

## Advanced Computer Networks

### Internal routing - distance vector protocols

Prof. Andrzej Duda  
duda@imag.fr

<http://duda.imag.fr>

1

## Contents

- Principles of internal routing
- Distance vector (Bellman-Ford)
  - principles
  - case of link failures
  - count to infinity
  - split horizon
- RIP
- RIP v2
- IGRP

2

## Routing algorithms

- Problem
  - find the **best** route to a destination
- What does it mean the best?
  - metric to measure how a route is good
  - hops
  - link capacity
  - performance measures: link load, delay
  - cost
- Graph optimization - Shortest Path
  - find the shortest path in a graph
  - shortest in the sense of a metric

3

## Main algorithms

- Distance vector (Bellman-Ford)
  - routers only know their local state
    - link metric and neighbor estimates
  - internal routing protocols (RIP, IGRP)
- Link state
  - knowledge of the global state
    - metrics of all links
    - global optimization (Shortest Path First - Dijkstra)
  - internal routing protocols (OSPF, PNNI (ATM))
- Path vector
  - knowledge of the global state
    - path: sequence of AS with attributes
    - global optimization and policy routing
  - external routing protocols (BGP)

4

## Routing protocols

|      | Internet                              | ISO   |
|------|---------------------------------------|-------|
| IGP  | distance vector: RIP, RIP v2,<br>IGRP |       |
|      | link state: OSPF<br>dual: EIGRP       | IS-IS |
| EGP  | EGP (obsolete)<br>BGP                 | IDRP  |
| host | ICMP Redirect                         | IS-ES |

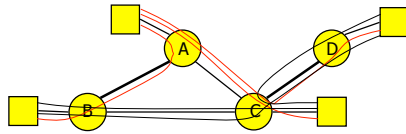
5

## Metrics

- Static - do not depend on the network state
  - number of hops
  - link capacity and static delay
  - cost
- Dynamic - depend on the network state
  - link load
  - current delay

6

## Traffic matrix



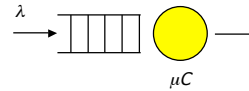
|   | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 8 | 2 | 3 |
| B |   | 0 | 6 | 4 |
| C |   |   | 0 | 1 |
| D |   |   |   | 0 |

|   | A | B  | C  | D   |
|---|---|----|----|-----|
| A |   | AB | AC | ACD |
| B |   |    | BC | BCD |
| C |   |    |    | CD  |
| D |   |    |    |     |

7

## Traffic

- Link model
  - queueing system M/M/1
    - exponentially distributed service and interarrival times



$$T = \frac{1}{\mu C - \lambda}$$

8

## Delay

- Parameters
  - 1 Mb/s and 0.5 Mb/s links
  - mean packet length  $1/\mu = 5$  Kbytes (40 000) bits
  - transmission time on 1 Mb/s link: 40 ms
  - transmission time on 0.5 Mb/s link: 80 ms

|    | $\lambda$ pq/s | C Mb/s | $\mu C$ pq/s | T      |
|----|----------------|--------|--------------|--------|
| AB | 8              | 1      | 25           | 58 ms  |
| AC | 5              | 0.5    | 12.5         | 133 ms |
| BC | 10             | 0.5    | 12.5         | 400 ms |
| CD | 8              | 1      | 25           | 58 ms  |

9

## Flooding

- Simple and robust routing
  - no need for routing tables
  - each packet duplicated on each outgoing link
  - packet duplication
  - duplicated packets destroyed at destination
  - robust - tolerates link or router failures
  - optimal in some sense
    - the first packet has found the shortest path to the destination
    - cannot be compared to the shortest path calculated by Link State - no packet duplication
- Problem
  - increased load due to packet duplication
- Used in OSPF to distribute link state information and in ad hoc routing protocols (AODV, OLSR)

10

## Distance vector

- Dynamic routing based on distributed estimation of the distance to the destination
  - uses the distributed algorithm by Bellman-Ford (dynamic programming)
  - each router receives aggregated information from its neighbors
  - estimates the local cost to its neighbors
  - computes the best routes
  - no global network states
- Distance
  - number of hops
  - delay

