Networking Technologies and Applications

Rolland Vida
BME TMIT

October 17, 2017

Transition to IPv6

- Routing services built on IP
  - RIPv6(ng), OSPFv6 (v3), BGPv6
- Network and transport layer protocols built on IP
  - TCPv6, UDPv6, RSVPv6
- Applications
  - Each application that was using directly the IPv4 addresses is not independent from the lower layers, so IPv6 support should be implemented in it
- Gradual transition
  - No „D-day”
- Expectations regarding transition
  - No transition dependencies
    - The transition of a given node can be done independently from the others
    - The most important aspect is backward compatibility
  - It should be as easy as possible for the end user
  - The different transition solutions should be appliable independently of each other
    - At least at the level of the different domains

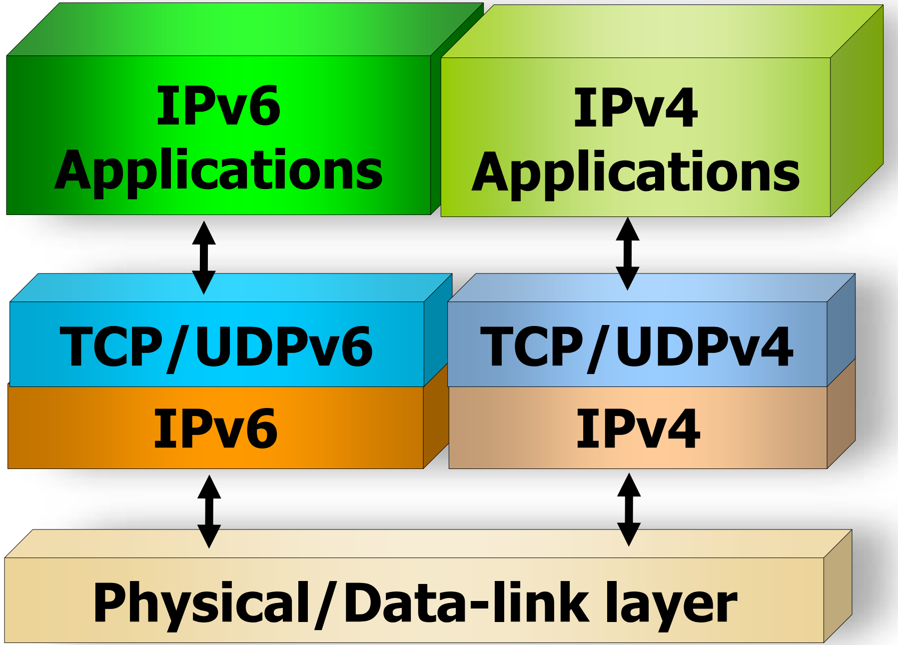
Transition solutions

- Dual Stack
  - Both IPv4 and IPv6 stack on the same device
- Tunnels
  - Initially tunneling IPv6 packets in IPv4 domains
  - Later, tunneling IPv4 packets in IPv6 domains
- Protocol translation
  - Headers containing protocol information should be translated into different protocol headers, based on certain translation rules
  - IPv6 <-> IPv4

Dual protocol stack

- The first step towards deploying IPv6 is deploying some nodes that support IPv6 (as well, next to IPv4)
  - They have a double stack strategy
    - Use IPv6 to communicate with other IPv6 systems
    - Can switch back to IPv4 mode to talk to IPv4 systems

Dual Stack


IPv6 Applications
IPv4 Applications
TCP/UDPv6
TCP/UDPv4
IPv6
IPv4
Physical/Data-link layer

