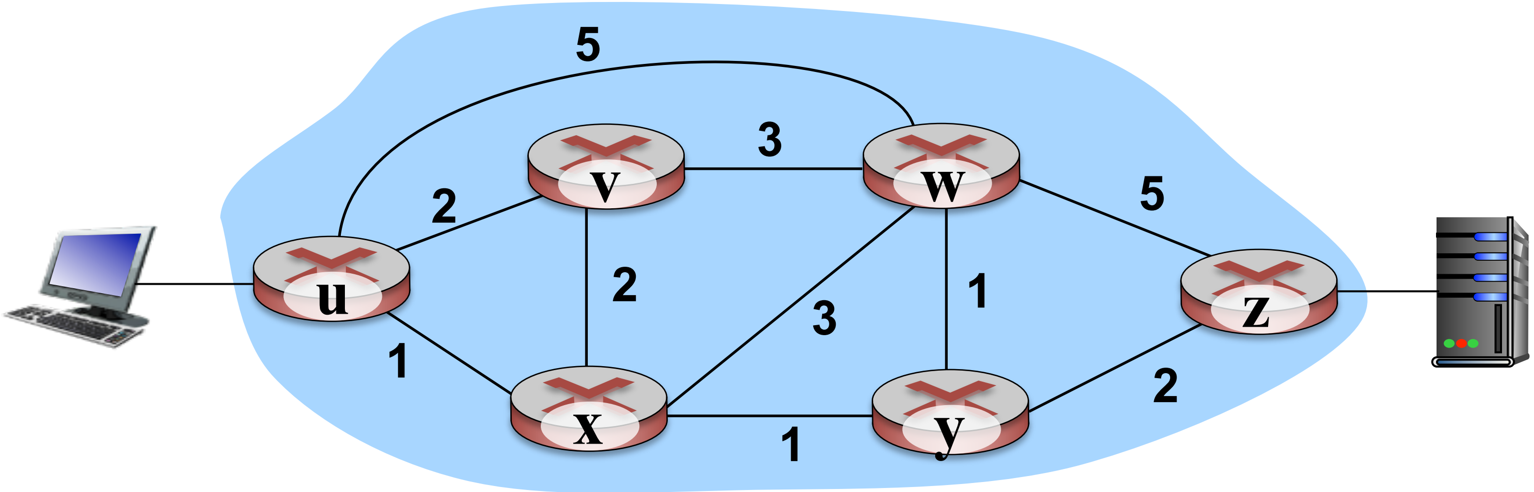
Software defined networking (SDN)

- Internet network layer: historically has been implemented via distributed, per-router approach
  - *monolithic* router contains switching hardware, runs proprietary implementation of Internet standard protocols (IP, RIP, IS-IS, OSPF, BGP) in proprietary router OS (e.g., Cisco IOS)
  - different "middleboxes" for different network layer functions: firewalls, load balancers, NAT boxes, ..
- ~2005: renewed interest in rethinking network control plane

