Computer Networking

Sixth edition

Chapter 5

The Network Layer

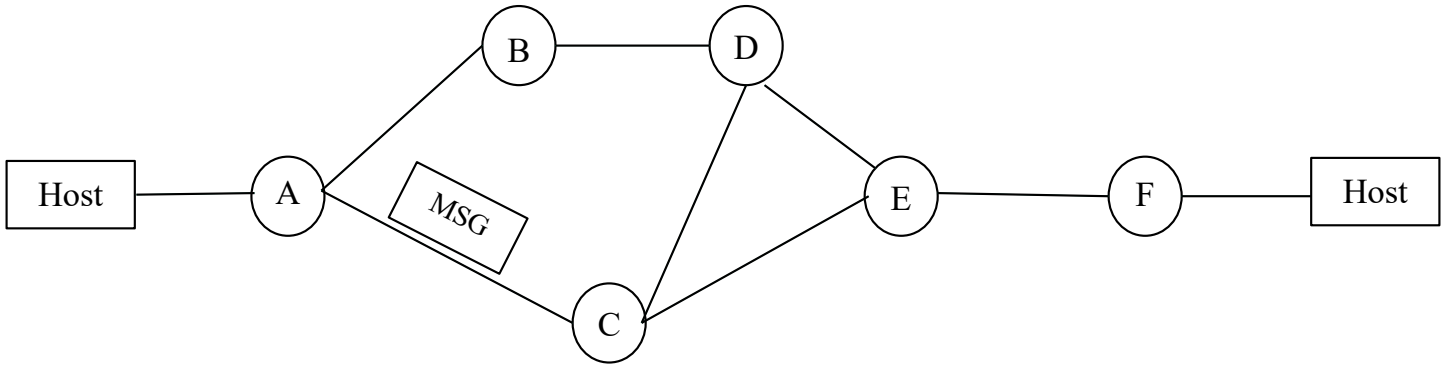
The Traditional View

Network Layer Design Issues

- Store and Forward Message Switching
- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets

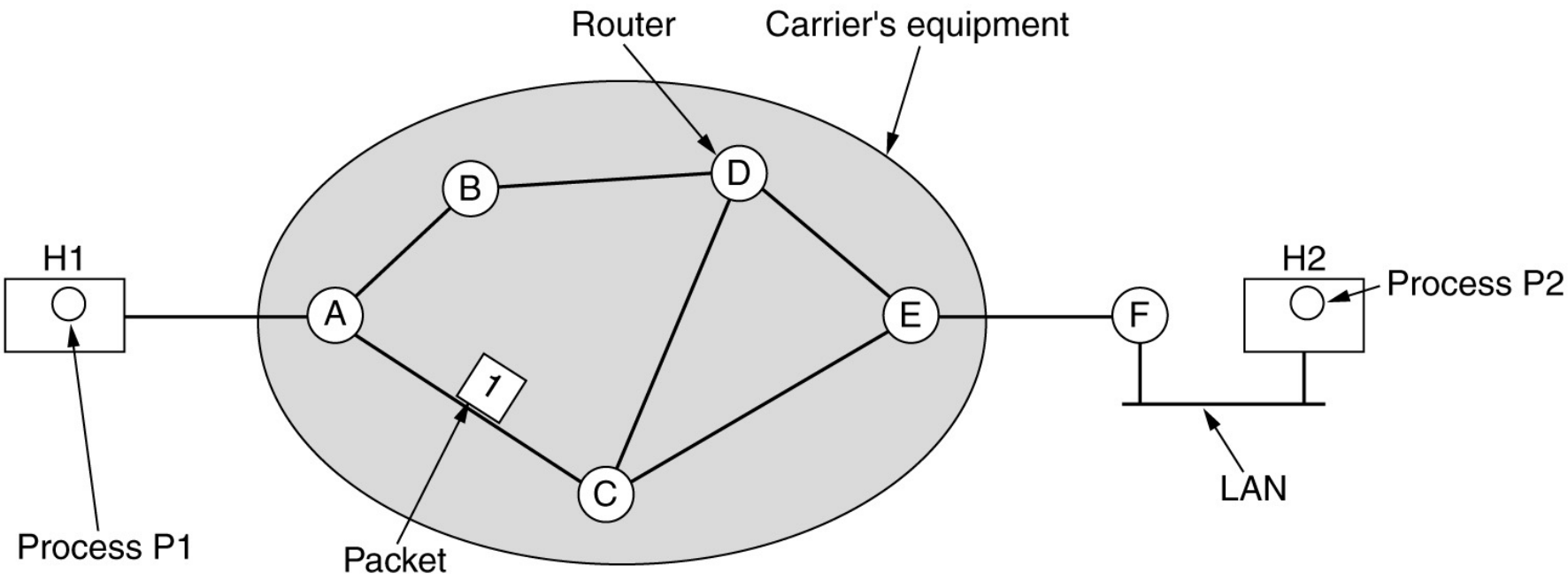
(They are Functions not Services.)

Store-and-Forward Message Switching



First there was Message Switching With Message Switching,
the Whole Message Was Moved from Node to Node
This is analogous in Operating Systems to
First-Come-First-Serve Batch Scheduling

Store-and-Forward Packet Switching



The environment of the network layer protocols.

In Operating Systems, this analogous to multiprogramming

Using Round-Robin Scheduling

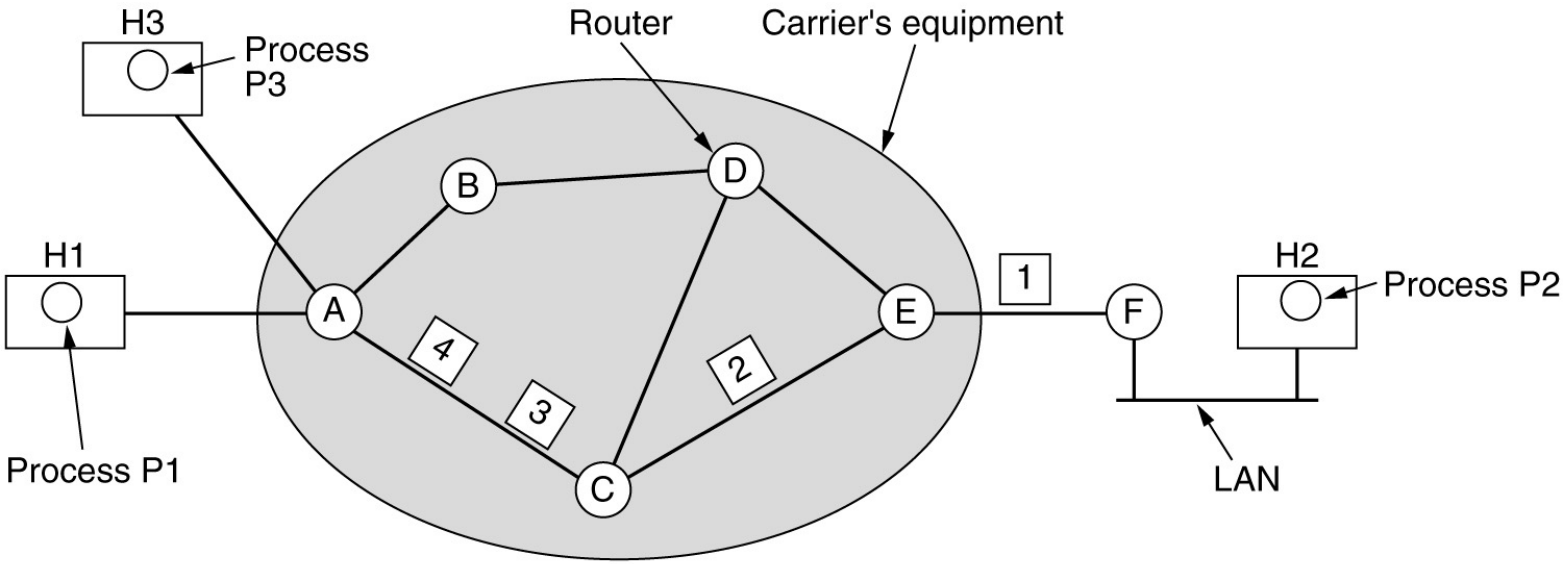
Services Provided to the Transport Layer

- Services independent of router technology.
- Transport layer shielded from number, type, topology of routers.
- Network addresses available to transport layer use uniform numbering plan
  - even across LANs and WANs

Not a good idea. Root of a lot of problems.

(?) Odd comment. Does T believe in Layers?

Implementation of Connection-Oriented



A's table

H1	1	C	1
H3	1	C	2

In Out

C's table

A	1	E	1
A	2	E	2

E's table

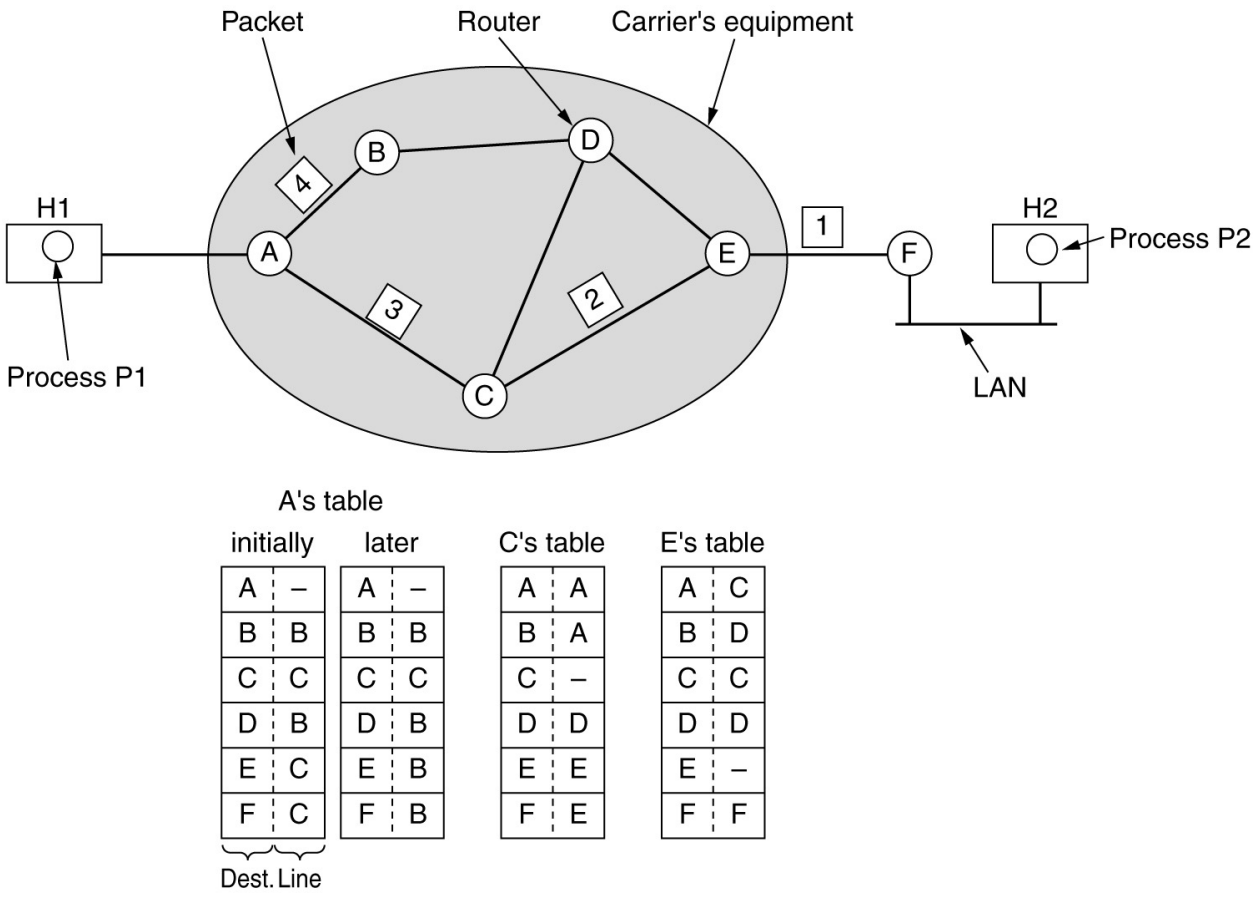
C	1	F	1
C	2	F	2

Routing within a virtual-circuit subnet.

Centralized routing, a router only knows how to route what goes through it.

Analogous to round-robin scheduling with contiguous memory allocation

Implementation of Connectionless



Dest.	Line (initially)
A	–
B	B
C	C
D	B
E	C
F	C

Dest.	Line (later)
A	–
B	B
C	C
D	B
E	B
F	B

Dest.	Line (C's table)
A	A
B	A
C	–
D	D
E	E
F	E

Dest.	Line (E's table)
A	C
B	D
C	C
D	D
E	–
F	F

Routing within a datagram subnet.

Decentralized routing: Every router can route any packet.

