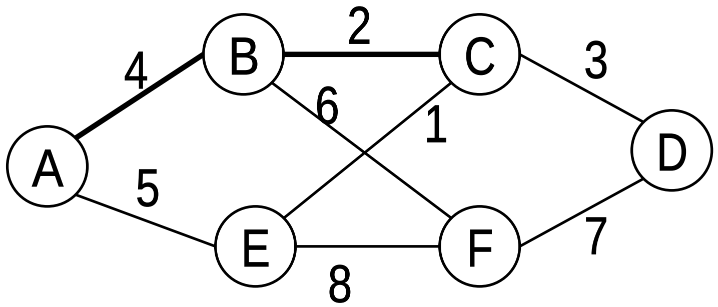
Router A

Packets received:

E	
a	5
c	1
f	8

F	
b	6
d	7
e	8

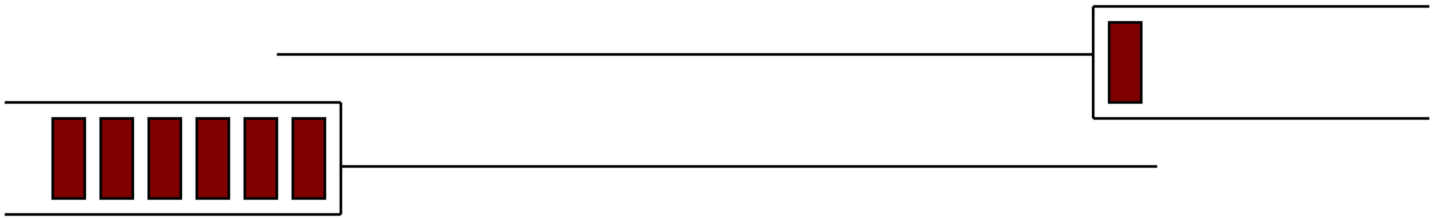
Network discovery



Confirm knowledge

Diverse cost per link

- ➔**Actually, the realistic case is a diverse cost per link**
  - ⇨in case of delay, queuing on one link is different from queuing on the reverse link!



No problem. The algorithm still works perfectly: every link will be represented twice: one packet per each direction!

Shortest Path Computation

- ➔ **Once discovered topology and link cost, shortest path trivially computed**
  - ⇨ Dijkstra algorithm

- ➔ **And routing tables built accordingly**

Advantages of link state protocols

- ➔ **Much faster convergence**
- ➔ **Do not need periodical update**
  - ⇨ but <u>event update</u>: packet transmission is triggered by link state changes
- ➔ **No count to infinity**
- ➔ **Do not need to transmit routing tables**
  - ⇨ only state of local links (but flooded)

Real-life protocols

- ➔ **Much more detailed**
- ➔ **Need to cope with all possible types of failures**
  - ⇨ link crashes and loss of flooding packets
    - ➔ use sequence number, ages
  - ⇨ router that “forget” to signal link state
  - ⇨ Inconsistent packet reports
    - ➔ due to route changes while network picture being worked out
- ➔ **Link state Internet protocols:**
  - ⇨ OSPF, OSPF-2, IS-IS (Intermediate System - Intermediate System)

OSPF features

- ➔ **OSPFv2: RFC 1247 (1991)**
- ➔ **Uses IP packets directly!**
  - ⇨ Uses own value in the protocol field of the IP header to allow demultiplexing at node
- ➔ **Can compute multiple routing tables for different TOS**
  - ⇨ min delay, max thr, max reliability, min cost
  - ⇨ no problem: based on cost associated to link!
- ➔ **Support subnets (uses netmasks)**
- ➔ **Allows authentication (via cleartext passwd transmission)**
- ➔ **When equal cost links found, OSPF uses load balancing**

How OSPF Works

- ➔ **Link failure detection**
  - ⇨ Not receiving HELLO message for long time
    - → Default, 40 seconds or 4 HELLO Intervals
- ➔ **If neighboring routers discover each other for the first time**
  - ⇨ Exchange their link-state databases
- ➔ **Synchronizing two neighbor's link-state databases**
  - ⇨ Default refresh information every 30 minutes