Traffic engineering: difficult traditional routing

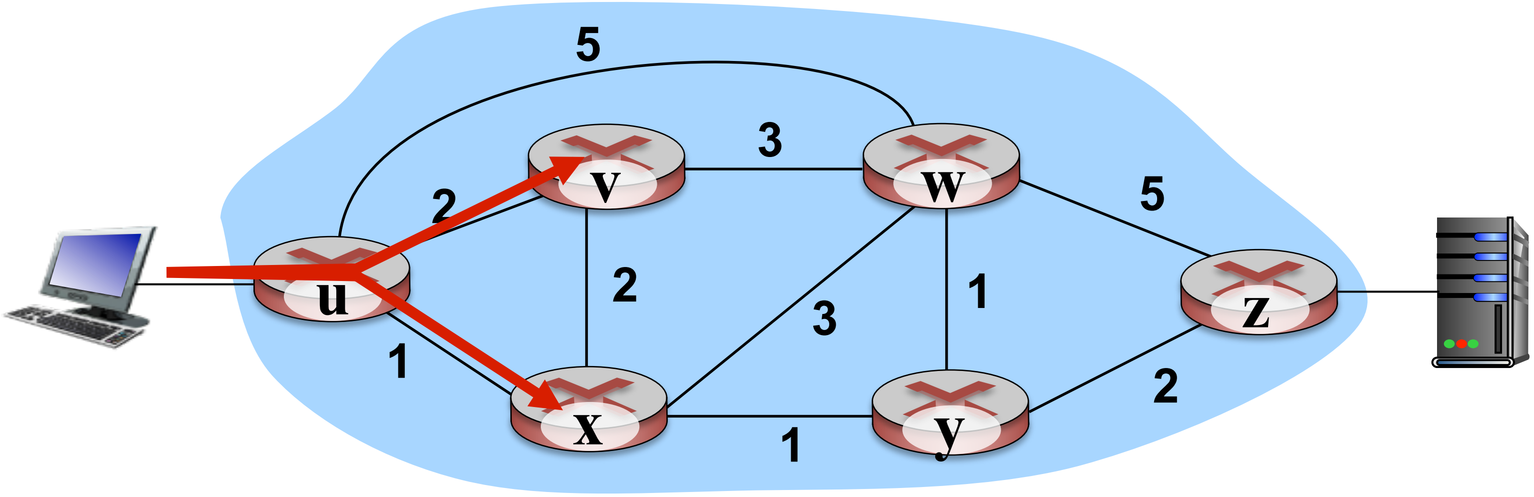


Q: **what if network operator wants u-to-z traffic to flow along *uvwz*, x-to-z traffic to flow *xwyz*?**

A: **need to define link weights so traffic routing algorithm computes routes accordingly (or need a new routing algorithm)!**

Link weights are only control "knobs": wrong!

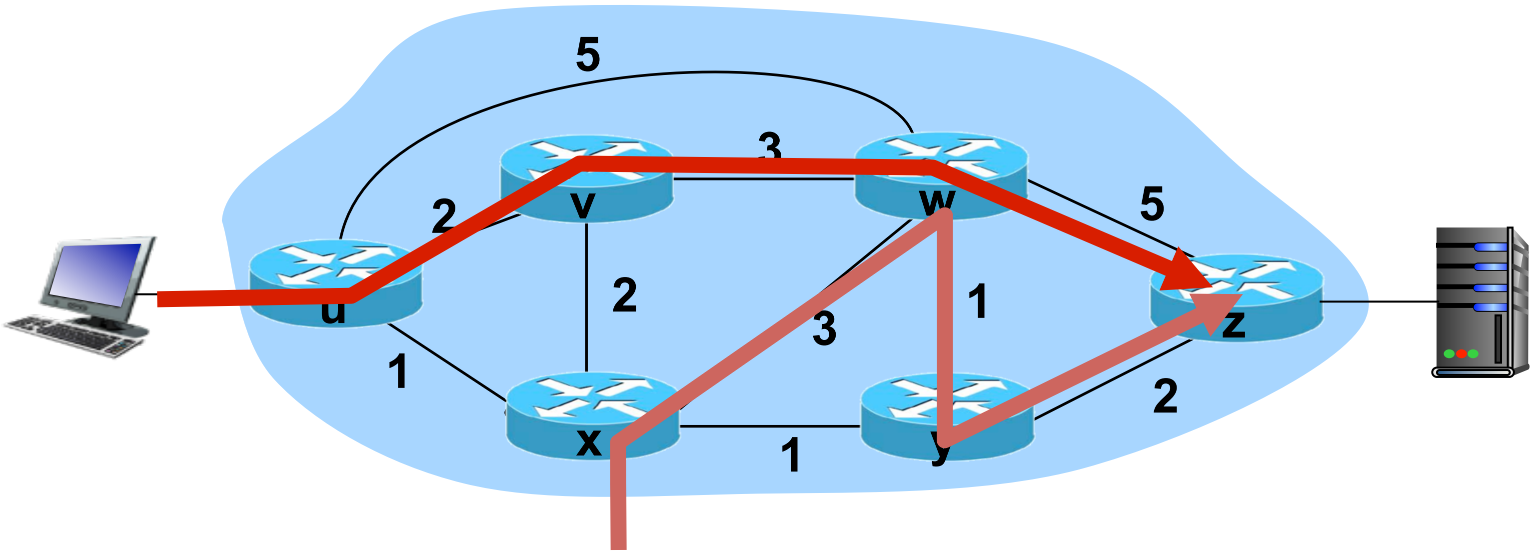
Traffic engineering: difficult



Q: **what if network operator wants to split u-to-z traffic along uvwz *and* uxyz (load balancing)?**