Analogous to dynamic resource allocation

Comparison of Virtual-Circuit and Datagram Subnets

Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

The Origins of Datagrams (1972)

- CYCLADES was building a network to do research on networks.
- First, they needed to work out what the minimal assumptions were.
  - Then consider what else was needed,
    - not unlike what we did with the IPC Model
- CYCLADES then adopted* Datagrams as a minimal PDU and with
  - End-to-end transport they had a minimal network.
  - Much to their surprise, nothing else was needed.
    - This was far simpler, more elegant than anything else being considered.
    - But assumed later there would need to be.
- But it was a long time before more was needed and by then
  - Datagrams had become a religion in the Internet.
- Almost immediately, there was a battle with the PTTs that had no use for connectionless. (ironically, especially with the French PTT)
- Datagrams are at the core of dynamic (stochastic) resource allocation.
  - Which has major advantages beyond merely being 'best effort.'

*Originally proposed by Donald Davies in 1966, but didn't have a chance to pursue it.

The Connection/Connectionless War

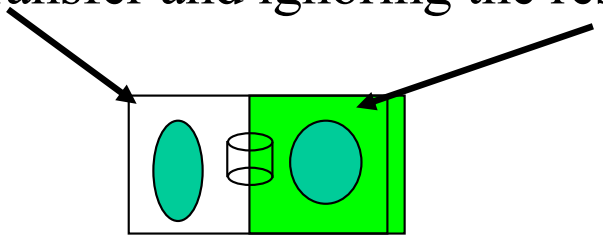
- The technical side of what was really an economic war.
  - The Layered Model invalidated both the PTT and IBM business models.
  - Connectionless removed the security blanket of determinism.
  - The war created a bunker mentality that made understanding hard.
    - All or nothing.
- And the Internet turned it into religion.
  - Dave Clark recognized that it is the minimal PDU and a building block
    - But never goes anywhere with it.
  - Classic sign they had heard about the idea but didn't grok it.
- Ultimately, the Internet only saw datagrams *as an end*.
- While CYCLADES knew they were *just the beginning*.
- For years, we saw it as the extremes of the amount of shared state.
  - Connections had lots of shared state; connectionless very little.
  - But there really wasn't anything in between.

Connections and Connectionless

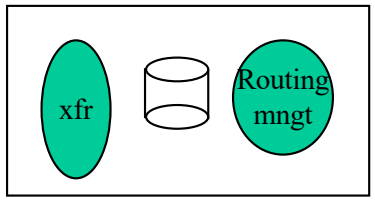
- Not a matter of religion
- The Choice of which one to use is a trade-off between static (reserved) and dynamic resource allocation, not reliability or error control as is often cited,
  - The more deterministic (less variance), the more connection-like;
  - The less deterministic (more variance), the more connectionless.
    - The use of CO/CL does not have to be end-to-end or even span the entire scope of a layer.
- As one moves down in the layers and in toward the backbone,
  - Traffic density increases and hence become more deterministic
    - traffic becomes more connection-like.
    - The trade-off advantages shift from storing state in the PDU to storing it in memory
- As one moves out from the backbone toward the periphery,
  - Traffic density decreases and becomes more stochastic
    - Traffic becomes more connectionless
    - The trade-off advantages shift from storing state in memory to storing it in the PDU
- As one moves in toward the backbone flows between intermediate points are more long-lived
  - Again, implying more connection-like
- But CL does have better resiliency and survivability properties.

Finding a Synthesis is Hard

- Let's look at this very carefully
- What makes connection-oriented so brittle to failure?
  - When a failure occurs, no one knows what to do.
  - Have to go back to the edge to find out how to recover.
- What makes connectionless so resilient to failure?
  - Everyone knows how to route everything!
- Just a minute! That means!
  - Yes, connectionless isn't minimal state, but maximal state.
    - The dumb network ain't so dumb.
  - Where did we go wrong?
- We were focusing on the data transfer and ignoring the rest:



We Need to Look at the Whole Picture



(A bit like doing a conservation of energy problem and getting the boundaries on the system wrong.)

- The amount of state is about the same, although where it is is different and the amount of replication is different.
- We have been distributing connectivity information to every Node in a layer, but
- We have insisted on distributing resource allocation information only on a 'need to know' basis, i.e., connection-like.
  - Even if we aren't too sure who needs to know.
- Now we have to work out how to do resource allocation more like how we do routing. (Left as an exercise.) ;-)

So, What Do We Know About CO/CL?

- It is a function of the layer. Should not be visible to applications.
  - Applications request a level of service, the layer determines how to provide it.
- Connectionless is characterized by the maximal dissemination of state information and dynamic resource allocation.
  - Remember what we learned about pooled vs static allocation?
- Connection-oriented mechanisms attempt to limit dissemination of state information and tends toward static resource allocation.
- Applications request the allocation of comm resources.
  - The layer determines what mechanisms and policies to use.
  - Tends toward CO when traffic density is high and deterministic.
  - CL when traffic density is low and stochastic.
  - Similarly, PDU size should increase as we move to the backbone.
    - Want to Forward More Stuff Less Often, Rather than Less Stuff More Often

So We See That

The Dumb Network ain't so dumb

Connectionless is maximal shared state, not minimal.

Misconception about Connection/Connectionless

- 'Connections' in Error and Flow Control Protocols are a Qualitatively Different Phenomena.
- The 'connection-ness' is the feedback mechanisms, not the static resource allocation.
  - Example of *reductio ad absurdum* going too far.
- There can be end-to-end feedback over connectionless.
  - Which there is.
- The Connection/Connectionless Debate is Purely Virtual-Circuit vs Datagram
  - There is really no debate. As traffic density increases it becomes more connection-like.
  - But it is important to retain the dynamic resource allocation.

Internetworking

- Internetworks: an overview
- How networks differ
- Connecting heterogeneous networks
- Connecting endpoints across heterogeneous networks
- Tunneling understanding the difference
- Internetwork routing: routing across multiple networks
- Supporting different packet sizes: packet fragmentation

How Networks Differ

Item	Some Possibilities
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	Present or absent; many different kinds
Error handling	Reliable, ordered, and unordered delivery
Flow control	Sliding window, rate control, other, or none
Congestion control	Leaky bucket, token bucket, RED, choke packets, etc.
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, by packet, by byte, or not at all

Some of the many ways networks can differ.

To some degree, all of this is irrelevant

How Networks Are Similar

They All Do IPC over a
Given Range of Capacity, QoS, and Scope.

(The Similarities are Much More Interesting and
Important Than the Differences.)

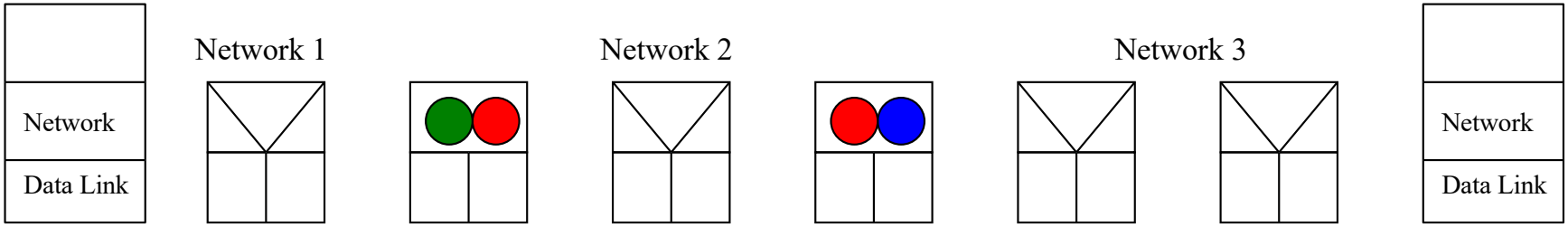
In the mid-70s, It was recognized that

There are 2 Ways To Do Internetworking

- *The DataComm Approach*: Do Protocol Conversion at the Gateways between Networks
  - Potential n^2 translation problem
  - This is reasonable, if the networks are similar.
  - Gets ugly if they aren't. Not future proof.
- *The Layered Approach*: Provide a common layer that is technology independent, and relays between networks.
  - Makes it an n x 1 problem, no translations
  - Each network provides a level of service to the "internetworking layer"
  - Implies that there is a minimal level of service required by the common layer.

ITU Chose the DataComm Approach

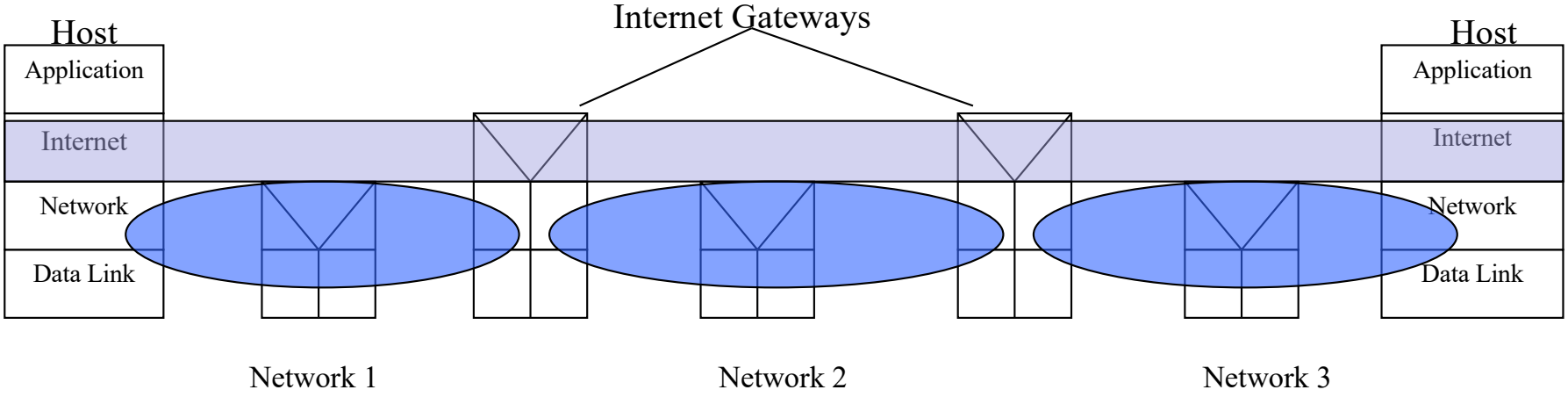
- Just hook them together, Convert one into the other.
  - Basically, the same as how the PSTN worked



- This has a potential n^2 problem. Converting everything to everything
- Quite reasonably, ITU chose this approach because
  - Retains the traditional datacom beads-on-a-string model; what they were familiar with.
  - They were interconnecting relatively similar X.25 networks, so protocol translation was feasible.
  - With no Transport Protocol that would relegate them to a commodity business, it preserved their business model.
    - Note that "layers" here are just modules. Basically beads-on-a-string with stripes.

The INWG Researchers chose the Layered Approach

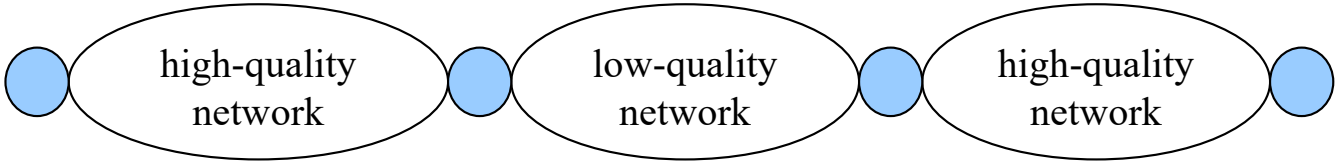
- Build a common layer over the different networks
  - Avoid the n^2 translation problem
- The upper layer requires at least a minimal service from the layers below.
  - If not, then a protocol is required to enhance the network's service.



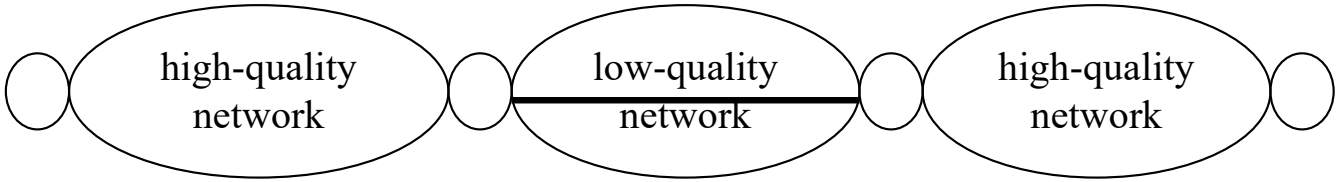
- The researchers in INWG assumed there would be a wide variety of potentially very different networks raising the specter of messy complex translations.
- Here Layers are a locus of distributed shared state, the elements of the layer are cooperating to do resource allocation.
- This is a distributed computing model.