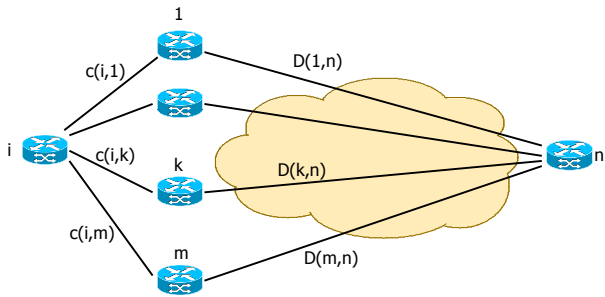
11

## Bellman-Ford algorithm

- Bellman-Ford algorithm
  - node  $i$  knows cost  $c(i,k)$  to its immediate neighbours ( $+\infty$  for most values of  $k$ )
  - distance  $D(i,n)$  is given by:  $D(i,n) = \min_k (c(i,k) + D(k,n))$
  - in the worst case, convergence after  $N-1$  iterations
- Distributed Bellman-Ford algorithm
  - initially:  $D(i,n) = 0$  if  $i$  directly connected to  $n$  and  $D(i,n) = +\infty$  otherwise
  - node  $i$  receives from neighbour  $k$  latest values of  $D(k,n)$  for all  $n$  (distance vector)
  - node  $i$  computes the best estimates
 
$$D(i,n) = \min_k (c(i,k) + D(k,n))$$

12

## Bellman-Ford algorithm



13

## Example of Bellman-Ford

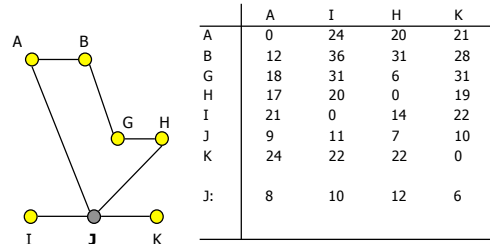


Table of J

|    |   |
|----|---|
| 8  | A |
| 20 | A |
| 18 | H |
| 12 | H |
| 10 | I |
| 0  | - |
| 6  | K |

computation of G :  $18+8=26$ ,  $31+10=41$ ,  $6+12=18$ ,  $6+31=37$   
 → choice of 18, H

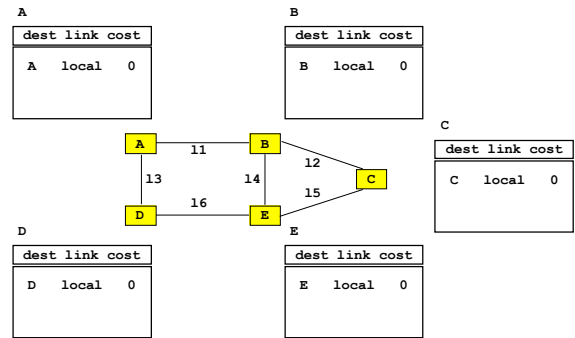
14

## Distance vector example

- Simple network
  - routers connected by links
  - destinations = subnetworks connected to routers
  - symmetric links
  - cost = number of hops

