

Lecture 11:

IPv6

CSE 123: Computer Networks
Alex C. Snoeren





The space crunch...

- Still running out of IP addresses... what to do?
- Two solutions
 - ◆ Network Address Translation – multiple multiple hosts on a single IP address (future class)
 - ◆ Get bigger addresses -> IPv6
- IPv6: 128bit addresses... we won't run out
 - ◆ 64bit routing prefix, 64bit host id

An IPv6 address (in hexadecimal)

2001:0DB8:AC10:FE01:0000:0000:0000:0000

↓ ↓ ↓ ↓ |

2001:0DB8:AC10:FE01:: Zeroes can be omitted

0010000000000001:0000110110111000:1010110000010000:1111111000000001:
0000000000000000:0000000000000000:0000000000000000:0000000000000000

IPv6 Addresses



- Colon-Hex notation
 - ♦ 8 groups of four HEX digits separated by colons, e.g.
 - » FEDC:0000:0000:0065:4321:0000:DEAD:BEEF
 - ♦ Can drop leading zeros:
 - » FEDC:0000:0000:65:4321:0000:DEAD:BEEF
 - ♦ Can even skip **first** sequence of all zeros w/ ::
 - » FEDC::65:4321:0000:DEAD:BEEF
 - ♦ Every IPv4 address is a IPv6 address:
 - » E.g., ::FFFF:222.173.190.239 (prepended w/zeros and FFFF)
- Network names expressed as prefix/length:
 - » FEDC::65:43/64

How many (hex) zeros
between FEDC and 65
in the last address?

- A. None
- B. 4
- C. 6
- D. 8

Address Types



- Each interface has multiple different addresses
 - ◆ Link local, prefixed with FE80::/10 (1111 1110 10)
 - » Used only for communication between adjacent IPv6 devices
 - » Packets are NOT forwarded by routers
 - » Automatically assigned upon boot
 - ◆ Unique local, prefixed with FC00::/7 (1111 110)
 - » Used only internal to one network
 - » Not routable on the global Internet
 - ◆ Global
 - » Like an IPv4 address

IPv6 vs IPv4 header



Ver.	Traffic Class	Flow Label	
Payload Length		Next Header	Hop Limit
Source Address			
Destination Address			

Ver.	Hdr Len	Type of Service	Total Length	
Identification			Flg	Fragment Offset
Time to Live	Protocol		Header Checksum	
Source Address				
Destination Address				
Options...				

- Gray bits are unique to each header
- Changes
 - Eliminate fragmentation-related fields
 - Eliminate header checksum
 - Added flow label
 - Quadruple size of addresses
 - IPv6 header (40 bytes) vs IPv4 (20 bytes)



Extension Headers

- Effectively a linked list of headers
 - ◆ The “next header” field is the pointer
- Two different types
 - ◆ [Destination](#), intended for the IP end point. E.g.,
 - » 44: Fragmentation Header (it's baaack!)
 - » 43: Routing header (dictates how to route the packet)
 - ◆ [Hop-by-hop](#), processed by each node on the path



IPv6 Transition is slow

- Need to support both protocols at the same time
 - ◆ Complicated... if a destination has both a IPv4 and IPv6 address which to use?
- Less need in developed world -> slower adoption
- That said
 - ◆ All major operating systems now support IPv6
 - ◆ All major router vendors
 - ◆ US Mobile carriers (e.g., Tmobile, Verizon, etc)
 - ◆ Offered as option by many US ISPs
- In your lifetime it is likely that IPv6-based addressing will start to dominate



Addressing Summary

- You can't route efficiently on flat address spaces
 - ◆ You'd need a table the size of all hosts on the Internet
 - ◆ You'd need to send updates about that table to everyone
- Network-layer addressing is done hierarchically
 - ◆ Routing prefix + host suffix
 - ◆ Originally, this split was done statically (class-based addressing)
 - ◆ Now it is done dynamically (CIDR)
 - ◆ Requires more complex forwarding table lookup
 - ◆ Allows contiguous chunks of address space to be aggregated (for the purposes of routing) into fewer prefixes

For Next Time



- IN PERSON STARTING MONDAY
- Midterm a WEEK FROM TODAY
- Last discussion before Midterm TUESDAY
- Read 3.3 (ARP) and 9.3 (DNS) for next Lecture

