Chapitre 5
Network Layer and Routing

Laurent Schumacher (UNamur)
Dernière mise-à-jour : 09 novembre 2020

UNIVERSITÉ DE NAMUR

Outline

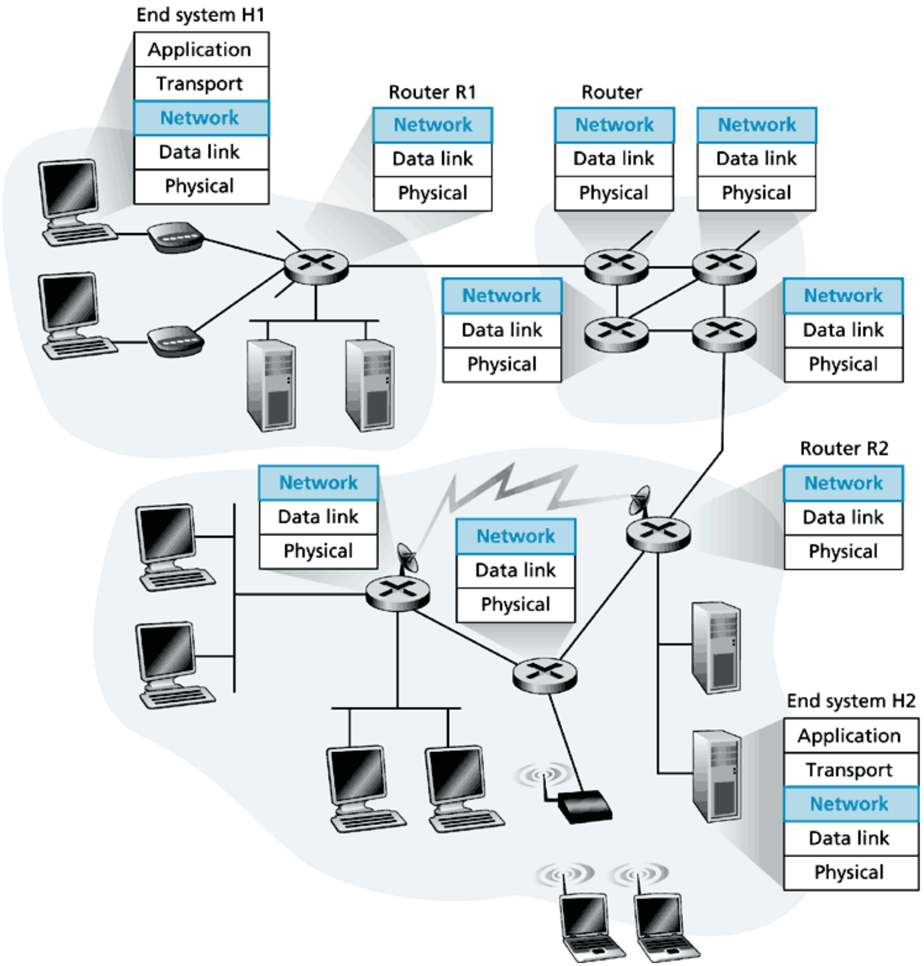
- Introduction
  - Forwarding and routing
  - Network-Layer services
  - Virtual circuit and datagram networks
- What's inside a router?
- The Internet Protocol (IP) – IPv6 and IPv4
- Routing principles
  - Link state vs. Distance Vector
  - Hierarchical routing
- Routing in the Internet
  - Intra-domain routing: RIP and OSPF
  - Inter-domain routing: BGP
- Broadcast and multicast routing

Forwarding

Routing

Network Layer Services

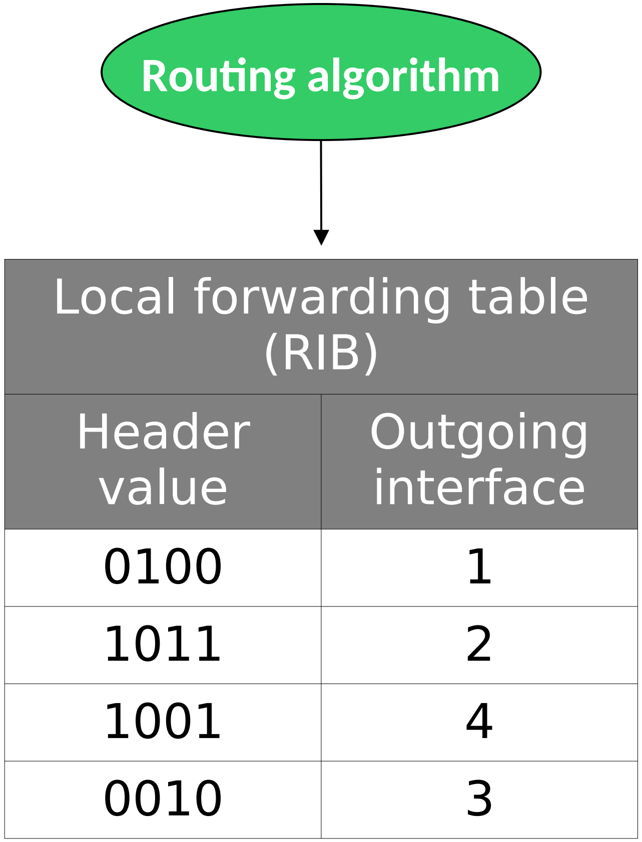
Functions



- Transport packets from sending to receiving hosts
- Network layer protocols in every host, router

Network Layer Services

Forwarding vs. Routing



Routing algorithm

Local forwarding table (RIB)	
Header value	Outgoing interface
0100	1
1011	2
1001	4
0010	3

Three important functions

1. Forwarding
   - Move packets from router's input to appropriate output
   - Local issue
2. Routing
   - Determine route from source to destination
   - Global issue
3. Call setup (Virtual Circuits, etc)

Network Layer Services
Service models

- Defines characteristics of E2E transport of data between two edges of the network
- Expectations
  - Loss-free delivery?
  - In-order delivery?
  - Guaranteed minimal bandwidth?
  - Preservation of inter-packet timing (no jitter)?
  - Congestion feedback to sender?
- Basic Internet offers a single service: best effort
- Other service models implemented in different architecture (ATM) or in Internet evolutions (IntServ, DiffServ)

Network Layer Services

Architecture and service models

Network architecture	Service model	Guarantees				Congestion feedback
		Bandwidth	No-loss	Ordering	Timing	
Internet	Best effort	No	No	No	No	No
	IntServ DiffServ	Yes	No	No	Yes	No
ATM	CBR	Guaranteed constant	Yes	Yes	Yes	No
	VBR	Guaranteed variable	Yes	Yes	Yes	No
	ABR	Guarantee minimum	No	Yes	No	Yes
	UBR	No	No	Yes	No	No

Network-Layer Services

Internet vs. ATM

Internet (Datagram)	ATM (Virtual Circuit)
Interconnected TCP/IP networks Data exchange between computers « Elastic » service, no strict timing requirements	Evolved from Public-Switched Telephone Networks (PSTN) Conversation between human beings Need for guaranteed service
Complexity at the edge : « smart end-systems (PC, handhelds, etc)	Complexity inside the network : dumb end-systems
Commoditization (bit pipe model) Services generate revenue	Great old days : operators moneytized connections

Network Layer Services

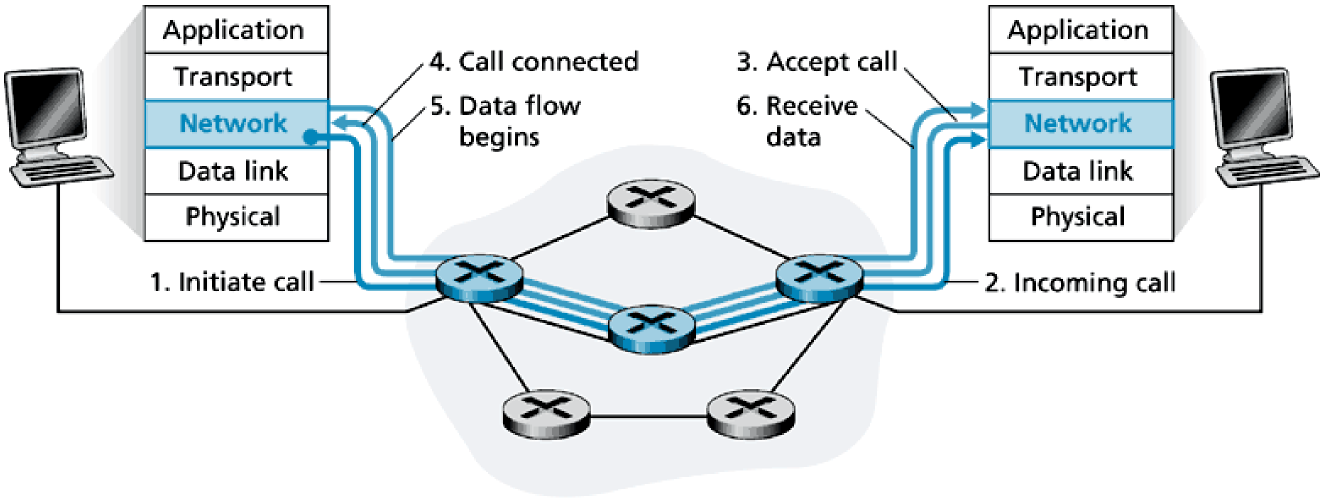
Virtual Circuit – Architecture

- Source-to-destination path behaves much like telephone circuit
- Three phases
  1. VC setup : sender indicates receiver to network layer and waits for a path to be traced. VC tables updated in switches.
  2. Data transfer : packets follow the VC path
  3. VC teardown : VC tables updated in switches
- Each packet carries VC identifier (not destination host ID)
- Every router on the path maintains “state” for each passing connection
- Link, router resources (bandwidth, buffers) may be reserved to VC setup

Network Layer Services

Virtual Circuit – Signalling

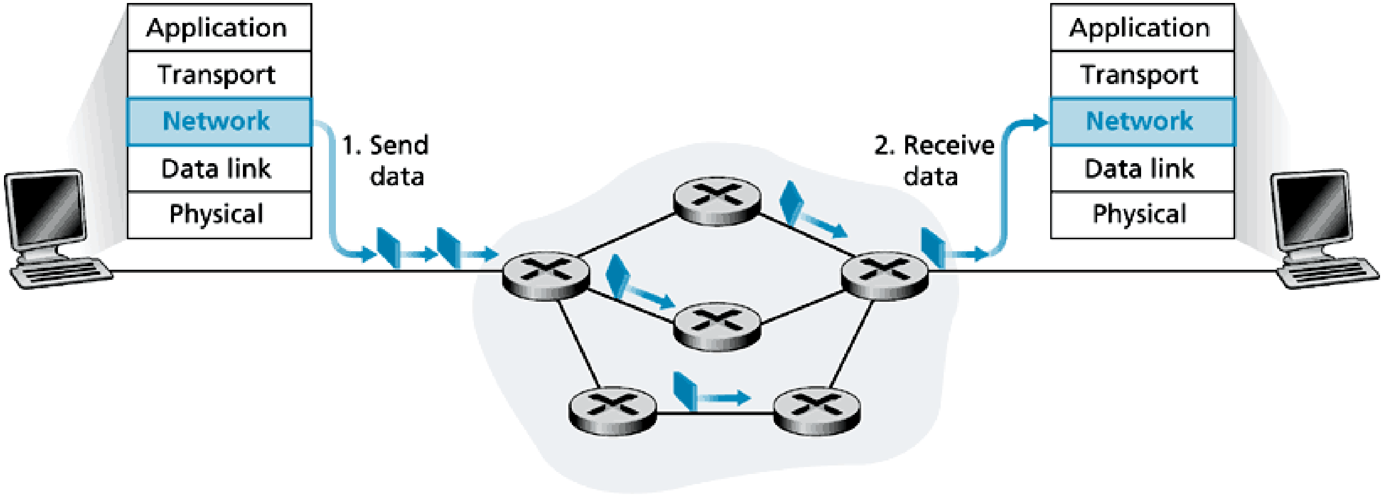
- All intermediates hosts involved in VC setup
  ≠ TCP handshake only involving end systems



- Signaling protocols
  - Used to setup, maintain and teardown VC
  - Used in legacy ATM, frame-relay, X.25
  - Not used in basic Internet, but in evolutions (MPLS)

Network Layer Services

Datagram – Internet model



- No call setup at network layer
- No state information about E2E connections in routers
  → No network-level concept of “connection”
- Packets forwarded using destination host address and forwarding tables in routers
- Packets may follow different paths
  → Unreliable, connectionless datagram delivery service (best-effort)

Network-Layer Services

Summary

Virtual Circuit	Datagram
Network-layer connection-oriented service	Network-layer connectionless service
MPLS Legacy: ATM, Frame Relay, X.25	Best-effort Internet

Outline

- Introduction
  - Forwarding and routing
  - Network-Layer services
  - Virtual circuit and datagram networks
- What’s inside a router?
- The Internet Protocol (IP) – IPv6 and IPv4

Forwarding

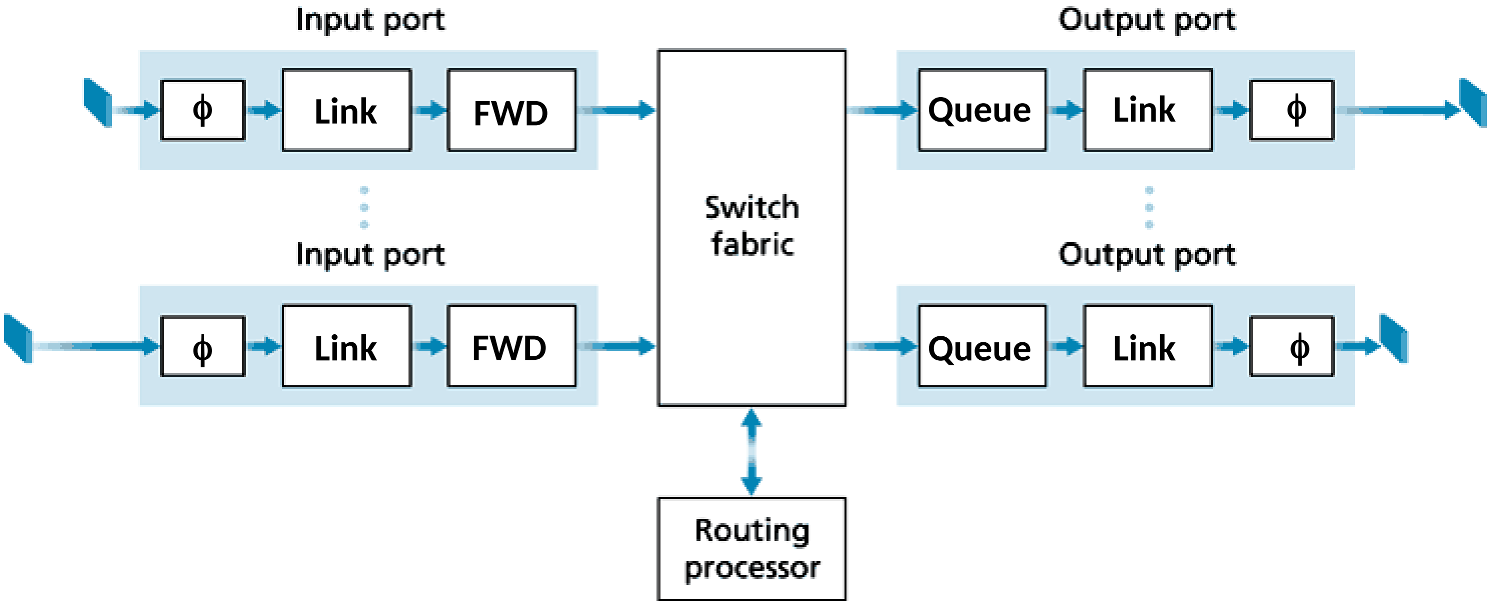
- Routing principles
  - Link state vs. Distance Vector
  - Hierarchical routing
- Routing in the Internet
  - Intra-domain routing: RIP and OSPF
  - Inter-domain routing: BGP
- Broadcast and multicast routing

Routing

Router architecture

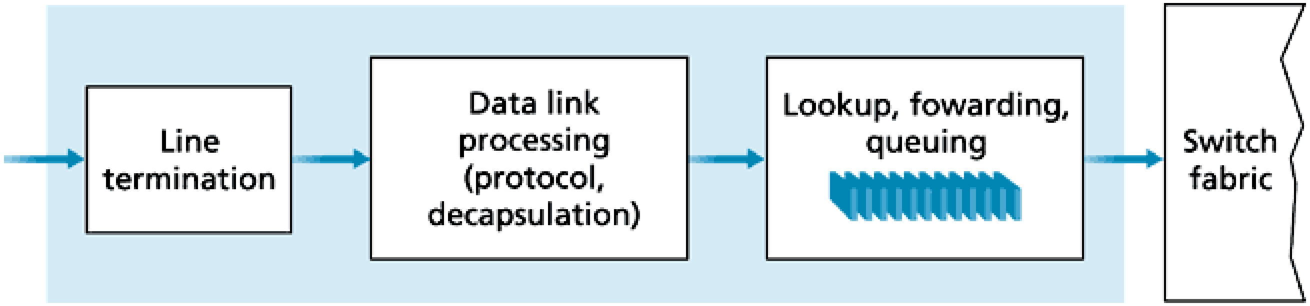
Overview

- Two key router functions
  1. Forwarding – Switch datagrams from incoming to outgoing link
  2. Routing – Run run routing algorithms like RIP, OSPF, BGP, etc



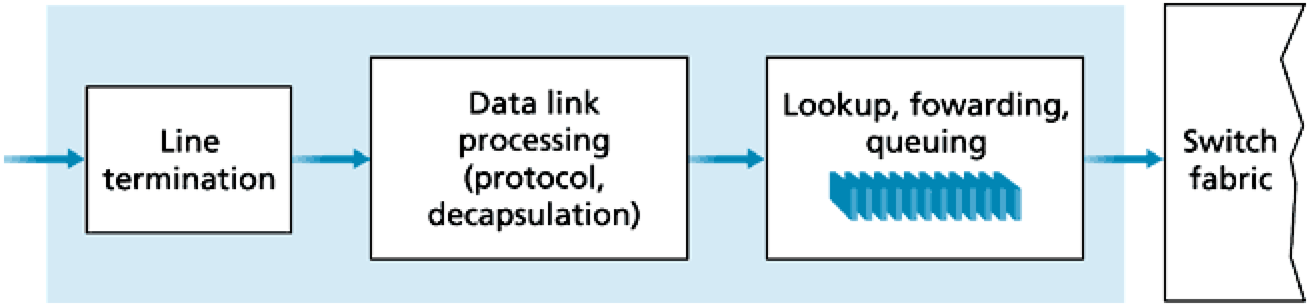
Router architecture

Input ports (1/2)



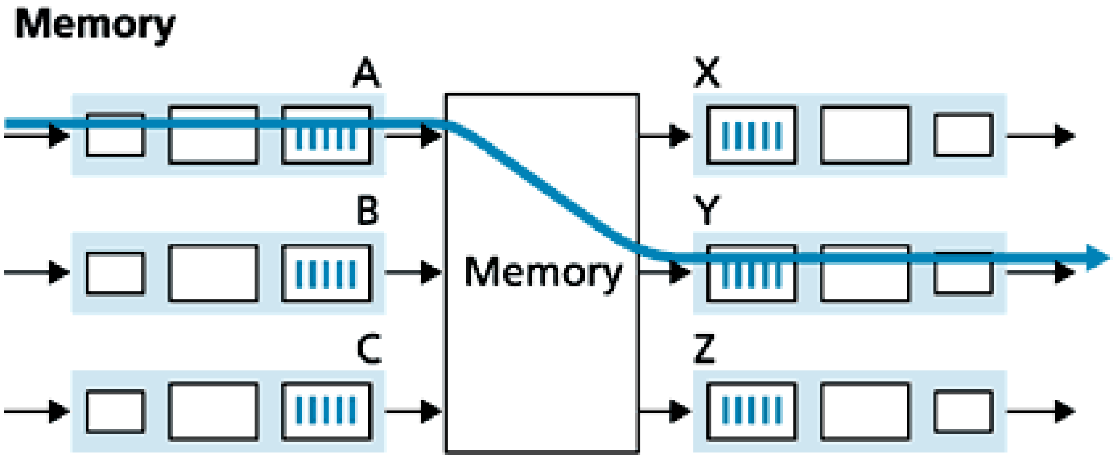
- Physical layer: bit-level reception
- Data link layer: e.g., Ethernet
- Network layer: decentralised switching
  - Given datagram destination, look-up output port using forwarding table in input port memory
  - Goal: complete input port processing at ‘line speed’
    - Lookup time < transmission time on input port
    - 256-Byte packet on 2.5 Gbps link = 0.819 µs
  - Queuing: if datagrams arrive faster than forwarding rate into switch fabric

Router architecture
Input ports (2/2)



- Network layer: decentralised switching (in dedicated routers)
  - Linear search through the forwarding table
  - Binary tree search in tree-structured forwarding table (128-level in IPv6, 32 in IPv4)
  - Content Adressable Memory (CAM): IP address used to index the forwarding table
  - Caching

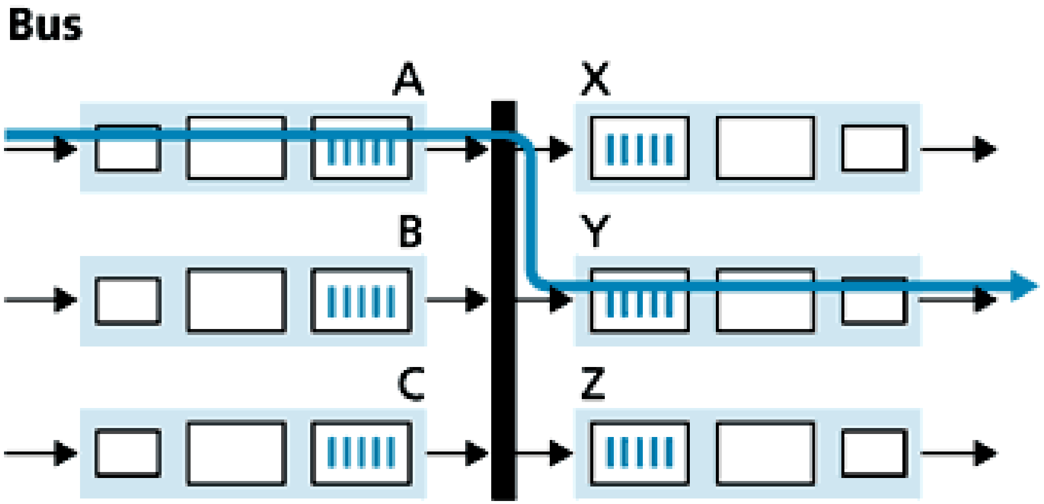
Router architecture
Switch fabric – In memory



- Packet copied in processor memory
  - Main processor analyses it and copies it to appropriate forward buffer → simplest approach, implementable on traditional PCs
  - Look-up already performed by input processor. Removal of a pointer from the receive queue, and copy of the value of the pointer to the appropriate forward queue → shared memory multiprocessor
- Switching bandwidth limited by memory performance

Router architecture

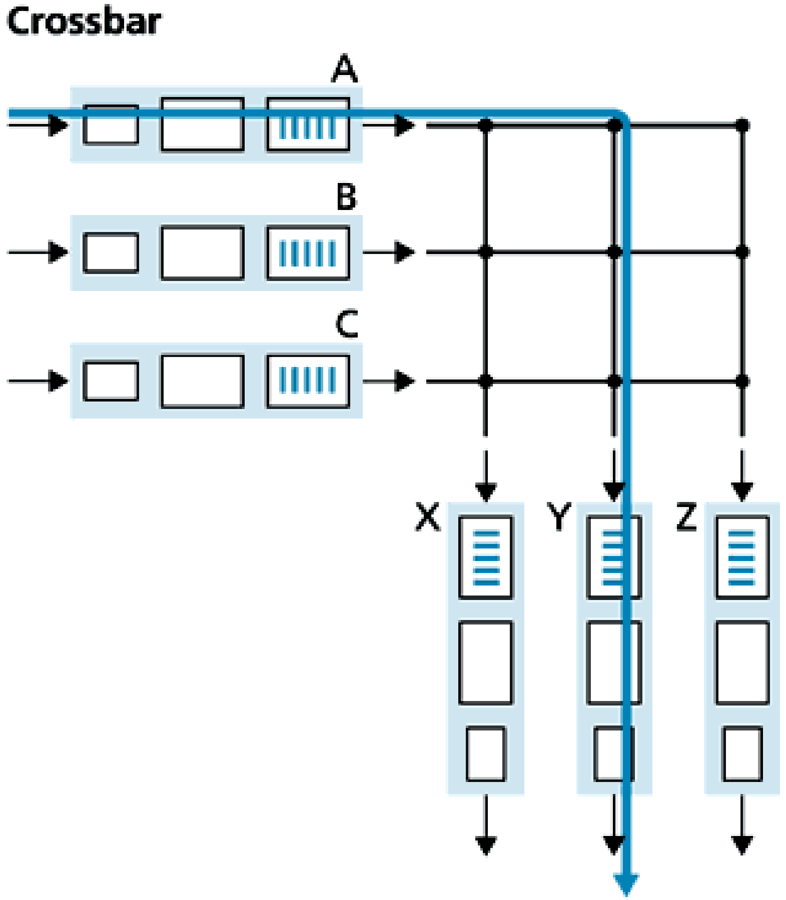
Switch fabric – Via a bus



- Direct transfer from input to output port via a shared bus
- Drawback: only one packet transferred at a time on the bus
- Switching bandwidth limited by bus speed