A: **can't do it (or need a new routing algorithm)**

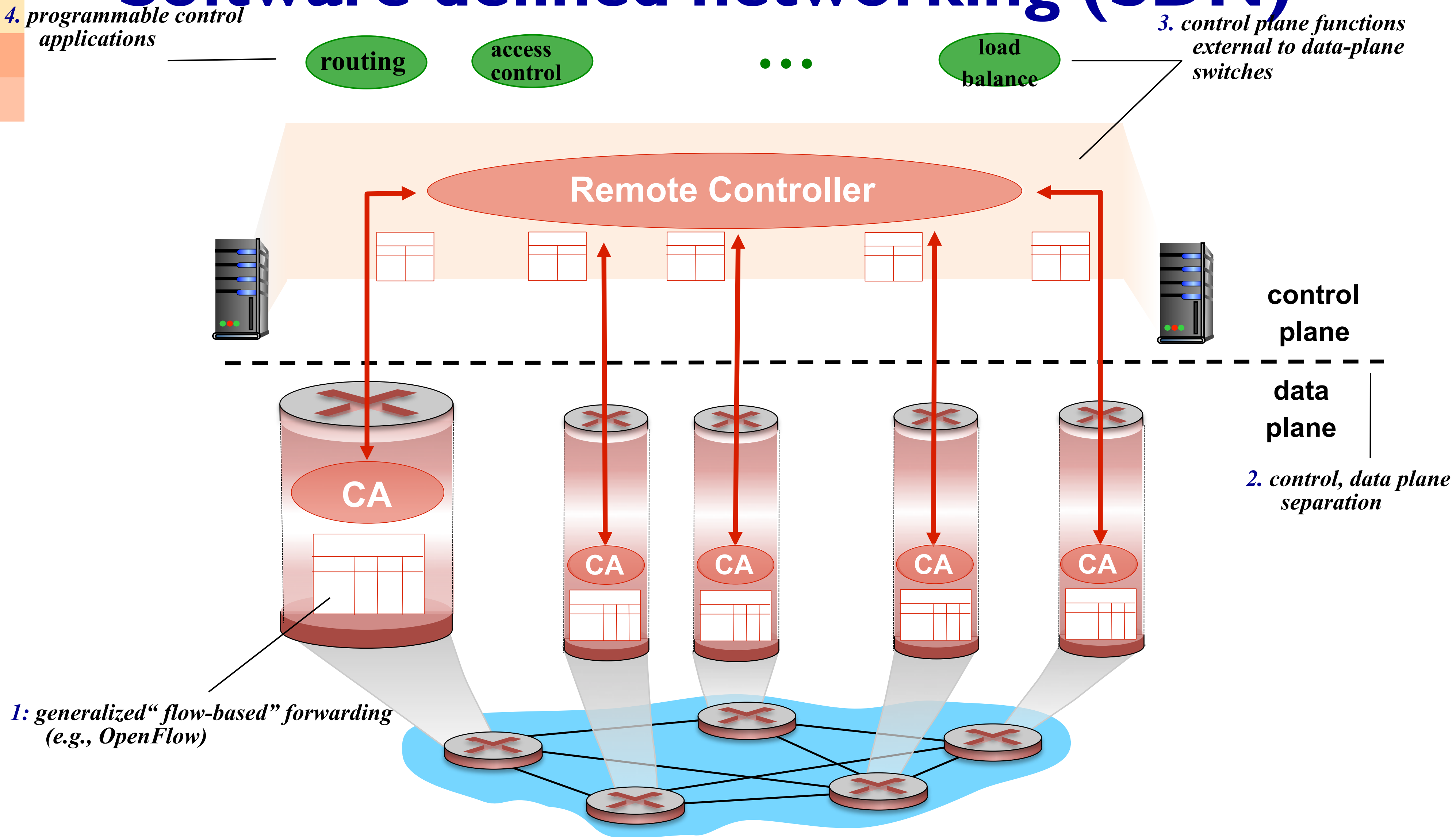
Traffic engineering: difficult



Q: **what if w wants to route blue and red traffic differently?**

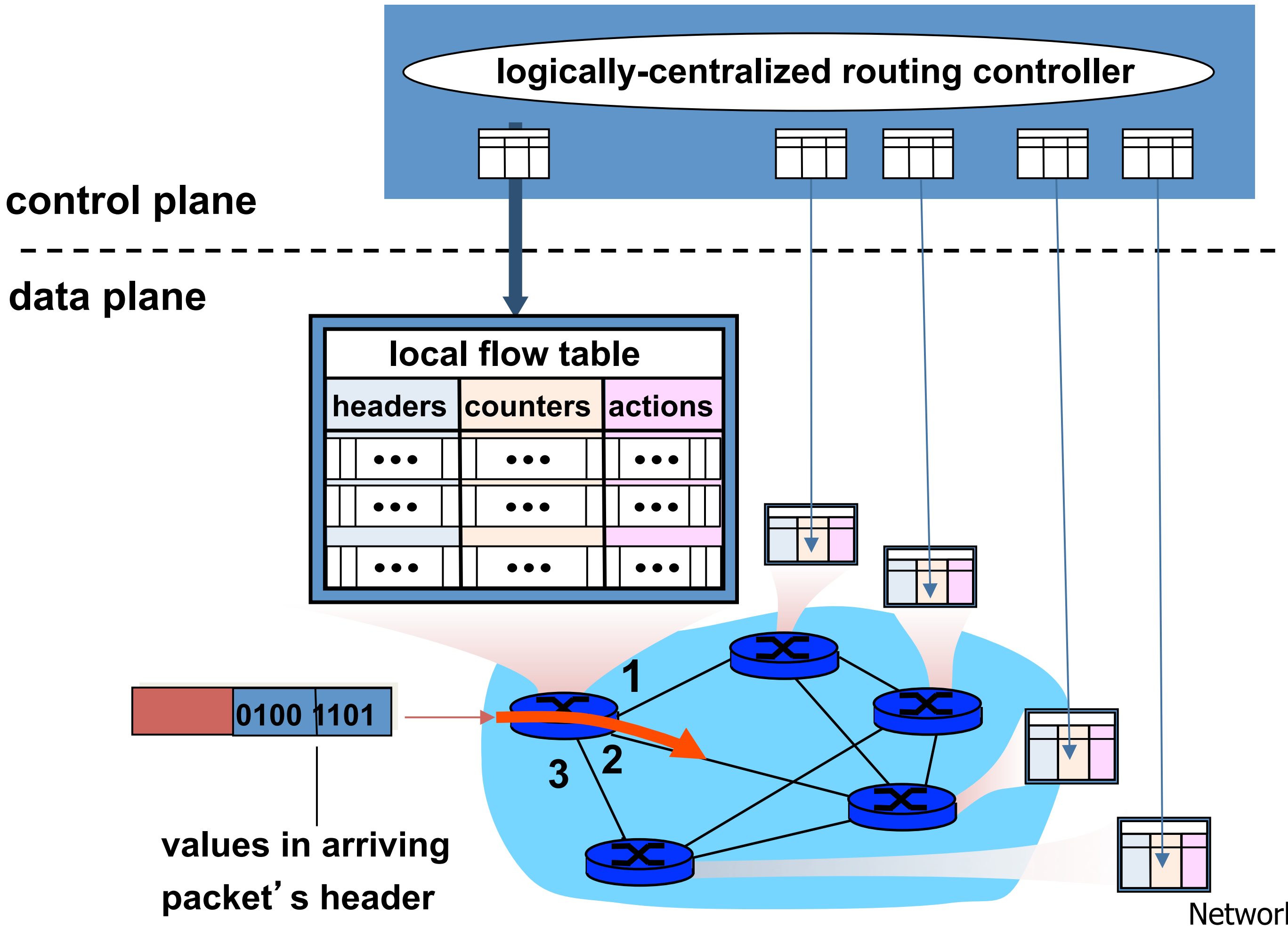
A: **can't do it (with destination based forwarding, and LS, DV routing)**

Software defined networking (SDN)



Generalized Forwarding and SDN

Each router contains a *flow table* that is computed and distributed by a *logically centralized* routing controller



OpenFlow: Flow Table Entries

Rule	Action	Stats

Packet + byte counters

1. Forward packet to port(s)
2. Encapsulate and forward to controller
3. Drop packet
4. Send to normal processing pipeline
5. Modify Fields

Switch Port	VLAN ID	MAC src	MAC dst	Eth type	IP Src	IP Dst	IP Prot	TCP sport	TCP dport

Link layer | Network layer | Transport layer

Examples

Destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	51.6.0.8	*	*	*	port6

IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	*	*	*	*	22	drop

do not forward (block) all datagrams destined to TCP port 22

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	128.119.1.1	*	*	*	*	drop

do not forward (block) all datagrams sent by host 128.119.1.1

IPv6

- IP was designed in the 1970's to support researchers and the military. The global use today wasn't predicted.
- Early design decisions led to predictions in the early 1990's of the "death of the Internet" due to
  - Shortage of Internet protocol (IP) addresses
  - Routing table explosion
  - Shortage of network numbers
- New capabilities needed:
  - Improved security (or even *some* security!)
  - Support for QoS (real-time services)
  - Better multicasting support
  - Mobile computing
- Two paths could be taken
  - Retrofit larger addresses and functionality onto IPv4
  - Develop a new version of IP