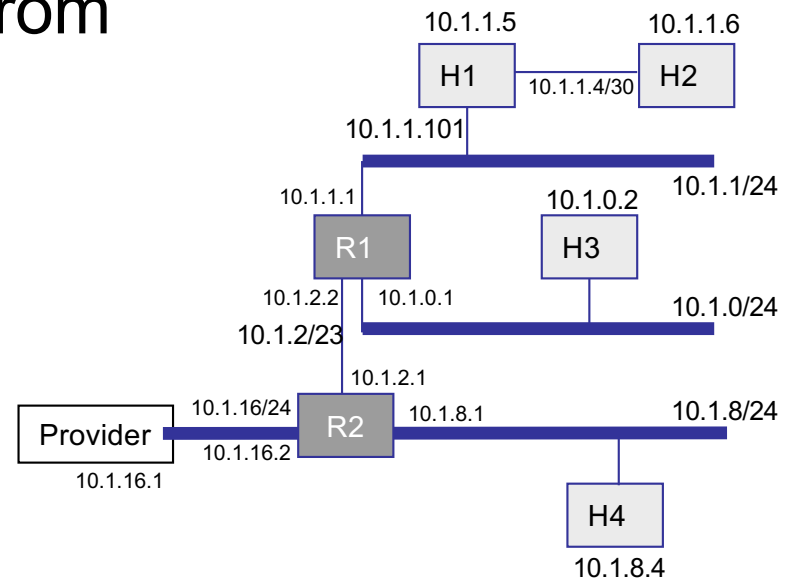


Forwarding example

- Packet to 10.1.1.6 arrives from Provider

What is the correct path?

- A. R2 – R1 – H3
- B. R2 – R1 – H2
- C. R2 – R1 – H1 – H2
- D. No path to destination



Forwarding table (1)



Forwarding table at R2

Destination	Next Hop
10.1.8.0/24	interface1
10.1.2.0/23	interface2
10.1.16.0/24	interface3



Forwarding table (2)



Forwarding table at R2

Destination	Next Hop
127.0.0.1	loopback
10.1.8.0/24	interface1
10.1.2.0/23	interface2
10.1.16.0/24	interface3

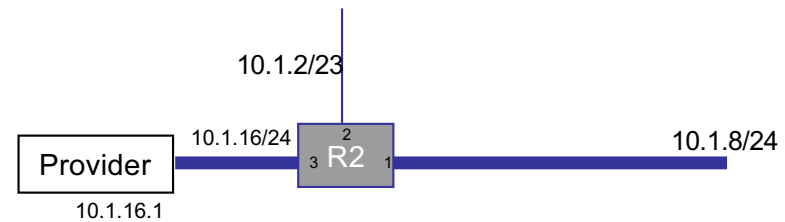


Forwarding table (3)



Forwarding table at R2

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.16.1
10.1.8.0/24	interface1
10.1.2.0/23	interface2
10.1.16.0/24	interface3

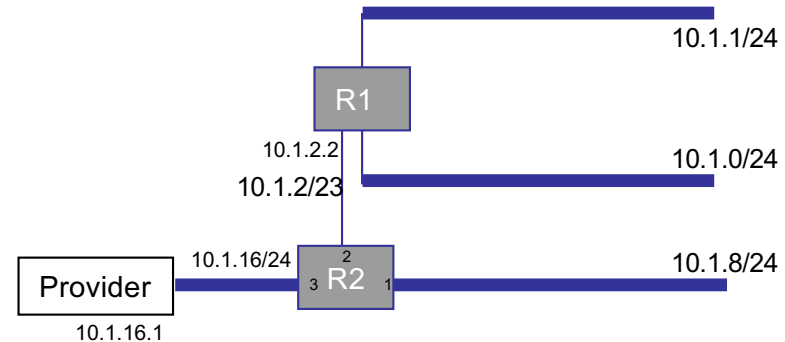


Forwarding table (4)



Forwarding table at R2

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.16.1
10.1.8.0/24	interface1
10.1.2.0/23	interface2
10.1.16.0/24	interface3
10.1.0.0/24	10.1.2.2
10.1.1.0/24	10.1.2.2

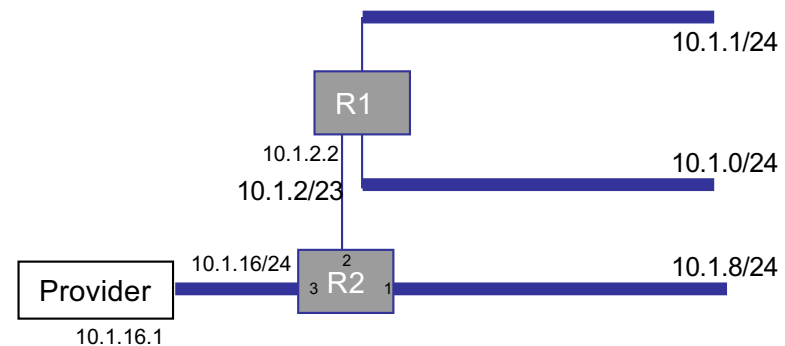


Forwarding table (5)



Forwarding table at R2

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.16.1
10.1.8.0/24	interface1
10.1.2.0/23	interface2
10.1.16.0/24	interface3
10.1.0.0/23	10.1.2.2