Router architecture

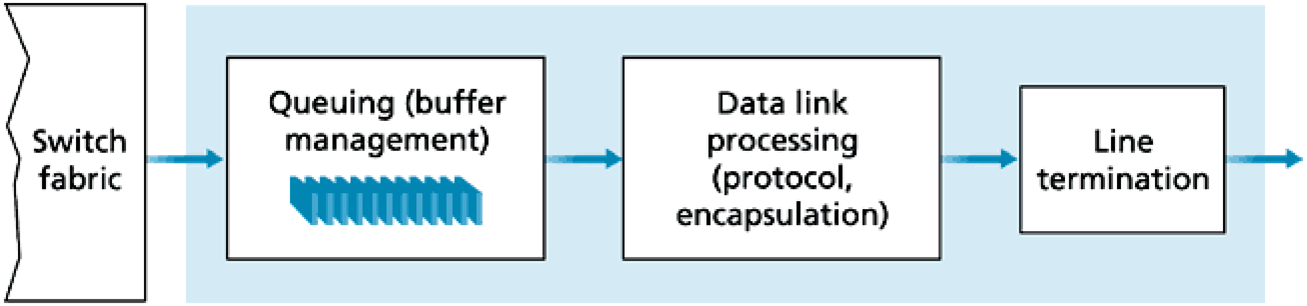
Switch fabric – Via interconnection network



- 2*N* buses connecting *N* input ports and *N* output ports
- Overcome bus bandwidth limitations
- Still blocking and queueing at input port if vertical bus busy

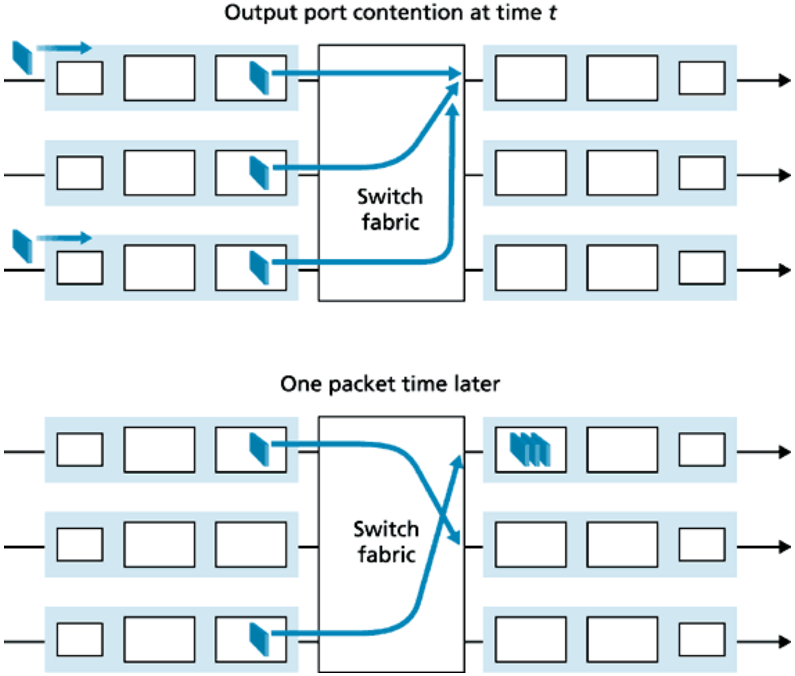
Router architecture

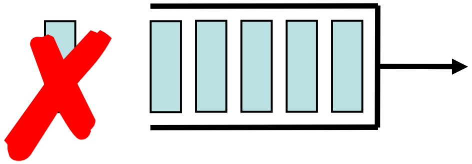
Output ports



- Network-layer
  - Queueing: required if switch fabric delivers packets at a rate greater than outgoing link rate
  - Scheduling: selection among queued datagrams for transmission
- Data link layer: e.g., Ethernet
- Physical layer: bit-level transmission

Router architecture
Output port queueing

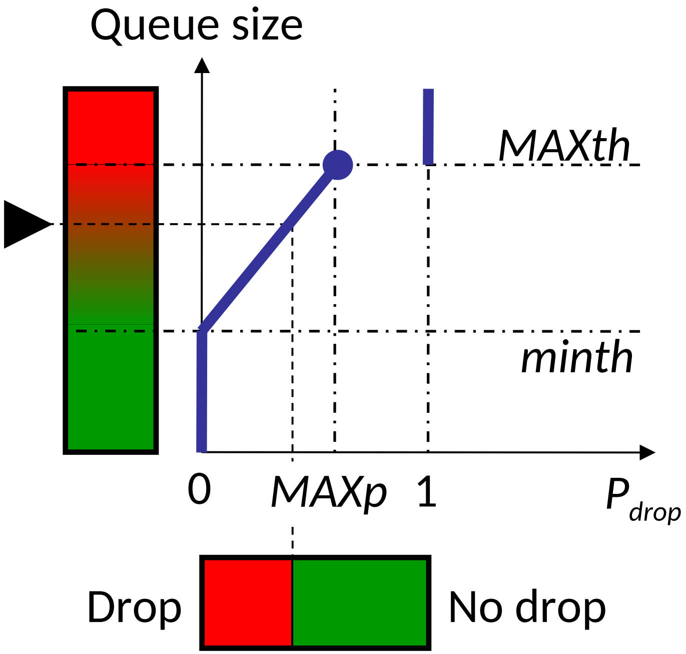




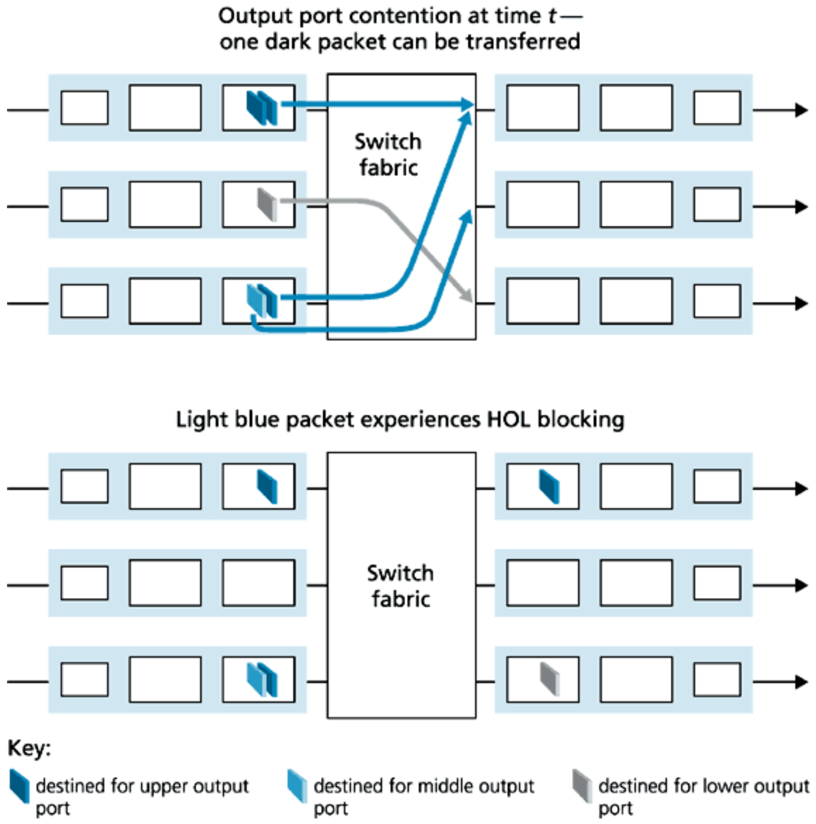
- Assume
  - *N* input ports
  - *N* output ports
  - A switch fabric *N* times as fast as line speed
- Switch fabric can forward packets incoming on *N* input ports to same output port
- But packets queue at output port
- If no memory left in buffer, loss (Drop Tail)

Router architecture
Random Early Detection (RED)

- Enable packet dropping before the buffer is full → signal congestion to source
- Probabilistic dropping of arriving packets.
- Drop probability P_{drop} increases as estimated average queue size grows.
  - Queue mostly empty recently → no drop
  - Queue mostly full recently → probabilistic dropping
- Two parts
  1. Estimate the average queue size
  2. Decide whether to drop a packet or not



Router architecture
Input port queueing



- Assume the switch fabric is not fast enough
- Packets queue at the input port
- Head-of-line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward

Outline

- Introduction
    - Forwarding and routing
    - Network-Layer services
    - Virtual circuit and datagram networks
- What’s inside a router?
- The Internet Protocol (IP) – IPv6 and IPv4
- Routing principles
    - Link state vs. Distance Vector
    - Hierarchical routing
- Routing in the Internet
    - Intra-domain routing: RIP and OSPF
    - Inter-domain routing: BGP
- Broadcast and multicast routing

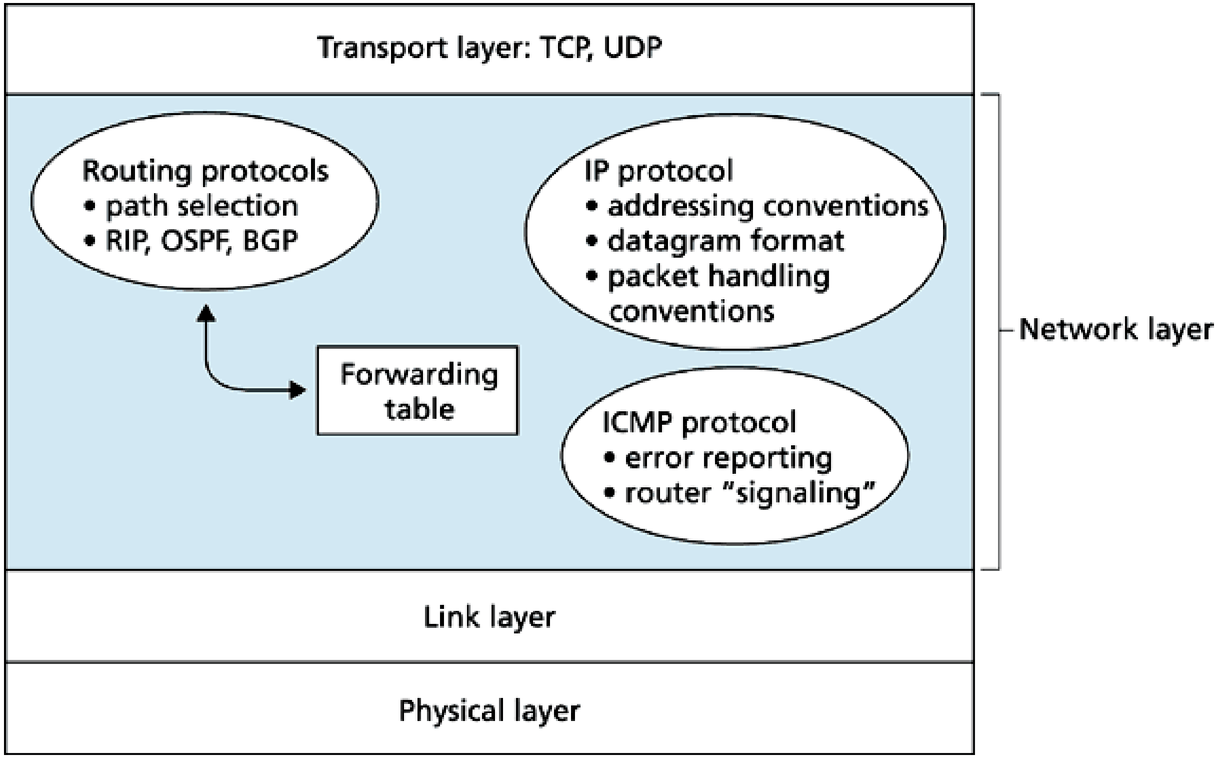
Forwarding

Routing

IP

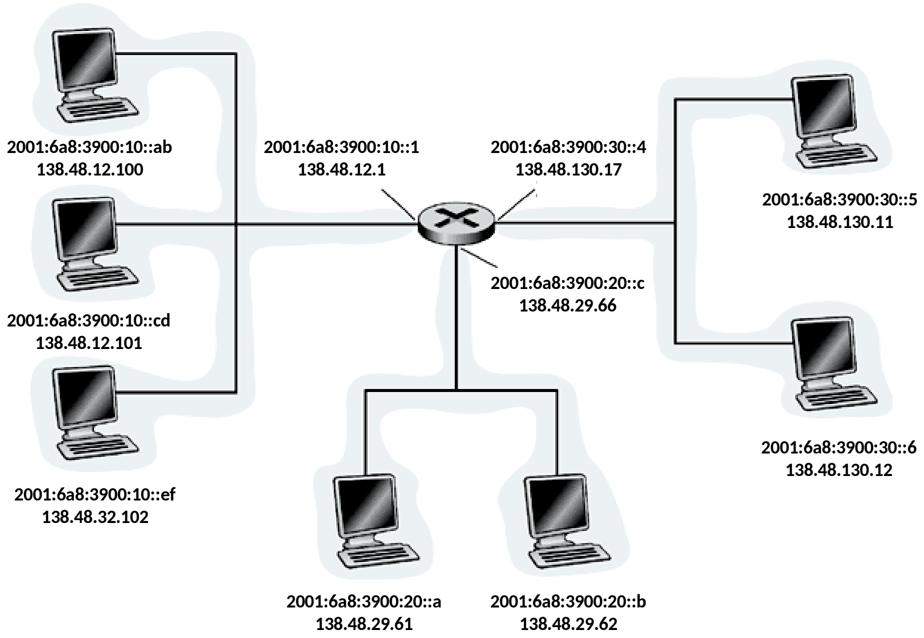
Network-Layer Functions

- Three main components: protocol, routing and signalling



IP

Introduction to addressing schemes

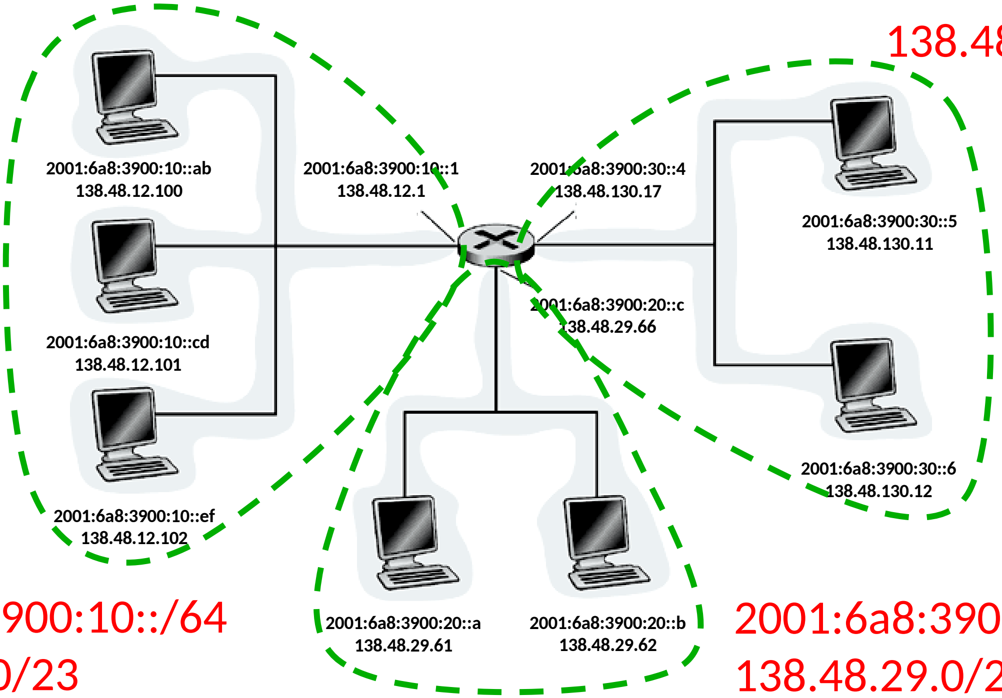


- Address ≠ Host
- Address Interface
- Routers and hosts typically have multiple interfaces
- An IP address associated with each interface
  → Multi-homing

- IPv6: 128-bit address, hexadecimal notation
  Example: 2001:6a8:3900:30::ef
- IPv4: 32-bit address, decimal notation
  Example: 138.48.32.100

IP

What's a « IP network »?



2001:6a8:3900:30::/64
138.48.130.0/23

2001:6a8:3900:10::/64
138.48.12.0/23

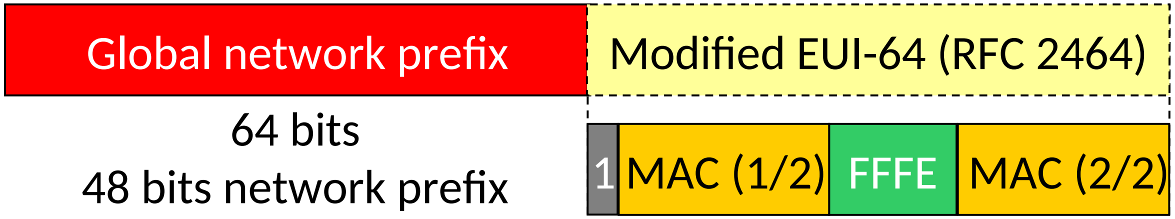
2001:6a8:3900:20::/64
138.48.29.0/23

- From IP address perspective, device interfaces with same network part of IP address can physically reach each other without router

IP

Network prefix – Network mask

IPv6 Global Unicast



Global network prefix

Modified EUI-64 (RFC 2464)

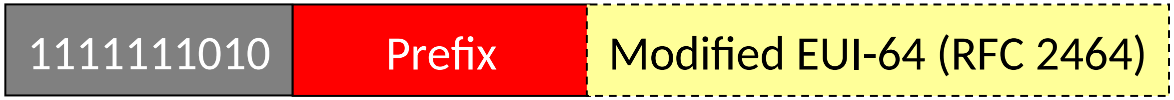
64 bits

48 bits network prefix
+ 16 bits sub-network prefix

1 | MAC (1/2) | FFFE | MAC (2/2)

23 bits 16 bits 24 bits

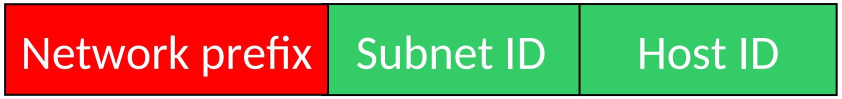
IPv6 Local Unicast (FE80::/10)



1111111010 | Prefix | Modified EUI-64 (RFC 2464)

54 bits

IPv4 Public Address (CIDR, /*m*)



Network prefix | Subnet ID | Host ID

m bits *n* bits (32-*m*-*n*) bits

IPv4 Private Address (RFC 6761)

10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16

IP

IPv4 Original Classful Address

- Inefficient use of address space
- e.g., class B large enough for 65,000 hosts, even if only 2,000 hosts in that network

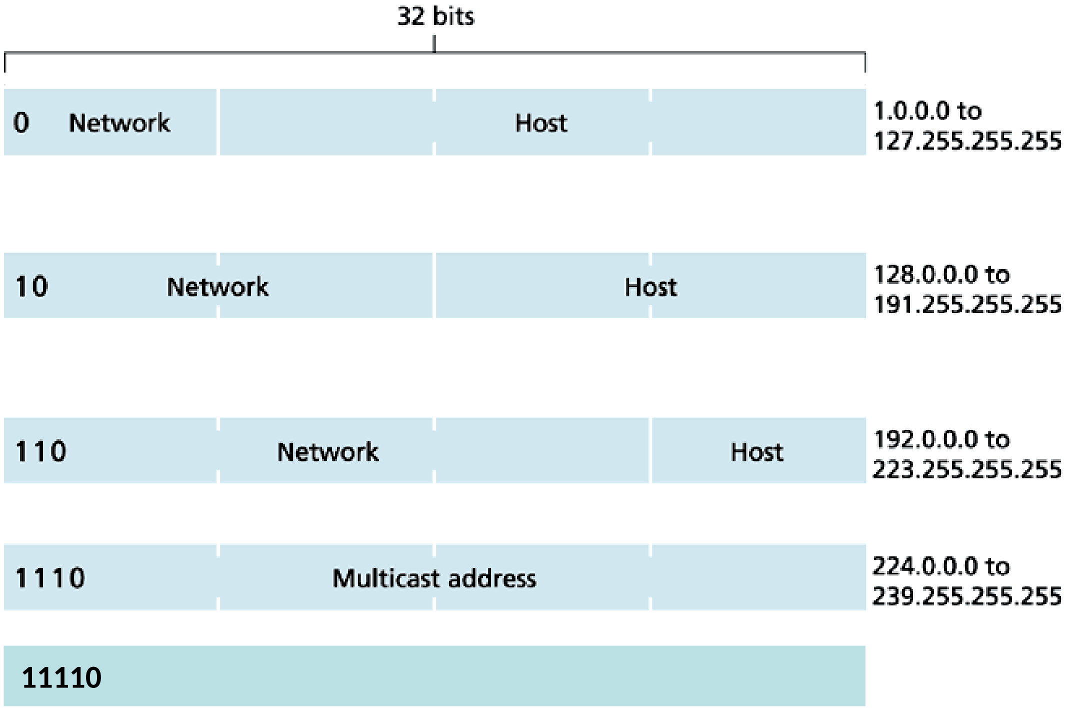
Class A
2^7 networks with 2^{24} interfaces

Class B
2^{14} networks with 2^{16} interfaces

Class C
2^{21} networks with 2^8 interfaces

Class D

Class E
(future use)



IP

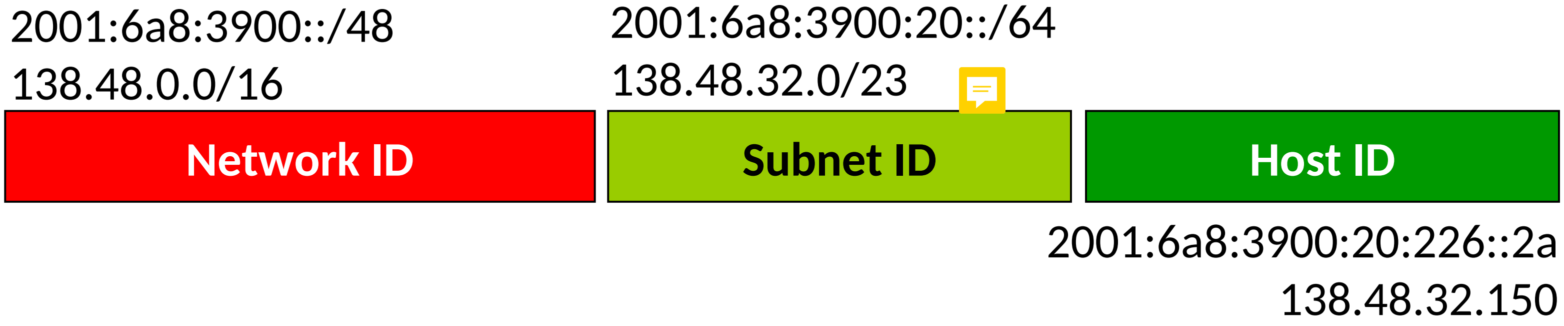
IPv4 Classless InterDomain Routing (CIDR)

- Network portion of address of arbitrary length
- Address format: a.b.c.d/x, where x is number of bits in network portion of address
- Routing decisions based on masking operations of the entire IP address (LMP – Longest Match Prefix)

IP

Subnetting

- Prefix can be further subdivided into subnetworks, for internal purposes
- Subnet hierarchy invisible from outside
- Routing tables of reduced size
  - Outside: a single address with *N* subnets better than *N* subnets
  - Inside: route to subnetworks instead of to machines



2001:6a8:3900::/48
138.48.0.0/16

2001:6a8:3900:20::/64
138.48.32.0/23

Network ID | **Subnet ID** | **Host ID**

2001:6a8:3900:20:226::2a
138.48.32.150

IP

Address types (RFC 4291, Feb. 2006)

Three main types of addresses

1. Unicast
    - An identifier for a single interface
    - A packet sent to a unicast address is delivered to the interface identified by that address.
2. Anycast
    - An identifier for a set of interfaces
    - A packet sent to an anycast address is delivered to the "nearest" interface identified by that address
3. Multicast
    - An identifier for a set of interfaces
    - A packet sent to a multicast address is delivered to all interfaces identified by that address

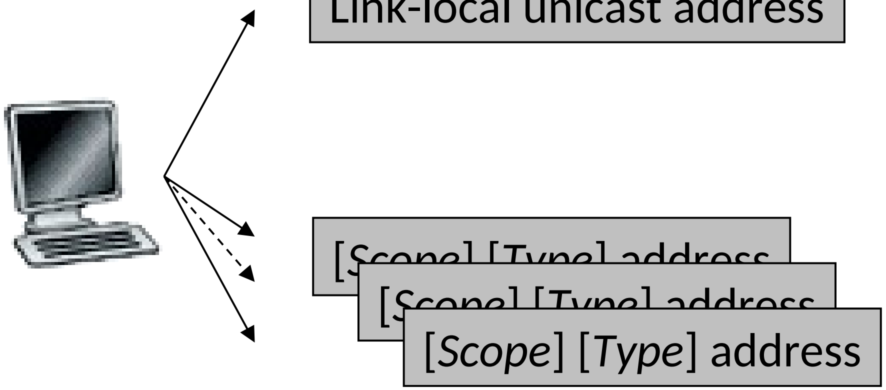
Broadcast (255.255.255.255) superseded by multicast

IP

Address types (RFC 4291, Feb. 2006)

Type	**Scope**
Unicast	Link-local
Anycast	Global
Multicast	
~~Broadcast~~	
Unspecified	
Loopback	