Rediscovered a Decade Later When Analyzing the Network Layer for OSI

- The well-known 7-layer model was adopted at the first meeting in March 1978 and frozen. After that, they had to work within that.
- They knew they would have to accommodate different networks of different quality and different technology.
- One of their concerns in the Network Layer deliberations was interworking over a less capable network:



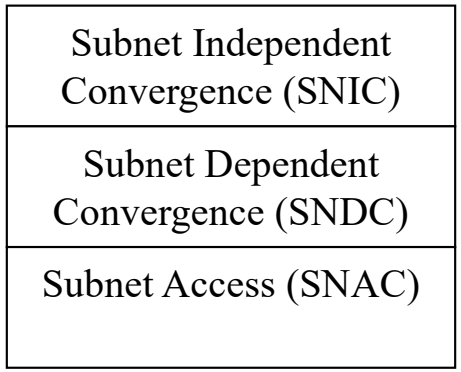
- Would need to enhance the less capable network with an additional protocol



This Implied . . .

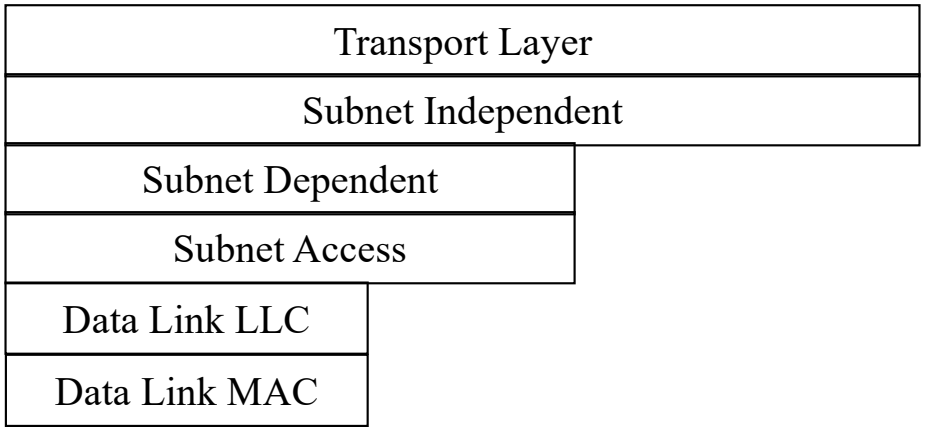
Sub-Dividing the Network Layer

- This concern and the recognition that there would be different networks interworking lead the computer companies to divide the Network Layer into three sub-layers, which might be optional depending on configuration:



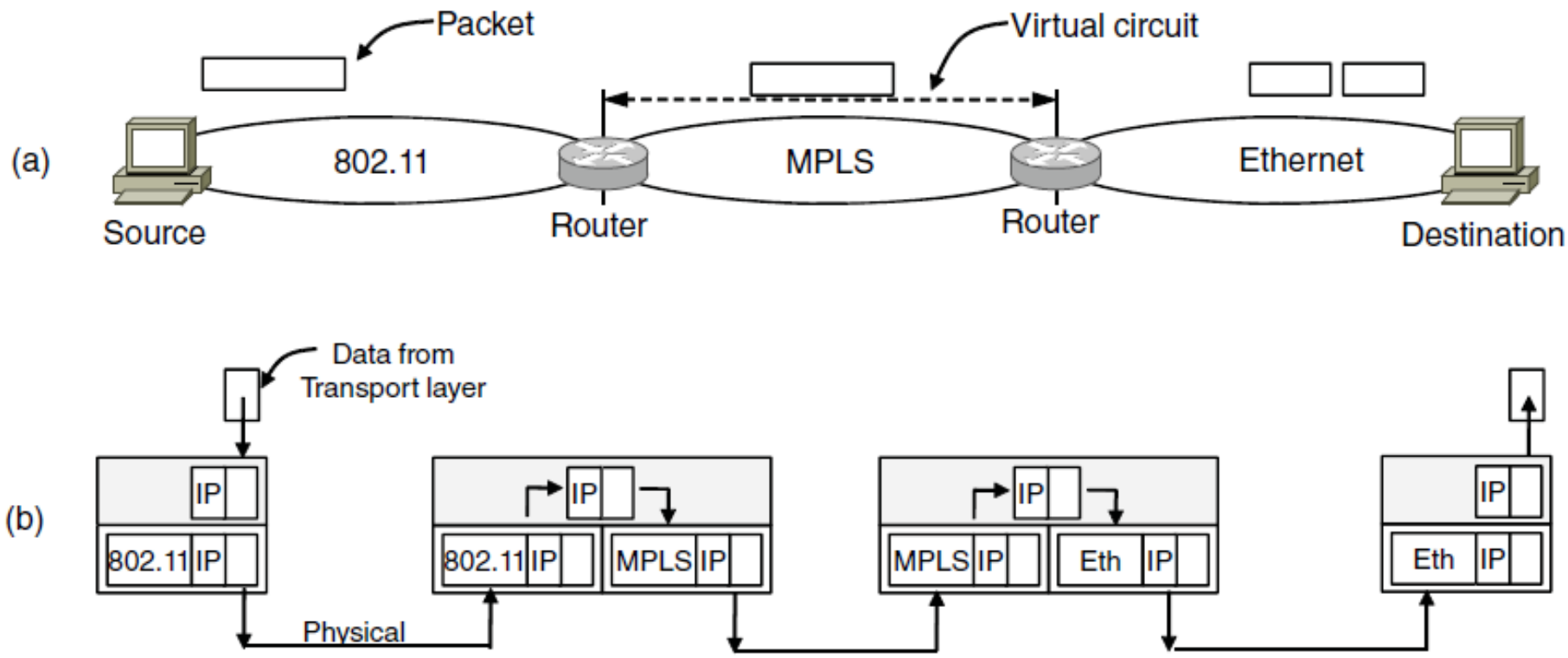
- Which . . . .

Which Brings Us to the INWG Model



- With the Transport Layer, OSI had rediscovered the INWG model.
- For different political reasons, OSI made a similar split to TCP/IP.
  - PTTs didn't want a Transport Layer at all.
  - This is independent confirmation. None of the OSI Network Layer group had been involved in INWG.
- And signs of a repeating structure.

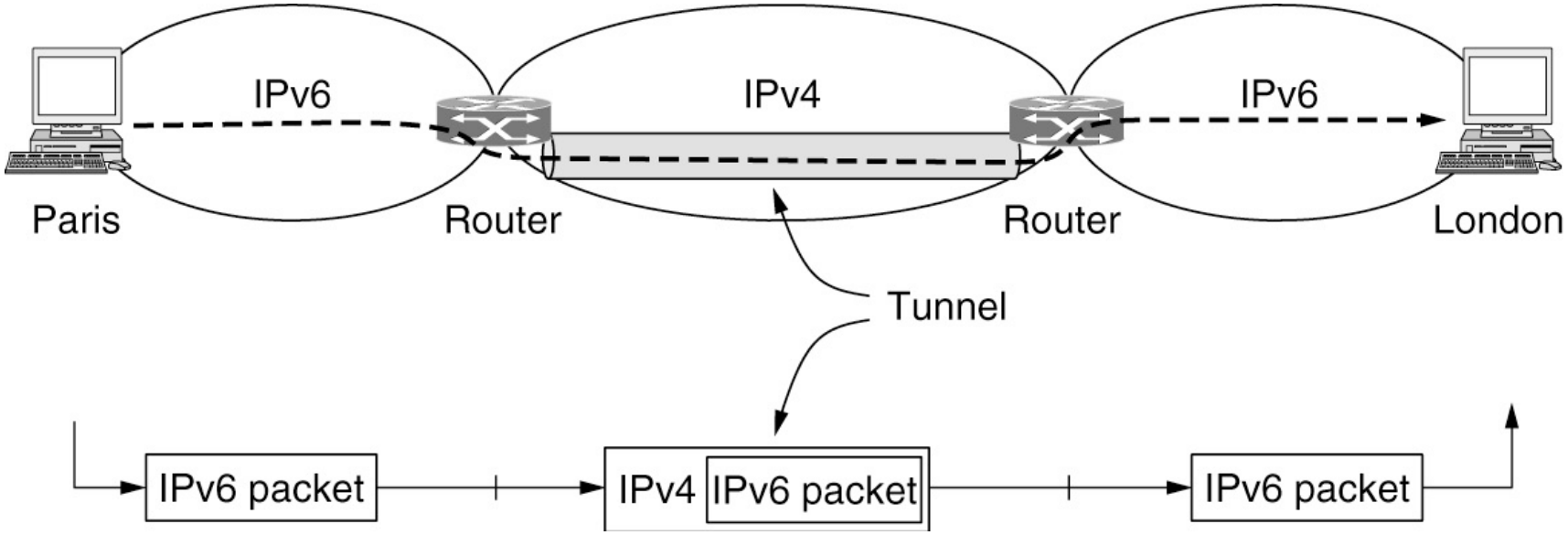
How Networks Can Be Connected



- A packet crossing different networks.
- Network and link layer protocol processing.
  - Where is the internet?
- Don't even think about protocol conversion. There be monsters there!

Connecting Endpoints Across Heterogeneous Networks (1 of 3)

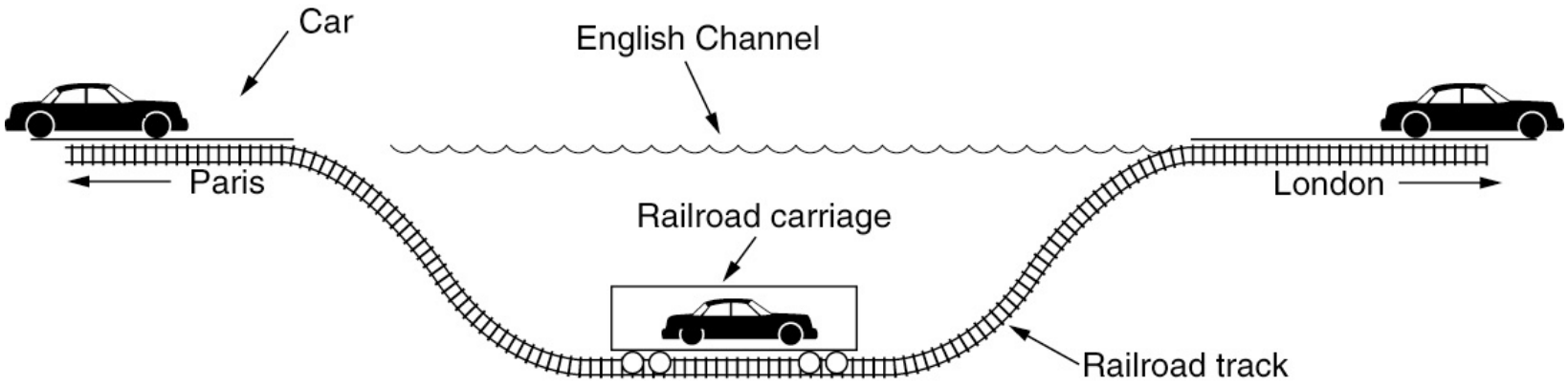
(no, actually this is tunneling)



Tunneling a packet from Paris to London

Connecting Endpoints Across Heterogeneous Networks (2 of 3)

(no, actually this is tunneling)

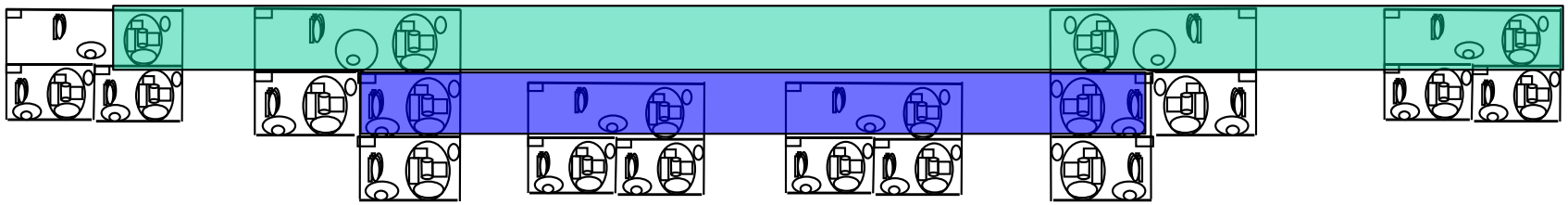


Tunneling a car from France to England

But the Internet doesn't 'tunnel' the whole car.