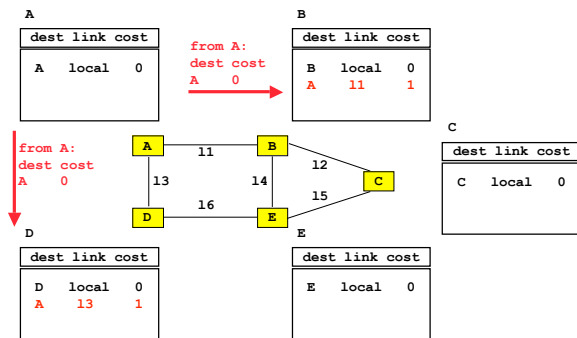
15

## Initialization



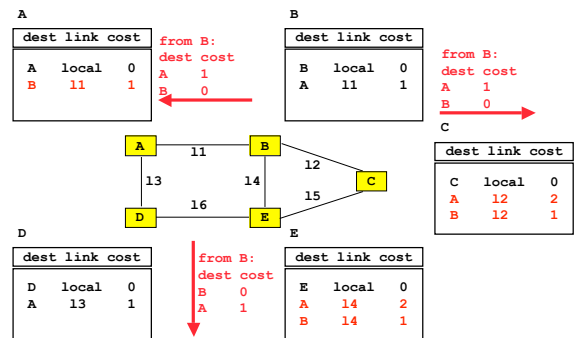
16

## Distance vector announcement



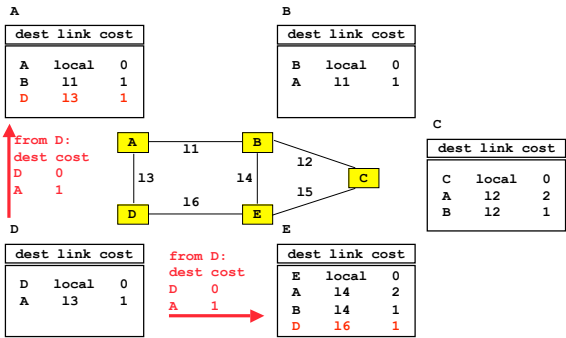
17

## Distance vector announcement



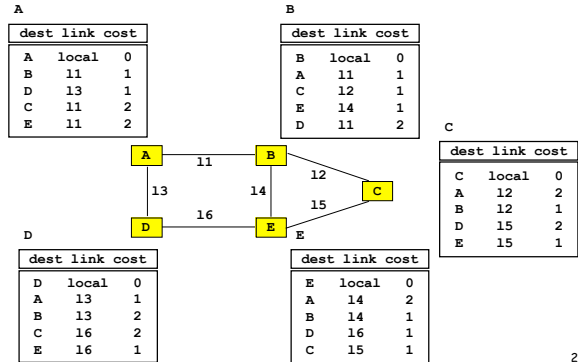
18

### Distance vector announcement



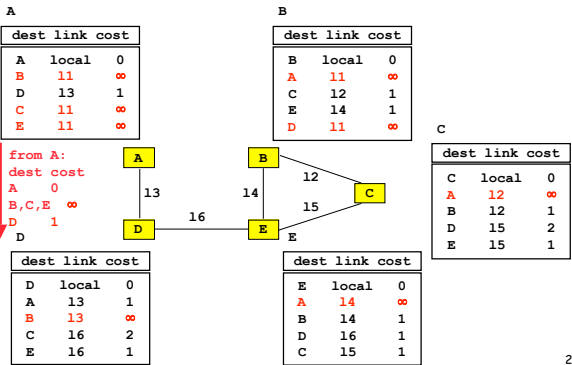
19

### Final



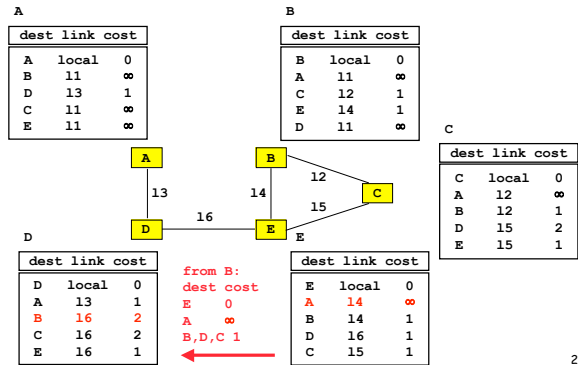
20

### Link failure



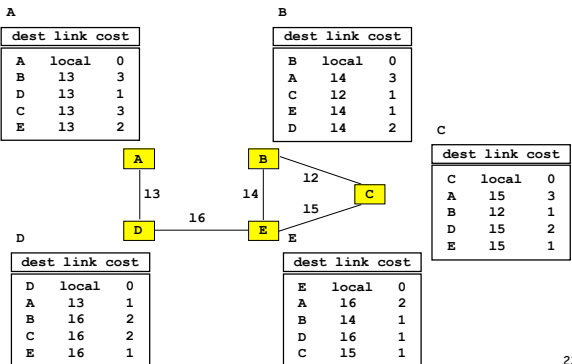
21

### Link failure



22

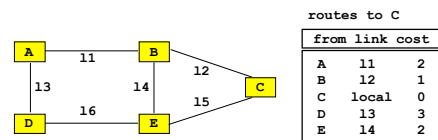
### Final state after failure



23

### Different link cost

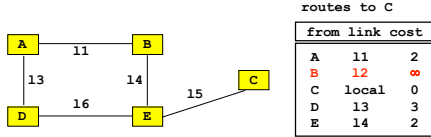
- Cost of link 5 = 5



24

## Link failure

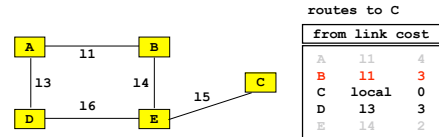
- Link 2 fails and B updates its table



25

## Link failure

- Just before B updates its table, A broadcasts its table with cost 2 to C
- B accepts this - link 1, cost 3 and sends updated vector to A and E

