

Networking Technologies and Applications

Rolland Vida
BME TMIT



Routing - Router



- **Routing**
 - Process through which the packets are directed to the destination node
 - Based on the routing tables and the used routing protocols, the internal routers determine the path
- **Router**
 - The node handling the routing process
 - Communicate with each other
 - Receive and store information from their neighbors
 - Create and maintain **routing tables**
 - Content: <destination address, outgoing interface> pairs



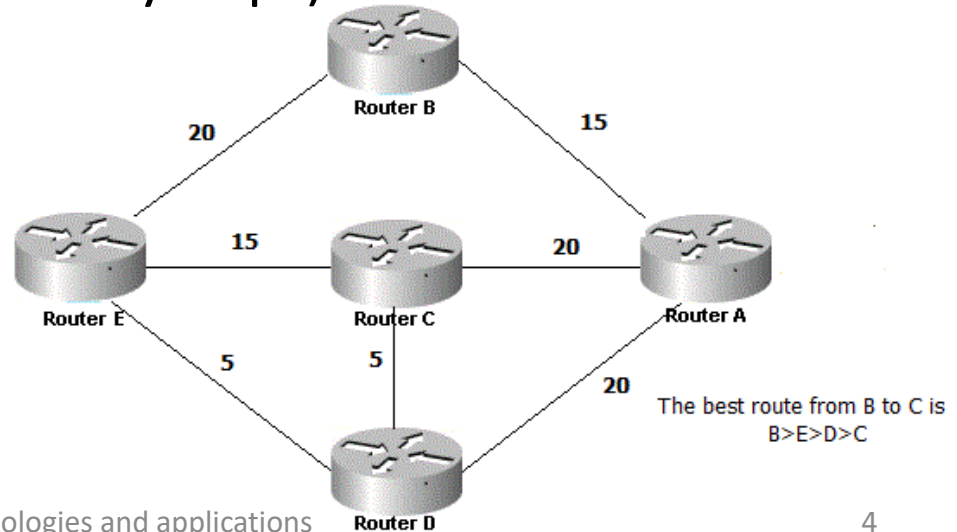
Router

- The router can be
 - a module of the operating system
 - Unix, Novell
 - Dedicated device (not only software, but hardware as well) – much faster
 - Cisco, Juniper, Alcatel-Lucent, Huawei, NEC, etc.
- **Capacity of a router**
 - How many packets can be transmitted in a time interval (packet/s)
 - E.g. Alcatel Lucent 7750 SR
 - 9.6 Tb/s, 10700 Mpps
 - Routing table – 22.000.000 (IPv4), 12.000.000 (IPv6)



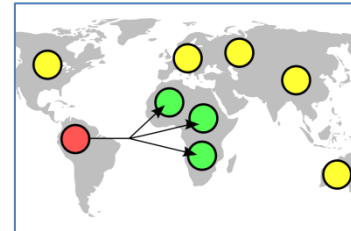
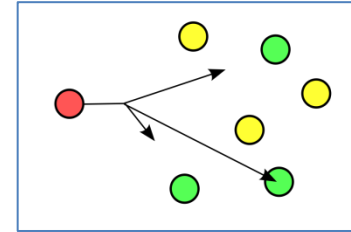
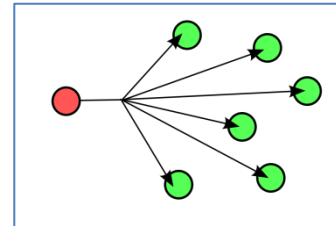
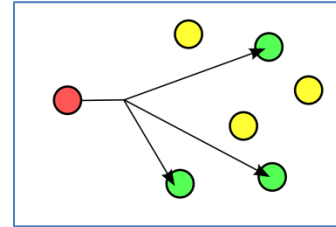
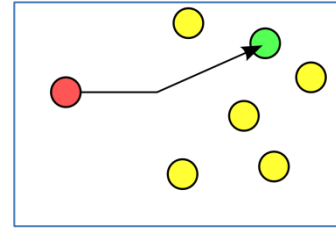
Tasks of a router

- Selecting the optimal path for a given packet
- Based on several aspects (metrics):
 - Length of the route (how many hops)
 - Cost
 - Bandwidth, speed
 - Reliability
 - Delay



Routing semantics

- **Unicast** – sending a packet to one specific destination
- **Anycast** – sending a packet to anyone (e.g., the closest one) from a group
- **Multicast** – sending a packet to a group
- **Geocast** – sending a packet to everyone in a given geographical area
- **Broadcast** – sending a packet to everyone in the (sub)network



Classification of routing protocols

- **Static:**
 - the routing table filled manually
 - Never refreshed automatically
- **Dynamic:**
 - The routers communicate with each other, routing tables are built dynamically, based on the current network topology
- **Single path:**
 - One single path stored towards each destination
- **Multi path:**
 - Many (or all) paths stored towards each destination
 - These protocols can handle load balancing

Classification of routing protocols

- **Flat:**
 - Each router knows about every destination
 - Old model, for smaller networks
- **Hierarchical:**
 - Routers do not know the path towards each destination
 - If an unknown destination address is seen, the packet is directed towards a well known direction (default route)
 - The size of the routing tables remains scalable
- **Inter-domain**
 - Responsible for routing the packet between domains
- **Intra-domain**
 - Responsible for routing inside a domain

Classification of routing protocols

- **Hop-by-hop:**
 - Each router decides where to forward the packet in an autonomous way
 - Based on (partial) topology information gathered from the neighbors
- **Source routing:**
 - The sender decides the route of the packet (and includes it in the IP packet header)
 - Routers only advertise availability information
 - Packets are just forwarded based on the header, no routing decision is taken
- They are intermediate solutions as well