Connecting Endpoints across Heterogeneous Networks, (3 of 3)
(not Tunneling)

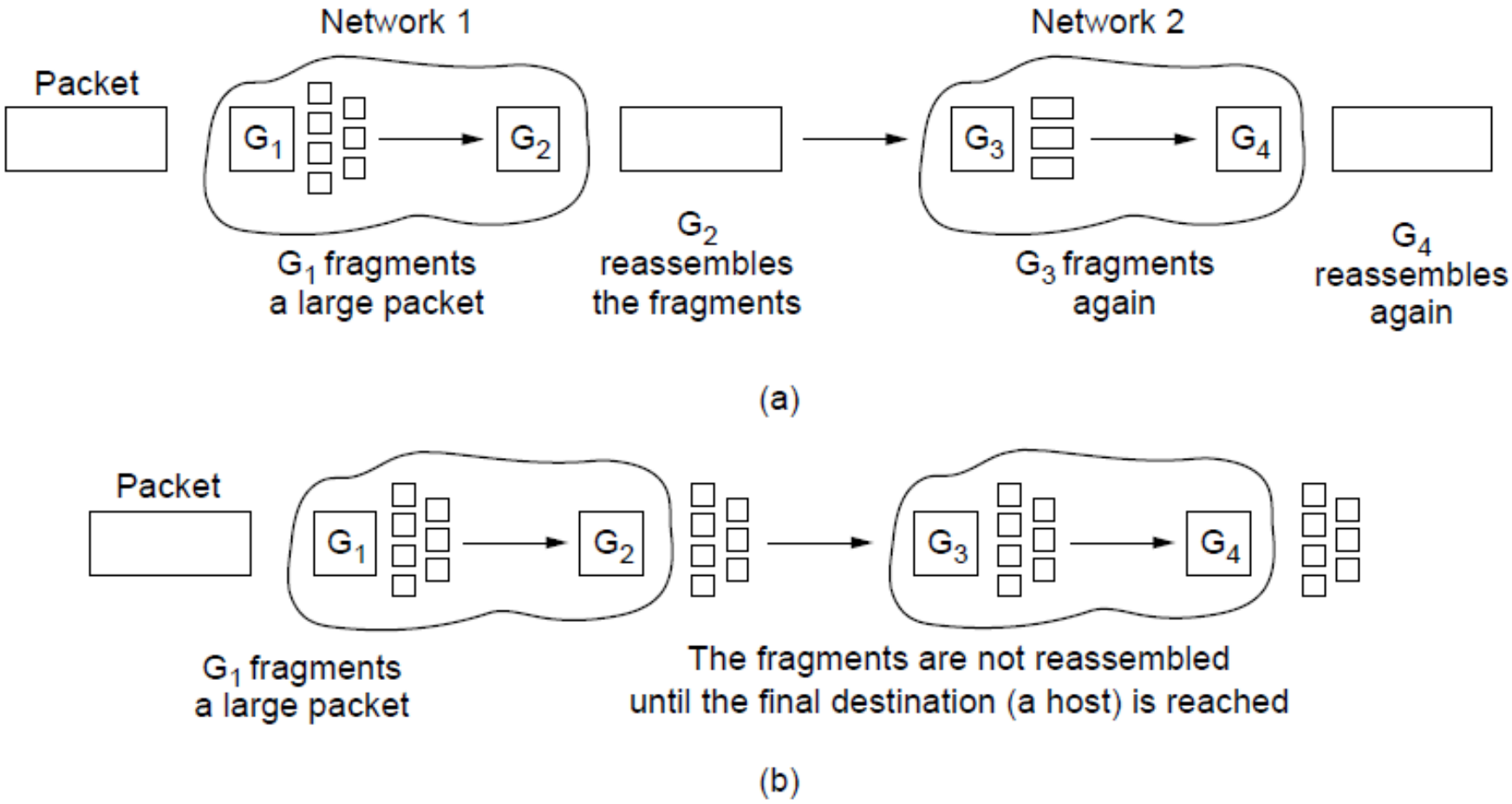
- Tunneling only "recurses" data transfer.
  - Routing remains flat, no scaling advantage.
- Must recurse the whole layer to gain the advantage.



IP Packet Fragmentation (1)

- Packet size issues:
  - Hardware
  - Operating system
  - Protocols
  - Compliance with (inter)national standard.
  - Reduce error-induced retransmissions (wrong)
    - Prevent packet occupying channel too long.

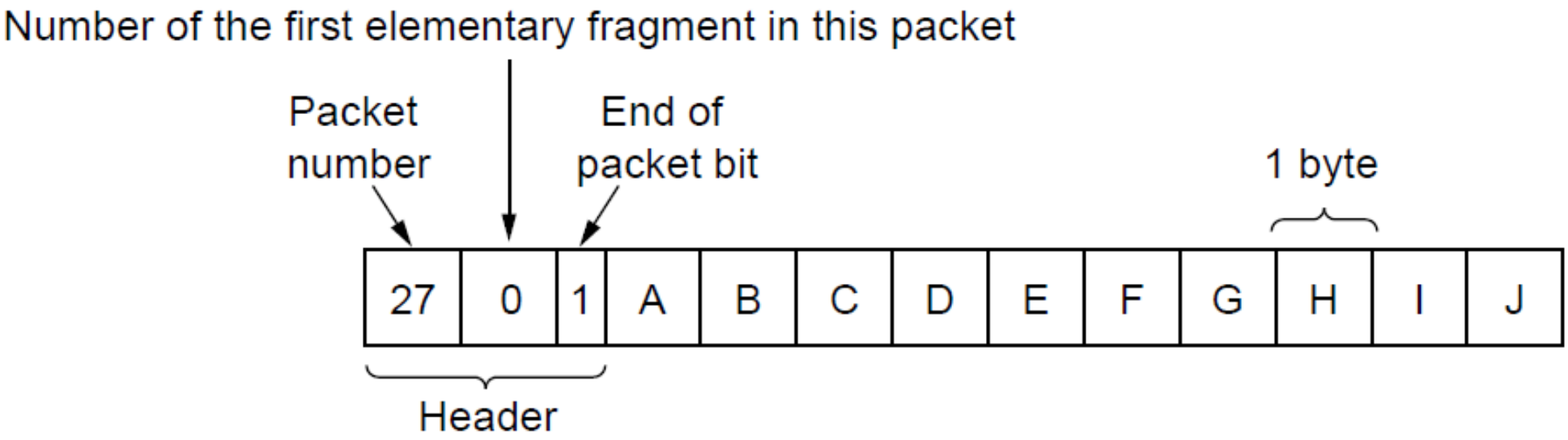
IP Packet Fragmentation (2)



(a) Transparent fragmentation.

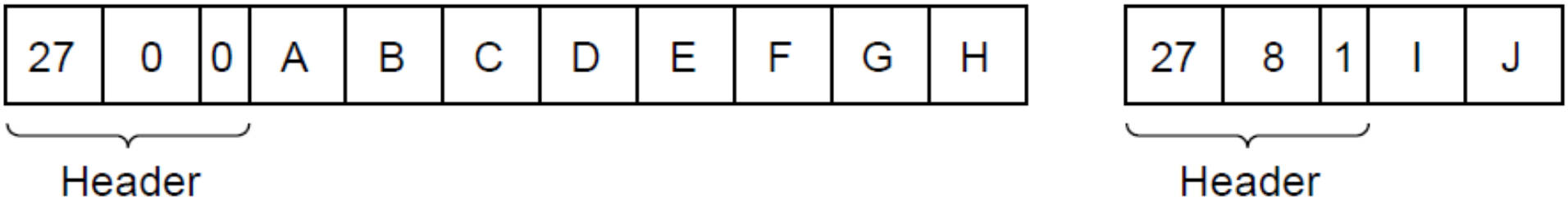
(b) Nontransparent fragmentation

IP Packet Fragmentation (3)



Fragmentation when the elementary data size is 1 byte.
(a) Original packet, containing 10 data bytes.

IP Packet Fragmentation (4)

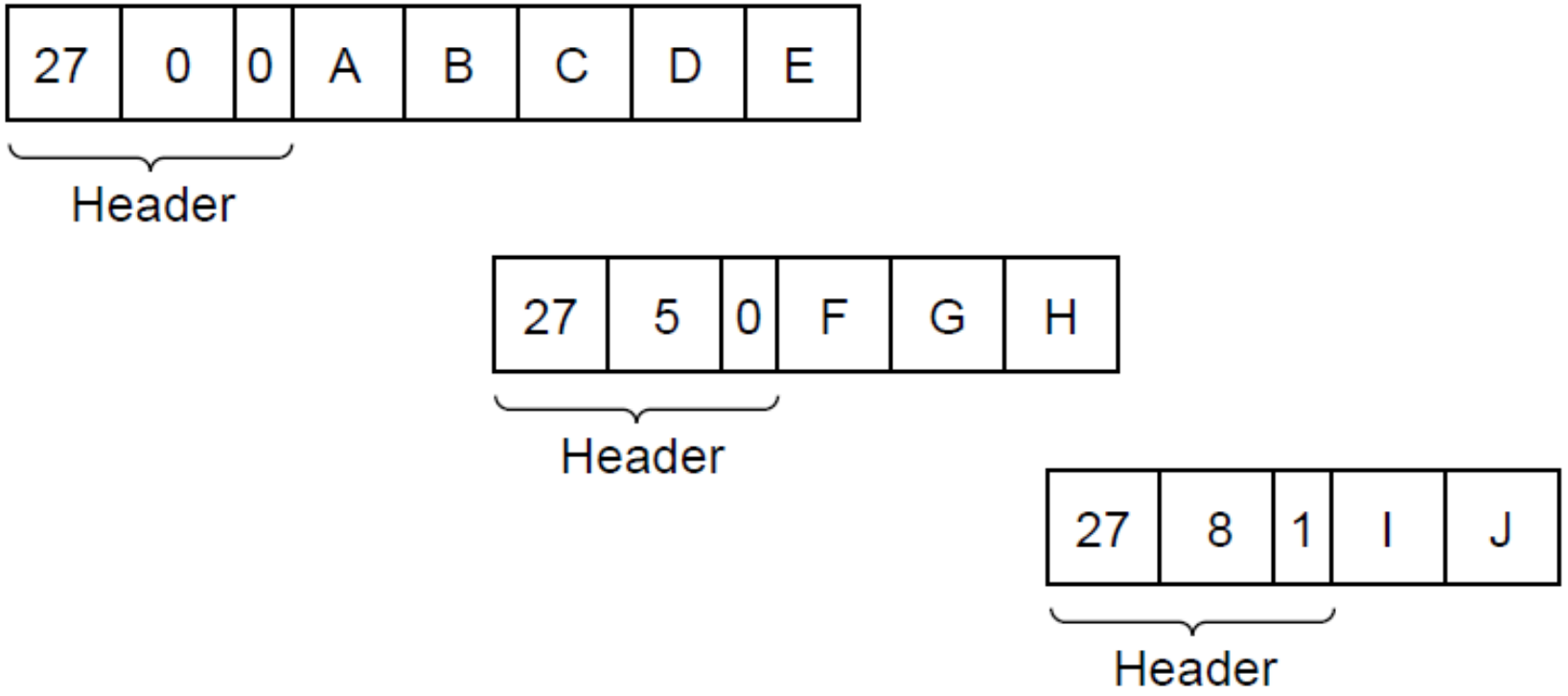


Fragmentation when the elementary data size is 1 byte

(b) Fragments after passing through a network

with maximum packet size of 8 payload bytes plus header.

IP Packet Fragmentation (5)



Fragmentation when the elementary data size is 1 byte

(c) Fragments after passing through a size 5 gateway.

IP Packet Fragmentation (6)

The Dirty Little Secret

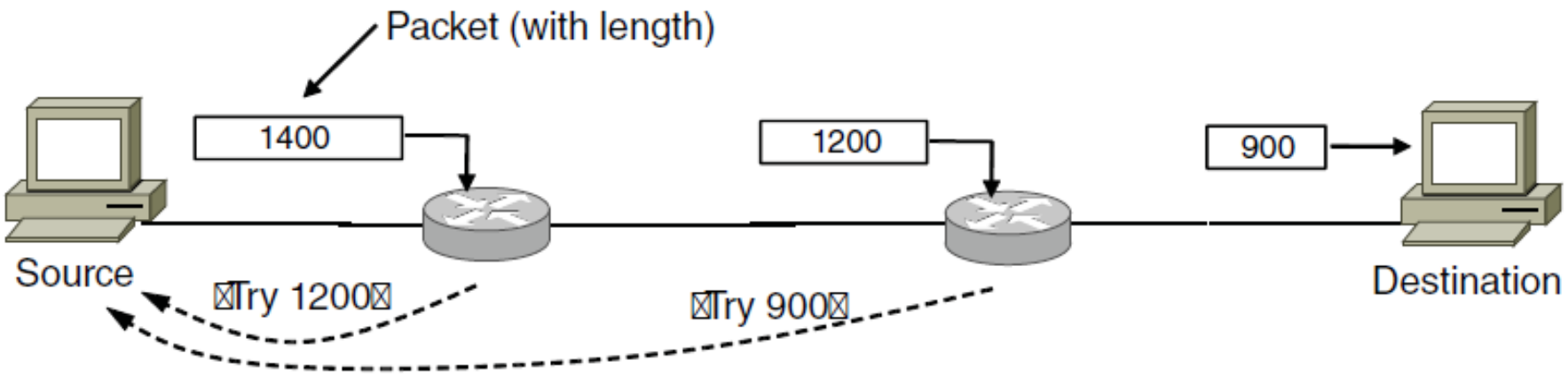
- It Doesn't Work
- As we will see with TCP/IP:
  1) Each packet needs an unique identifier to know what fragments belong to which packets.
  2) At some point, the packet is broken into n fragments. There is a probability, p, that at least one fragment is lost.
  3) The fragments that do arrive must be held for at least one MPL.