How OSPF Works

- ➔**The direct connected routers detect state change of the link**
  - ⇨Trigger the Link State Update to neighbors
  - ⇨Compute the shortest path
- ➔**Other routers flood the updates to whole network**
  - ⇨Use sequence number to detect redundant updates
  - ⇨Confirm the updates (Link State Acknowledge)
  - ⇨Compute the shortest path

OSPF --- Message Types

- **HELLO: Type 1**
  - Identify neighbors
  - Elect a designated route for a multi-access network
  - To find out about an existing designated router
  - "I am alive" signal

OSPF --- Message Type

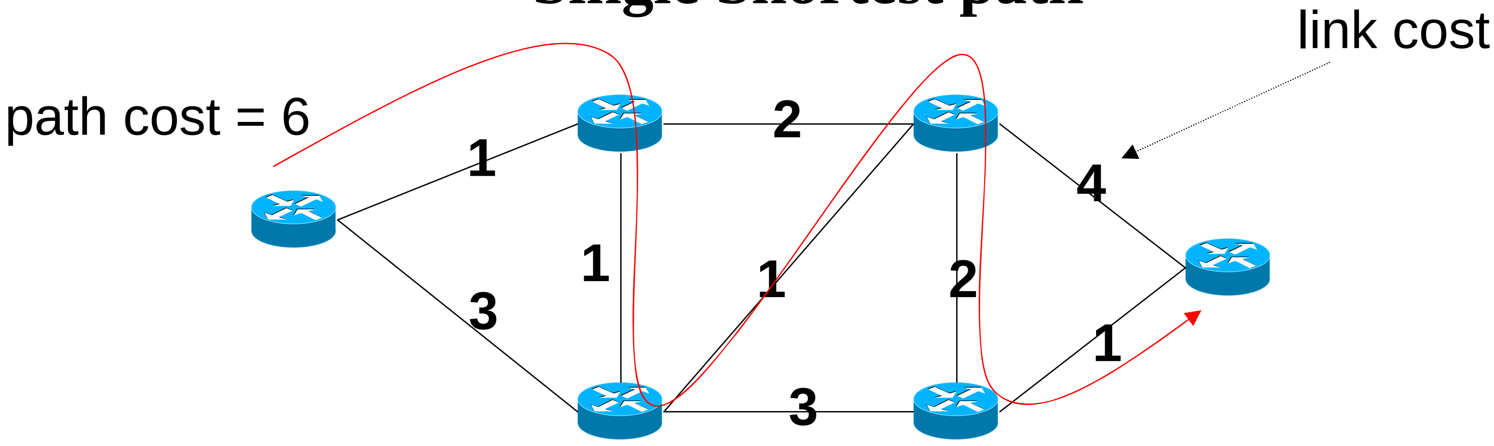
- ➔**Database Description: Type 2**
  - ⇨Exchange information during initialization
    - →So that a router can find out what data is missing from its topology database
- ➔**Link State Request: Type 3**
  - ⇨Ask for data that a router has discovered is missing from its topology databases or to replace data that is out of date