Not to be forwarded by router



Link-local unicast address

[Scope] [Type] address

[Scope] [Type] address

[*Scope*] [*Type*] address

RFC under revision: draft-ietf-6man-rfc4291bis

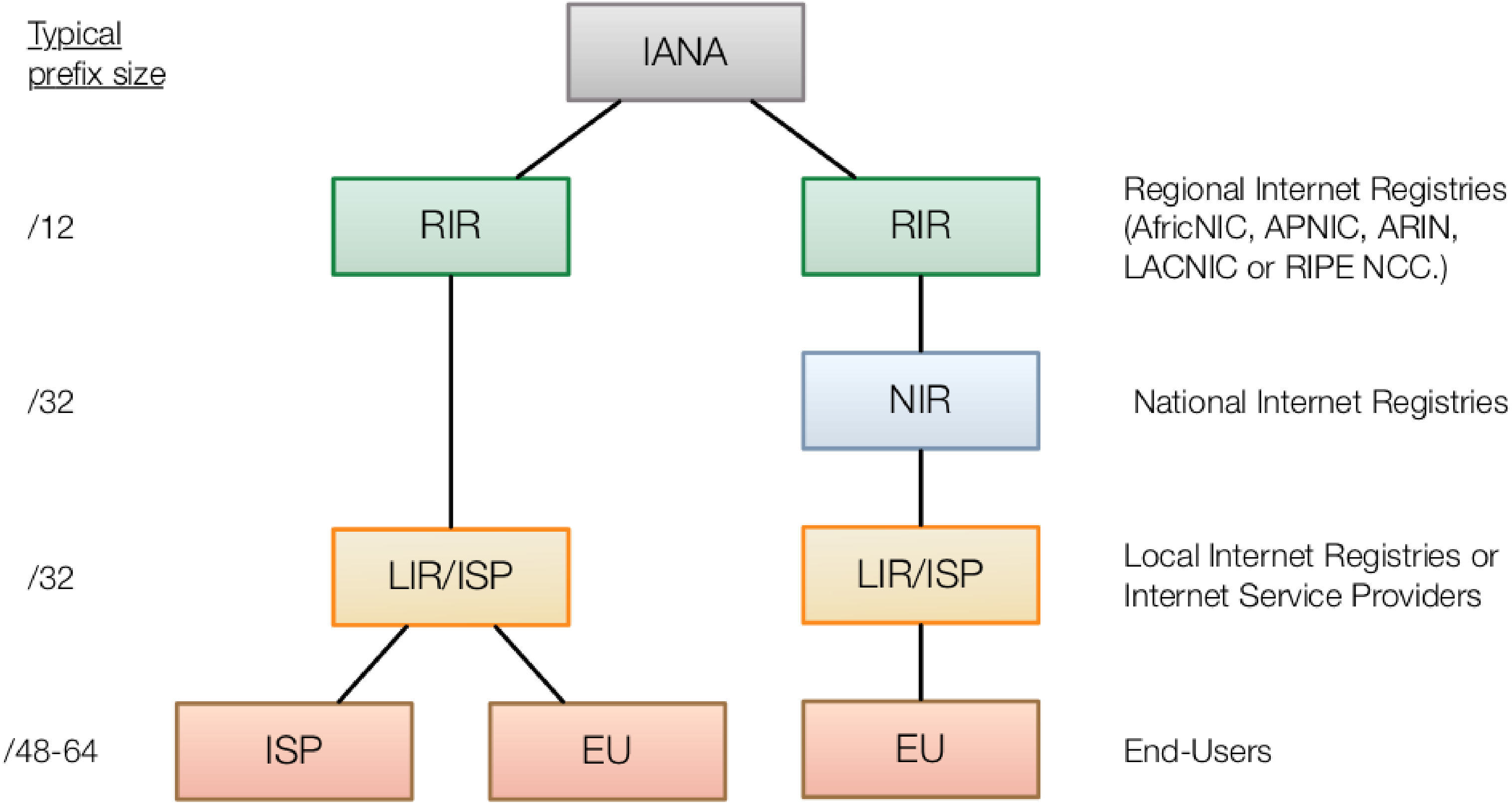
IP

Special addresses

Type	IPv6 (RFC 4291)	IPv4 (RFC 6761)
Unspecified	::/128	0.0.0.0
Loopback	::1/128	127.0.0.1
Multicast	FF00::/8	224.0.0.0 to 239.255.255.255
Broadcast	FF02::1	255.255.255.255
All Routers Multicast	FF02::2	
Private	a.k.a. Link-local FE80::/10	10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16

IP

IPv6 Address Allocation Hierarchy and Policies on Sizes



Allowing IP Networks to be Securely Renumbered and Shared,
Damien Leroy, PhD thesis, 2011

IP

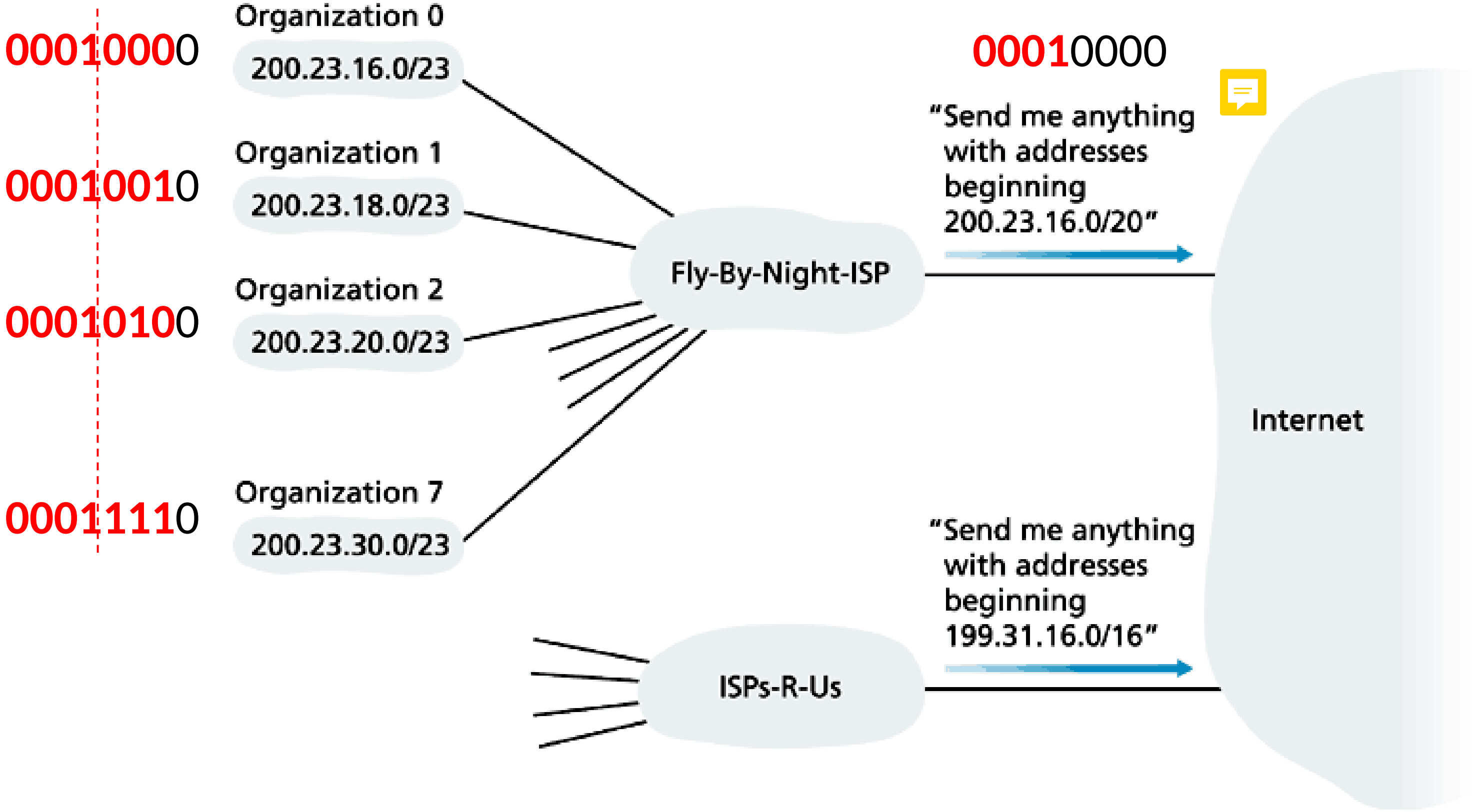
Prefix allocation

- ISPs get prefix from IANA through registries
- Two commercial strategies
  1. Provider provisioning (PP): ISP owns prefix. Renumbering required when changing ISP.
  2. Provider Independent (PI): customer owns prefix. Routing update required when changing ISP.
- Prefix allocated to maximise Route Aggregation

IP

Route Aggregation

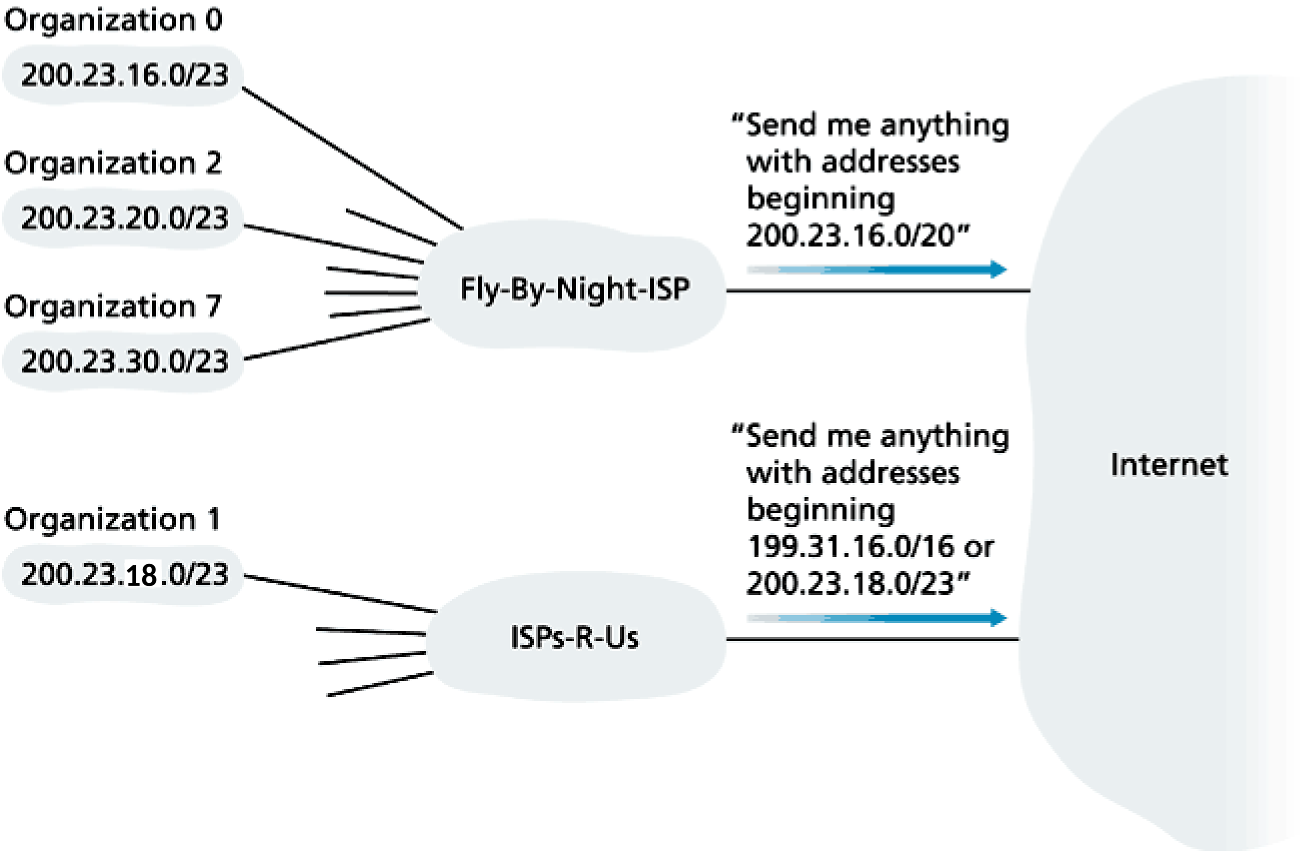
- Hierarchical addressing allows efficient advertisement of routing information



IP

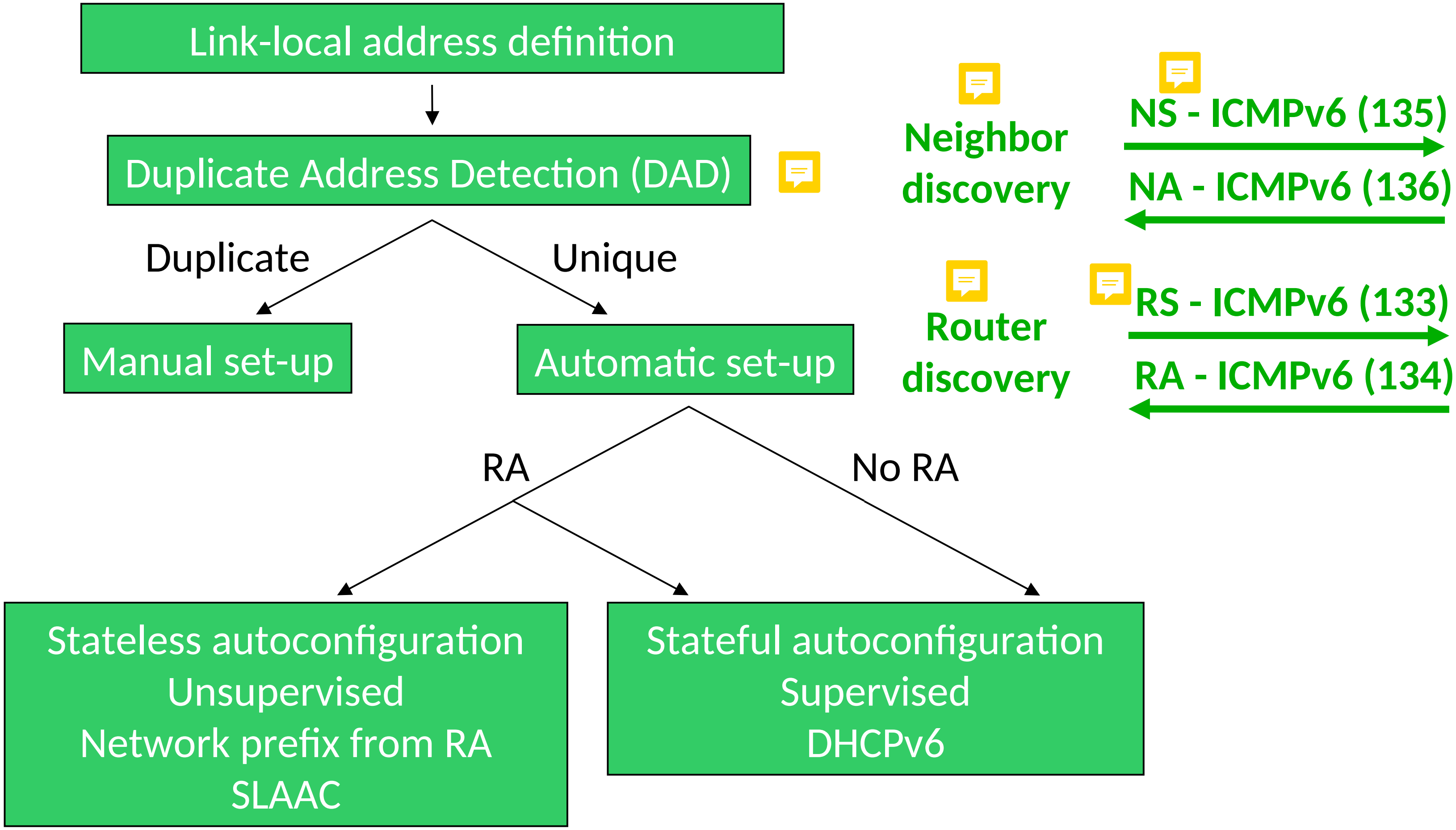
Route Aggregation with PI prefix

- Organisation 1 uses another ISP but keeps prefix
- New ISP has to advertise a specific route



IP

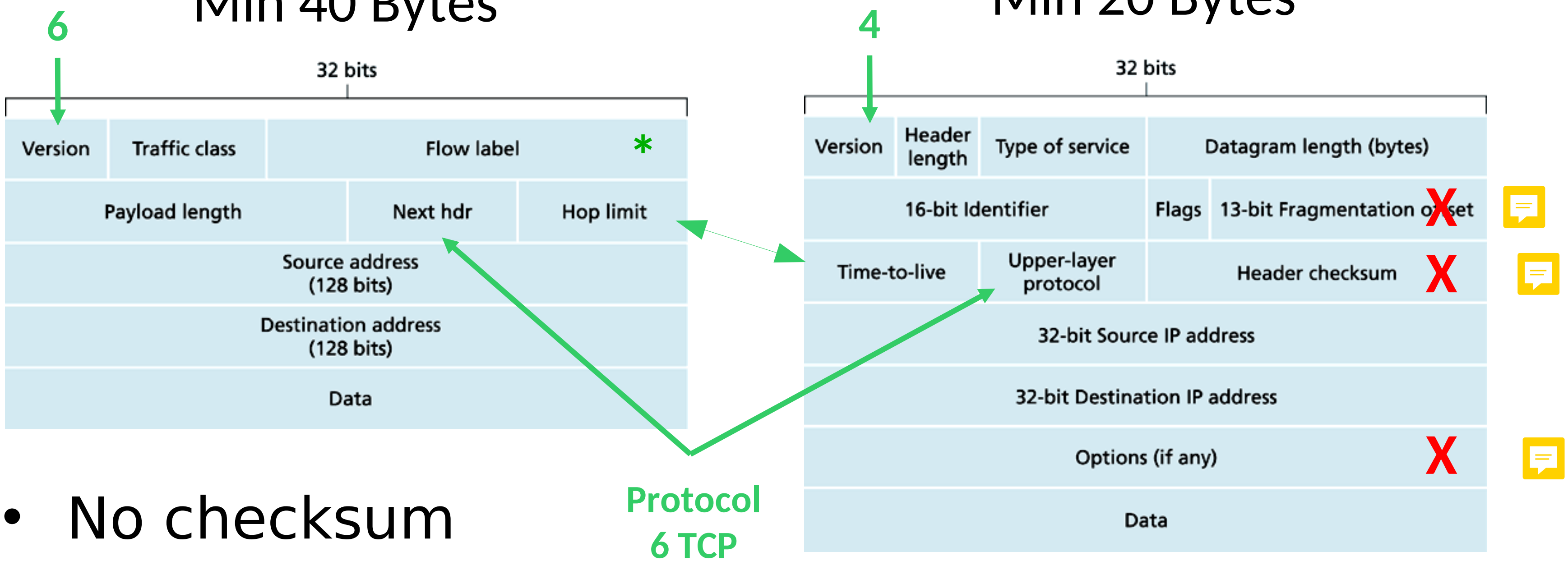
IPv6 Address Autoconfiguration (SLAAC)



IP
Comparison of datagram headers

IPv6
Min 40 Bytes

IPv4
Min 20 Bytes



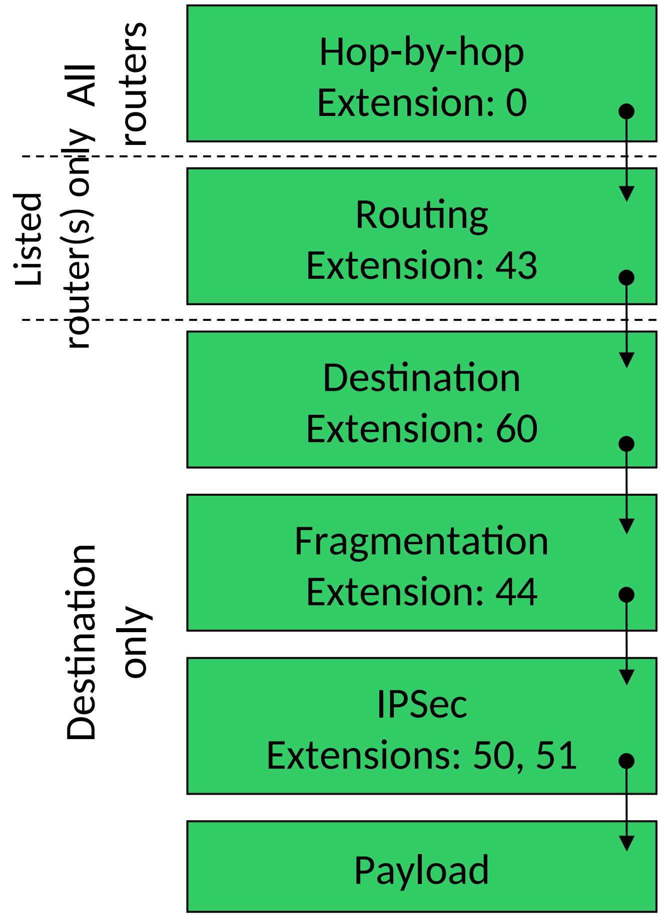
- No checksum
- No fragmentation
- No option
- Flow support (*)

Protocol
6 TCP
17 UDP
50, 51 IPSec
58 ICMPv6
89 OSPF
+ Extensions

Effective exhaustion of unallocated IPV4 pool announced on February 3rd, 2011

IP

IPv6 Extensions



All routers

Listed router(s) only

Destination only

Hop-by-hop
Extension: 0

- Padding – Packet size multiple of 64 bits
- Router alert – Payload check request
- Jumbogram – Packet length exceeds length field

Routing
Extension: 43

Enables loose source routing

Destination
Extension: 60

- Padding
- Mobility
- Tunnelling

Fragmentation
Extension: 44

Enables handling source fragmentation

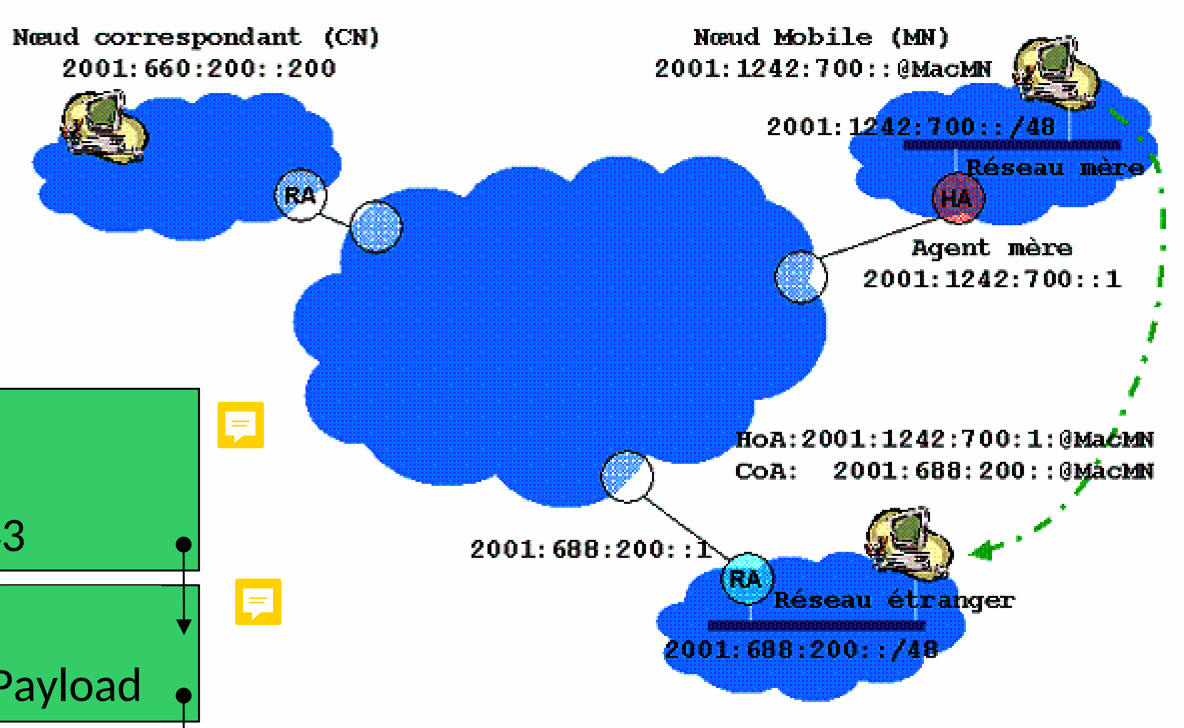
IPSec
Extensions: 50, 51

Confidentiality

Payload

IP

Mobility in IPv6



From: CoA
To: CN
Extension: 43

From: HoA
Extension: Payload

Payload

IPSec
Introduction

- Two different goals
  1. Confidentiality at network layer – Sending host encrypts the data in IP datagram
  2. Authentication at network layer – Destination host can authenticate source IP address
- Two security protocols
  1. Authentication Header (AH) protocol
  2. Encapsulation Security Payload (ESP) protocol
- Each protocol supports two modes of use
  1. Transport mode: the protocols provide protection primarily for upper layer protocols (E2E)
  2. Tunnel mode: the protocols are applied to tunneled IP packets (router to router)

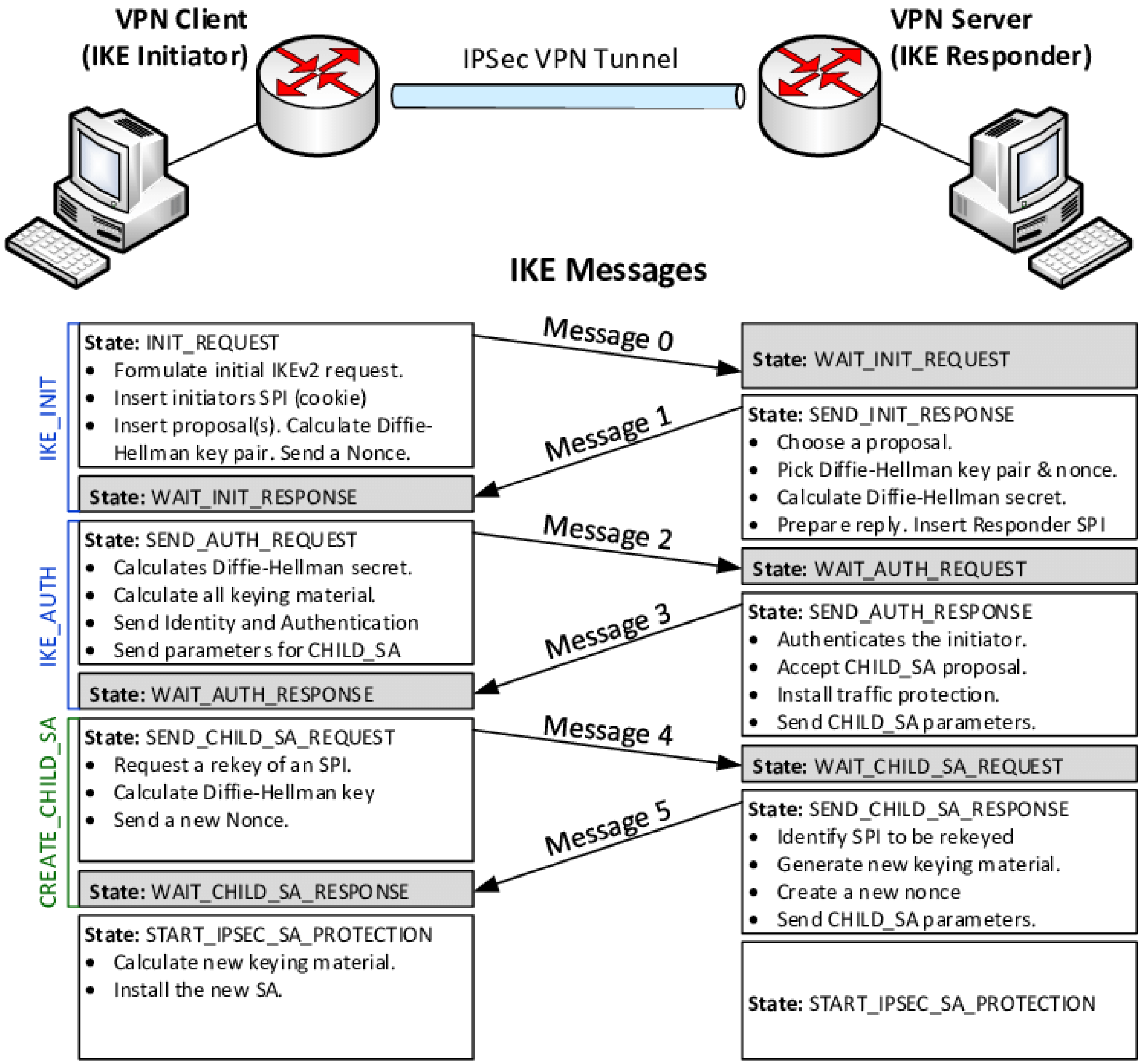
IPSec

Handshake and Security Association (SA)

- For both AH and ESP and before sending secure datagrams, source and network hosts handshake (IKE, ISAKMP/Oakley)
  - Authenticate endpoints
  - Create a network-layer connection (Security Association)
    - Choose cryptographic algorithms and keys
    - Reset sequence numbers to zero
- Security Association (SA)
  - Simplex, e.g. unidirectional, and logical connection
  - Uniquely defined by a 3-uple
  - Security protocol identifier (AH/ESP)
  - Source IP address of simplex connection
  - 32-bit Security Parameter Index (SPI), pseudo-random or manually set
- Contradicts connectionless essence of net layer

IPSec

Handshake and Security Association (SA)



State: INIT_REQUEST
- Formulate initial IKEv2 request.
- Insert initiators SPI (cookie)
- Insert proposal(s). Calculate Diffie-Hellman key pair. Send a Nonce.