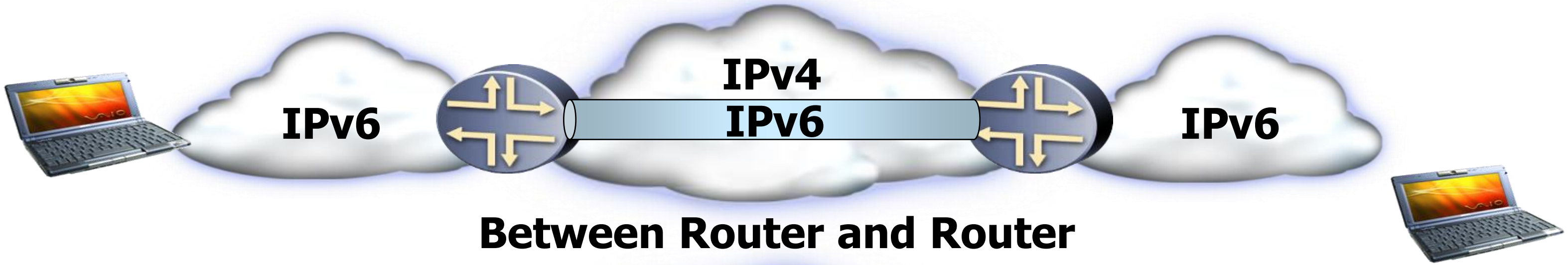
Dual stack

- Advantages
  - Easy to install, configure, maintain
  - The entire functionality of IPv6 can be exploited
  - Any two nodes can communicate exclusively with IPv4 or IPv6 packets
  - Transparent transition for the end users
- Drawbacks
  - Not scalable: each node should have an IPv6 and an IPv4 address, the limitation of the IPv4 address domain obstructs its spreading
  - The size of the routing tables is increased in the routers
  - Not flexible: no communication possibility between nodes speaking just IPv4 or just IPv6

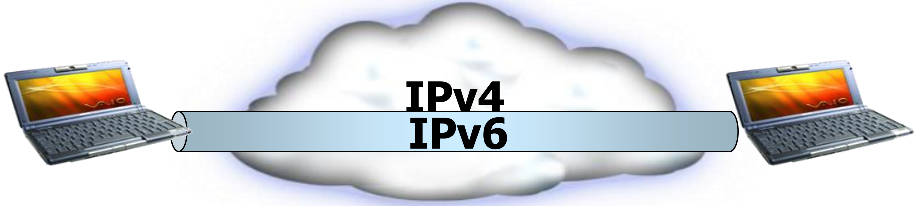
Tunneling

- IPv6 packet encapsulated inside an IPv4 packet
- The tunnel endpoints manage the encapsulation
- The process transparent to the intermediate nodes
- Configured tunnels
  - The tunnel endpoints are explicitely configured
  - They are dual stack nodes
  - Should use global IPv4 addresses, no NAT between the endpoints
- Automatic tunnels
  - The tunnel endpoint are automatically discovered by the network
  - Tunnel Brokers (RFC3053)
  - 6to4 (RFC3056)
  - ISATAP (Intra-Site Automatic Tunnel Addressing Protocol)
  - 6over4 (RFC2529)
  - Teredo: support tunnels through IPv4 NAT

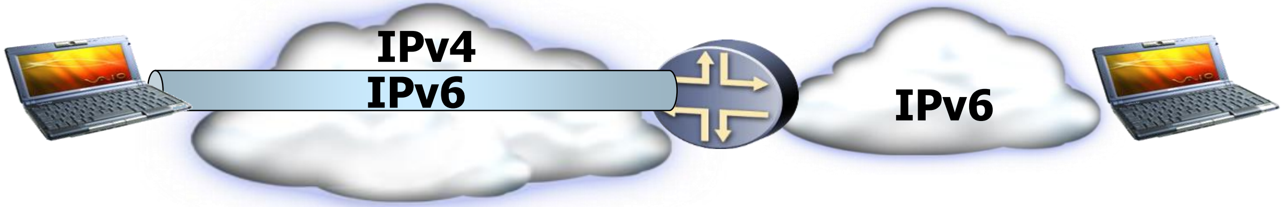
Tunneling



Between Router and Router



Between Host and Host



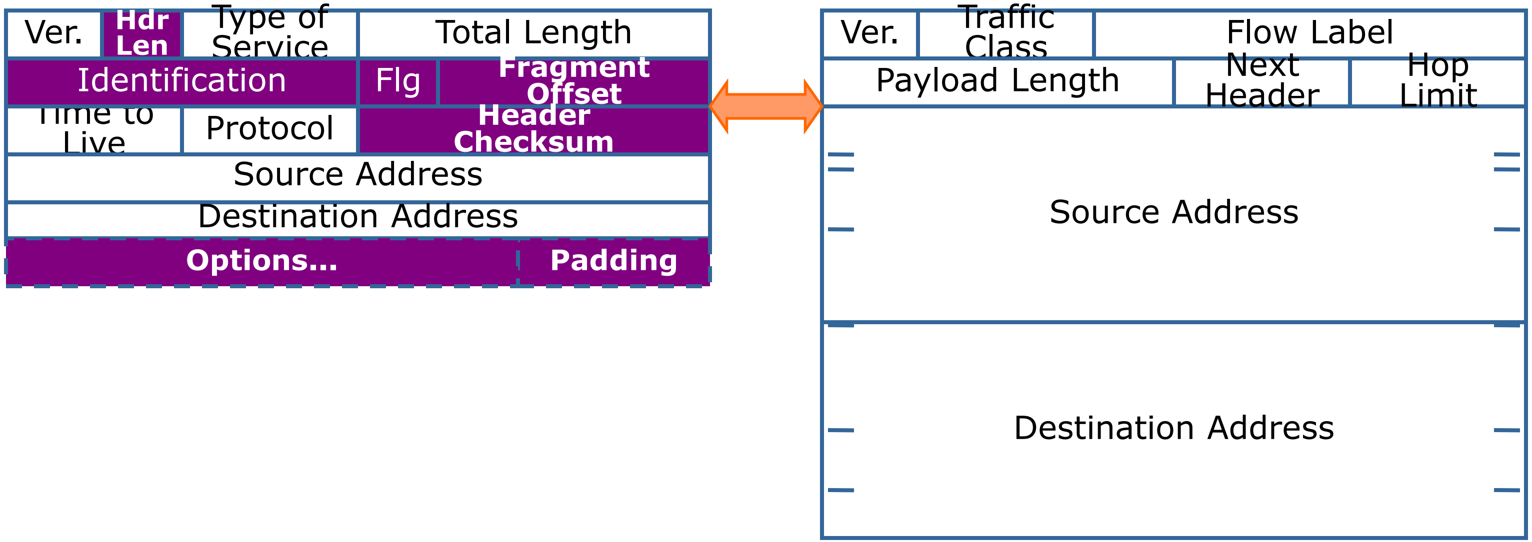
Between Host and Router / Router and Host