     But IP doesn't do retransmission!
  4) As we will see, Transport is going to re-transmit the *packet* after 1 RTT + ε << MPL, and hand it to the Network Layer.
  5) Which the network layer will dutifully assign a new identifier to and send. (See step 2).
- The result is obvious. The host may have several copies of the same packet and can't know they are copies or that some may complete others. Consuming needed buffers and holding up delivery to the application.

Packet Fragmentation (7)

There is a Patch!

Path MTU Discovery



Or

Doc, It hurts when I do this!

Doc: Then don't do it.

Just One Problem:

(This Doesn't Work Either)

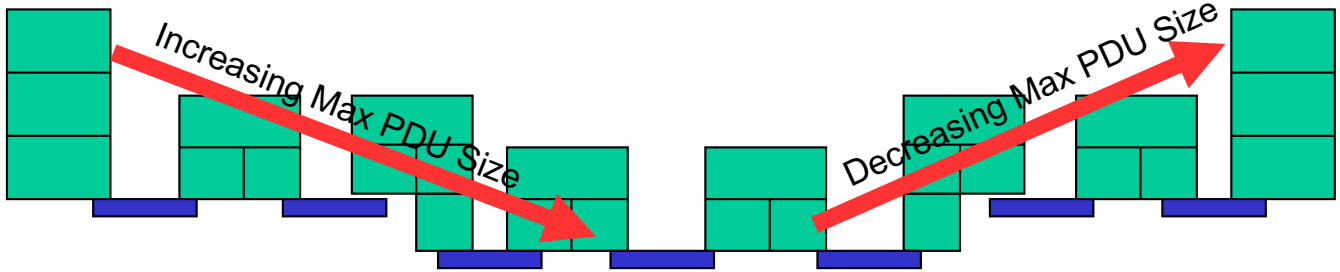
Many sites block ICMP because of DoS attacks

And it is an architectural mess!

IP Packet Fragmentation (8)

Conclusion

- Find a Solution: Part of it is in the Transport Layer
- PDU Size is a basic parameter of the layer (in the layered model).
- Moving toward the backbone, (higher traffic density) PDU size should increase.
  - Should be relaying more stuff less often, not less stuff more often.
  - Small PDU sizes at the edge, larger toward the center.
- Moving toward the backbone should tend toward more stuff going to the same place (intermediate point in the network.
  - May not be lots of traffic between Chelmsford and Lake Forest, but constant traffic between the Boston region and the Chicago region.



- So What is the Final Take-away?
  - Fragmentation is an indication of a poorly designed internet
  - Why did T spend so much time on it?

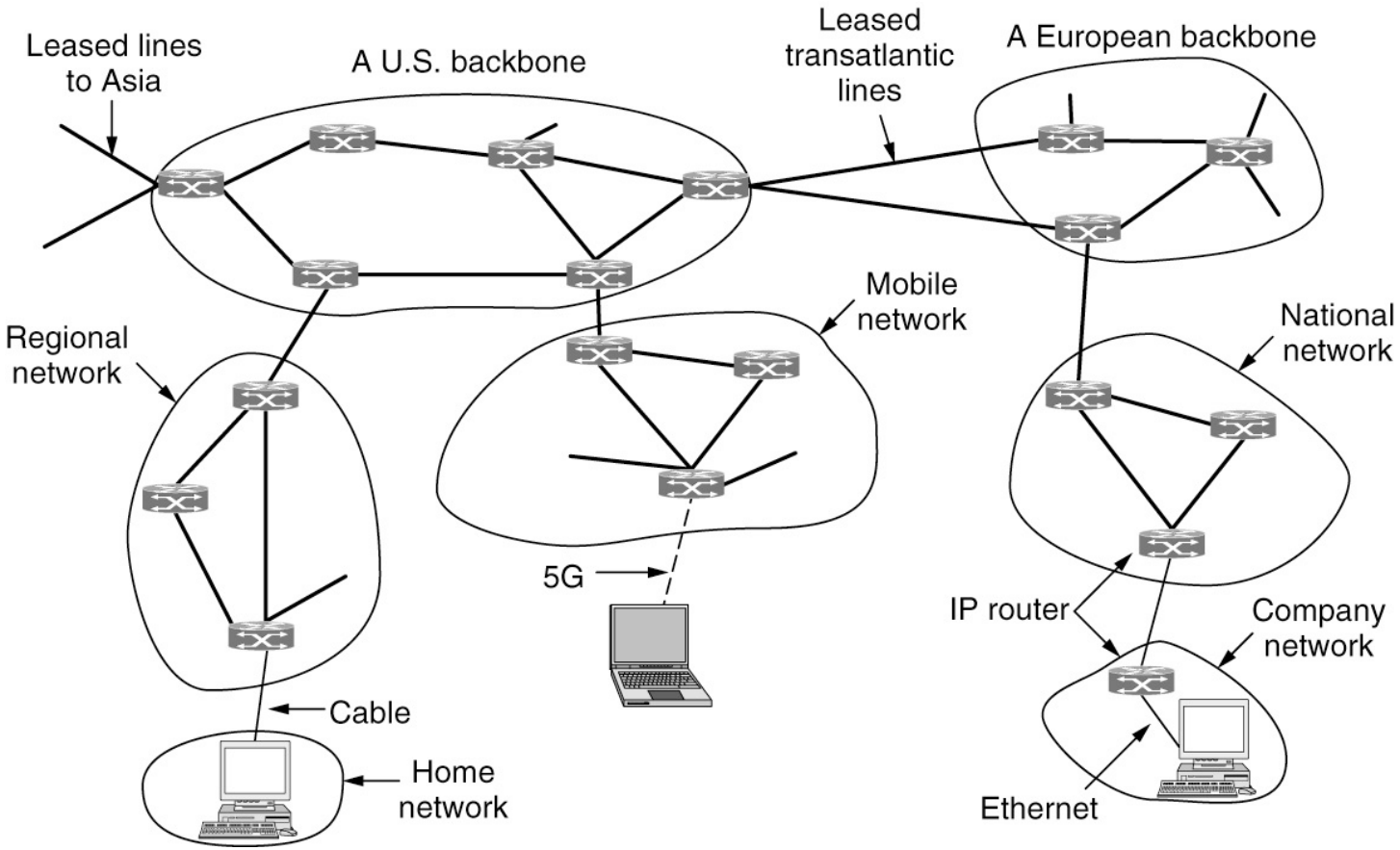
The Network Layer in the Internet (1 of 3)

- Top 10 principles
  - Make sure it works
  - Keep it simple
  - Make clear choices
  - Exploit modularity
  - Expect heterogeneity
  - Avoid static options and parameters
  - Look for a good design; it need not be perfect
  - Be strict when sending and tolerant when receiving
  - Think about scalability
  - Consider performance and cost

The Network Layer in the Internet (2 of 3)

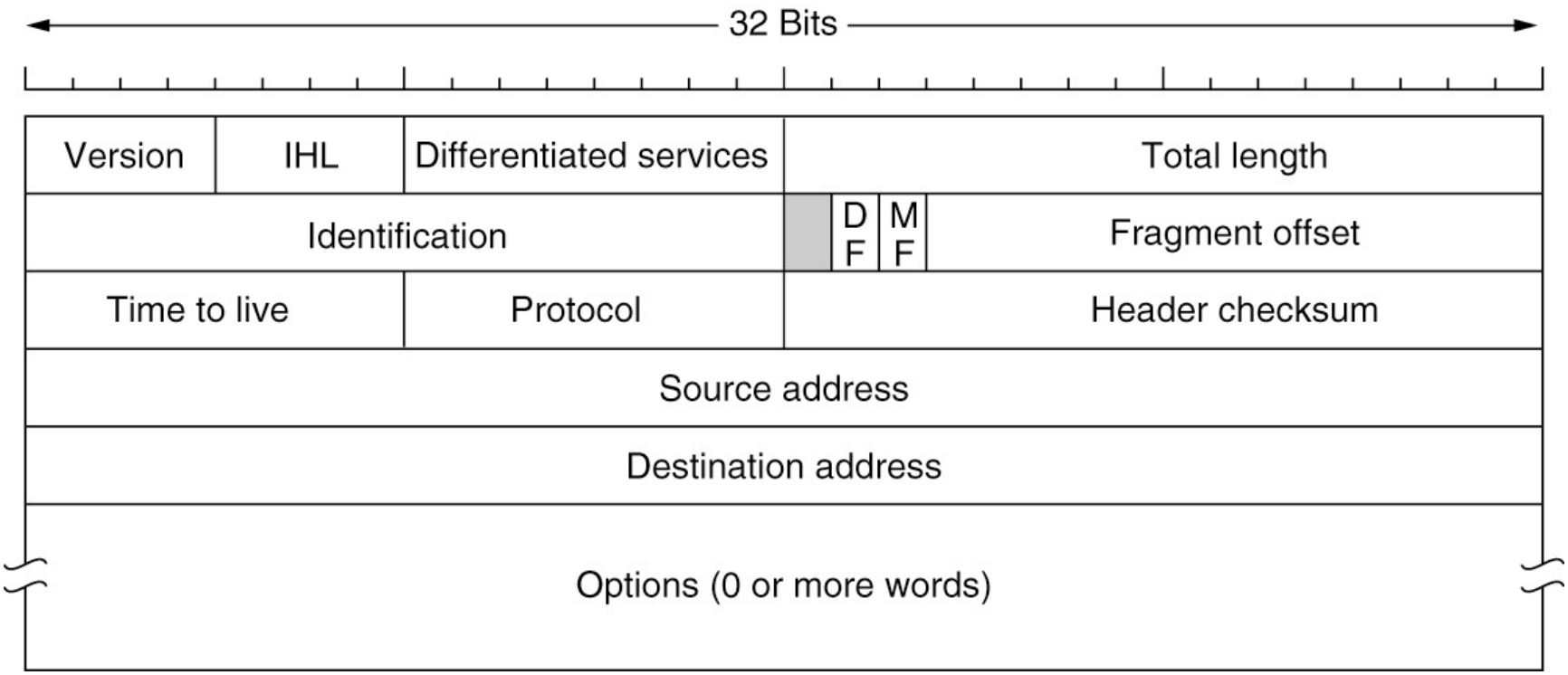
- The IP Version 4 Protocol
- IP Addresses
- IP Version 6
- Internet control protocols
- OSPF—An interior gateway routing protocol
- BGP—The exterior gateway routing protocol
- Internet multicasting

The Network Layer in the Internet (3 of 3)



The Internet is an interconnected collection of many networks.

The IP Version 4 Protocol (1 of 2)



The IPv4 (Internet Protocol version 4) header

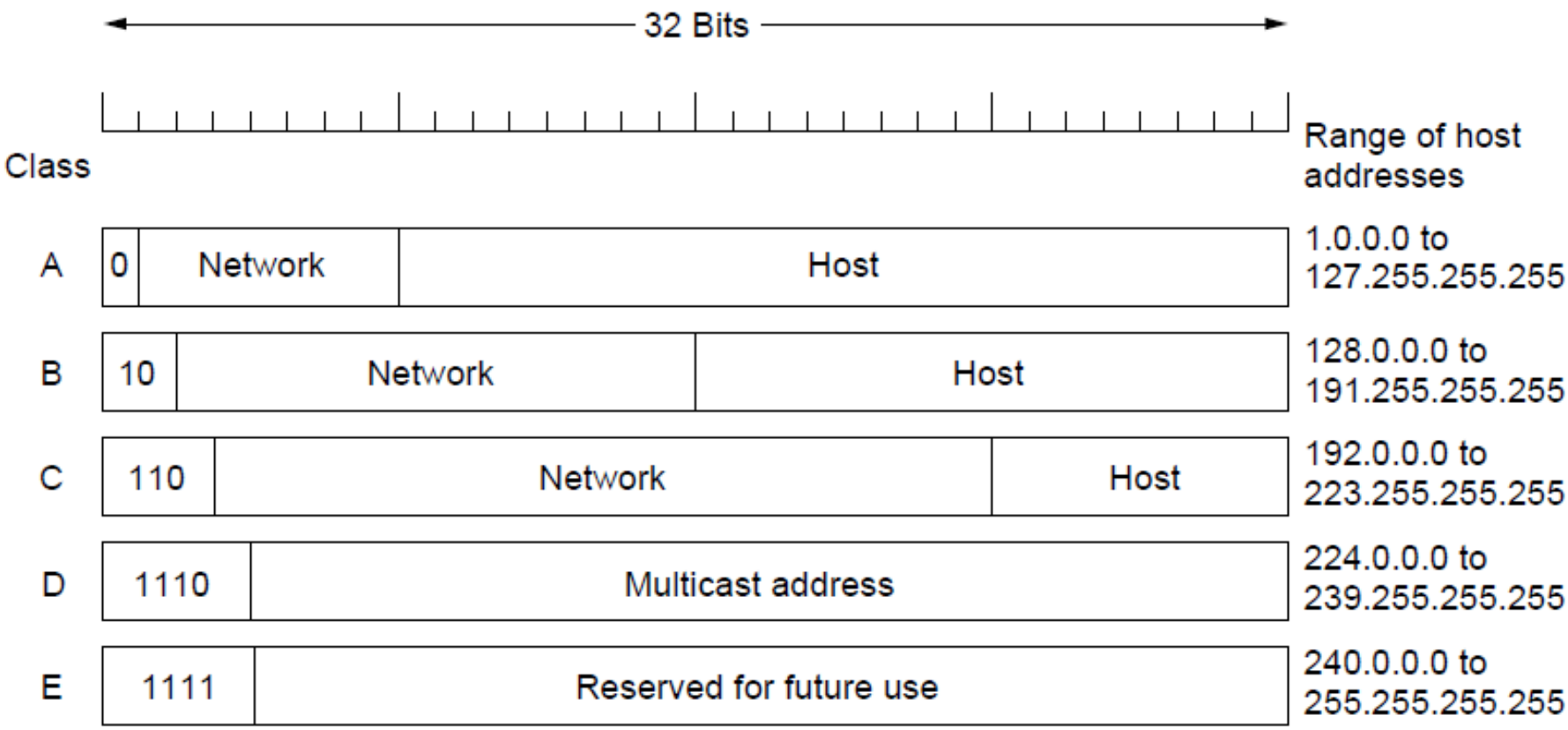
The IP Protocol (2)

Option	Description
Security	Specifies how secret the datagram is
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed
Record route	Makes each router append its IP address
Timestamp	Makes each router append its address and timestamp

Some of the IP options.