State: WAIT_INIT_REQUEST

State: SEND_INIT_RESPONSE
- Choose a proposal.
- Pick Diffie-Hellman key pair & nonce.
- Calculate Diffie-Hellman secret.
- Prepare reply. Insert Responder SPI

State: WAIT_INIT_RESPONSE

State: SEND_AUTH_REQUEST
- Calculates Diffie-Hellman secret.
- Calculate all keying material.
- Send Identity and Authentication
- Send parameters for CHILD_SA

State: WAIT_AUTH_REQUEST

State: SEND_AUTH_RESPONSE
- Authenticates the initiator.
- Accept CHILD_SA proposal.
- Install traffic protection.
- Send CHILD_SA parameters.

State: WAIT_AUTH_RESPONSE

State: SEND_CHILD_SA_REQUEST
- Request a rekey of an SPI.
- Calculate Diffie-Hellman key
- Send a new Nonce.

State: WAIT_CHILD_SA_REQUEST

State: SEND_CHILD_SA_RESPONSE
- Identify SPI to be rekeyed
- Generate new keying material.
- Create a new nonce
- Send CHILD_SA parameters.

State: WAIT_CHILD_SA_RESPONSE

State: START_IPSEC_SA_PROTECTION
- Calculate new keying material.
- Install the new SA.

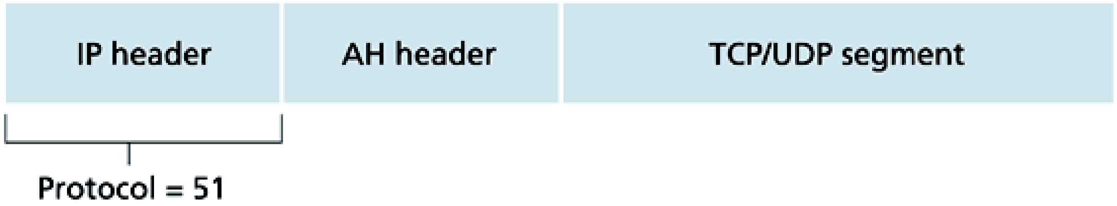
State: START_IPSEC_SA_PROTECTION

Source : https://www.researchgate.net/profile/Shaimaa_Abdel_Hakeem

IPSec

Authentication Header (AH, RFC 4302)

- Provides source host authentication and data integrity, but not confidentiality
- Having established SA, source can send secure datagrams
- Secure datagrams include the AH header

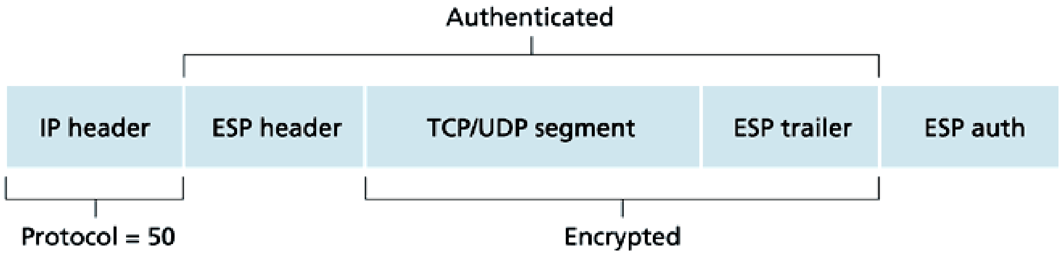


- AH header include
  - Next header (substitute for IP protocol header set at 51)
  - SPI
  - 32-bit sequence number, to prevent playback and person-in-the-middle attacks
  - Authentication data, digital signature of the IP datagram

IPSec

Encapsulation Security Payload (ESP, RFC 4303)

- Provides source host authentication, data integrity, and confidentiality

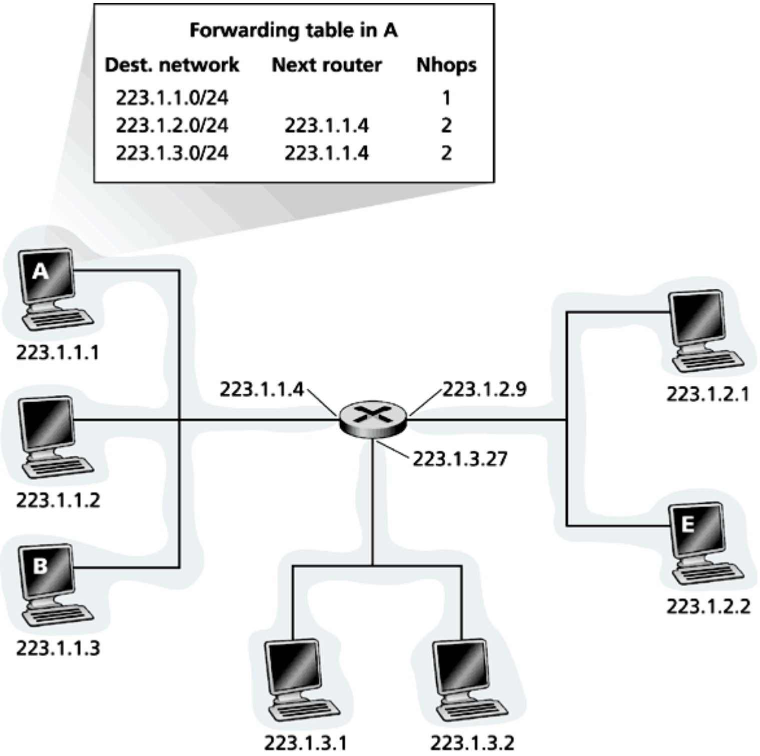


- ESP header
  - SPI
  - 32-bit sequence number, to prevent playback and person-in-the-middle attacks
- ESP trailer – Next header (substitute for IP protocol header set at 50)
- ESP authentication field – Authentication data, digital signature of the IP datagram

IP

Moving datagram to destination (1/4)

Misc fields	**223.1.1.1**	**223.1.1.3**	**Data**



Forwarding table in A

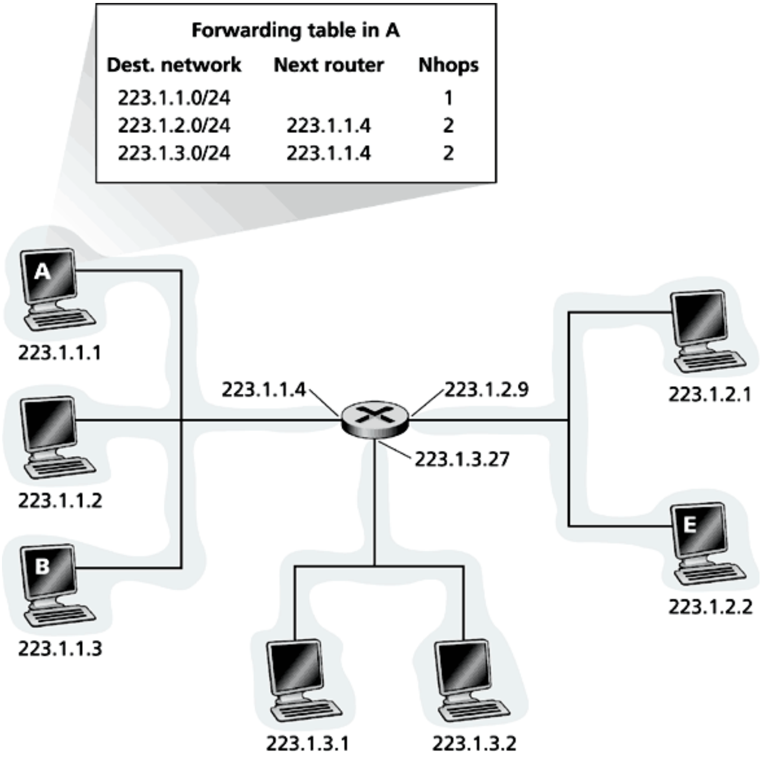
Dest. network	Next router	Nhops
223.1.1.0/24		1
223.1.2.0/24	223.1.1.4	2
223.1.3.0/24	223.1.1.4	2

- Starting at A, send IP datagram addressed to B
- Look up network address of B in A’s RIB (LMP)
- Find B is on same network as A
- Link layer will send datagram directly to B inside link-layer frame
- A and B are directly connected

IP

Moving datagram to destination (2/4)

Misc fields	**223.1.1.1**	**223.1.2.2**	**Data**



Forwarding table in A

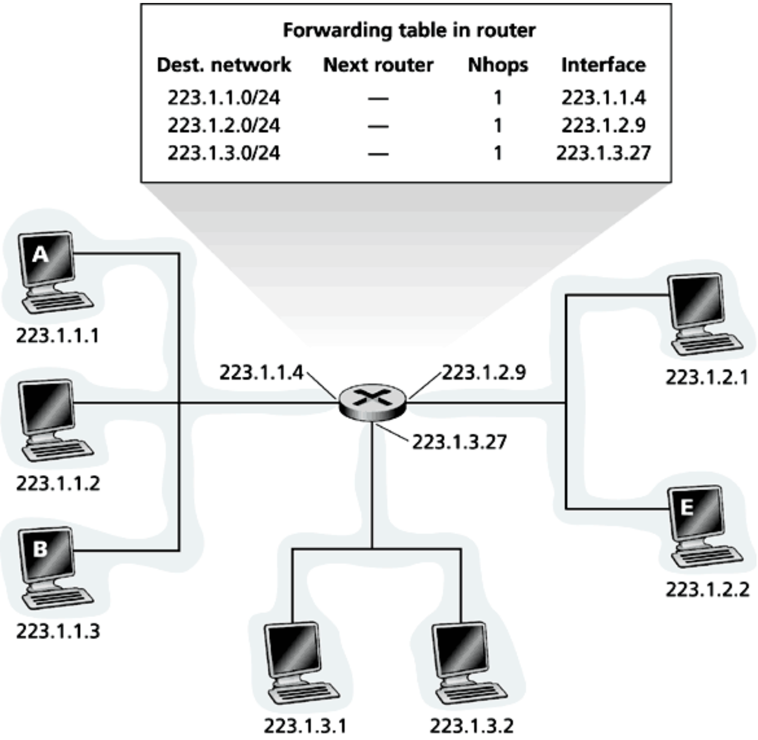
Dest. network	Next router	Nhops
223.1.1.0/24		1
223.1.2.0/24	223.1.1.4	2
223.1.3.0/24	223.1.1.4	2

- Starting at A, send IP datagram addressed to E
- Look up network address of E in A’s RIB
- A and E are not directly connected
- Next hop router to E is 223.1.1.4
- Link layer sends datagram to router 223.1.1.4 inside link-layer frame

IP

Moving datagram to destination (3/4)

Misc fields	**223.1.1.1**	**223.1.2.2**	**Data**

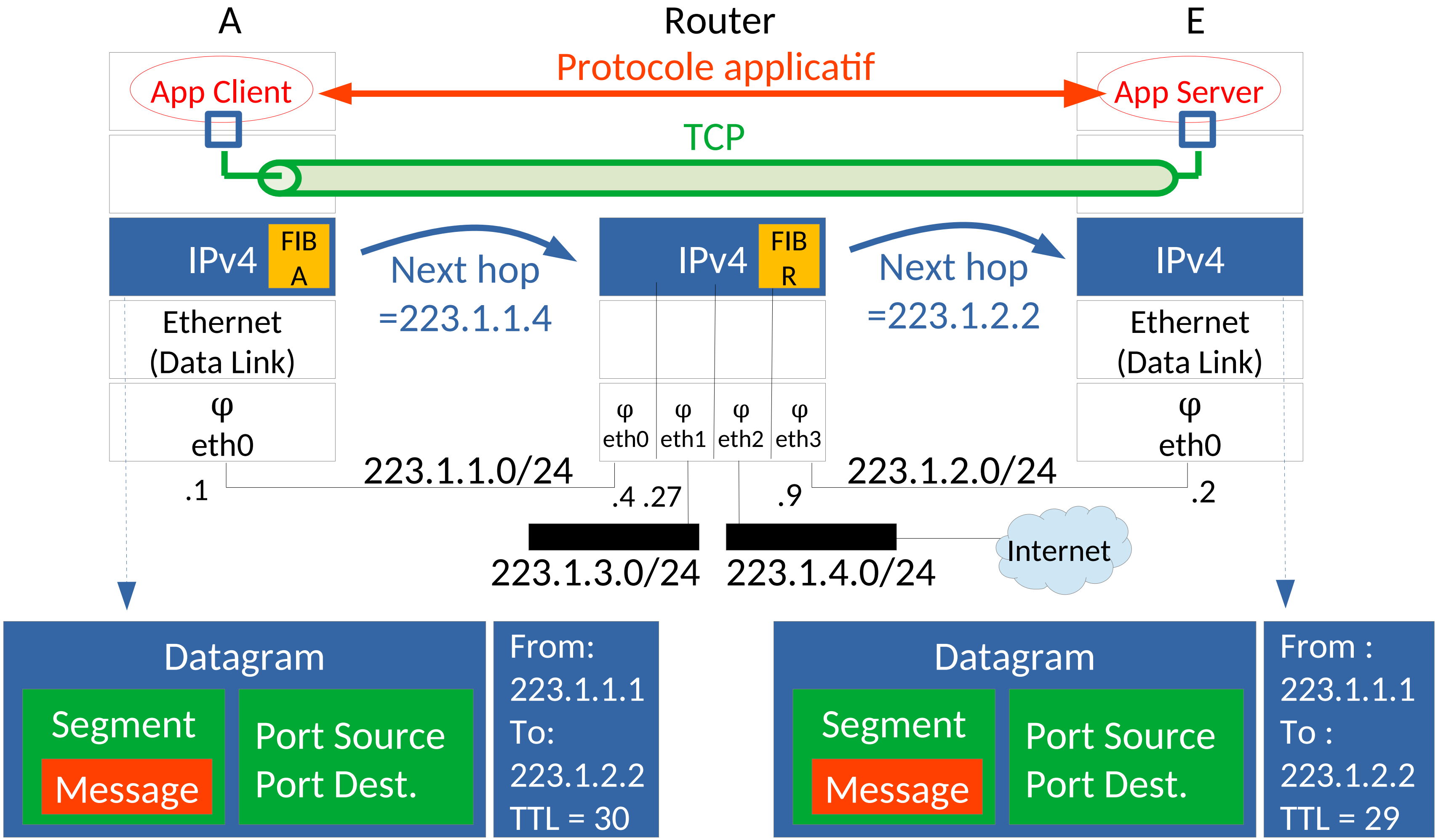


Forwarding table in router

Dest. network	Next router	Nhops	Interface
223.1.1.0/24	—	1	223.1.1.4
223.1.2.0/24	—	1	223.1.2.9
223.1.3.0/24	—	1	223.1.3.27

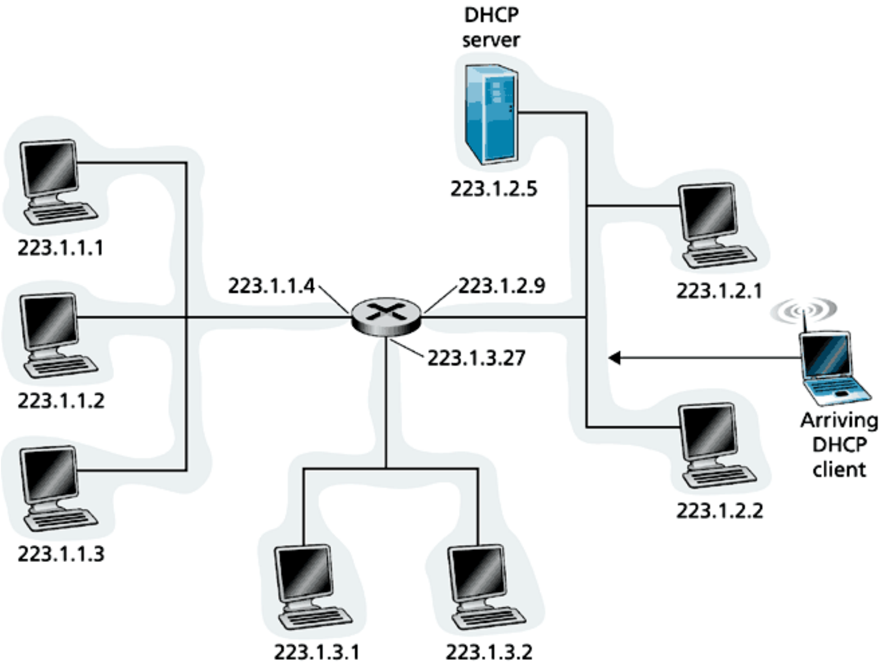
- Arriving at 223.1.4, destined for 223.1.2.2
- Look up network address of E in router’s RIB
- Router and E directly attached
- Link layer sends datagram to 223.1.2.2 inside link-layer frame via interface 223.1.2.9
- Datagram reaches E

IP
Moving datagram to destination (4/4)



IP

Dynamic Host Configuration Protocol (DHCP, 1/2)

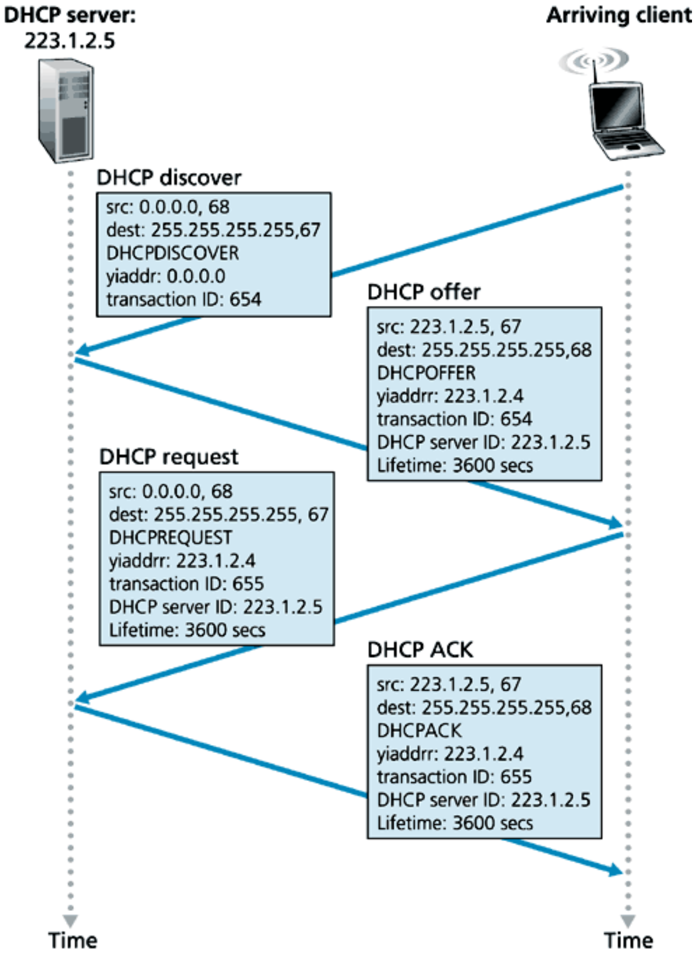


DHCP Sample Capture on Wireshark Wiki

- Application layer
- Allows host to dynamically obtain its IP address from network server when it joins network
  - Allows reuse of addresses (only hold address while connected and “on”)
  - Can renew its lease on address in use
  - Support for mobile users who want to join network

IP

Dynamic Host Configuration Protocol (DHCP, 2/2)



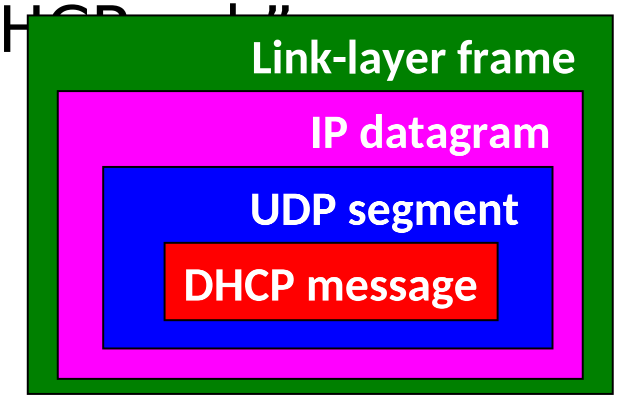
src: 0.0.0.0, 68
dest: 255.255.255.255,67
DHCPDISCOVER
yiaddr: 0.0.0.0
transaction ID: 654

src: 223.1.2.5, 67
dest: 255.255.255.255,68
DHCPOFFER
yiaddrr: 223.1.2.4
transaction ID: 654
DHCP server ID: 223.1.2.5
Lifetime: 3600 secs

src: 0.0.0.0, 68
dest: 255.255.255.255, 67
DHCPREQUEST
yiaddrr: 223.1.2.4
transaction ID: 655
DHCP server ID: 223.1.2.5
Lifetime: 3600 secs

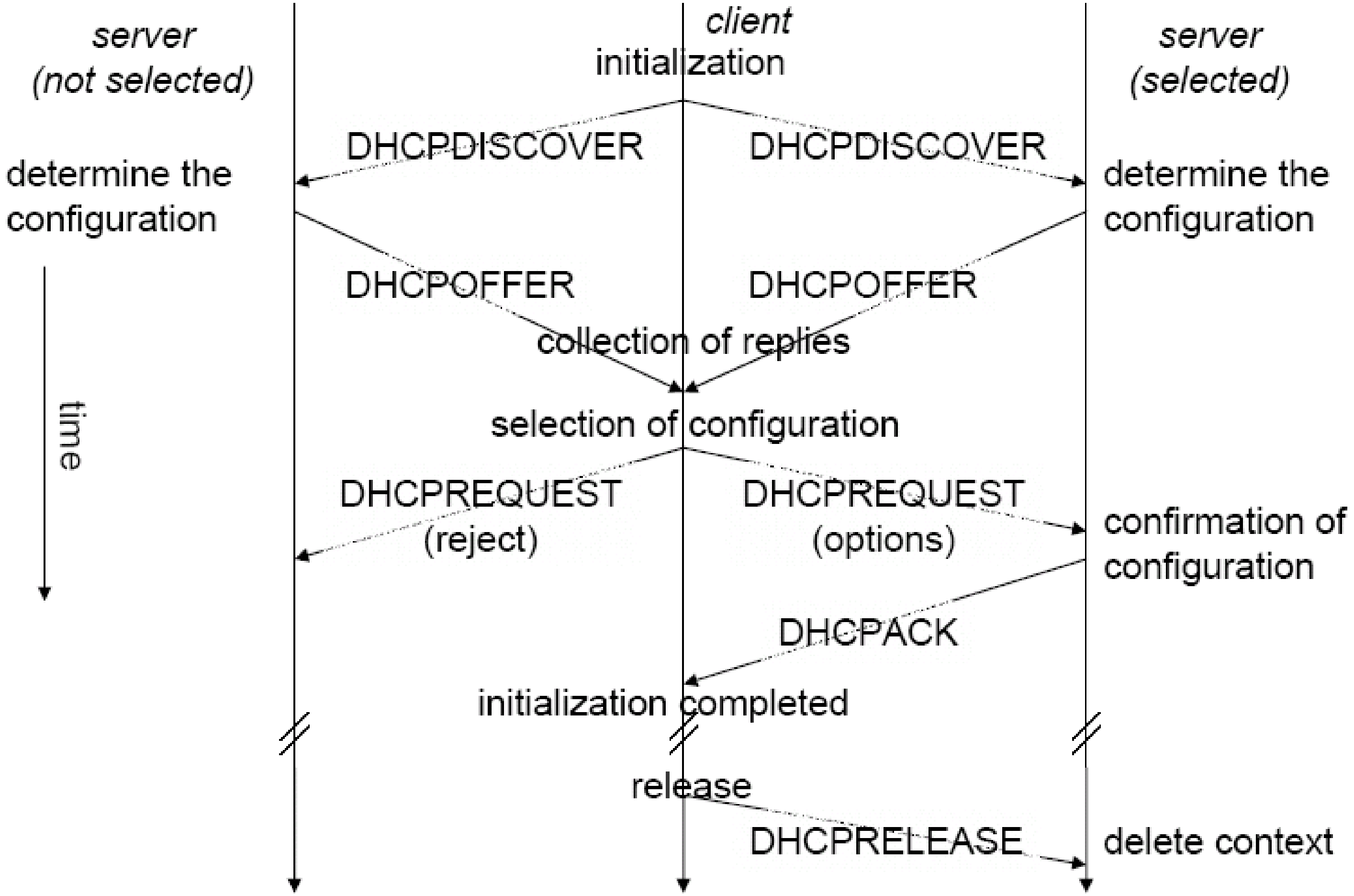
src: 223.1.2.5, 67
dest: 255.255.255.255,68
DHCPACK
yiaddrr: 223.1.2.4
transaction ID: 655
DHCP server ID: 223.1.2.5
Lifetime: 3600 secs