OSPF --- Message Type

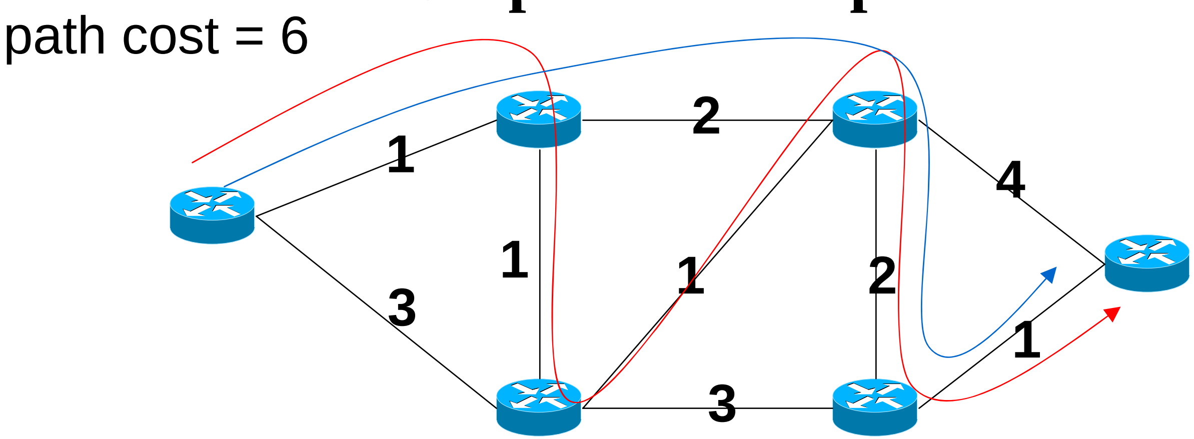
- ➔**Link State Update: Type 4**
  - ⇨Used to reply to a link state request and also to dynamically report changes in network topology
- ➔**Link State ACK: Type 5**
  - ⇨Used to confirm receipt of a link state update
  - ⇨Sender retransmit until an update is ACKed