Classification of routing protocols

- **Distance vector protocols**
 - Routers communicate only with their neighbors
 - Each routers tells its neighbor:
 - What is the cost of the route he knows to a given destination
 - Does not specify what is that route, who is the next hop
 - Routers gather the ads from their neighbors, and choose the node that advertised the cheapest route
 - Packets are directed towards this neighbor
 - They add their own cost, and advertise the updated route information

Classification of routing protocols

- Link state protocols

1. Discover the network topology
2. Find the shortest path in this graph

Routers advertise the status of their interfaces (i.e., the costs of their links)

- Information is exchanged with all the other routers in the network
- Everyone builds his own network topology
 - Everyone builds the same topology

Distance-Vector Protocols

Bellman-Ford protocols

Classical Bellman-Ford algorithm

$d_{ij} :=$ the cost of link i - j (infinity, if no link)

- Real cost, delay, packet loss rate, etc.

- **Additivity**

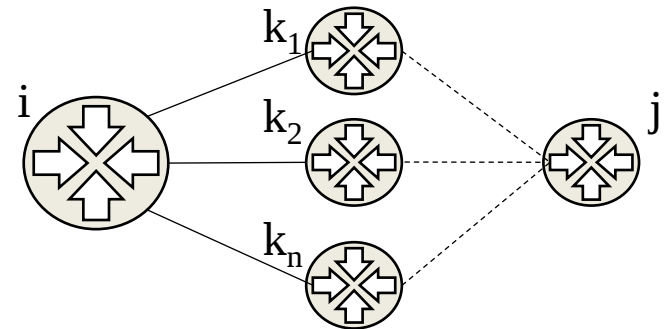
- The cost of a route is the sum of the costs of the links composing that route

$D_{ij} :=$ minimum cost between i and j

Bellman equation:

$D_{ii} = 0$, for each i

$D_{ij} = \min_k \{d_{ik} + D_{kj}\}$



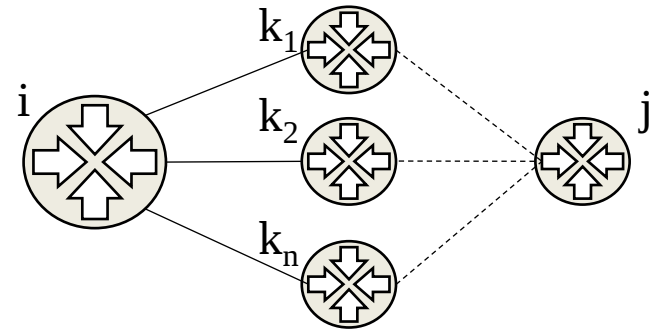
Distributed Bellman-Ford Algorithm

$D_{kj}^i(t)$ = minimal distance from k to j –ig, that router i is aware of at time t

$D_{ii} = 0$, for each i

$D_{ij}(t) = \min_k \{d_{ik} + D_{kj}^i(t)\}$

- The algorithm can run autonomously in each router



Distance-vector protocols

- RIPv1 (RFC 1058, '88)
 - Routing Information Protocol
 - Rest In Pieces ☺
- RIPv2 (RFC 2453, '98)
- RIPv6 (RFC 2080, '97)
 - IPv6 version
- EIGRP
 - Enhanced Interior Gateway Routing Protocol
 - Cisco proprietary standard

Distance Vector protocols

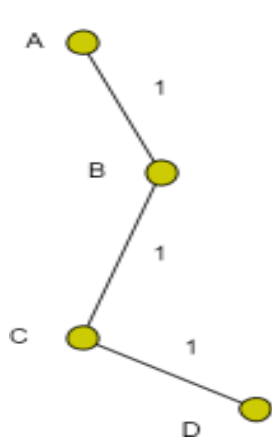
- Store distance vectors for each route
 - Data triples:
 - Destination
 - Cost
 - Next hop node (where to forward)
 - Periodically refreshed among neighbors
 - Update messages (2 parts):
 - Destination, cost
 - If a router learns about a better path, it updates its table
 - Learns about a new neighbor, or learns a better path from an old neighbor
 - The information spreads (slowly)

Properties

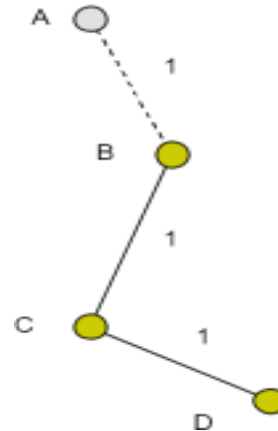
- Simple, but not perfect:
 - Link costs can change
 - Links can be broken
 - Cost of a broken link set to infinity
 - An integer value that is larger than any real possible value (by default, 16 for RIP)
 - In case of topology change, routing tables are refreshed gradually
 - Periodically (e.g., each 30 s) update message sent
 - If 6 updates are missed, cost set to infinity
 - Neighbors also update their entries
 - Converges, but slowly
 - Can be used only in small networks

Counting to infinity

- When advertising the costs of reaching a destination, costs can be incremented endlessly



B	C	D
1	2	3



B	C	D
3	2	3

B	C	D
3	4	3

B	C	D
5	4	5

Solution

- Split horizon method
 - If C learns a route from B, it will not advertise it back to B
- Poisoned Reverse method
 - If C learns a route from B, it will advertise it back to B with a cost set to infinity