- Host broadcasts “DHCP discover” message
- DHCP server(s) respond with “DHCP offer” message
- Host requests IP address with “DHCP request” message
- DHCP server sends address in “DH



IP

Competition between two DHCP servers



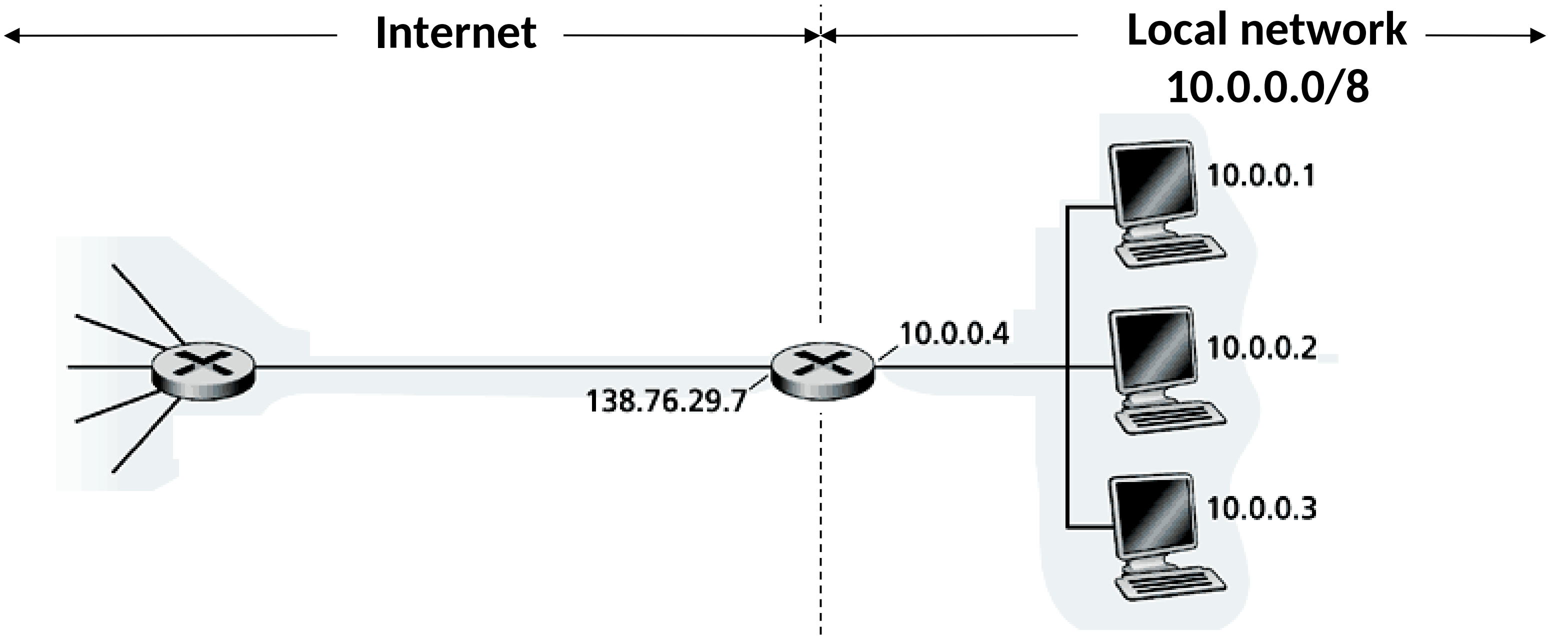
IP

Network Address Translation with Port Translation (NAT-P)

- Context : shortage of IPv4 addresses
- Local network uses just one IP address as far as outside world is concerned.
- A.k.a. masquerading (source) and IP-forwarding (destination) in Linux
- Advantages
  - Faces address shortage: just one IP address is used for all devices.
  - Eases (re-)addressing
    - Addresses of devices can be changed in local network without notifying outside world.
    - Another ISP can be selected without changing addresses of devices in local network.
    - No need to be allocated range of addresses from ISP.
  - Network obfuscation and topology hiding: devices and services inside local network not explicitly addressable, visible by outside world (a security plus).

IP

Network Address Translation with Port Translation (NAT-P)



All datagrams leaving local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0.0/8 address for source, destination (as usual)

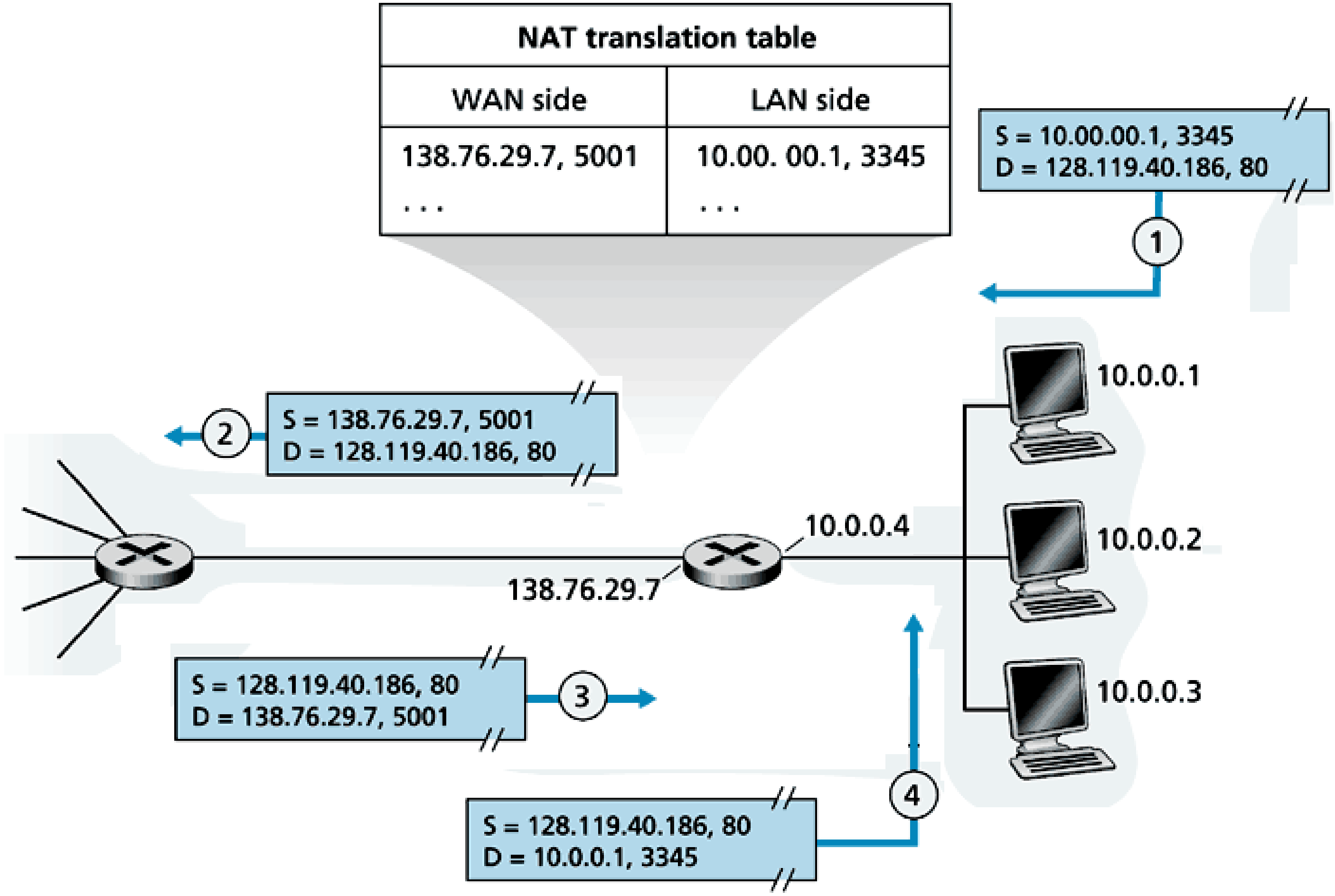
IP

NAT-P – Implementation

- Outgoing datagrams
  - Replace (source IP address, port number) of every outgoing datagram to (NAT IP address, new port number)
  - Remote clients/servers will respond using (NAT IP address, new port number) as destination address/port pair
- NAT translation table : mapping (source IP address, port number) to (NAT IP address, new port number)
- Incoming datagrams : replace (NAT IP address, new port number) in destination fields of every incoming datagram with corresponding (source IP address, port number) stored in NAT table

IP

NAT-P – Example



NAT translation table	
WAN side	LAN side
138.76.29.7, 5001	10.00. 00.1, 3345
. . .	. . .

IP

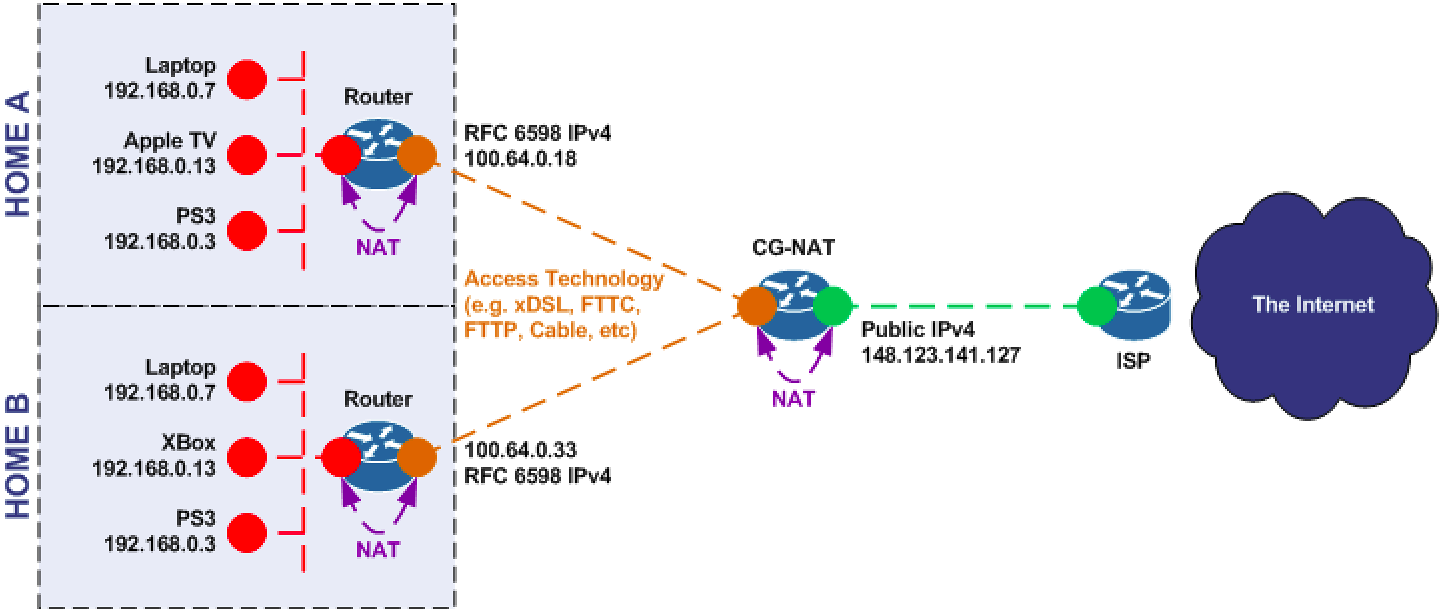
NAT-P – Controversy

- Routers should only process up to layer 3
  - Port numbers are part of layer 4 headers
  - Port numbers become an extended version of the IP address (draft-rosenberg-internet-waist-hourglass-00, February 2008)
- Violates E2E principle
- Inside computers can not behave as servers without tricks (passive FTP, P2P connection reversal, STUN, TURN, etc.)
- (Pseudo-)push services actually pull content: RSS clients regularly poll server for updates
- Address shortage should instead be solved by IPv6…although some claim NAT advantages also apply to IPv6 as well (RFC 5902, July 2010)

IP

Carrier-Grade NAT (CGN)

- Residential NAT: 16-bit port-number field → 60,000+ simultaneous connections with a single LAN-side address
- Carrier-Grade NAT (RFC 6598): IPv4 prefix 100.64.0.0/10 used for facing address shortage



Source : http://netnix.org/2013/09/22/the-long-road-to-ipv6/

IP

Internet Control Messaging Protocol (ICMP)

- Used by hosts, routers, gateways to communication network-level information
  - Error reporting: unreachable host, network, port, protocol
  - Echo request/reply (used by ping)
- Network-layer “above” IP: ICMP messages carried as IP payload
- ICMP message
  - Type
  - Code
  - First 8 bytes of IP datagram causing error
- Traceroute implemented with ICMP messages

Type	Code	Description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - rarely used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

IP

ICMPv6

- Replaces Address Resolution Protocol (ARP) in Layer 2 (Link)
- Probes Path Maximum Transfer Unit (PMTU)
- Additional messages
  - Neighbor discovery (RFC 2461, 3971)
  - Mobility (RFC 3775)
- Multicast support (RFC 2710, 3810)

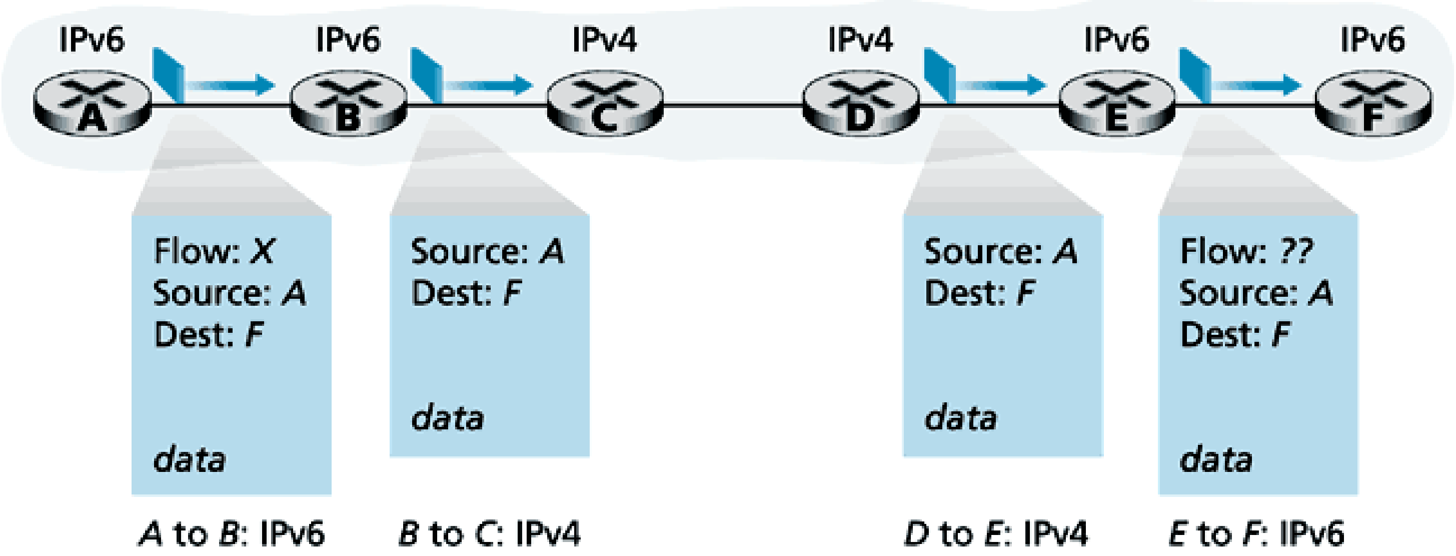
IP

Transition from IPv4 to IPv6

- Not all routers can be upgraded simultaneous
  - No “flag day”
  - The networks have to operate with mixed IPv4 and IPv6 routers
- Two main proposed approaches (RFC 2893)
  - Dual Stack: some routers with dual stack (IPv4, IPv6) can “translate” between formats
  - Tunneling: IPv6 datagram carried as payload in IPv4 datagram among IPv4 routers

Transition from IPv4 to IPv6

Dual Stack



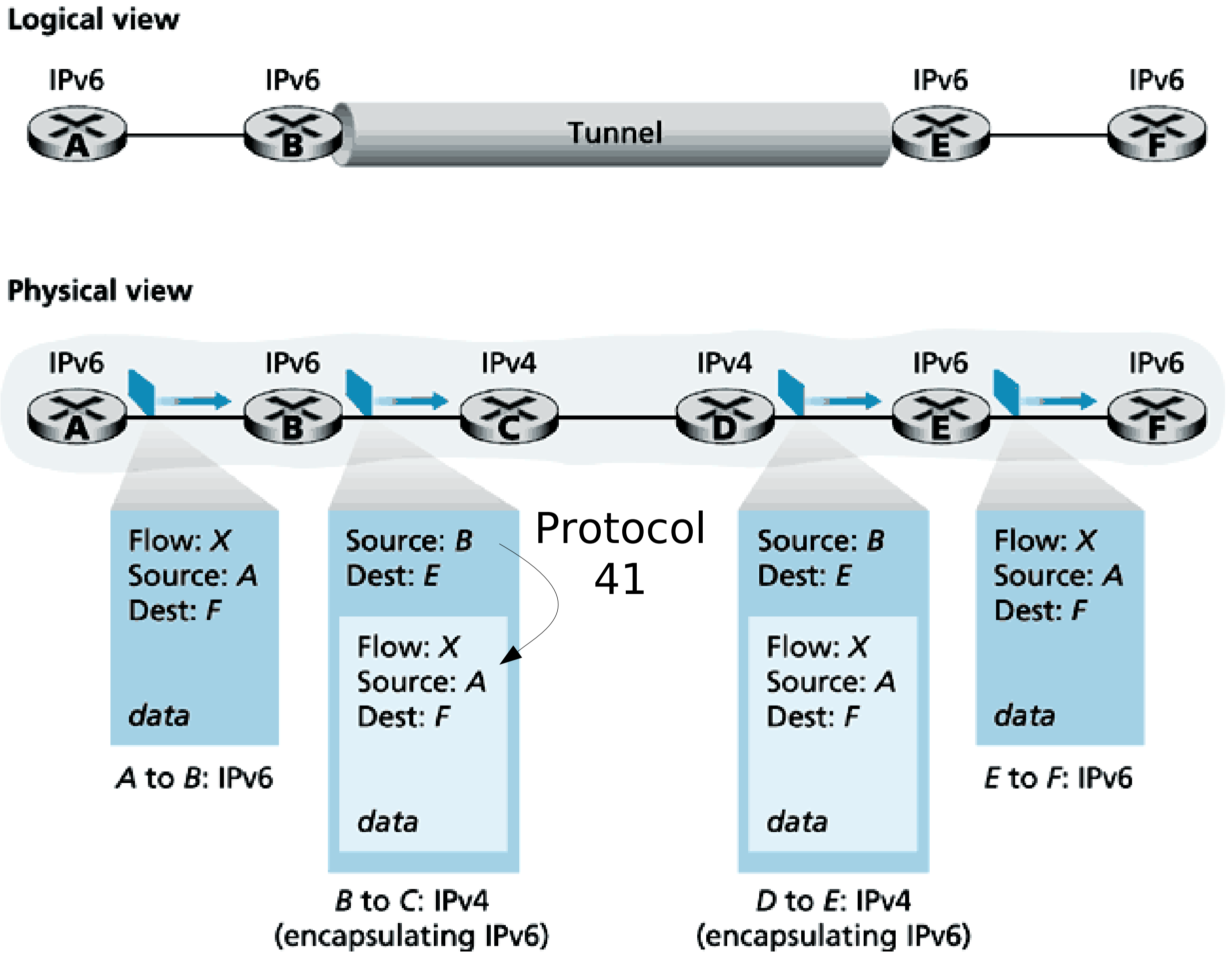
IPv6-specific fields (e.g. flow) are lost when converting to IPv4

Since September 2003 the BELNET backbone is dual-stack (IPv6 range 2001:6a8::/32)

Since April 2009 UNamur is dual-stack (IPv6 range 2001:6a8:3900::/48)

Transition from IPv4 to IPv6

Tunnelling (RFC 7059, November 2013)



Protocol 41

IPv6

in IPv4

- Protocol = 41
- Manual (6in4, GRE, SEAL)
- Automatic (6to4, 6rd)

IPv6

in UDP

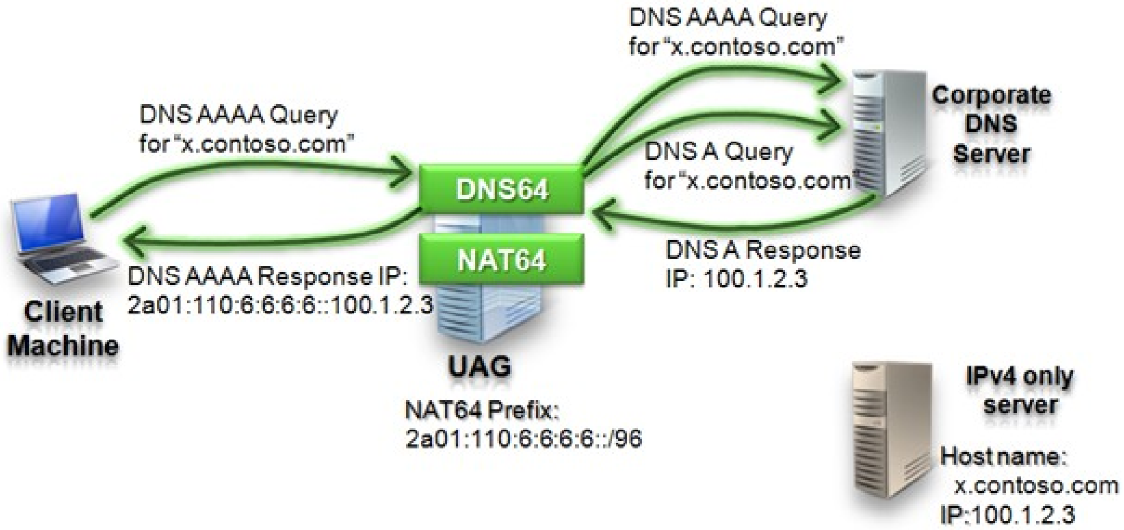
in IPv4

- NAT-friendly
- Teredo, LISP, 6bed4

IP

Transition from IPv4 to IPv6 – DNS64 + NAT64

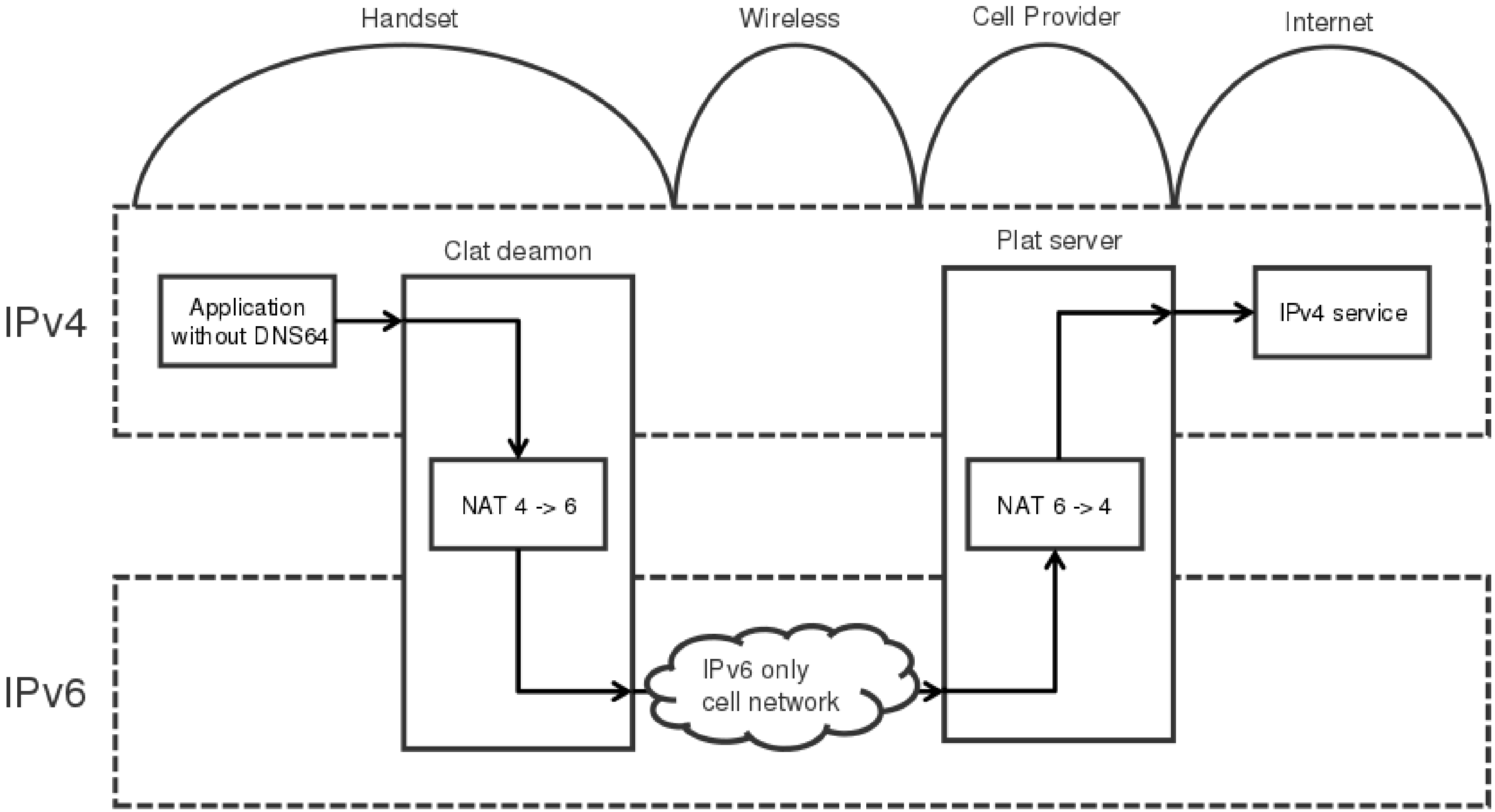
- Solution to enable an IPv6 client to connect to an IPv4 server
- Gateway plays two roles
  - DNS64: translates AAAA query into A query
  - NAT64: generates an IPv6-compliant internal address for the IPv4 server
- Standardised as RFC 6146 and RFC 6147 (April 2011)