Translators

- Network layer translators
  - SITT (Stateless IP/ICMP Translator Algorithms) (RFC2765)
  - NAT-PT (Network Address Translator-Protocol Translator) (RFC2766)
  - BIS (Bump int the Stack) (RFC2767)
- Transport layer translator
  - TRT (Transport Relay Translator) (RFC3142)
- Application layer translators
  - BIA (Bump in the API) (RFC3338)
  - SOCKS64 (RFC3089)
  - ALG (Application Level Gateway)

Network layer translators

- The IPv4 messages are translated into IPv6 messages, and vice-versa (especially the headers)



Routing - Router

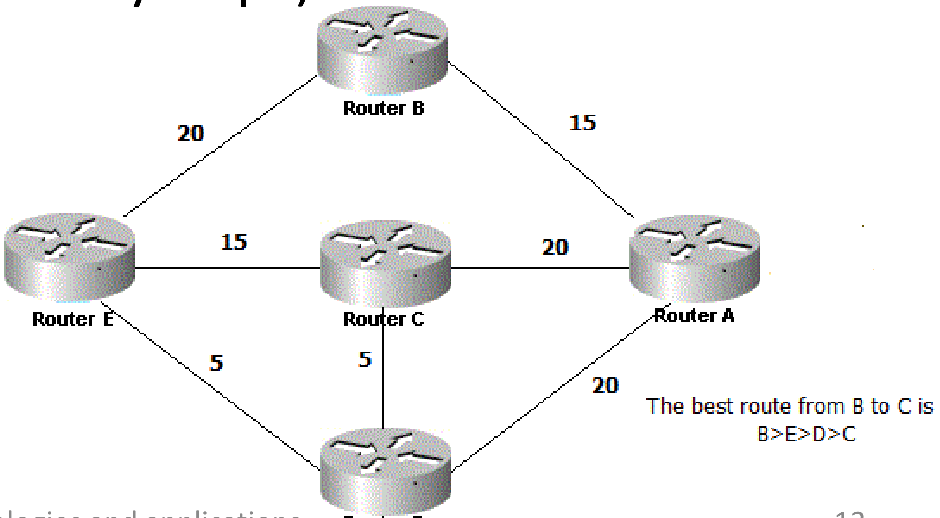
- Routing
  - Process through which the packets are directed to the destination node
  - Based on the routing tables and the used routing protocols, the internal routers determine the path
- Router
  - The node handling the routing process
  - Communicate with each other
  - Receive and store information 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)

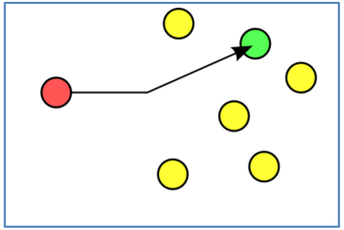
Tasks of a router

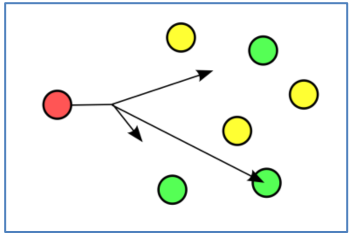
- Selecting the optimal path for a given packet
- Based on several aspects (metrics):
  - Length of the route (how many hops)
  - Cost
  - Bandwidth, speed
  - Reliability
  - Delay