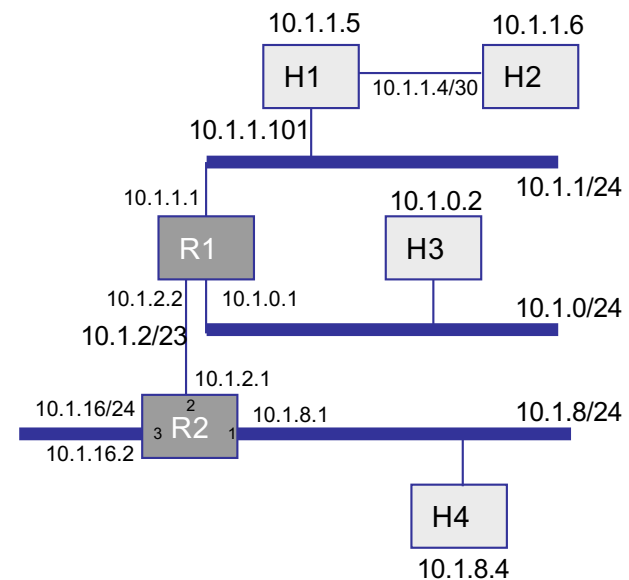
Forwarding example (2)

- Packet to 10.1.1.6

Forwarding table at R2

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.16.1
10.1.8.0/24	interface1
10.1.2.0/23	interface2
10.1.16.0/24	interface3
10.1.0.0/23	10.1.2.2

- Matches 10.1.0.0/23





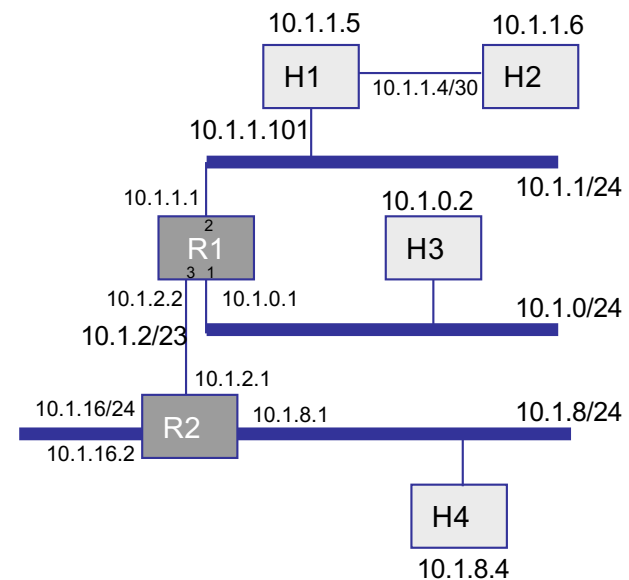
Forwarding example (3)

- Packet to 10.1.1.6

Forwarding table at R1

	Destination	Next Hop
	127.0.0.1	loopback
A	Default or 0/0	10.1.2.1
	10.1.0.0/24	interface1
B	10.1.1.0/24	interface2
C	10.1.2.0/23	interface3
D	10.1.1.4/30	10.1.1.101

What is the next hop?





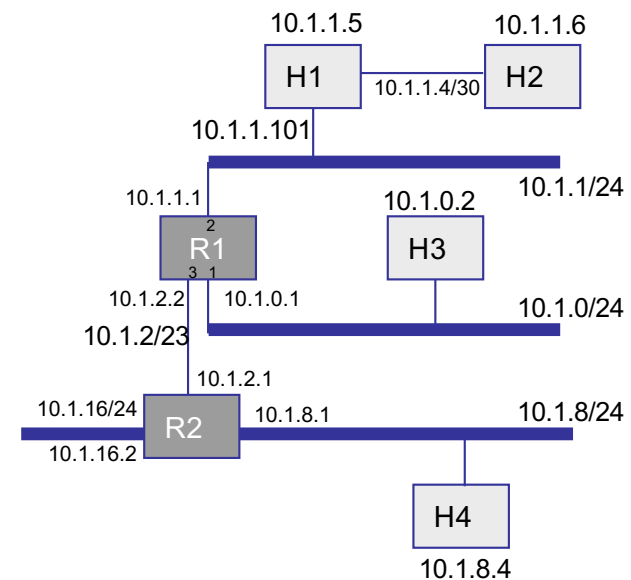
Forwarding example (3)

- Packet to 10.1.1.6

Forwarding table at R1

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.2.1
10.1.0.0/24	interface1
10.1.1.0/24	interface2
10.1.2.0/23	interface3
10.1.1.4/30	10.1.1.101

- Matches 10.1.1.4/30
 - Longest matching prefix!





Forwarding example (4)

- Packet to 10.1.1.6
- Direct route
 - Longest prefix match

Forwarding table at H1

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.1.1
10.1.1.0/24	interface1
10.1.1.4/30	interface2