IP

Transition from IPv4 to IPv6 – 464XLAT

- Standardised as RFC 6877 (April 2013)

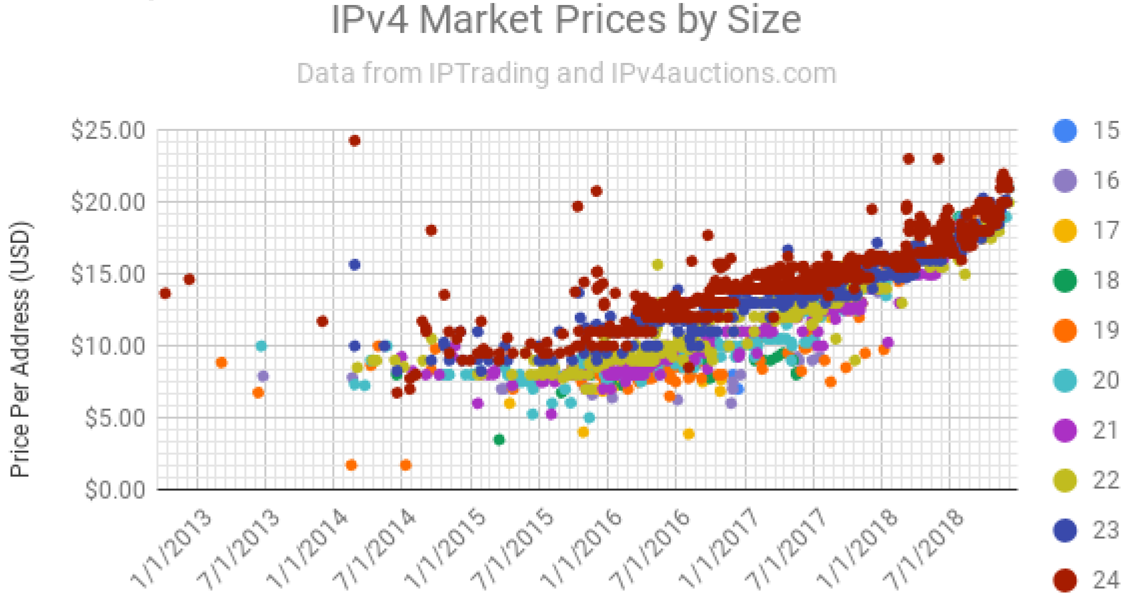


Source : IPv6 @ Telenet, Hans Thienpondt, December 2014

IP

Transition from IPv4 to IPv6 – Status

- Google probe thanks to Eric Vyncke
- Is the transition to IPv6 a « Market Failure »?
  - Opinion by Geoff Huston (APNIC, Sep’09)
  - Prices in the IPv4 market have been rising (Feb’19)



IPv4 Market Prices by Size

Data from IPTrading and IPv4auctions.com

Lee Howard, Retevia

IP

Transition from IPv4 to IPv6 – A lesson

- Enormously difficult to introduce new network-layer protocols (IPv6, multicast, etc)
- Very easy to deploy new application protocols
- Human analogy: changing house foundations vs. wall painting

Outline

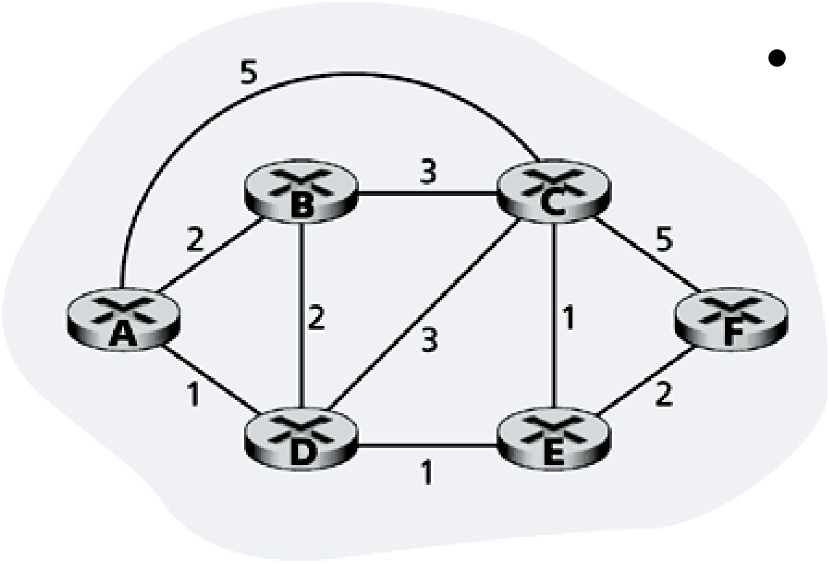
- Introduction
  - Forwarding and routing
  - Network-Layer services
  - Virtual circuit and datagram networks
- What’s inside a router?
- The Internet Protocol (IP) – IPv6 and IPv4
- Routing principles
  - Link state vs. Distance Vector
  - Hierarchical routing
- Routing in the Internet
  - Intra-domain routing: RIP and OSPF
  - Inter-domain routing: BGP
- Broadcast and multicast routing

Forwarding

Routing

Routing principles
Introduction

- Routing protocol determines "good" path (sequence of routers) through network from source to destination



- Graph theory
  - Nodes are routers
  - Edges are physical links
- "Good" path
  - Typically minimum cost path
  - Other definitions possible
    - Link cost: delay, monetary cost, or congestion level
    - All cost the same → least-cost path = shortest path

Routing principles
Classification

- “Link state” algorithms
  - Centralised/global
  - Each router has complete topology, link cost information
  - Each router can build least-cost tree whose root is itself
- “Distance vector” algorithms
  - Decentralised
  - Router knows its neighborhood
    - Physically-connected neighbouring routers
    - Link costs to neighbouring routers
  - Iterative process of computation, by exchanging information with neighbours

Routing principles
Link state algorithm – Dijkstra

- Net topology and link costs known to all nodes
  - Accomplished via “link state broadcast”
  - All nodes have same information
- Each node computes least cost paths to all other nodes
- Example: Dijkstra’s algorithm
  - Iterative
  - After *k* iterations, least cost paths are known to *k* destinations

Routing principles
Dijkstra's algorithm

Notation

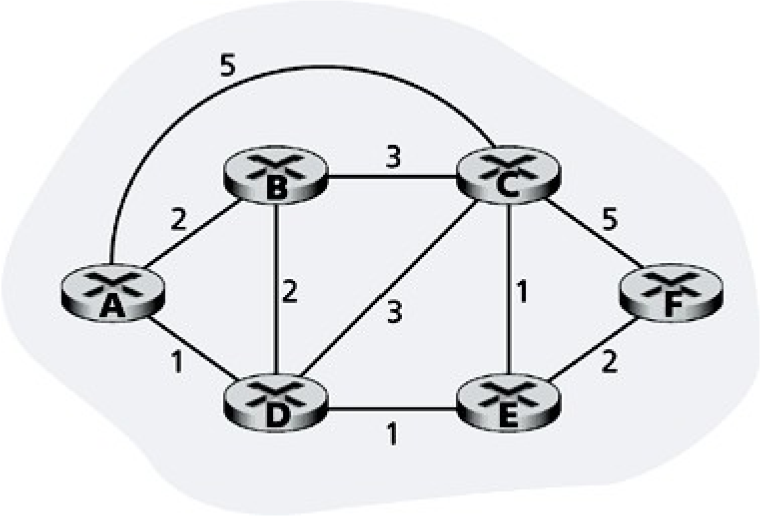
- *c(i,j):* link cost from node *i* to *j*. Cost ∞ if not direct neighbours
- *D(v):* current value of cost of path from source to *v*
- *p(v):* previous node along path from source to *v*
- *N*: set of nodes whose least cost path definitively known

```
Initialization
  N = {A}
  for all nodes v
    if v adjacent to A
      then D(v) = c(A,v)
      else D(v) = infinity

Loop
  find w not in N such that D(w) is a minimum
  add w to N
  update D(v) for all v adjacent to w and not in N:
     D(v) = min( D(v), D(w) + c(w,v) )
  /* new cost to v is either old cost to v or known
  shortest path cost to w plus cost from w to v */
until all nodes in N
```

Routing principles

Dijkstra’s algorithm – Example



E picked arbitrarily

Step	N	D(B), p(B)	D(C), p(C)	D(D), p(D)	D(E), p(E)	D(F), p(F)
Init	A	2, A	5, A	1, A	∞	∞
1	A, D	2, A	4, D	1, A	2, D	∞
2	A, D, E	2, A	3, E	1, A	2, D	4, E
3	A, D, E, B	2, A	3, E	1, A	2, D	4, E
4	A, D, E, B, C	2, A	3, E	1, A	2, D	4, E
5	A, D, E, B, C, F	2, A	3, E	1, A	2, D	4, E

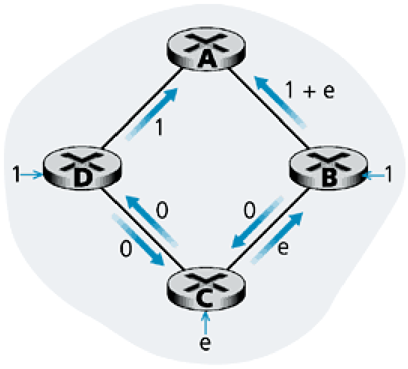
Routing principles

Dijkstra’s algorithm – Discussion

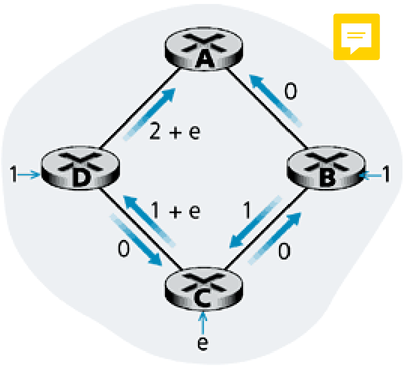
- Routing
  - For each node, the forwarding table gives the predecessor on the least-cost path
  - For each predecessor, the table gives its own predecessor
  - … until ones reaches the next-hop router
- Algorithm complexity
  - Assume n nodes
  - At each iteration, need to check all nodes w not in N
  - (n-1) + (n-2) + … = n*(n+1) / 2 comparisons → $O(n^2)$
  - More efficient implementations possible → O(n log n)
- Drawback: oscillations possible

Routing principles

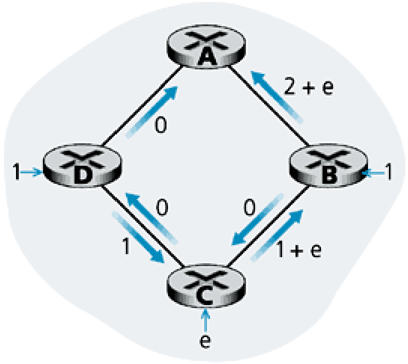
Dijkstra’s algorithm – Oscillation



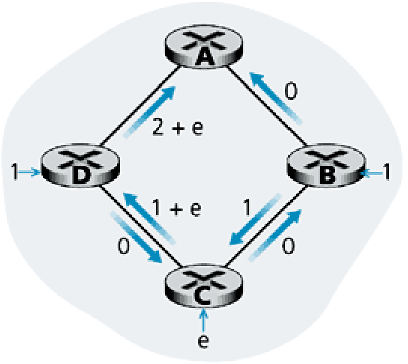
a. **Initial routing**



b. ***B*, *C* detect better path to *A*, clockwise**



c. ***B*, *C*, *D* detect better path to *A*, counterclockwise**



d. ***B*, *C*, *D*, detect better path to *A*, clockwise**

- Link cost = amount of carried traffic
- Assymmetric costs due to assymmetric traffic
  - B and D sends 1 to A
  - C sends e to A
- Solution
  - Trivial and useless: link cost != link traffic
  - Avoid (self-) synchronisation

Routing principles

Distance vector algorithms

- Iterative
  - Continues until neighbouring nodes stop exchanges
  - Self-terminating
- Asynchronous: Route updates due to local changes or update from neighbour
- Distributed: each node communicates only with direct neighbours
- Distance table data structure: each node has its own
  - Row for each possible destination
  - Column for each directly-attached neighbour to node

X	**Z**
Y	$D^X(Y,Z)$

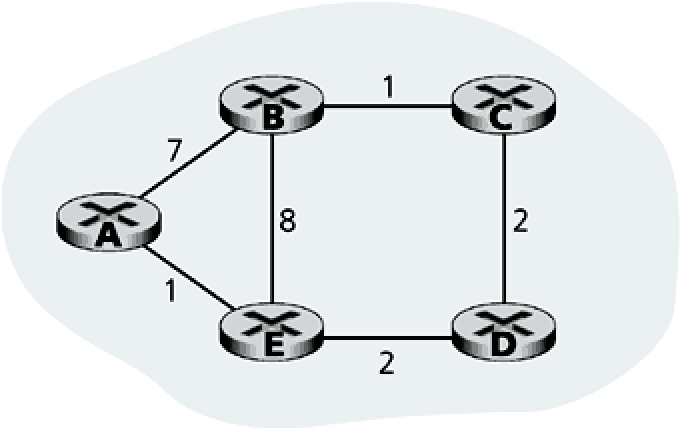
$$D^X(Y,Z) = \text{distance from } X \text{ to } Y \text{ via } Z \text{ as next hop}$$

$$= c(X,Z) + min_w \left[D^Z(Y,w) \right]$$

Routing principles

Distance table – Example

- Each node must know the cost of the least-cost path of its neighbours to each destination



Distance table → **Routing table**

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

E	Next hop	Cost
A	A	1
B	D	5
C	D	4
D	D	2

$$D^E(A,D) = c(E,D) + \min_w[D^D(A,w)] = 2 + 3$$

$$D^E(A,B) \mathrel{!=} 15$$

- The column with the circled entry identifies the next-hop to destination along the least-cost path

Routing principles

Distance table – Bellman-Ford algorithm

At each node X

```
Initialization:
  for all adjacent nodes V:
    $D^X(*,V)$ = infinity  /* operator * means "for all rows" */
    $D^X(V,V)$ = c(X,V)
  for all destinations, Y
    send $\min_W D^X(Y,W)$ to each neighbour /* W over neighbours */

loop
  wait (until link cost change to neighbour V or update from V)

  if (c(X,V) changes by d)
    for all destinations Y:  $D^X(Y,V)$ =  $D^X(Y,V)$ + d

  else if (update received from V w.r.t. destination Y)
    for the single destination Y: $D^X(Y,V)$ = c(X,V) + newval

  if we have a new $\min_W D^X(Y,W)$ for any destination W
    send new value of $\min_W D^X(Y,W)$ to all neighbours

forever
```

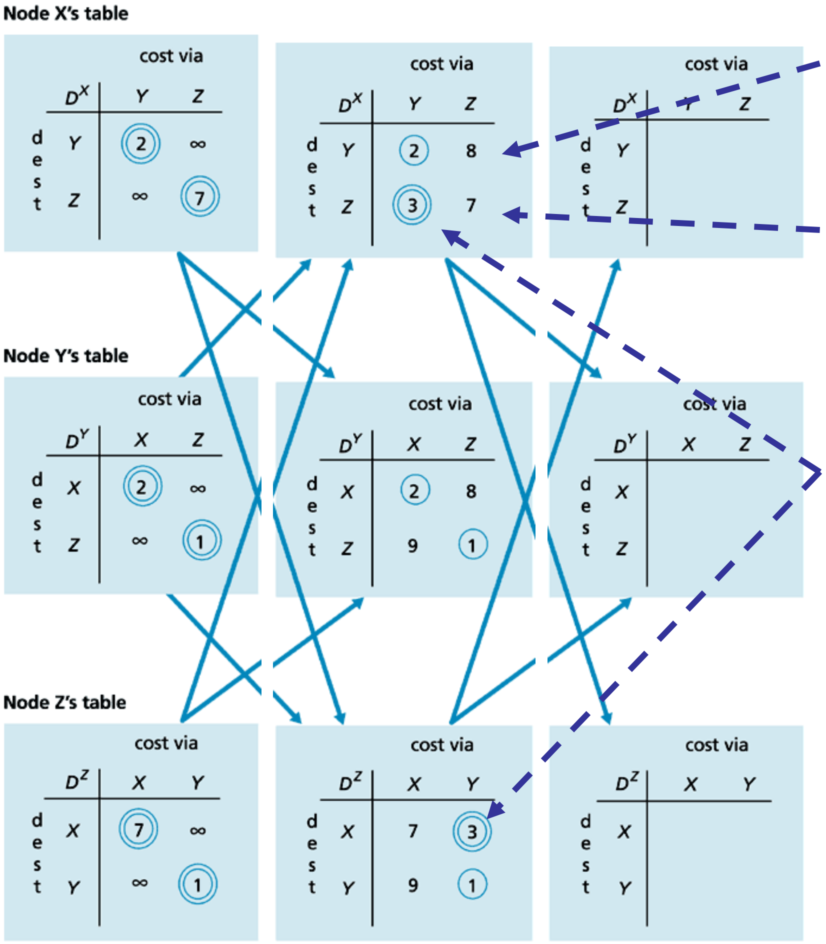
Cost change

Update from V

Update from X

Routing principles

Distance table – Example



Node X's table

	cost via	
D^X	Y	Z
Y	2	∞
Z	∞	7

	cost via	
D^X	Y	Z
Y	2	8
Z	3	7

	cost via	
D^X	Y	Z
Y		
Z		

Node Y's table

	cost via	
D^Y	X	Z
X	2	∞
Z	∞	1

	cost via	
D^Y	X	Z
X	2	8
Z	9	1

	cost via	
D^Y	X	Z
X		
Z		

Node Z's table

	cost via	
D^Z	X	Y
X	7	∞
Y	∞	1

	cost via	
D^Z	X	Y
X	7	3
Y	9	1

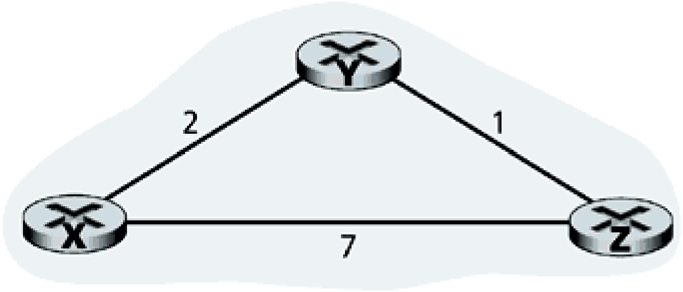
	cost via	
D^Z	X	Y
X		
Y		

$$\begin{aligned} D^X(Y,Z) &= c(X,Z) + min_w \left| D^Z(Y,w) \right| \\ &= 7+1 \\ &= 8 \end{aligned}$$

$$\begin{aligned} D^X(Z,Y) &= c(X,Y) + min_w \left| D^Y(Z,w) \right| \\ &= 2+1 \\ &= 3 \end{aligned}$$

Lower $\min_w[D^X(Y,w)]$

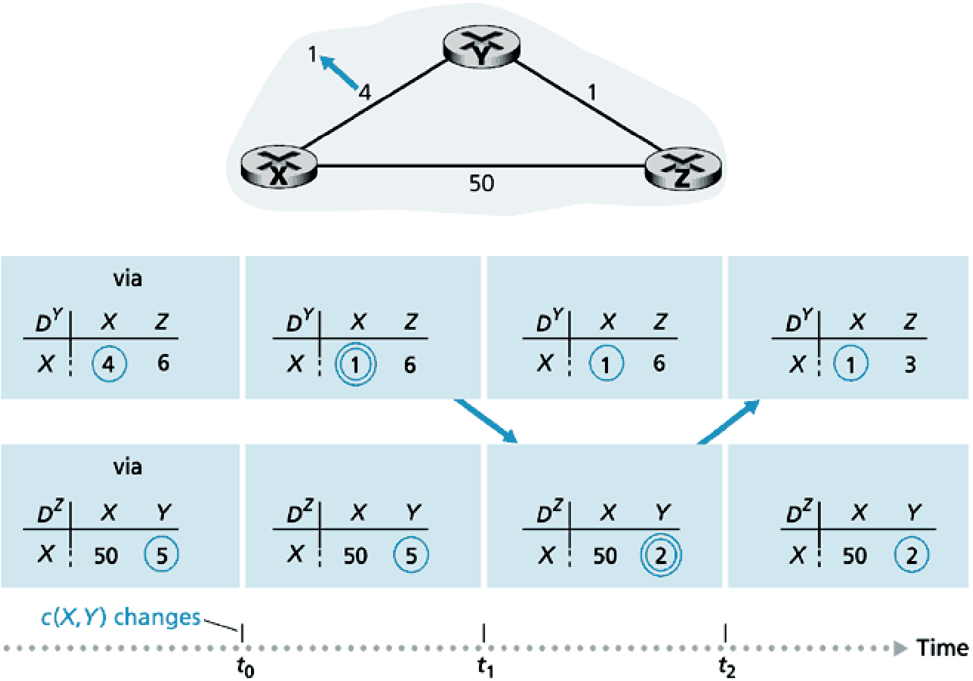
Updates sent to neighbours



Routing principles

Distance table – Link cost changes

- Node detects local link cost change
- Updates distance table
- If cost change in least cost path, notify neighbours

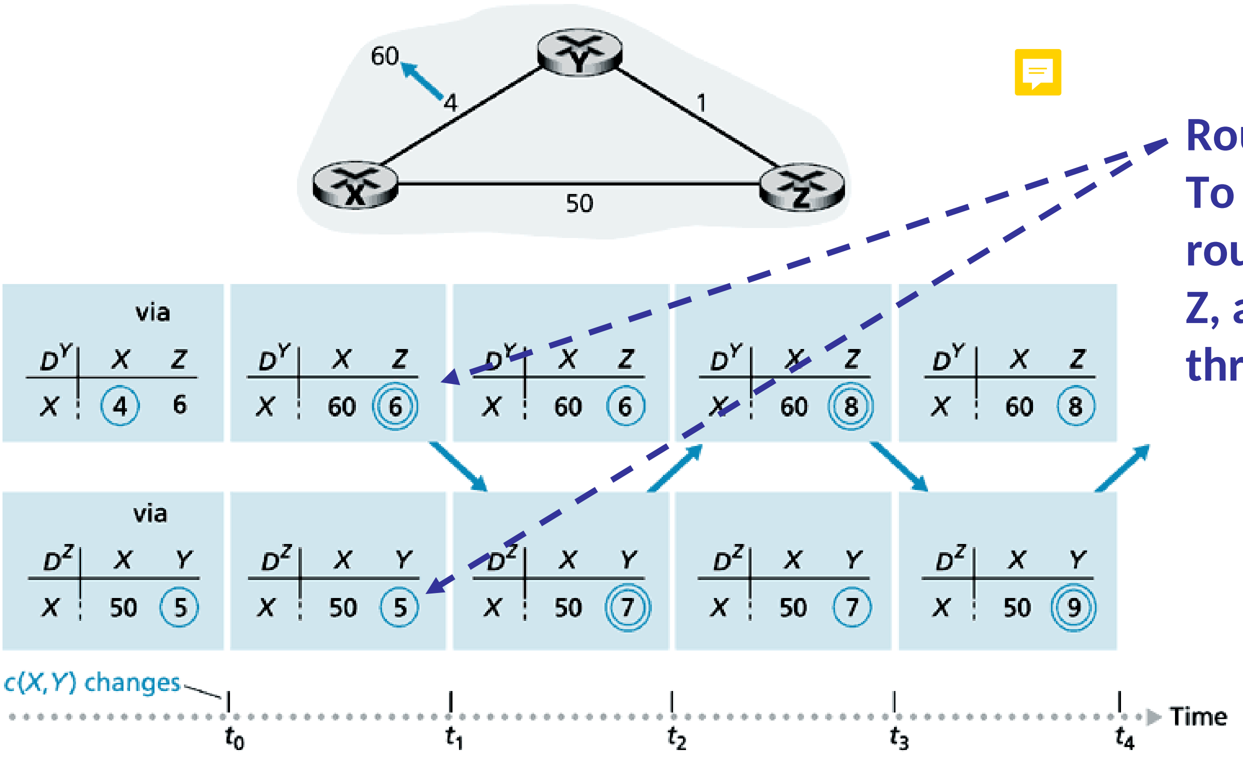


No more updates

Routing principles

Distance table – « Count to infinity » problem

- Good news travel fast, bad news travels slow



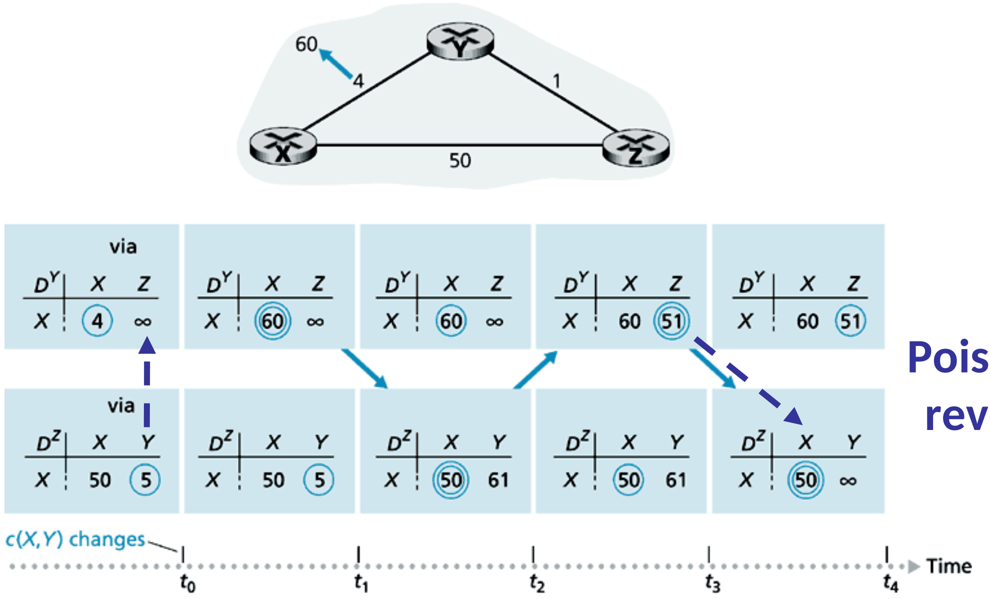
Routing loop
To get to X, Y routes through Z, and Z routes through Y

- Loop broken after 44 iterations

Routing principles

Distance table – Adding « Poisoned Reverse »

- If Z routes through Y for X, Z announces Y an infinite distance to X



Poisoned reverse

Poisoned reverse

- Unfortunately, poisoned reverse does not solve « count to infinity » for more complex networks

Routing principles

Link State vs. Distance Vector

	Link State	**Distance Vector**
Signalling traffic	With *n* nodes, *E* links, O(*nE*) messages sent each update	Exchange between neighbours only, and only if better least-cost path
Convergence speed	O(n^2) algorithm May have oscillations	Can converge slowly Risk of routing loops « Count to infinity » problem
Robustness	Each node computes only its own table Robustness against other node’s malfunction	Incorrect node calculation can propagate through the whole network

- No clear winner. Both approaches used.