- **Early proposals**
  - **CLNP (Connectionless Network Protocol)**
  - **SIP (Simple IP)**
  - **Pip(Paul's Internet protocol)**
  - **SIPP (SIP plus Pip) became IPv6 or IPng**

Expanding the address space

- By 1996, all of the class A networks, 62% of class B and 37% of class C networks were allocated. Predicted to run out of addresses in 2008.
- The temporary solution is Classless InterDomain Routing (CIDR)
- With the push towards IP enabled devices this is likely to be only a temporary patch. IPv6 aims for a more permanent solution.
- How big is "big enough" for the address space?
  - Current 32 bit space allows *billions* of addresses.
  - Inefficiencies of allocation are almost inevitable.
  - May want to identify experimental networks in a different address spaces.
  - Can we predict future use? Are we setting ourselves up to need IPv7?
- Should the address be variable length?

IPv6 addressing

- 128 bits was a compromise
  - Fixed addresses are easier to manage and program.
- There is nearly 1 IP address for every molecule on the earth's surface.
  - Even with ***in***efficient allocation there should be at least 1000 addresses per square metre.
- 128 bit addresses are written in hex: x:x:x:x:x:x:x:x
  - Each x is 16-bits = 4 hex digits
  - Leading zeros are not required
  - Sequence of zero fields given by "::"
- Examples:

1080:0:0:0:8:800:200C:417A	=	1080::8:800:200C:417A
FF01:0:0:0:0:0:0:43	=	FF01::43
0:0:0:0:0:0:0:1	=	::1

IPv6 Routing

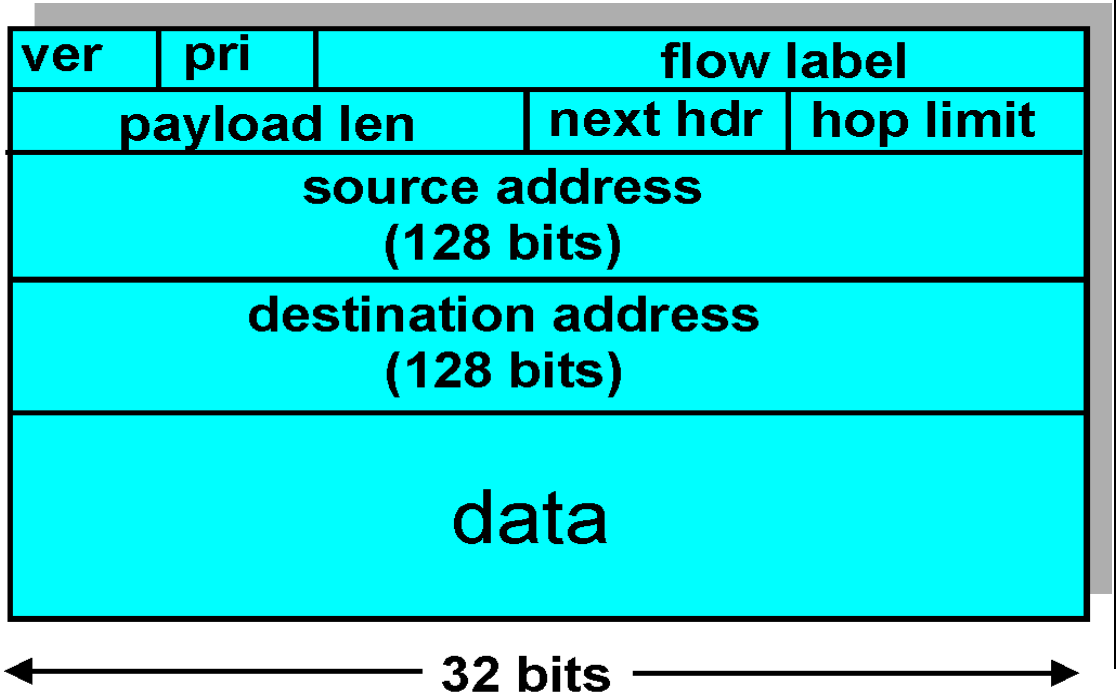
- Each interface has an address (but possibly more than one):
  - Unicast – identify single interface
  - Anycast (or "cluster")– identify one out of number of interfaces ie. any one will do (eg. nearest host)
  - Multicast – identify set of interfaces, all of which are to receive message
- Intention in IPv6 addresses is to support ***hierarchical routing***
  → prefixes will **identify** registries, providers etc.
- Example: hierarchical organisation:

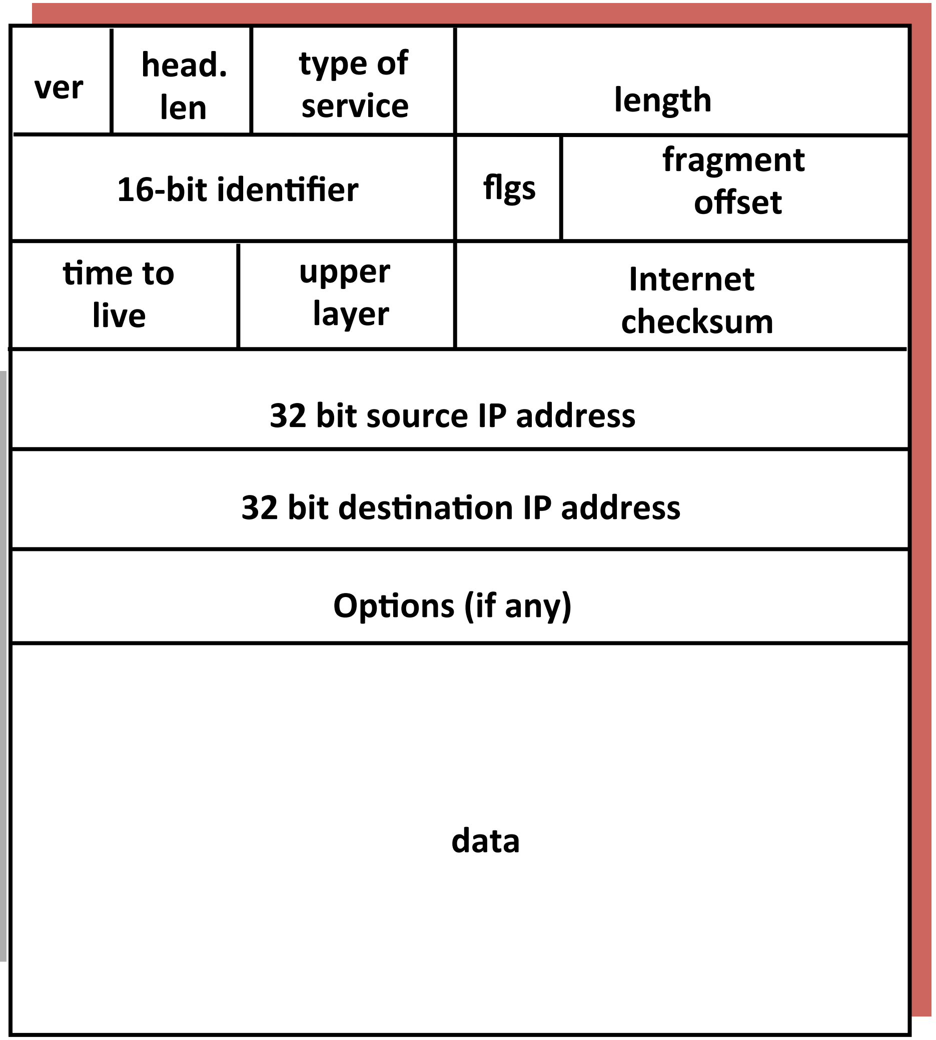
s	n	m	128-s-n-m
subscriber prefix	area id	subnet id	interface id

- **This will help routing tables**
- Other possibilities:
  - global provider-based unicast address
  - geographic-based unicast address

IPv6 Header

- IPv6 8 fields in base header vs 13 fields in IPv4
  - Faster processing
  - Simpler management
  - More flexibility





IPv4

Extension Headers

- This base header is followed by a number of optional ***extension headers***
  - Still allows flexibility
- Each header specifies code of next header/data component
- Extension headers commonly specify:
  - type of header and length
  - response if the router can't process the header — ignore the header, skip the packet, skip the packet and report an error

Extension Header	**Description**
Hop-by-hop Options	Misc. information for routers
Routing	Full, or partial, route to follow
Fragmentation	Management of datagram fragments
Authorisation	Verification of Sender's Identity
Encrypted Security Payload	Information about encrypted Contents
Destination Options	Additional Info for destination