IP Addresses

Pre-CIDR formats



IP address formats

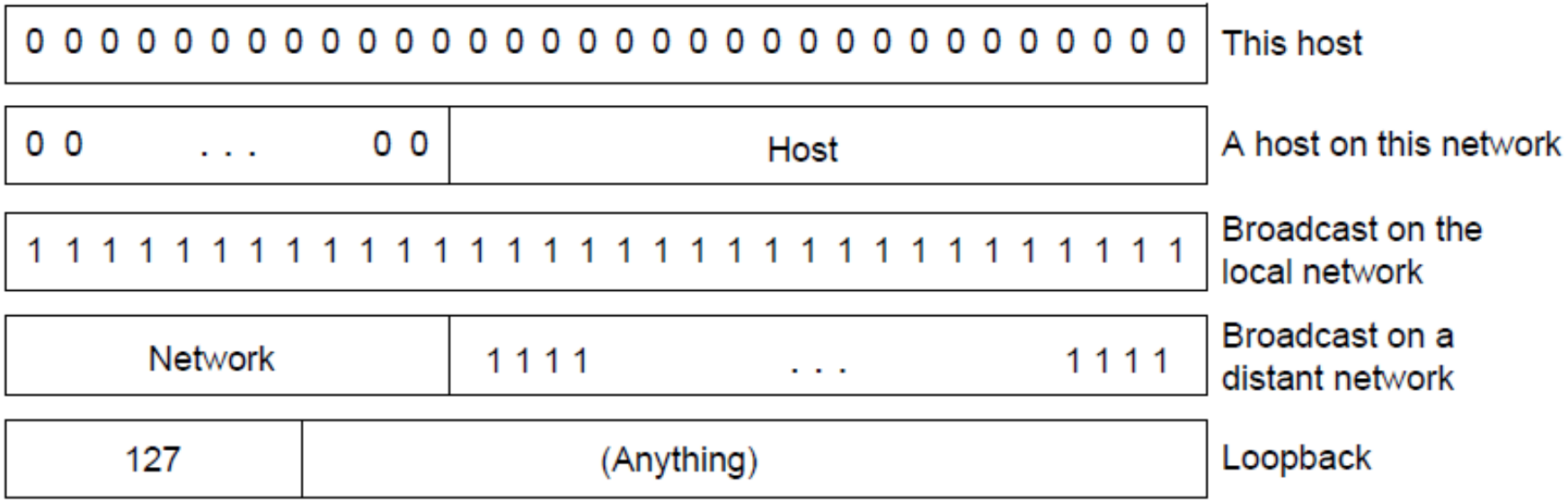
IPv4 Class A Allocations

Address Block	Registry - Purpose	Date
000/8	IANA - Reserved	Sep 81
001/8	IANA - Reserved	Sep 81
002/8	IANA - Reserved	Sep 81
003/8	General Electric Company	May 94
004/8	Bolt Beranek and Newman Inc.	Dec 92
005/8	IANA - Reserved	Jul 95
006/8	Army Information Systems Center	Feb 94
007/8	IANA - Reserved	Apr 95
008/8	Bolt Beranek and Newman Inc.	Dec 92
009/8	IBM	Aug 92
010/8	IANA - Private Use	Jun 95
011/8	DoD Intel Information Systems	May 93
012/8	AT&T Bell Laboratories	Jun 95
013/8	Xerox Corporation	Sep 91
014/8	IANA - Public Data Network	Jun 91
015/8	Hewlett-Packard Company	Jul 94
016/8	Digital Equipment Corporation	Nov 94
017/8	Apple Computer Inc.	Jul 92
018/8	MIT	Jan 94
019/8	Ford Motor Company	May 95
020/8	Computer Sciences Corporation	Oct 94
021/8	DDN-RVN	Jul 91
022/8	Defense Information Systems Agency	May 93
023/8	IANA - Reserved	Jul 95
024/8	ARIN - Cable Block	May 01
	(Formerly IANA - Jul 95)	
025/8	Royal Signals and Radar Establishment	Jan 95
026/8	Defense Information Systems Agency	May 95
027/8	IANA - Reserved	Apr 95
028/8	DSI-North	Jul 92
029/8	Defense Information Systems Agency	Jul 91
030/8	Defense Information Systems Agency	Jul 91
031/8	IANA - Reserved	Apr 99
032/8	Norsk Informasjonsteknologi	Jun 94
033/8	DLA Systems Automation Center	Jan 91
034/8	Halliburton Company	Mar 93
035/8	MERIT Computer Network	Apr 94
036/8	IANA - Reserved	Jul 00
	(Formerly Stanford University - Apr 93)	
037/8	IANA - Reserved	Apr 95
038/8	Performance Systems International	Sep 94
039/8	IANA - Reserved	Apr 95
040/8	Eli Lily and Company	Jun 94
041/8	IANA - Reserved	May 95
042/8	IANA - Reserved	Jul 95
043/8	Japan Inet	Jan 91
044/8	Amateur Radio Digital Communications	Jul 92
045/8	Interop Show Network	Jan 95
046/8	Bolt Beranek and Newman Inc.	Dec 92
047/8	Bell-Northern Research	Jan 91
048/8	Prudential Securities Inc.	May 95
049/8	Joint Technical Command	May 94
	Returned to IANA	Mar 98
050/8	Joint Technical Command	May 94
	Returned to IANA	Mar 98
051/8	Deparment of Social Security of UK	Aug 94
052/8	E.I. duPont de Nemours and Co., Inc.	Dec 91
053/8	Cap Debis CCS	Oct 93
054/8	Merck and Co., Inc.	Mar 92
055/8	Boeing Computer Services	Apr 95
056/8	U.S. Postal Service	Jun 94
057/8	SITA	May 95
058/8	IANA - Reserved	Sep 81
059/8	IANA - Reserved	Sep 81
060/8	IANA - Reserved	Sep 81

IPv4 Class A Allocations

061/8	APNIC - Pacific Rim	Apr 97
062/8	RIPE NCC - Europe	Apr 97
063/8	ARIN	Apr 97
064/8	ARIN	Jul 99
065/8	ARIN	Jul 00
066/8	ARIN	Jul 00
067/8	ARIN	May 01
068/8	ARIN	Jun 01
069-079/8	IANA - Reserved	Sep 81
080/8	RIPE NCC	Apr 01
081/8	RIPE NCC	Apr 01
082-095/8	IANA - Reserved	Sep 81
096-126/8	IANA - Reserved	Sep 81
127/8	IANA - Reserved	Sep 81
128-191/8	Various Registries	May 93
192/8	Various Registries - MultiRegional	May 93
193/8	RIPE NCC - Europe	May 93
194/8	RIPE NCC - Europe	May 93
195/8	RIPE NCC - Europe	May 93
196/8	Various Registries	May 93
197/8	IANA - Reserved	May 93
198/8	Various Registries	May 93
199/8	ARIN - North America	May 93
200/8	ARIN - Central and South America	May 93
201/8	Reserved - Central and South America	May 93
202/8	APNIC - Pacific Rim	May 93
203/8	APNIC - Pacific Rim	May 93
204/8	ARIN - North America	Mar 94
205/8	ARIN - North America	Mar 94
206/8	ARIN - North America	Apr 95
207/8	ARIN - North America	Nov 95
208/8	ARIN - North America	Apr 96
209/8	ARIN - North America	Jun 96
210/8	APNIC - Pacific Rim	Jun 96
211/8	APNIC - Pacific Rim	Jun 96
212/8	RIPE NCC - Europe	Oct 97
213/8	RIPE NCC - Europe	Mar 99
214/8	US-DOD	Mar 98
215/8	US-DOD	Mar 98
216/8	ARIN - North America	Apr 98
217/8	RIPE NCC - Europe	Jun 00
218/8	APNIC - Pacific Rim	Dec 00
219/8	APNIC	Sep 01
220/8	APNIC	Dec 01
221-223/8	IANA - Reserved	Sep 81
224-239/8	IANA - Multicast	Sep 81
240-255/8	IANA - Reserved	Sep 81

IPv4 Addresses



Special IP addresses

First Internet Addressing Crisis

More About This Next Time

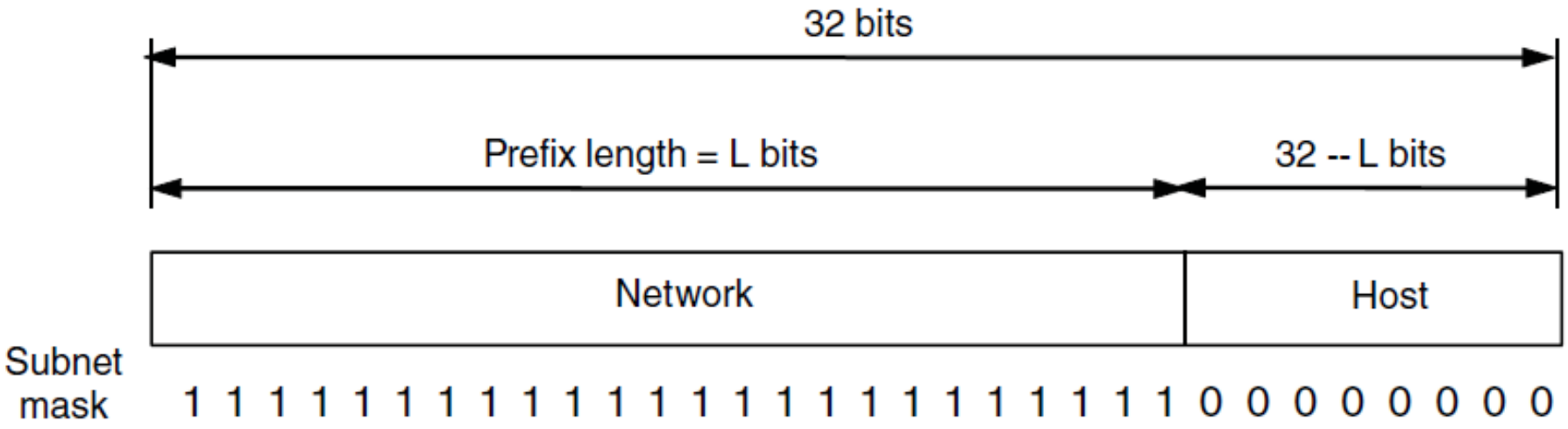
- There were Three Problems
  - Exponential Growth of Router Tables
  - Route on the node to solve Multihoming
  - Impending Shortage of Addresses
- Immediate Steps to Moderate the Problem
  - Create Private Address Space and Use Network Address Translation (NATs)
  - Tighten-Rules for Handing out Large Blocks of Addresses
    - Refer most to Tier 1 Providers
  - Give Tier 1 providers large blocks for their customers
    - Allows Addresses to be aggregated
  - Move to Classless Inter-Domain Routing
  - Recommend a Replacement for IPv4

IP Addresses

- Classful and special addressing
- Prefixes
  - A contiguous block of IP address space
- Subnets
- CIDR—Classless InterDomain Routing
- NAT—Network Address Translation

IP Addresses (1 of 7)