Routing principles
Hierarchical routing

- Ideal routing study so far
  - Homogeneous routers
  - Network “flat”
- Not true in practice
- Two main issues
  - Scalability
    - With millions of destinations
    - Can not store all destinations in routing tables
    - Routing table exchange would swamp links
- Administrative autonomy
  - Internet = network of networks
  - Each network admin may want to control routing in its own network

Routing principles
Autonomous Systems (AS)

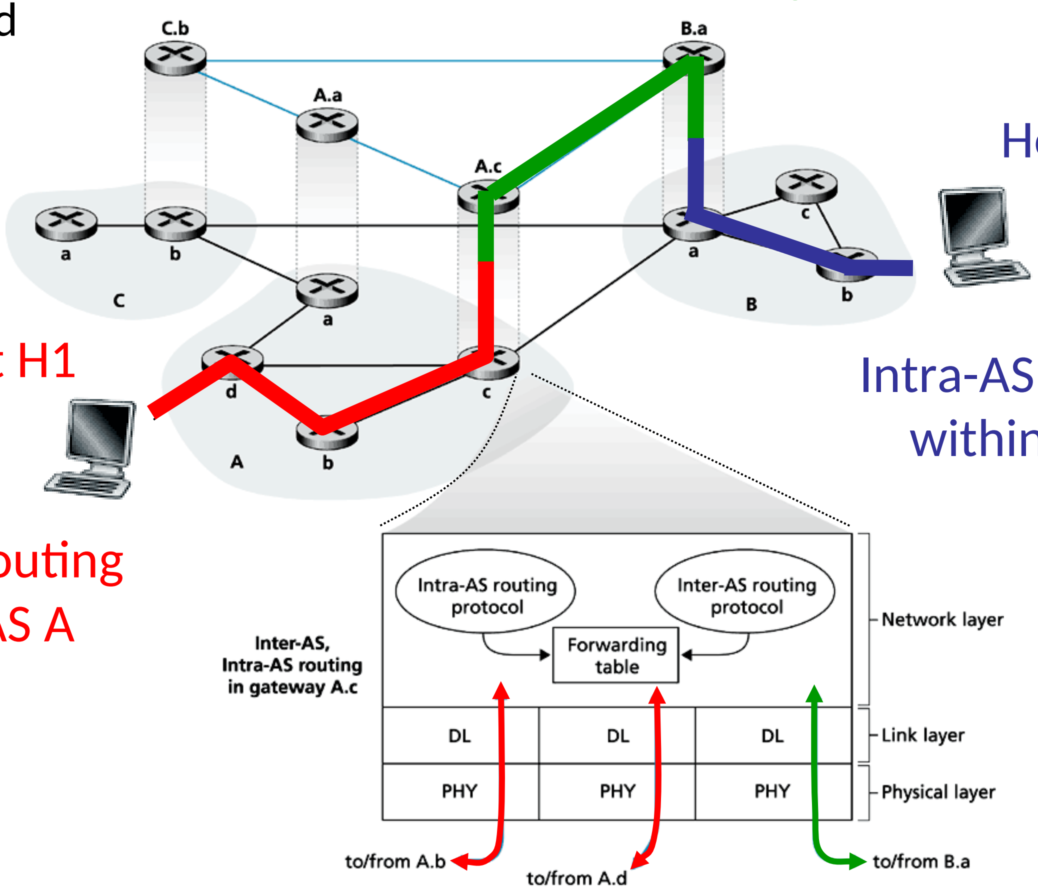
- Aggregate routers into regions called Autonomous Systems (AS)
- Routers in same AS
    - Run same routing protocol
    - Intra-AS routing protocol
    - Routers in different ASs can run different intra-AS routing protocols
- Gateway routers
    - Special routers in AS
    - Run intra-AS routing protocol with all other routers in AS
    - Also responsible for routing to destinations outside AS
    - Run inter-AS routing protocol with other gateway routers

Routing principles

Autonomous System (AS)

Gateway routers A.a, A.c, B.a and C.b

Inter-AS routing between A and B



Host H2

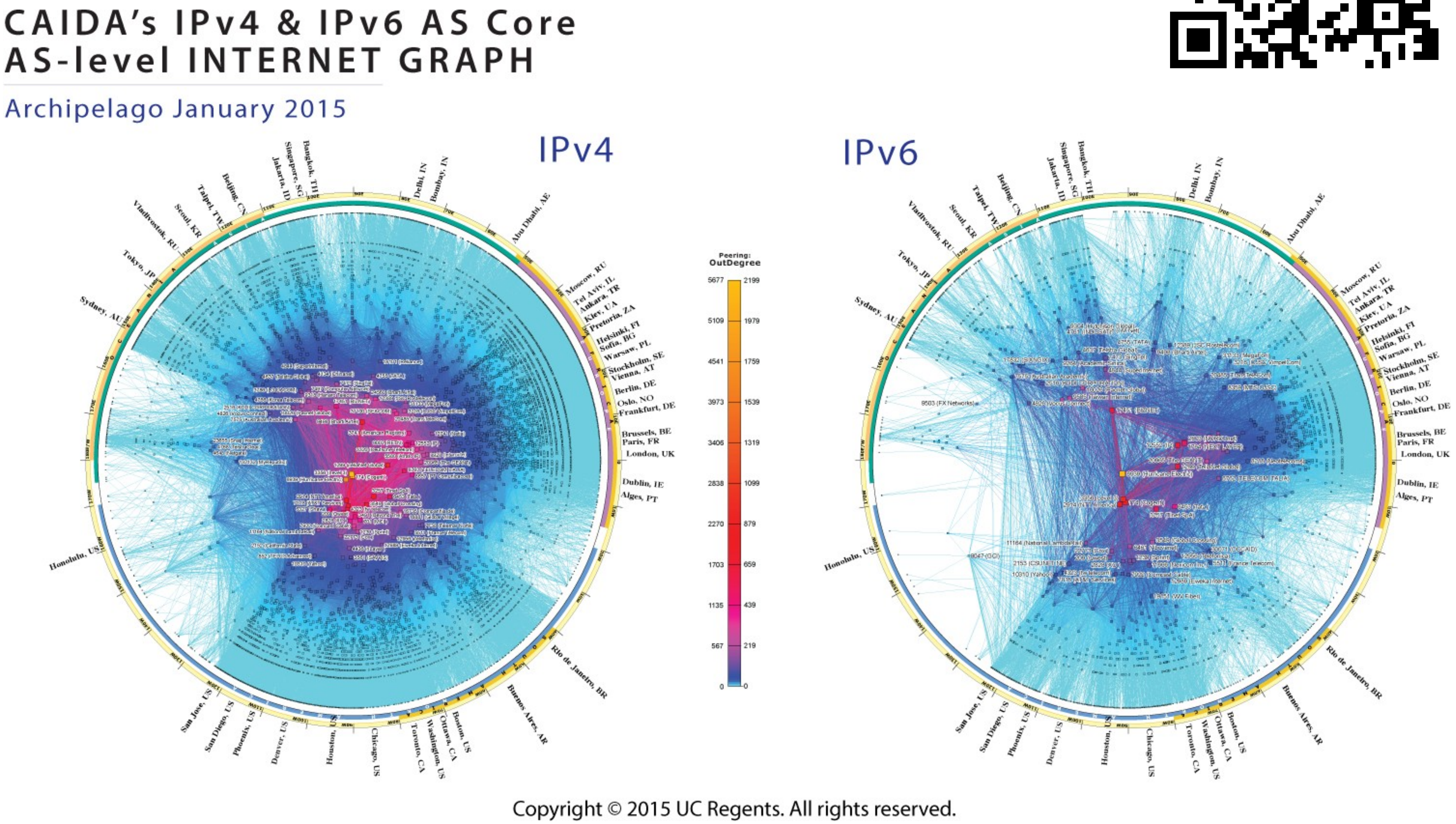
Host H1

Intra-AS routing within AS B

Intra-AS routing within AS A

Routing principles

IPv6 and IPv4 AS Core from CAIDA

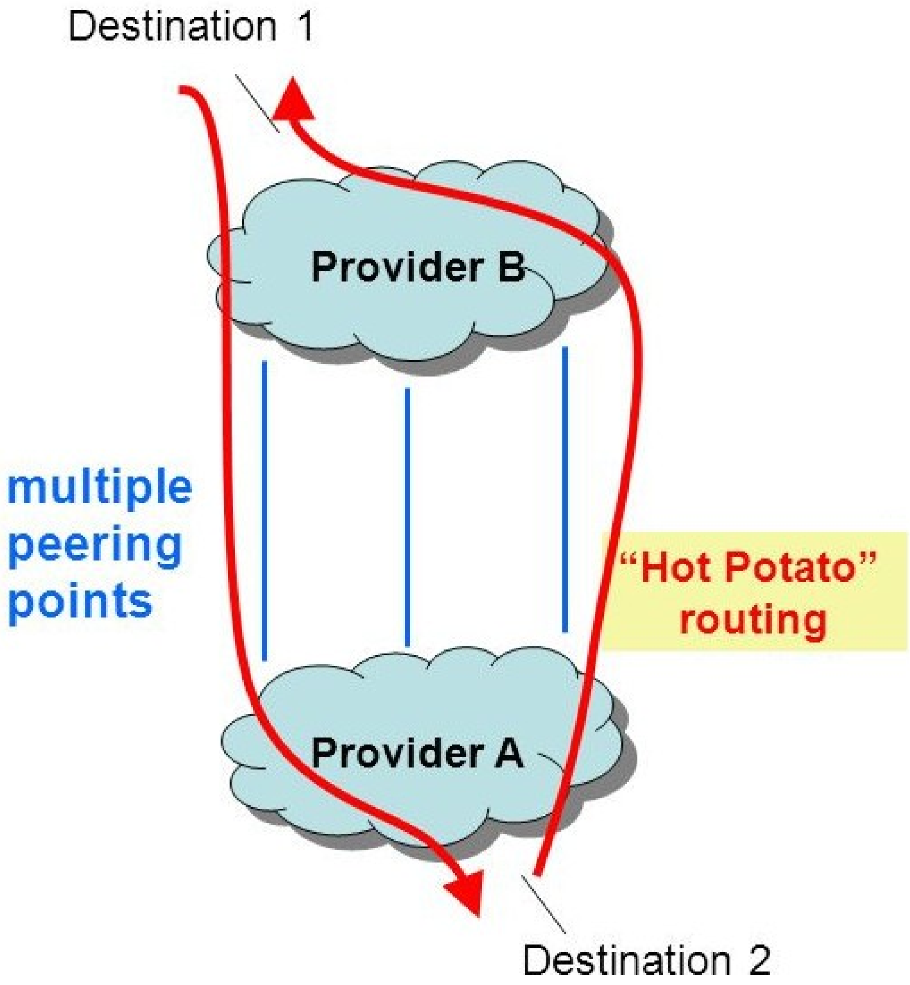


CAIDA's IPv4 & IPv6 AS Core
AS-level INTERNET GRAPH

Archipelago January 2015

Routing principles

Hot Potato Routing



- Host in ISP A willing to contact external subnet x through ISP B
- If x reachable through several equal cost routes, forward to closest ISP A gateway router
- Alternative: traffic engineering (TE)

Outline

- Introduction
  - Forwarding and routing
  - Network-Layer services
  - Virtual circuit and datagram networks
- What’s inside a router?
- The Internet Protocol (IP) – IPv6 and IPv4
- Routing principles
  - Link state vs. Distance Vector
  - Hierarchical routing
- Routing in the Internet
  - Intra-domain routing: RIP and OSPF
  - Inter-domain routing: BGP
- Broadcast and multicast routing

Forwarding

Routing

Routing in the Internet
Introduction

The Global Internet consists of Autonomous Systems (AS) interconnected with each other

- Stub AS: only source/destination
  - Pure stub: one connection to other ASs
  - Multihomed stub: multiple connections to other ASs, no transit however
- Provider AS, hooking many ASs together
  - Transit:access to every publicly reachable destination provided for a fee
  - Peering: customer traffic is exchanged between two networks and the access provided it is only to each other's network and customers

Two-level routing

- Intra-AS (a.k.a. Interior Gateway Protocol)
  - Administrator responsible for routing strategy within own network
  - Examples: RIP, OSPF, IS-IS
- Inter-AS (a.k.a. Exterior Gateway Protocol)
  - Unique standard for inter-AS routing: BGP

Routing in the Internet

Intra-AS Routing - Routing Information Protocol (RIP)

- Distance vector algorithm: neighbouring routers exchange information
- Metric
    - Link cost = 1
    - Number of hops (MAX = 15)
    - Use limited to AS fewer than 15 hops in diameter
- RIP Response Message/Advertisement
    - Routing updates exchanged every 30 s
    - Each advertisement list of up to 25 destination networks within AS, as well as the distance to them

Routing in the Internet

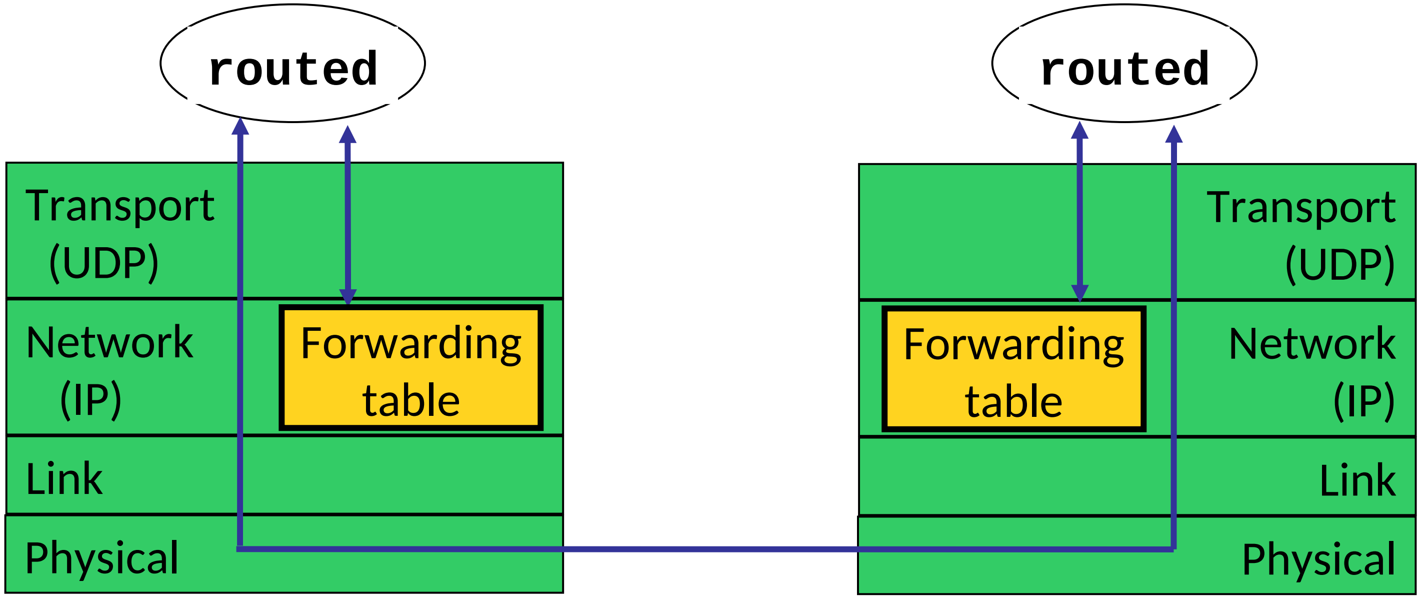
Intra-AS Routing – Link failure in RIP

- If no advertisement heard after 180 s, neighbour/link declared dead
- Routes via neighbour invalidated
- New advertisements sent to neighbours
- Neighbours in turn send out new advertisements (if tables changed)
- Failure information quickly propagates to entire network
- Poisoned reverse used to prevent loops

Routing in the Internet

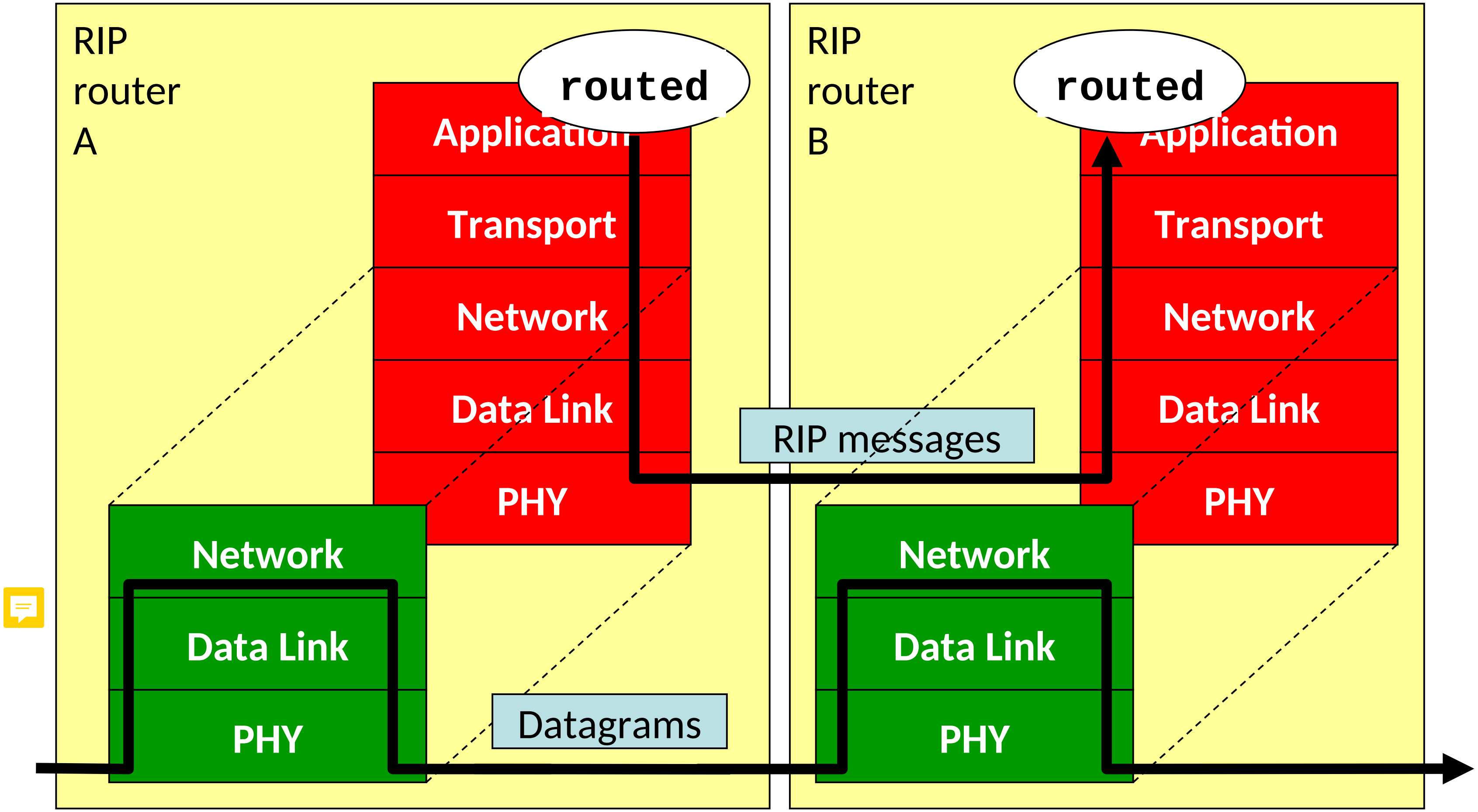
Intra-AS Routing – RIP Implementation (1/3)

- RIP requests/responses exchanged as UDP segments
- Daemon **`routed`** executes RIP protocol in the application layer to update forwarding tables of network layer



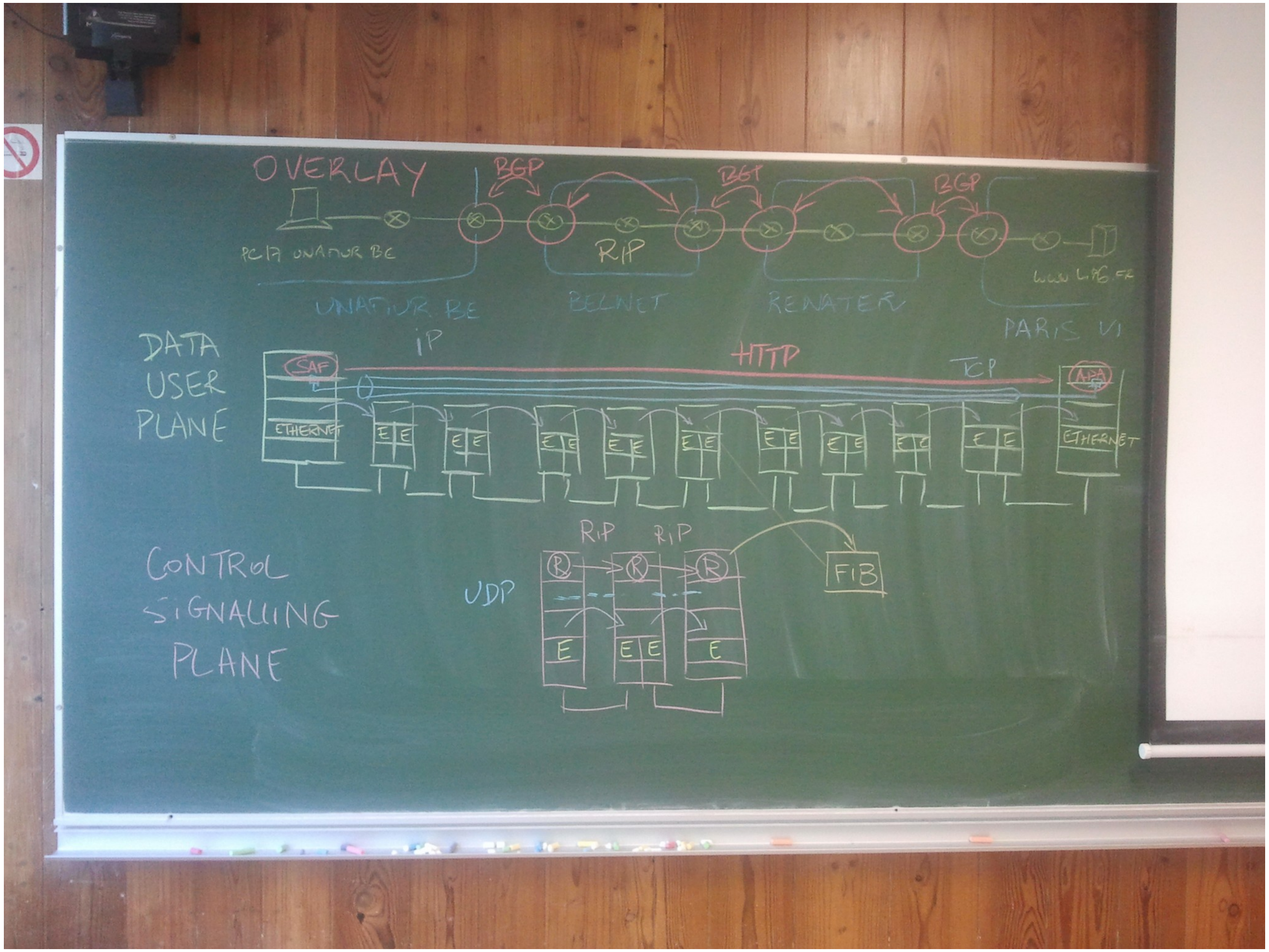
Routing in the Internet

Intra-AS Routing – RIP Implementation (2/3)