Fragmentation

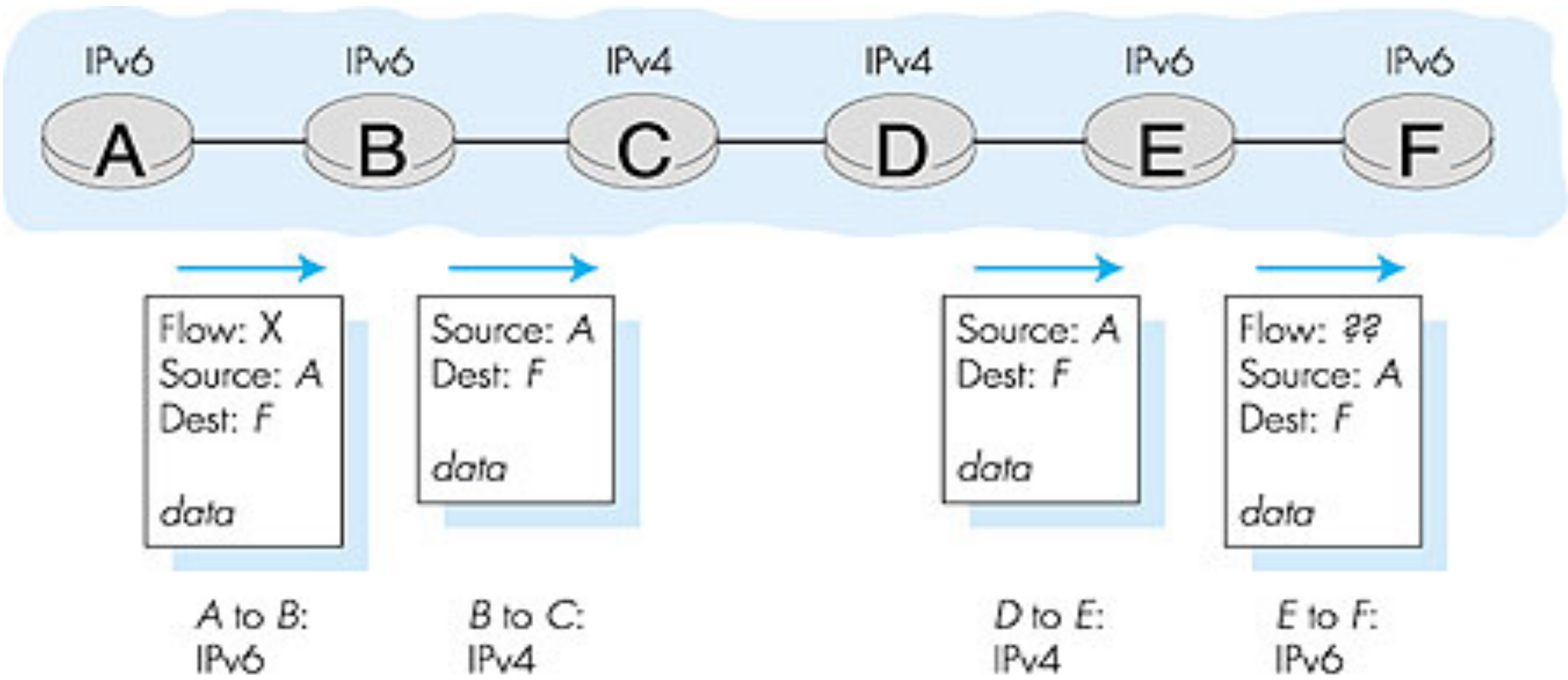
- Fragmentation information is no longer in every header, but only in special extension headers
- Fragmentation is no longer performed at intermediate routers
  - The source host should choose datagram size so fragmentation is not necessary
  - Source host needs to run "path MTU discovery”