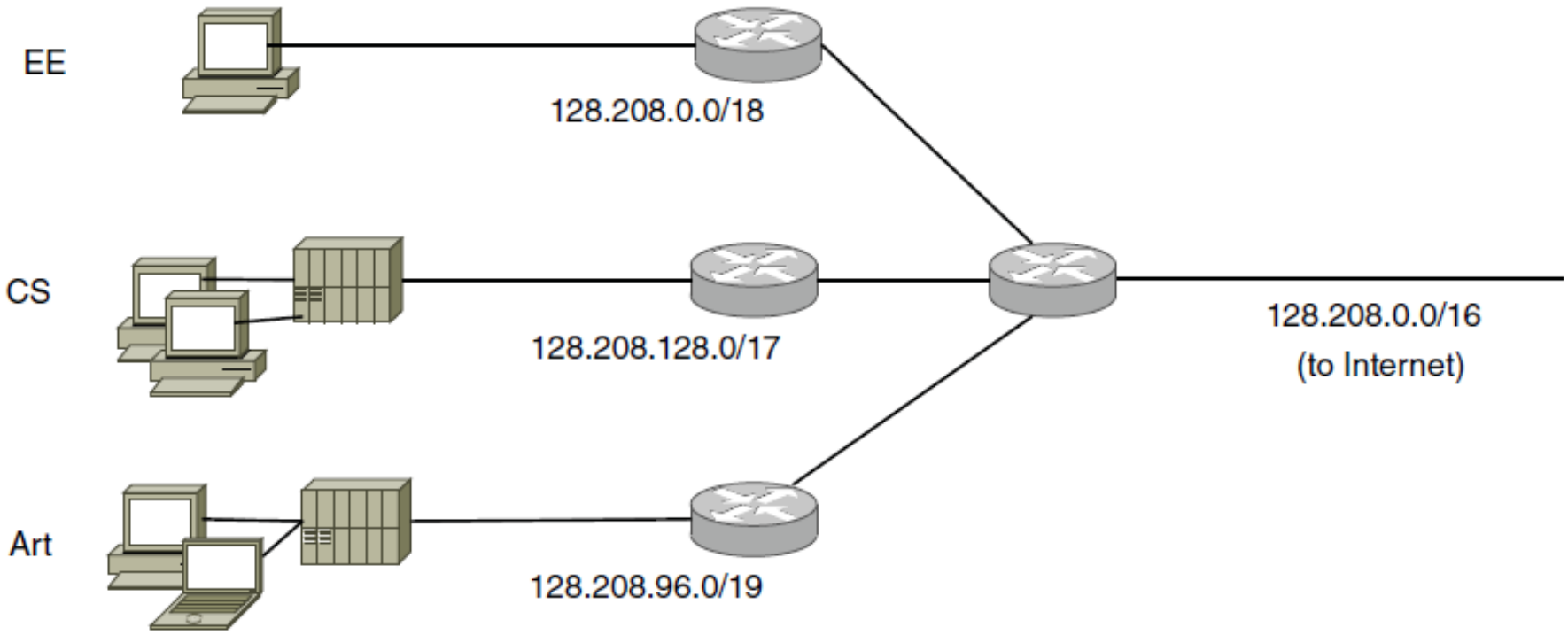
CIDR- Classless Interdomain Routing



An IP prefix and a subnet mask.

Start treating IP addresses as addresses

IP Addresses (2 of 7)



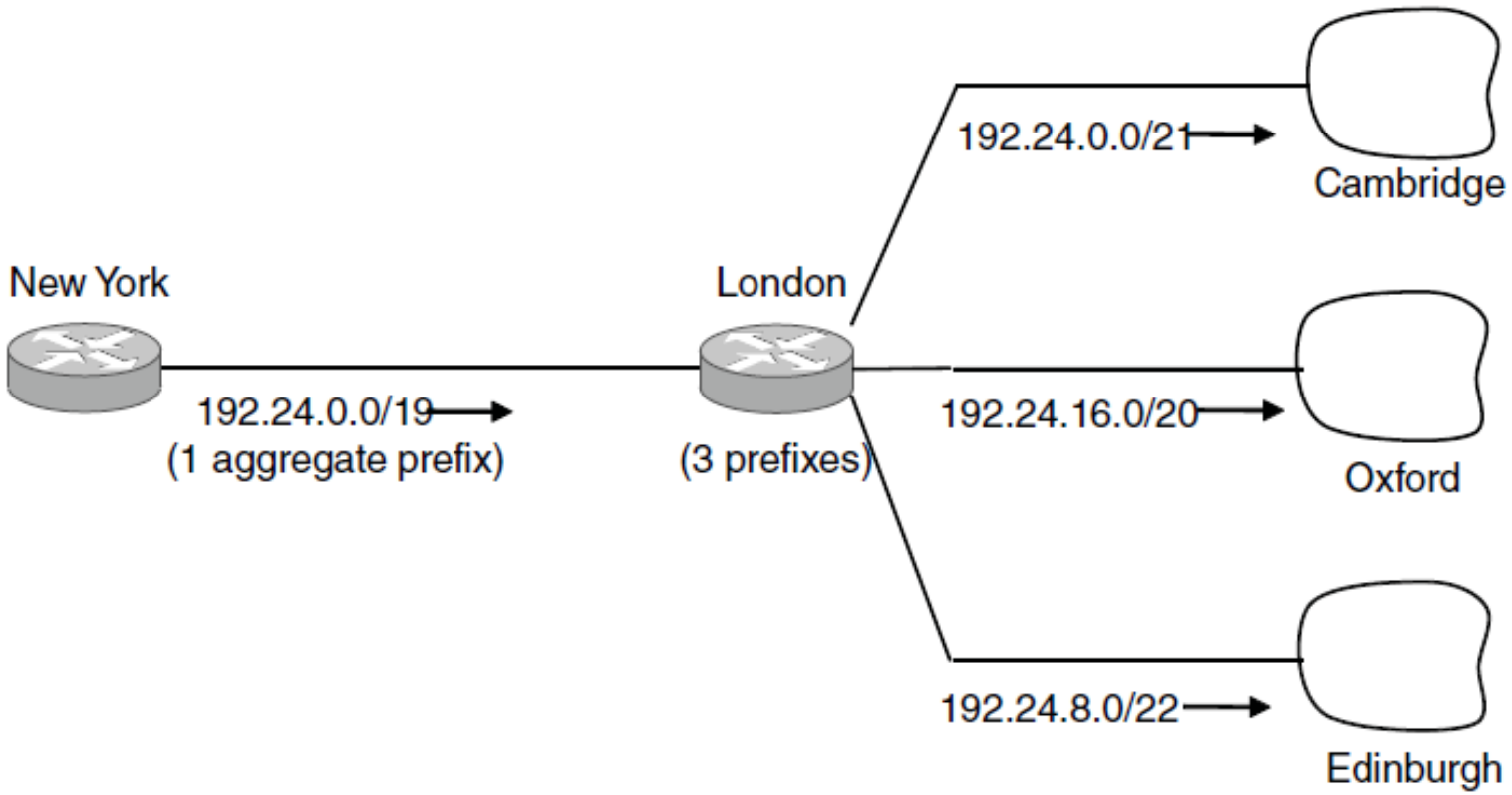
Splitting an IP prefix into separate networks with subnetting.

IP Addresses (3 of 7)

University	First address	Last address	How many	Prefix
Cambridge	194.24.0.0	194.24.7.255	2048	194.24.0.0/21
Edinburgh	194.24.8.0	194.24.11.255	1024	194.24.8.0/22
(Available)	194.24.12.0	194.24.15.255	1024	194.24.12/22
Oxford	194.24.16.0	194.24.31.255	4096	194.24.16.0/20

A set of IP address assignments

IP Addresses (4 of 7)



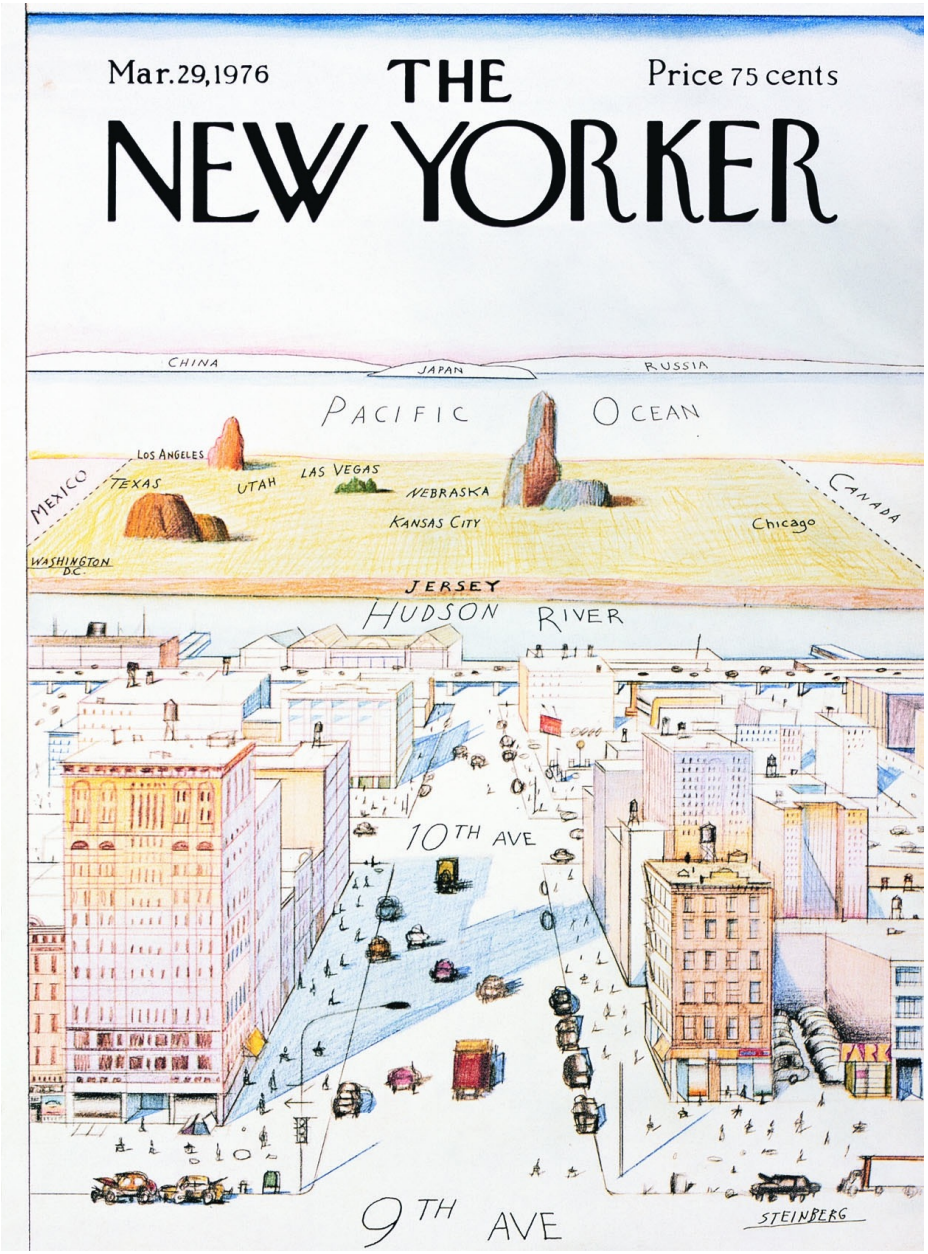
Aggregation of IP prefixes

The Internet sees this as Aggregation.

It is really making the addresses location-dependent.

IP Addresses (5 of 7)

What's Really Going On

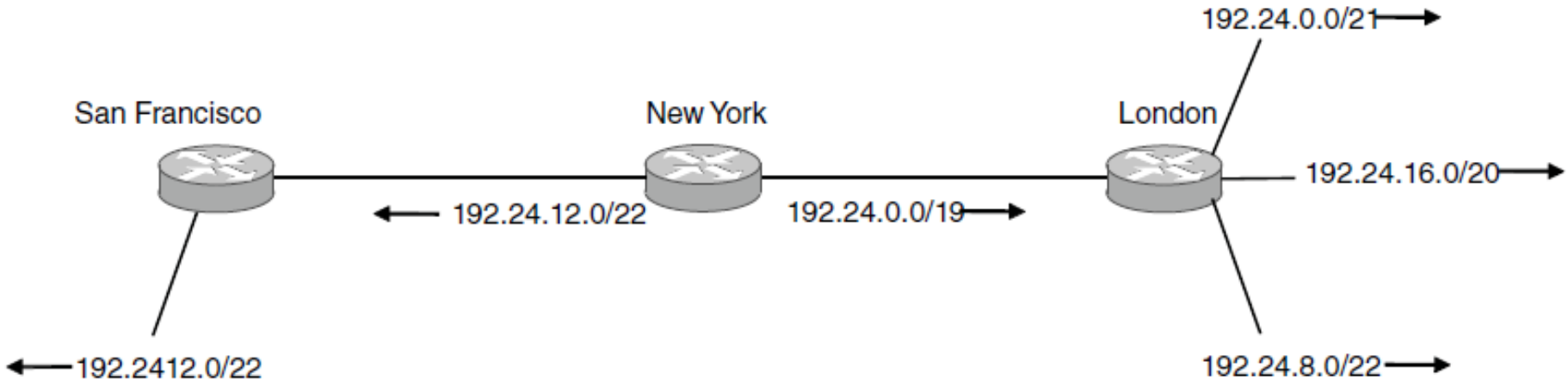


IP Addresses (6 of 7)

What's Really Going On for Each Router

- Aggregation is pre-processing the network graph relative to this router to create a new graph with fewer nodes (the New Yorker view)
- Sets of Addresses are being aggregated into a single node of the new graph.
- Then the routing algorithm is run on this new graph.
- This works because all of the aggregated addresses would have been forwarded to the same next hop anyway.
- The graph that each router runs the routing algorithm on is potentially a different graph.
  - Notice I have not used the word 'topology' once. Because they aren't.
    - If you don't know what a topology is, don't use the word.