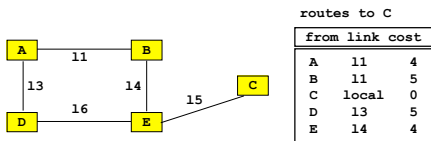


26

- Loop between A and B!
- Bounce effect** - transient loop until stable state

## Towards a stable state

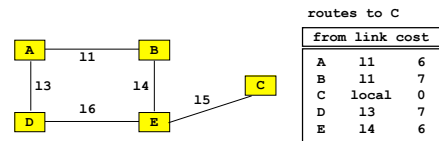
- C sends its vector but it is ignored because cost = 5
- A and E broadcast their vectors, B and D increase the costs



27

## Towards a stable state

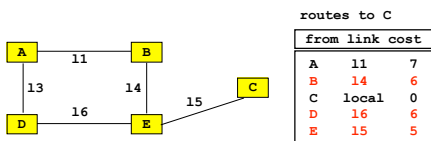
- B and D broadcast their vectors, the cost is increased



28

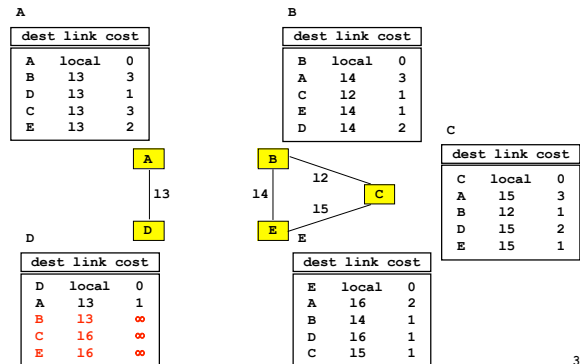
## Stable state

- C broadcasts its vector, table becomes now
- No loop!
- Slow convergence - increase by 2 for each cycle



29

## Equal link costs - link failures



30

### Counting to infinity

A

| dest | link  | cost |
|------|-------|------|
| A    | local | 0    |
| B    |       | 13   |
| D    |       | 13   |
| C    |       | 13   |
| E    |       | 13   |

from A:  
dest cost

|     |   |
|-----|---|
| A   | 0 |
| B,C | 3 |
| D   | 1 |
| E   | 2 |

D

| dest | link  | cost |
|------|-------|------|
| D    | local | 0    |
| A    |       | 13   |
| B    |       | 13   |
| C    |       | 13   |
| E    |       | 13   |

- Loop between A and D
- Exchange of routes, costs increase by 2 each cycle
- Convergence to a stable state
  - $\infty$  = large number
  - e.g. RIP:  $\infty$  = 16

31

### Split horizon

- Minimize the effects of bouncing and counting to infinity
- Rule
  - if A routes packets to X via B, it does not announce this route to B

32

### Example of split horizon

A

| dest | link  | cost |
|------|-------|------|
| A    | local | 0    |
| B    |       | 13   |
| D    |       | 13   |
| C    |       | 13   |
| E    |       | 13   |

B

| dest | link  | cost |
|------|-------|------|
| B    | local | 0    |
| A    |       | 14   |
| C    |       | 12   |
| E    |       | 14   |
| D    |       | 14   |

C

| dest | link  | cost |
|------|-------|------|
| C    | local | 0    |
| A    |       | 15   |
| B    |       | 12   |
| D    |       | 15   |
| E    |       | 15   |

D

| dest | link  | cost |
|------|-------|------|
| D    | local | 0    |
| A    |       | 13   |
| B    |       | 13   |
| C    |       | 16   |
| E    |       | 16   |

E

| dest | link  | cost |
|------|-------|------|
| E    | local | 0    |
| A    |       | 16   |
| B    |       | 14   |
| D    |       | 16   |
| C    |       | 15   |

33

### Split horizon

A

| dest | link  | cost |
|------|-------|------|
| A    | local | 0    |
| B    |       | 13   |
| D    |       | 13   |
| C    |       | 13   |
| E    |       | 13   |

from A:  
dest cost

|   |   |
|---|---|
| A | 0 |
|---|---|

D

| dest | link  | cost |
|------|-------|------|
| D    | local | 0    |
| A    |       | 13   |
| B    |       | 13   |
| C    |       | 16   |
| E    |       | 16   |