  e.g. send sequence of datagram sizes to target until they don't arrive

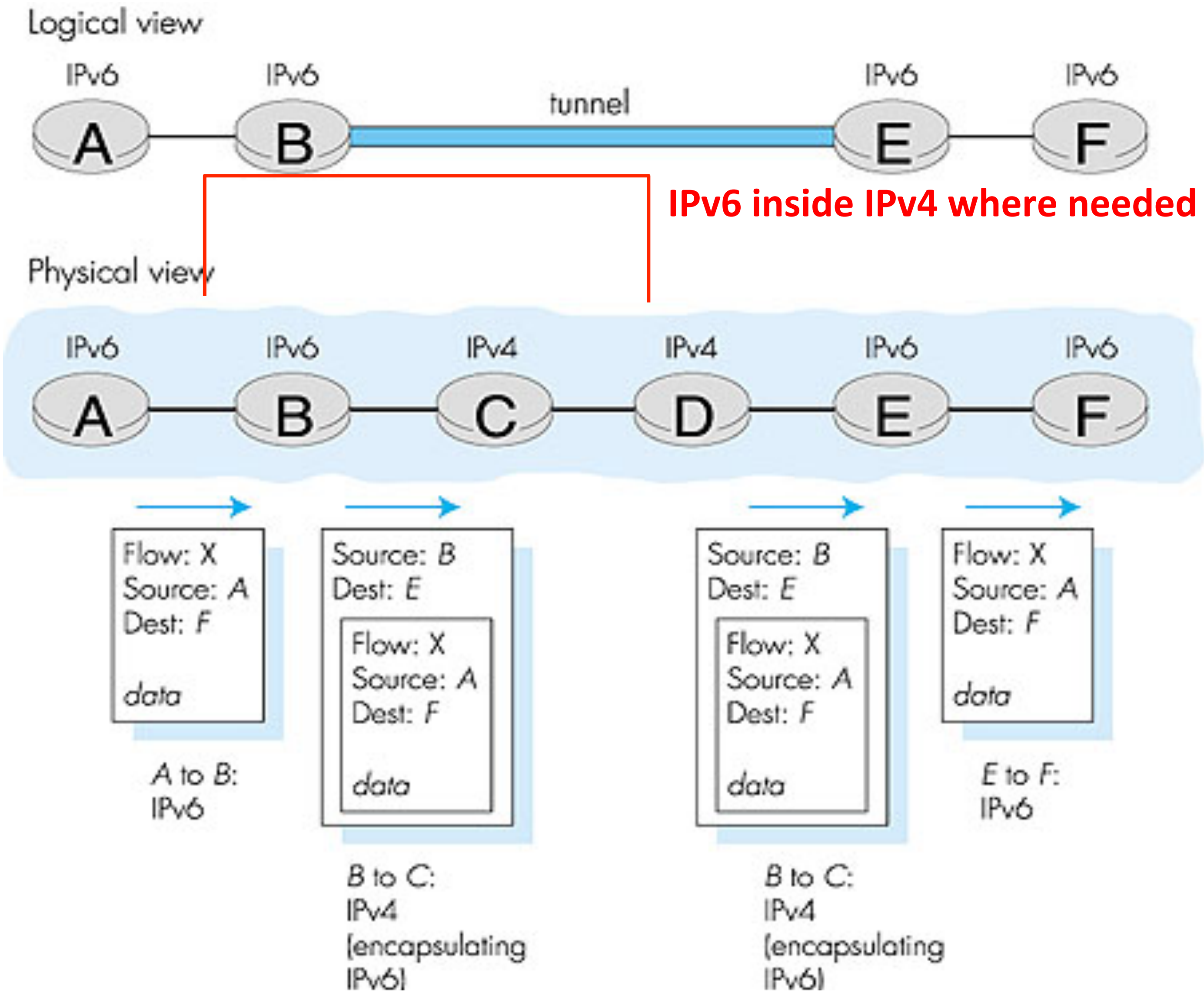
Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneously
  - no "flag days"
  - How will the network operate with mixed IPv4 and IPv6 routers?
- Two proposed approaches:
  - *Dual Stack*: some routers with dual stack (v6, v4) can "translate" between formats
  - *Tunneling:* IPv6 carried as payload in IPv4 datagram among IPv4 routers
- Changes in networking technologies *always* proceed by evolution, rather than revolution

Dual Stack Approach



Tunneling


Logical view
IPv6
IPv6
IPv6
IPv6
tunnel
A
B
E
F
IPv6 inside IPv4 where needed
Physical view
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
A
B
C
D
E
F
Flow: X
Source: A
Dest: F
data
A to B:
IPv6
Source: B
Dest: E
Flow: X
Source: A
Dest: F
data
B to C:
IPv4
(encapsulating
IPv6)
Source: B
Dest: E
Flow: X
Source: A
Dest: F
data
B to C:
IPv4
(encapsulating
IPv6)
Flow: X
Source: A
Dest: F
data
E to F:
IPv6

Tunneling

- Tunneling is a generically useful technique in networking…
  - Note that IPv6 payloads could be anything!
  - We could tunnel arbitrary protocols
- This is how experimentation is done –
  - New protocols are tunneled across existing infrastructure
  - Can spread new protocols across the network this way
  - Can link in "new" devices or "old" devices this way
  - Can implement secure networks this way (see later…)

IPv6 Summary

- What has changed
  - Simpler, fixed length header
    - No fragmentation in routers
    - Options in extension headers
    - No checksum
  - 128 bit (IPv4 32 bit) addresses, *with hierarchy*
  - Additional support for
    - Multicast and anycast routing
    - Security
    - Mobile hosts and autoconfiguration
    - Real Time applications