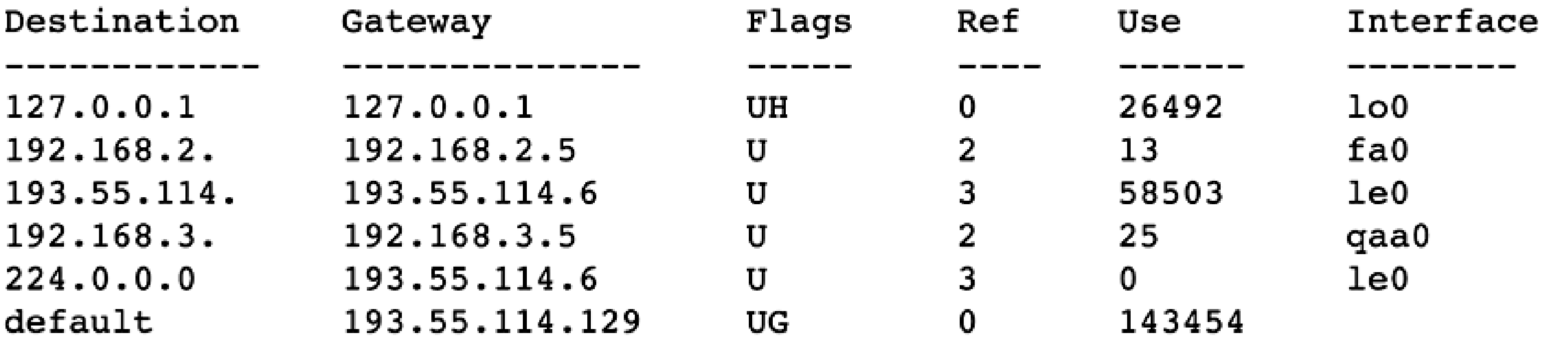
Routing in the Internet

Intra-AS Routing – RIP Implementation (3/3)



Routing in the Internet

Intra-AS Routing – RIP table



```
Destination      Gateway             Flags    Ref    Use       Interface
------------     --------------      -----    ----   ------    ---------
127.0.0.1        127.0.0.1           UH       0      26492     lo0
192.168.2.       192.168.2.5         U        2      13        fa0
193.55.114.      193.55.114.6        U        3      58503     le0
192.168.3.       192.168.3.5         U        2      25        qaa0
224.0.0.0        193.55.114.6        U        3      0         le0
default          193.55.114.129      UG       0      143454
```

- Three attached class C networks (LANs) via interfaces **`fa0`**, **`le0`** and **`qaa0`**
- Router only knows routes to attached LANs
- Default router 193.55.114.129 used to “go up”
- Multicast address 224.0.0.0
- Loopback interface 127.0.0.1
- **U** = Up, **G** = Gateway, **H** = Complete host address

Routing in the Internet

Intra-AS Routing – Open Shortest Path First (OSPF)

- Link State algorithm
  - Flooding of link state information
  - Advertisements disseminated to entire AS
  - Dijkstra least-cost path algorithm
  - Shortest path tree → routing table
- OSPF vs. RIP
  - Conceived as successor of RIP with advanced features
  - OSPF with unitary link cost = RIP
  - Cost determination is a matter of *policy* (unitary cost → minimum hop counting, inverse link capacity, etc.)
  - OSPF messages directly over IP (rather than TCP or UDP like RIP)

Routing in the Internet

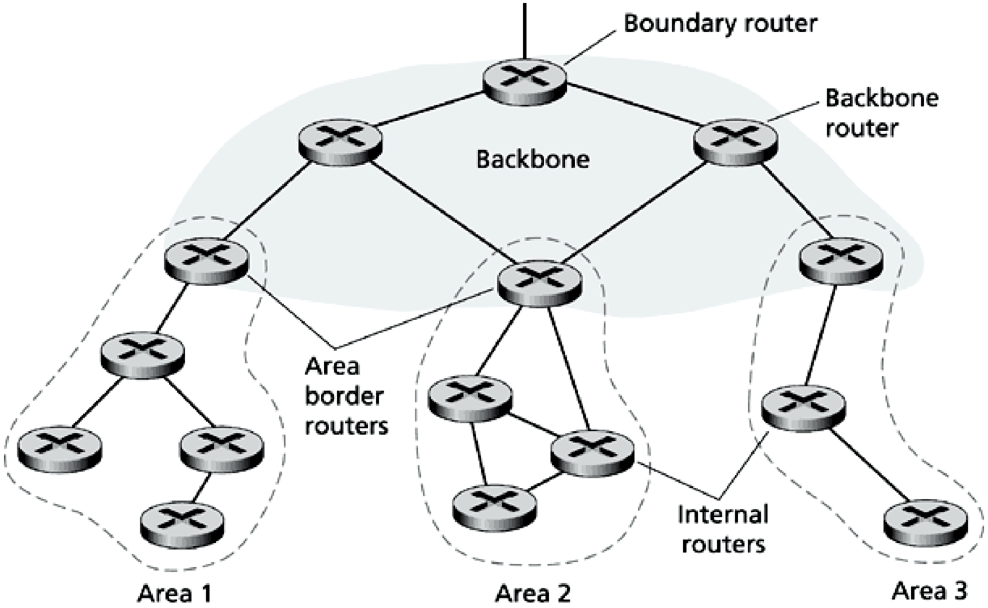
Intra-AS Routing – OSPF Advanced Features

- Security
    - All OSPF messages authenticated to prevent malicious intrusion
    - Only trusted routers can participate
- Multiple same-cost paths allowed (only one selected path in RIP)
- Integrated uni- and multicast support. Multicast OSPF (MOSPF) uses same topology data base as OSPF.
- Support for hierarchy within a single routing domain

Routing in the Internet

Intra-AS Routing – OSPF Hierarchy Support

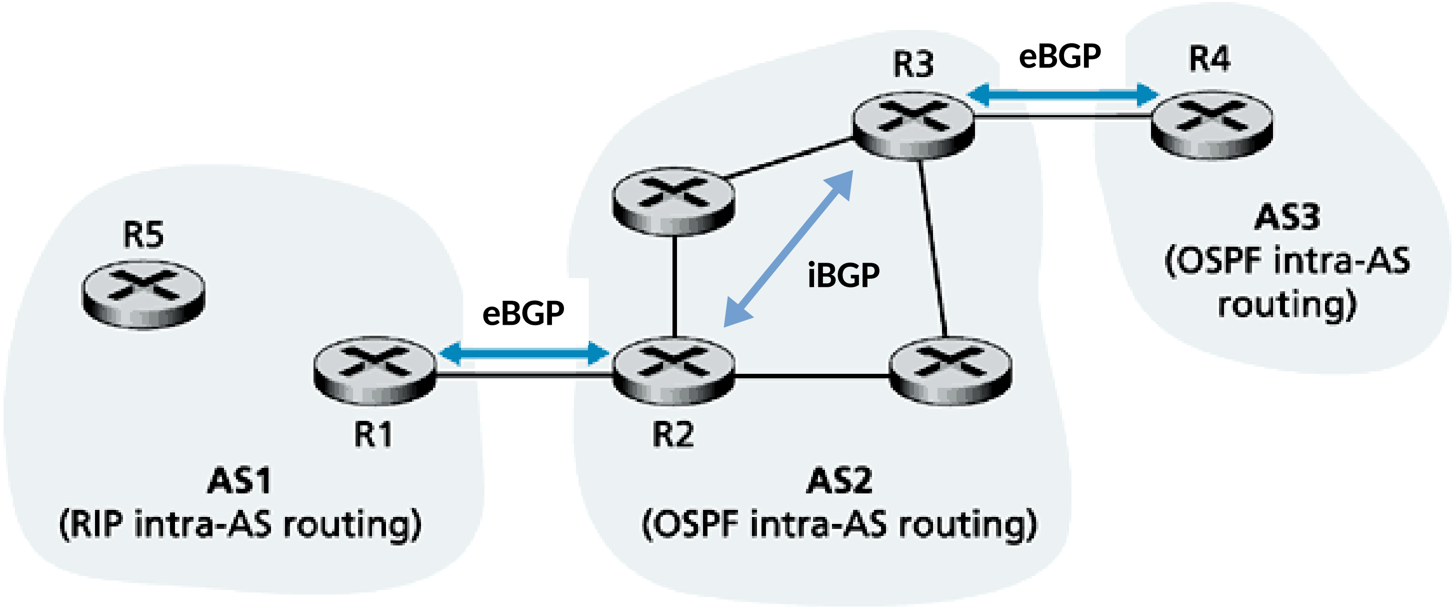
- OSPF AS can be configured into areas
- Each area runs OSPF internally
- Internal structure of the area invisible from outside
- Area border routers route packets outside the area
- Backbone area route traffic between the areas and to other ASs



Routing in the Internet

Inter-AS Routing – Border Gateway Protocol (BGP)

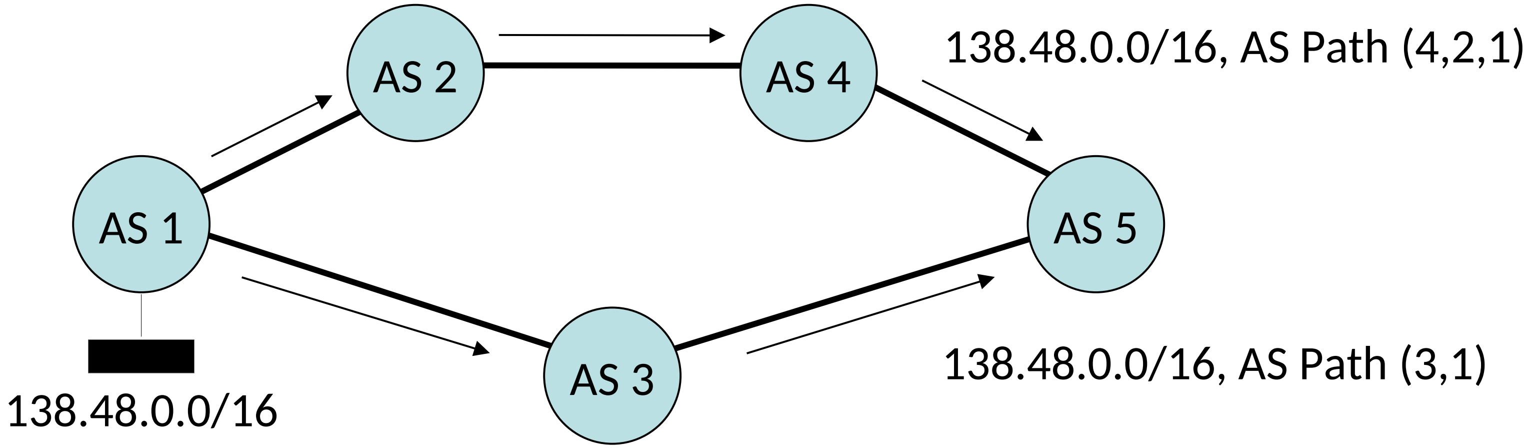
- The de facto standard
- Path Vector protocol
  - Similar to Distance Vector protocol
  - BGP peers exchange detailed path information (list of ASs to destination) over TCP, port 179



Routing in the Internet

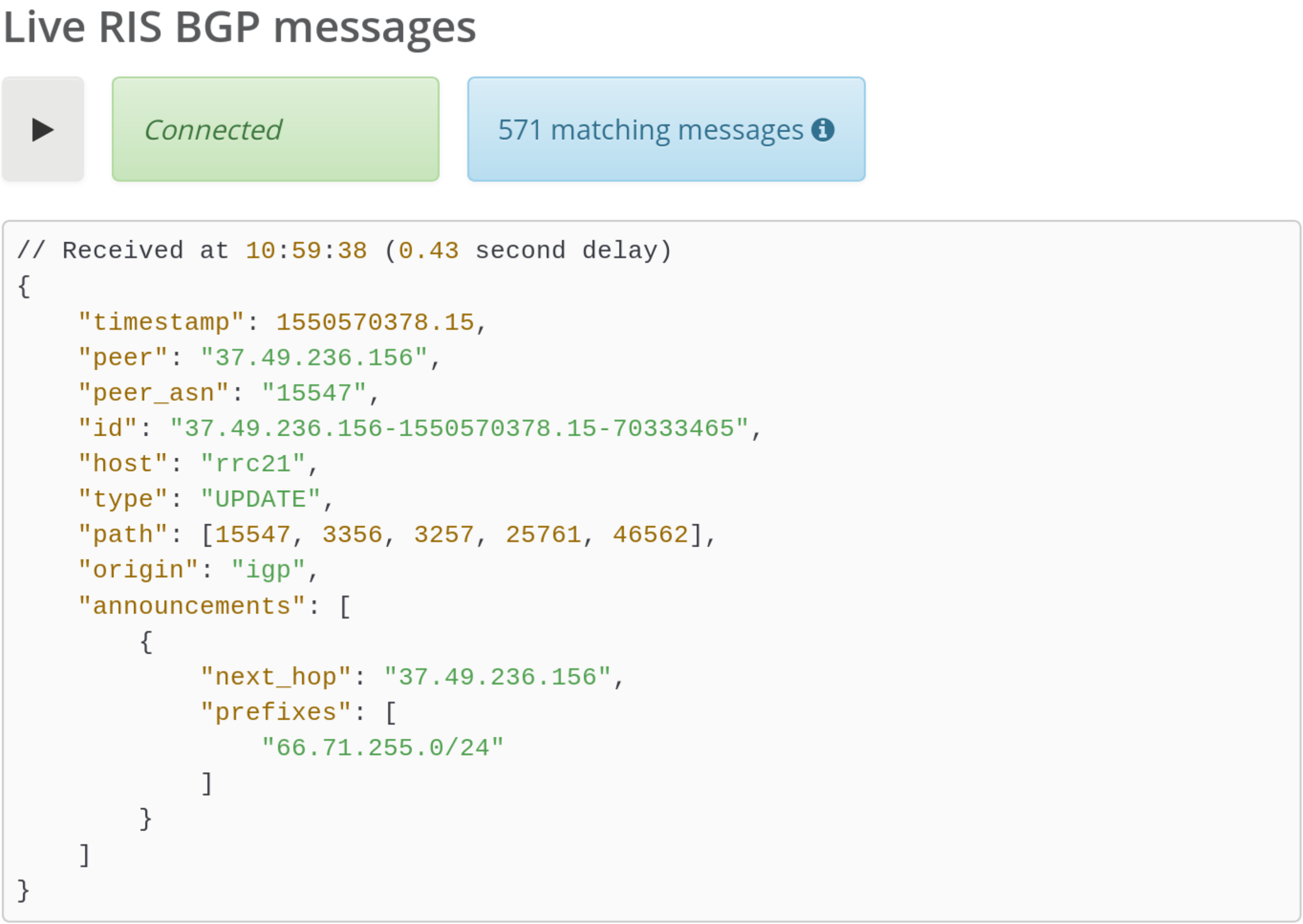
Inter-AS Routing – Path Vector Protocol

- AS identified with 4-Byte Autonomous System Number (ASN) assigned by one of four Regional Internet Registries (RIR)
- BGP routes to CDIRised prefixes, not individual hosts
- AS Path = ASN1, ASN2, ASN3, etc.

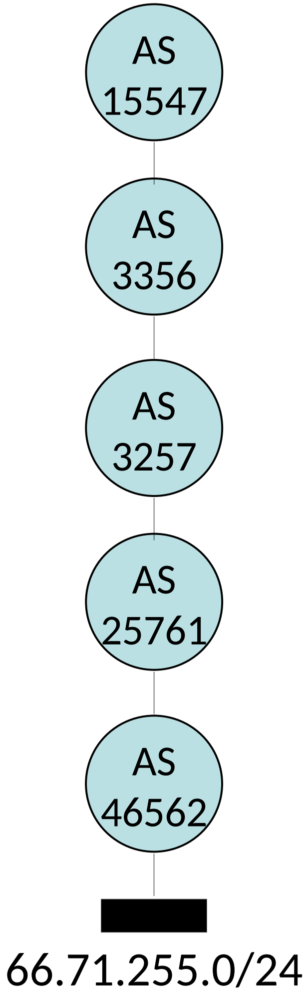


Routing in the Internet

Inter-AS Routing – Routing Information Service Live

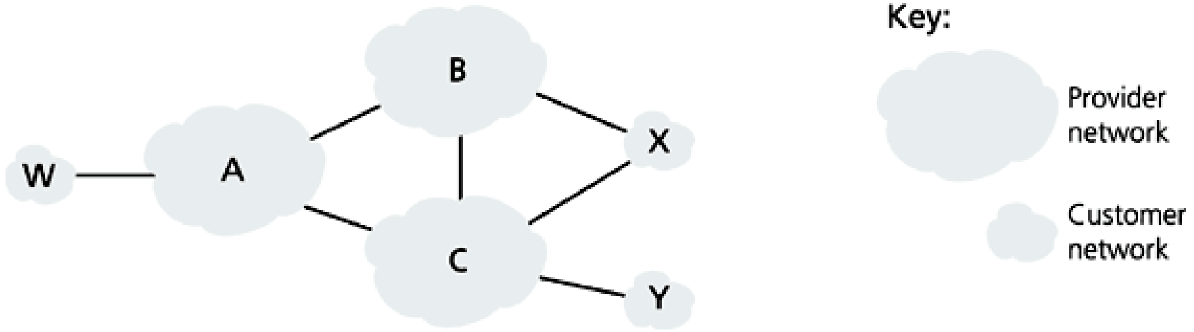


```
// Received at 10:59:38 (0.43 second delay)
{
    "timestamp": 1550570378.15,
    "peer": "37.49.236.156",
    "peer_asn": "15547",
    "id": "37.49.236.156-1550570378.15-70333465",
    "host": "rrc21",
    "type": "UPDATE",
    "path": [15547, 3356, 3257, 25761, 46562],
    "origin": "igp",
    "announcements": [
        {
            "next_hop": "37.49.236.156",
            "prefixes": [
                "66.71.255.0/24"
            ]
        }
    ]
}
```



Routing in the Internet

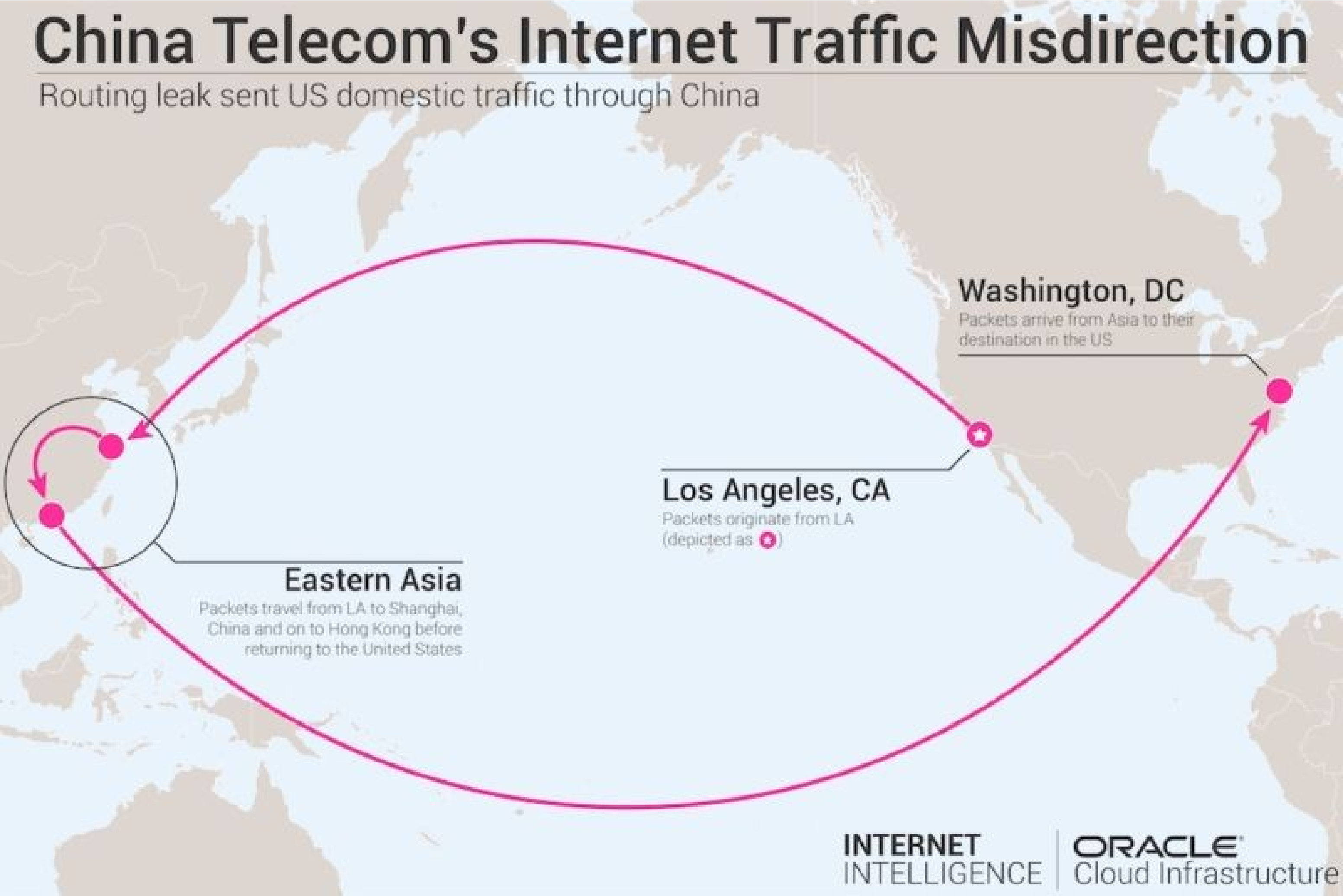
Inter-AS Routing – BGP example



- Stub network's viewpoint
    - X is multi-homed stub network
    - Could forward traffic between B and C
    - To avoid it, X does not advertise any path but to itself
- Backbone provider network's viewpoint
    - X customer → B advertises route BAW to X
    - C competitor → rule of thumb: B does not advertise BAW to C because CW traffic is not having source/destination in any B's customer network

Routing in the Internet

Inter-AS Routing – BGP hijacking



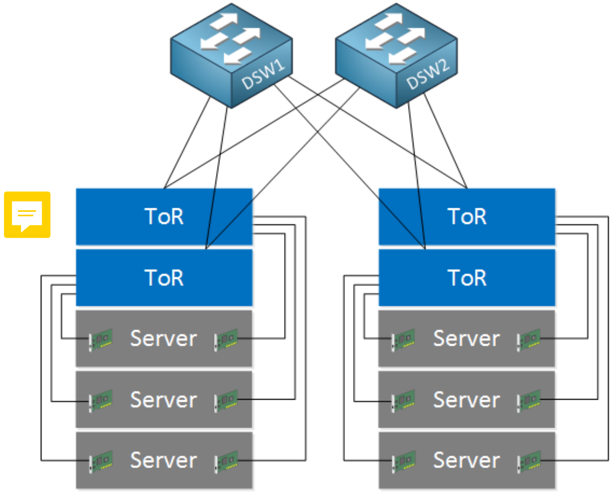
https://arstechnica.com/information-technology/2018/11/strange-snafu-misroutes-domestic-us-internet-traffic-through-china-telecom/

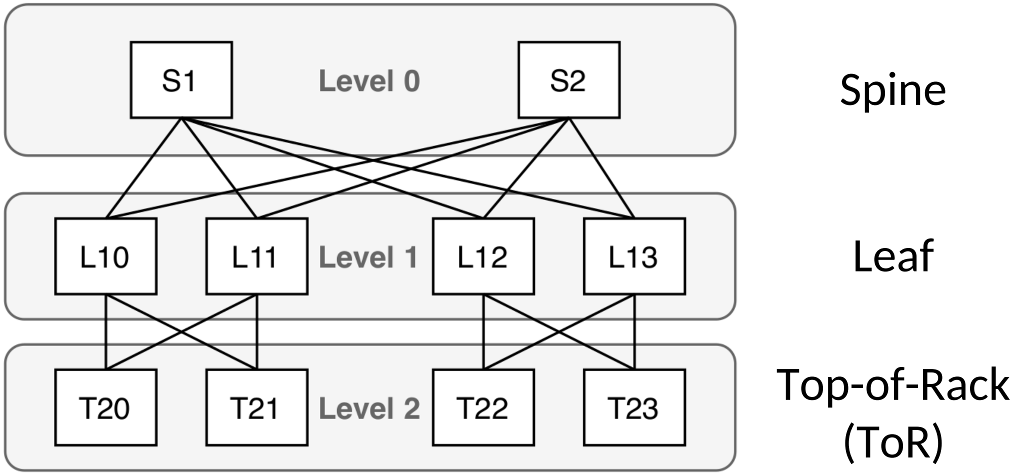
Routing in the Internet

Intra-AS vs. Inter-AS routing

	Intra-AS routing	Inter-AS routing
Link State	OSPF (TCP 89)	
Distance Vector	RIP (UDP 520)	BGP (TCP 179)
Policy	Under same administrative control Not so sensitive	Enable policy-based routing decisions
Scalability	If network becomes too large, split into several ASs	Should manage large number of networks
Performance	More focused on performance	Policy greater concern than quality of routes

Routing in Data Centers (DC)





- Scalability issue
  - OSPF impaired by flooding mechanism
  - BGP preferred for its AS-Path filtering feature
- For instance, prevents “valley” L10 → S1 → L11 → S2 by giving same ASN to S1 and S2
  - S2 rejects paths through S1

Summary

- Overall topics
  - Forwarding and routing
  - Network-Layer services
  - Virtual circuit and datagram networks
- What’s inside a router?
- The Internet Protocol (IP) – IPv6 and IPv4
- Routing principles
  - Link state vs. Distance Vector
  - Hierarchical routing
- Routing in the Internet
  - Intra-domain routing: RIP and OSPF
  - Inter-domain routing: BGP

Forwarding

Routing

Review questions

- Suppose an application generates messages of 40 Bytes of data every 20 ms, and each message gets encapsulated in a TCP segment, and then in an IP datagram. What percentage of each datagram will be overhead, and what percentage will be application data, in both IPv4 and IPv6?
- Compare IPv4 and IPv6 headers
- Compare and contrast link state and distance vector routing algorithms