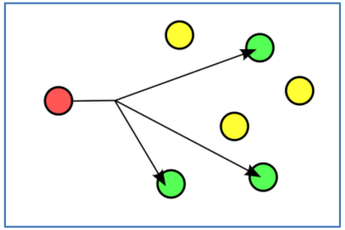


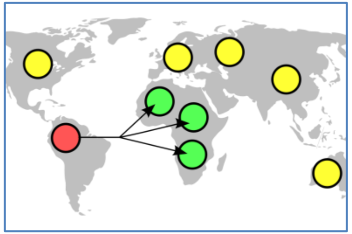
Routing semantics

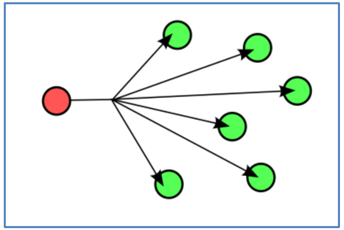
- **Unicast** – sending a packet to one specific destination
- **Anycast** – sending a packet to anyone (e.g., the closest one) from a group
- **Multicast** – sending a packet to a group
- **Geocast** – sending a packet to everyone in a given geographical area
- **Broadcast** – sending a packet to everyone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

- The 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

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

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

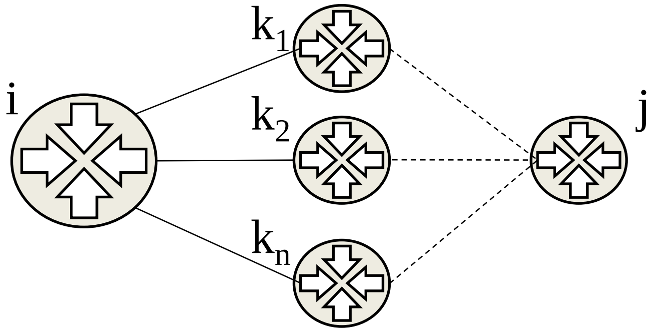
- **Additivity**
  - The cost of a route is the sum of the costs of the links composing that route

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

Bellman equation:

$\mathbf{D_{ii}} = 0$, for each *i*

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



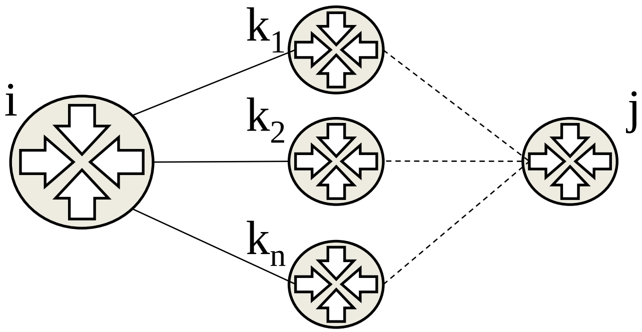
Distributed Bellman-Ford Algorithm

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

$\mathbf{D_{ii}} = 0$, for each *i*

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

- 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)
- RIPng (RFC 2080, ‘97)
  - IPv6 version
- EIGRP
  - Enhanced Interior Gateway Routing Protocol
  - Cisco proprietary standard

Distance Vector protocols

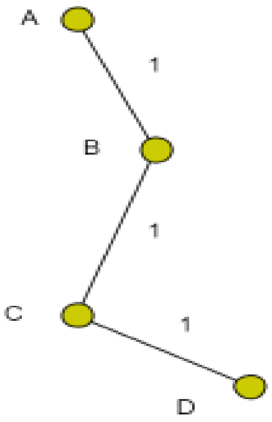
- Store distance vectors for each route
  - Data triples:
    - Destination
    - Cost
    - Next hop node (where to forward)
  - Periodically refreshed among neighbors
    - Update messages (2 parts):
      - Destination, cost
    - If a router learns about a better path, it updates its table
     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

Properties

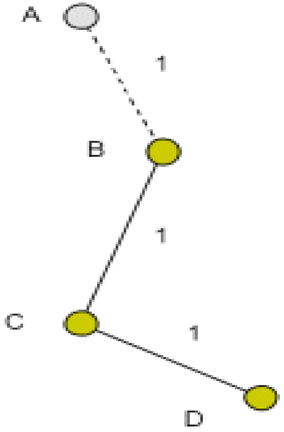
- Simple, but not perfect:
  - Link costs can change
  - Links can be broken
    - Cost of a broken link set to infinity
      - An integer value that is larger than any real possible value (by default, 16 for RIP)
  - In case of topology change, routing tables are refreshed gradually
    - Periodically (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