- Split horizon cuts the process of counting to infinity

34

### Split horizon

A

| dest | link  | cost |
|------|-------|------|
| A    | local | 0    |
| B    |       | 13   |
| D    |       | 13   |
| C    |       | 13   |
| E    |       | 13   |

from D:  
dest cost

|       |          |
|-------|----------|
| D     | 0        |
| B,C,E | $\infty$ |

D

| dest | link  | cost |
|------|-------|------|
| D    | local | 0    |
| A    |       | 13   |
| B    |       | 13   |
| C    |       | 16   |
| E    |       | 16   |

- Split horizon cuts the process of counting to infinity

35

### Split horizon may fail

B

| dest | link  | cost |
|------|-------|------|
| B    | local | 0    |
| A    |       | 14   |
| C    |       | 12   |
| E    |       | 14   |
| D    |       | 14   |

from E:  
dest cost

|   |          |
|---|----------|
| A | $\infty$ |
| B | 1        |
| C | 1        |
| D | $\infty$ |

C

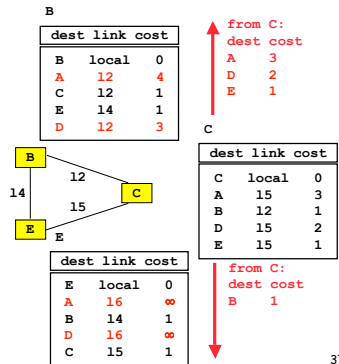
| dest | link  | cost |
|------|-------|------|
| C    | local | 0    |
| A    |       | 15   |
| B    |       | 12   |
| D    |       | 15   |
| E    |       | 15   |

E

| dest | link  | cost |
|------|-------|------|
| E    | local | 0    |
| A    |       | 16   |
| B    |       | 14   |
| D    |       | 16   |
| C    |       | 15   |

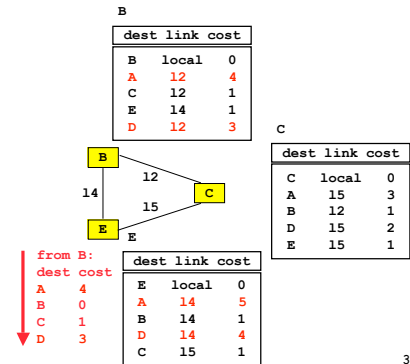
36

## Split horizon may fail



37

## Split horizon may fail



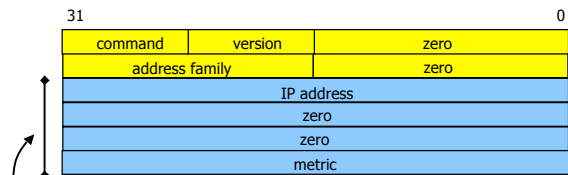
38

## RIP v1

- Distance vector protocol
- Metric - hops
- Network span limited to 15
  - ∞ = 16
- Split horizon
- Destination network identified by IP address
  - no prefix/subnet information - derived from address class
- Encapsulated as UDP packets, port 520
- Largely implemented (routed on Unix)
- Broadcast every 30 seconds or when update detected
- Route not announced during 3 minutes
  - cost becomes ∞

39

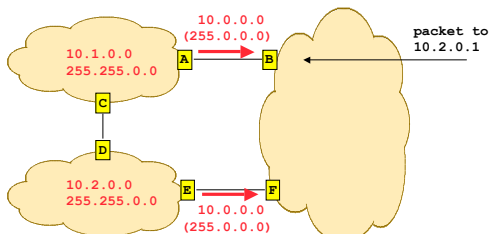
## Message format



- May be repeated 25 times
- Command
  - REQUEST - 1 (sent at boot to initialize)
  - RESPONSE - 2 (broadcast each 30 sec)

40

## Missing netmask



- A and E can forward to 10.0.0.0
- Packet to 10.2.0.1 can go through F or B
  - if sent to B, it goes through A and C
- If link C-D broken, no route to destination