Some remarks: Multiple shortest path

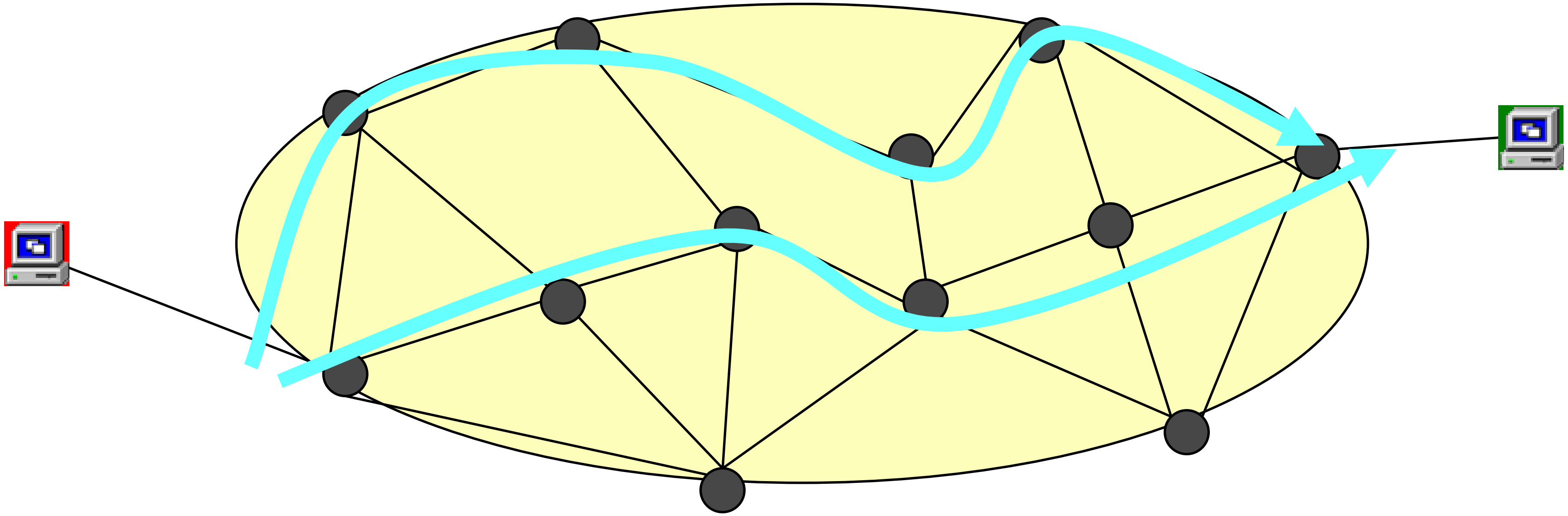
Single Shortest path

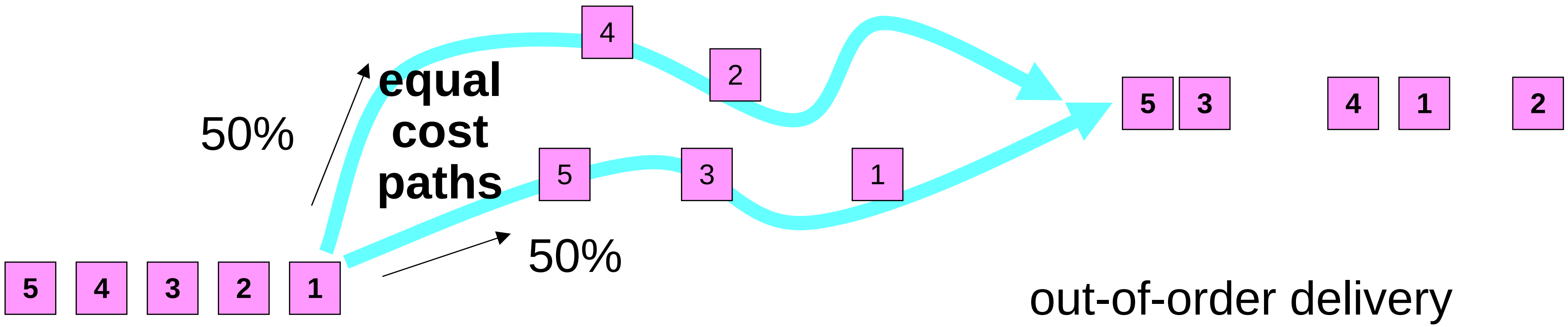


Multiple Shortest paths



Some remarks:
Load Balancing drawbacks

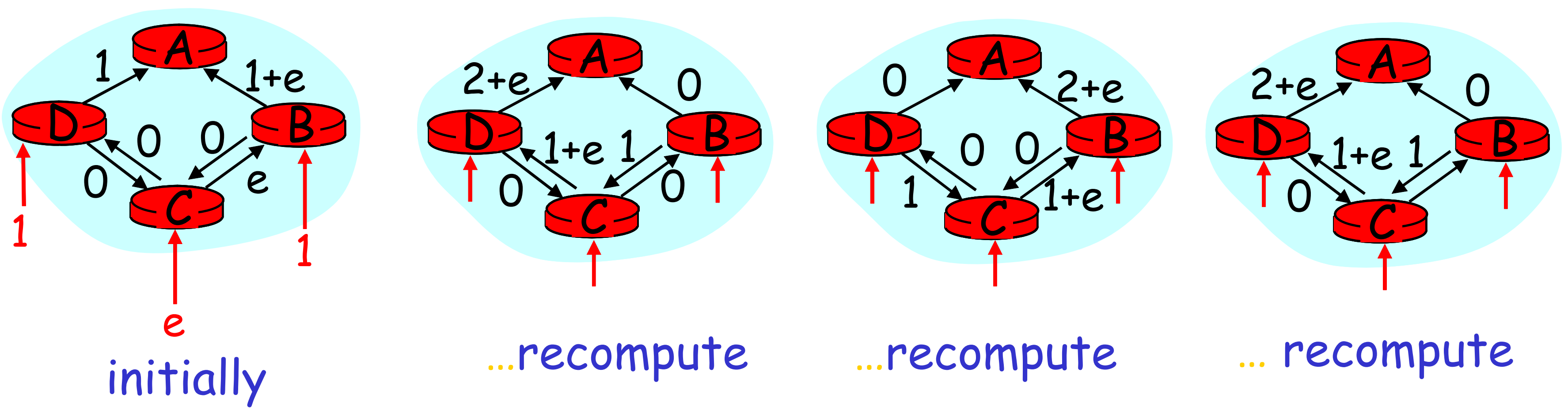




out-of-order delivery

Some remarks:
risks of traffic dependent costs

➔Route oscillation



Università degli Studi di Palermo

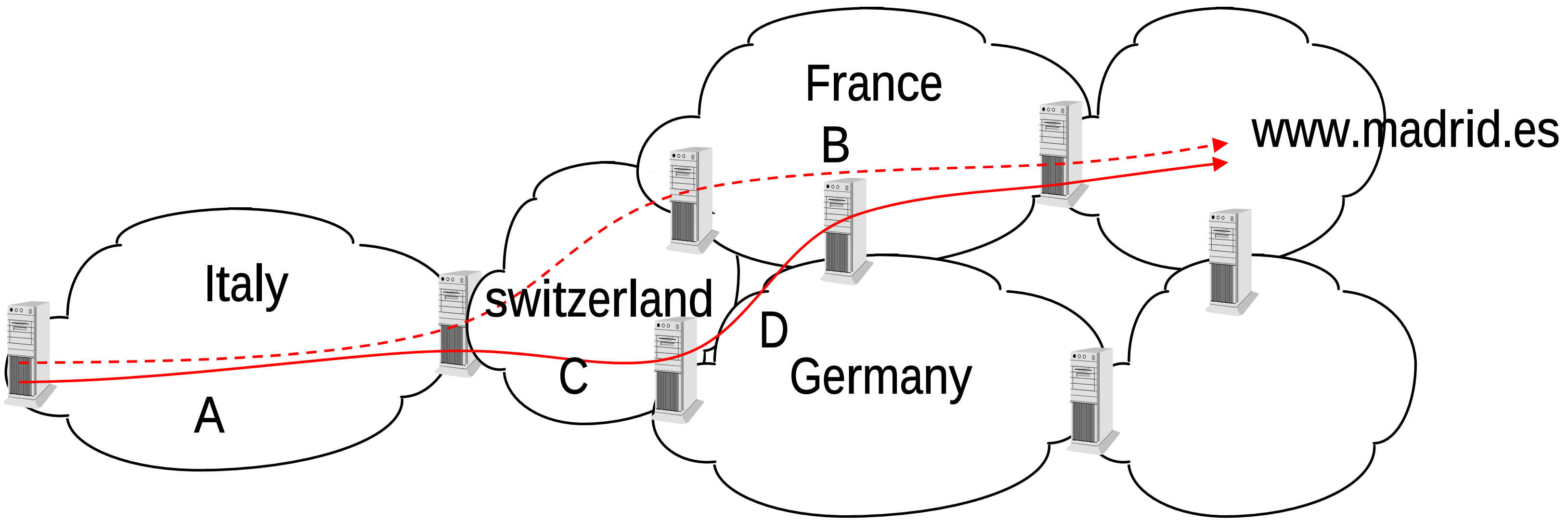
Exterior Gateway routing Protocols (EGPs)

Exterior Gateway Protocols

- ➔ **Have several "non technical" problems**
  - ⇨ policies: avoid routes which may be strategically critical
    - ➔ e.g. allowing IP packets to transit over corporate links
    - ➔ e.g. avoiding crossing critical states (midwest, etc)
  - ⇨ policies: manually configured
- ➔ **efficiency is secondary**
  - ⇨ most concern is reachability of networks
  - ⇨ a further efficiency problem: different ASs may use different metrics that cannot be compared…
- ➔ **First protocol used: EGP (!)**
  - ⇨ worked only for star topologies!!
- ➔ **Now replaced by BGP (v4)**

BGP view

Routers connected only when there is a network interconnecting them



How to compare path costs?

A,C may use RIP (hops), D may use OSPF delay, B may use IS-IS

Network classification by BGP

- **Stub Networks**
  - have only one BGP router
  - cannot be used for transit
- **Multiconnected Networks**
  - have more BGP routers
  - Might be used for transit, but refuse
- **Transit Networks**
  - backbones willing to handle packets of third parties (eventually with restrictions)

Inter BGP Router connections

- ➔ **Via TCP**
  - ⇨ to provide reliable information exchange
  - ⇨ and hide all the features of the underlying network(s)
- ➔ **Keepalive message exchange**
  - ⇨ periodically (about 30s), to check if peer router is up
- ➔ **"Distance" Vector approach, but exchange FULL path information**
  - ⇨ not only next hop information
  - ⇨ not vulnerable to count to infinity

Refer to RFC 1654 (BGP-4) for further information