41

## RIP v2 (RFC 2453)

- Subnetworks
  - take into account CIDR prefixes and netmasks
- Authentication
- Multicast
  - 224.0.0.9 mapped to MAC 01-00-5E-00-00-09
  - on LAN only, no need for IGMP

42

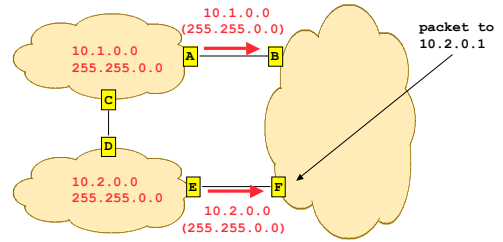
## Message format

|                |         |           |  |
|----------------|---------|-----------|--|
| 31             |         | 0         |  |
| command        | version | unused    |  |
| address family |         | route tag |  |
| IP address     |         |           |  |
| netmask        |         |           |  |
| next router    |         |           |  |
| metric         |         |           |  |

- Command, version unchanged
- One address family - authentication
- Next router
  - used at the border of different routing domains (e.g. RIP and OSPF)
- Route tag
  - for external routes (used by BGP)

43

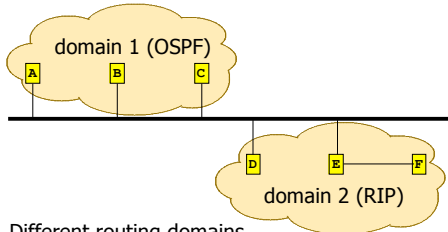
## Announcing netmasks



- E can forward to 10.2.0.0
- Packet to 10.2.0.1 can go through F

44

## Routing domains



- Different routing domains
  - e.g. routers under different administrations that run different routing protocols (RIP, OSPF)
- If A wants to send a packet to F, it goes through D and E
- When announcing F, D adds E as **next router**

45

## Simple authentication

|                      |         |                         |  |
|----------------------|---------|-------------------------|--|
| 31                   |         | 0                       |  |
| command              | version | unused                  |  |
| xFFFF                |         | authentication type = 2 |  |
| password on 16 bytes |         |                         |  |

- Configuration of gated (/etc/gated.conf)
 

```
rip yes {
    interface all
    version 2 multicast
    authentication simple "qptszwmz"
}
```

46

## MD5 authentication

|                         |         |                         |  |
|-------------------------|---------|-------------------------|--|
| 31                      |         | 0                       |  |
| command                 | version | unused                  |  |
| xFFFF                   |         | authentication type = 3 |  |
| packet length           | key Id  | auth. length            |  |
| increasing sequence no. |         |                         |  |
| zero                    |         |                         |  |
| zero                    |         |                         |  |
| route info              |         |                         |  |
| xFFFF                   |         | x01                     |  |
| seal                    |         |                         |  |

47

## MD5 authentication

- Seal
  - MD5 digest on the message using a shared secret
  - sequence number avoids replay attacks
- Configuration of gated (/etc/gated.conf)
 

```
rip yes {
    interface all
    version 2 multicast
    authentication md5 "qptszwmz"
}
```

48

## IGRP (Interior Gateway Routing Protocol)

- Proprietary protocol by CISCO
- Metric that estimates the global delay
- Maintains several routes of similar cost
  - load sharing
- Takes into account netmasks
- No limit of 15
  - number of routers included in messages
- Broadcast every 90 sec

49

## Metric example



- **Metric**
  - $Trans = 10000000 / \text{Bandwidth}$  (time to send 10 Kb)
  - $delay = (\text{sum of Delay}) / 10$
  - $m = [K_1 * Trans + (K_2 * Trans) / (256 - \text{load}) + K_3 * \text{delay}]$
  - default:  $K_1=1, K_2=0, K_3=1, K_4=0, K_5=0$
  - if  $K_5 \neq 0, m = m * [K_5 / (\text{Reliability} + K_4)]$
- **Bandwidth in Kb/s, Delay in  $\mu s$** 
  - At Venus: Route for 172.17/16: Metric =  $10000000 / 784 + (20000 + 1000) / 10 = 14855$
  - At Saturn: Route for 12./8: Metric =  $10000000 / 224 + (20000 + 1000) / 10 = 46742$

50

## Conclusion

- Main distance vector protocols
- Largely deployed (Unix BSD **routing**)
- Simplicity
- Slow convergence
- Not suited for large and complex networks
  - Link State protocols should be used instead

51

52