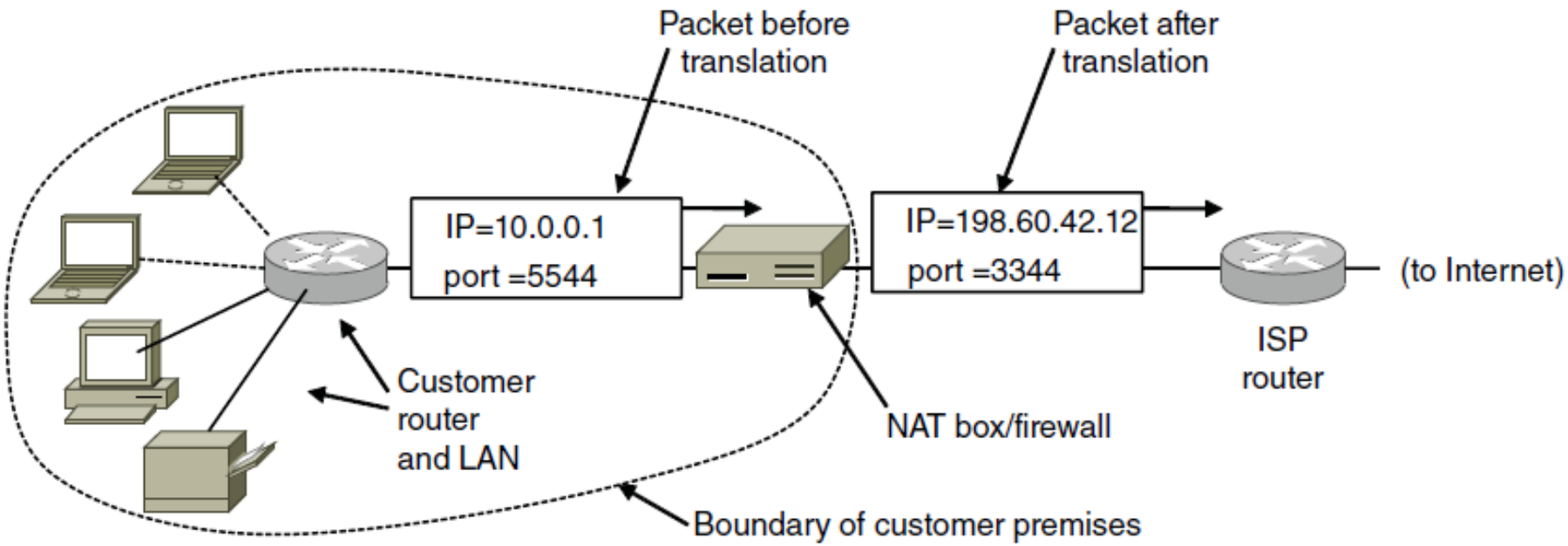
IP Addresses (7 of 7)



Longest matching prefix routing at the New York router.

NAT (1 of 3)

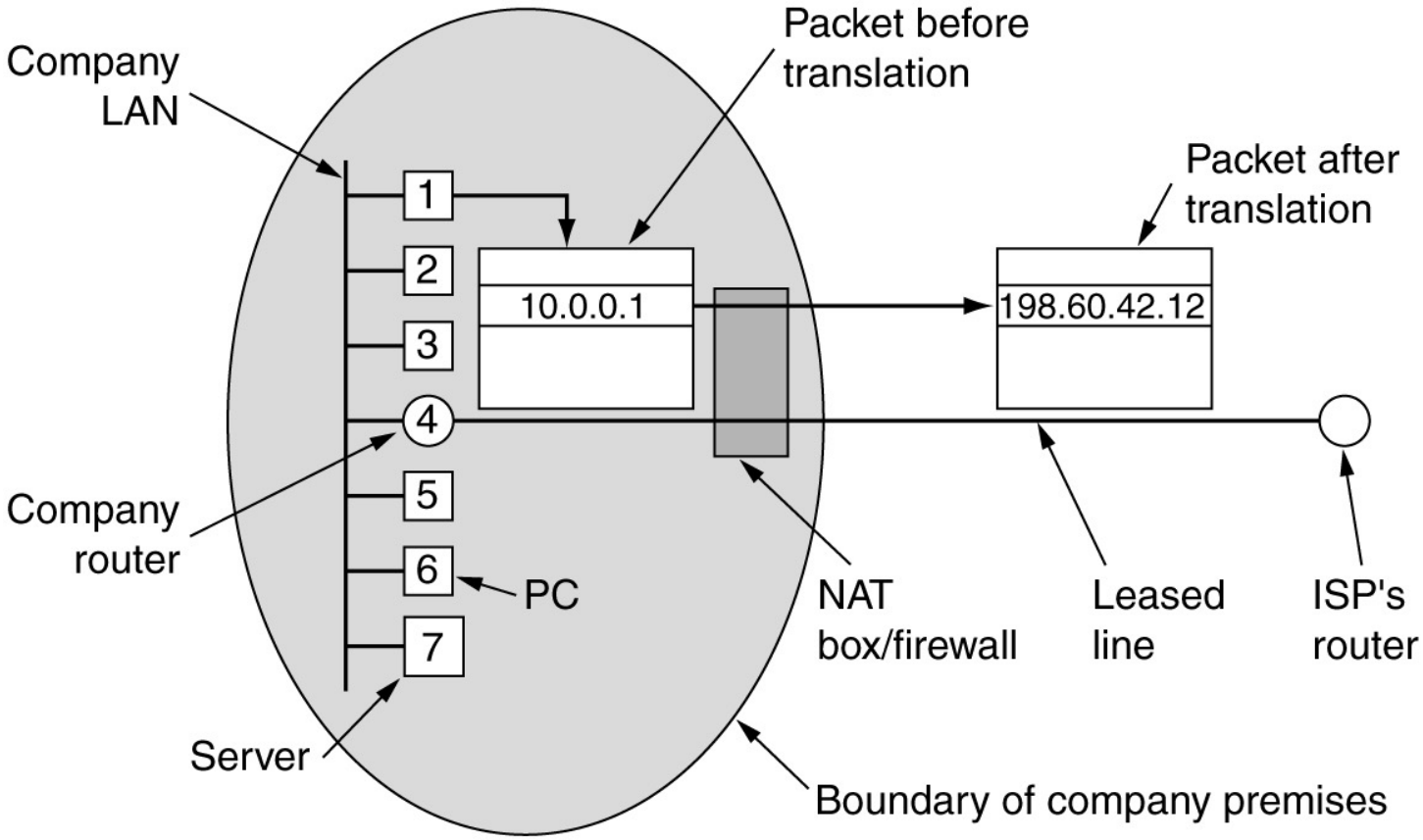
Network Address Translation



Placement and operation of a NAT box.

NAT (2 of 3)

Network Address Translation



Placement and operation of a NAT box.

NAT (3 of 3)

What's Wrong with NATs

- Everything under a single address space.
  - Most sites do not want every address globally visible.
- Makes the Internet more connection-like
  - Artifact of incomplete addressing architecture.
- Violates Layering
  - So does the Protocol ID field in IP
- Forces use of TCP or UDP
  - Not really. No matter what protocol is on top it will need port-ids.
- Some applications pass IP addresses
  - Arrrrgh!!! Passing IP addresses in application protocols is like passing physical memory addresses in Java!
- NATs only break broken architectures

IP Version 6 (1 of 3)

- The main IPv6 header
- Extension headers
- Controversies

IP Version 6 (2 of 3)
(the spin doctor version)

- IPv6 major goals
  - Support billions of hosts
  - Reduce routing table size (only realized after choosing the design)
  - Simplify the protocol (then why all the RFCs?)
  - Provide better security (hype, same as IPv4)
  - Attention to type of service (still debated and not clear)
  - Aid multicasting (same flawed definition)
  - Roaming host without changing address (then it isn't an address)
  - Allow future protocol evolution (design thwarts that)
  - Permit coexistence of old and new protocols for years
    - (Yes! The only field in common with IPv4 is the version field)

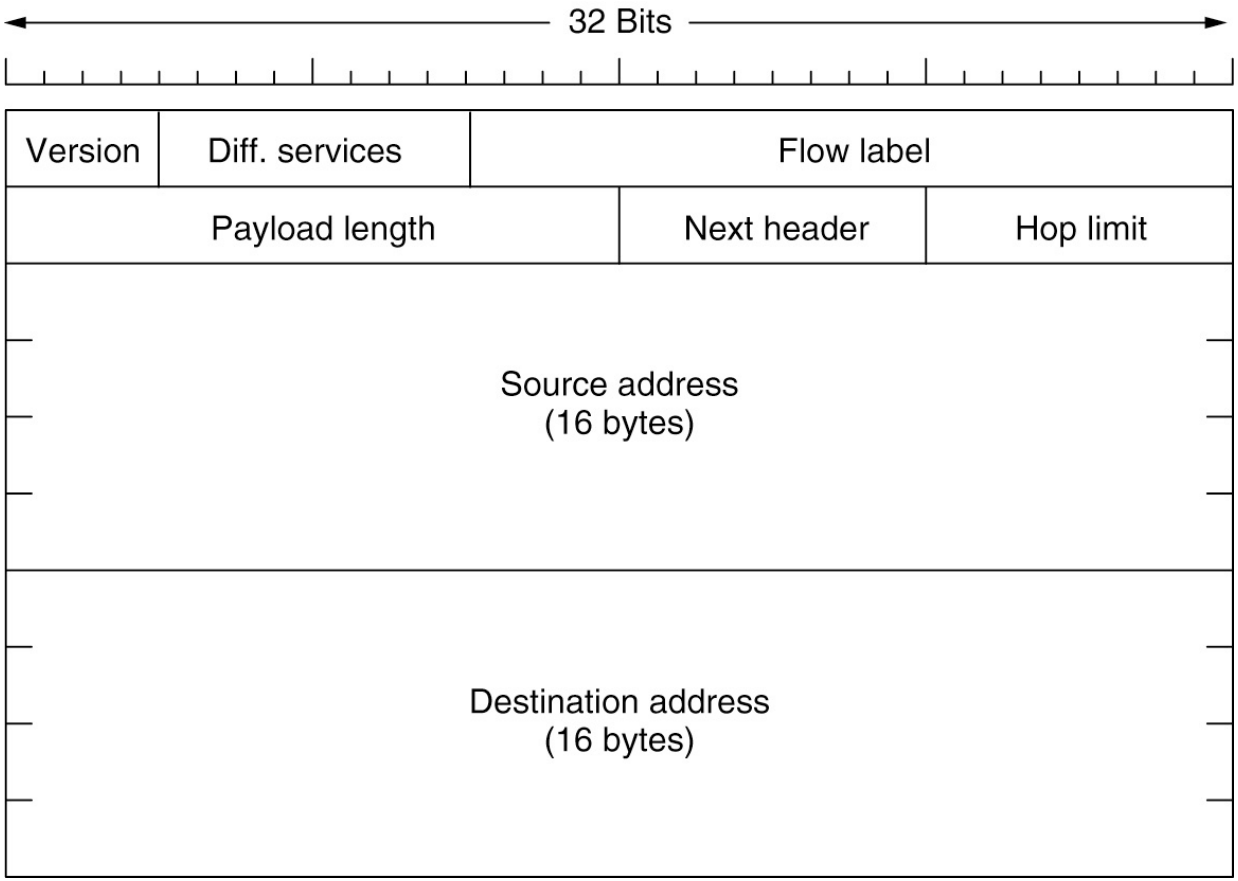
IP Version 6 (3 of 3)
(still mostly spin)

- IP version 6 improvements
  - Longer addresses than IPv4 (Yes! But is 64 bits enough)
  - Simplification of the header (You said that)
  - Better support for options (LOL! Two id lists for the same field)
  - Big advance is in security (Same as IPv4)
  - Quality of service (Same as IPv4)

The "Advantages" of IPv6
(Reality)

- More addresses - True, but 64 bits, not 128.
- Header Simplification - Gee whiz! 2% of the problem. Can't skip options.
- Better option support - (which aren't optional) ignored by most routers
  - Any option is handled by slow path. Pushes nets toward consistency
- Improved security - same as IPv4
  - Ground rule: Anything developed before transition must work with both.
- Does not address the real problem: Router table size - This is now a crisis.
- Transition requires a NAT - Once you have a NAT, you don't need v6
- The real problem with adoption - No benefit to whoever has to pay for the transition.
- No private addresses - Had to reverse themselves on this. (twice)
  - Much to their chagrin this one of the big points for v6!
- There are going to be scaling problems: Route calculation.
- Consequently, IETF has resorted to spin.

The Main IPv6 Header



The IPv6 fixed header (required)

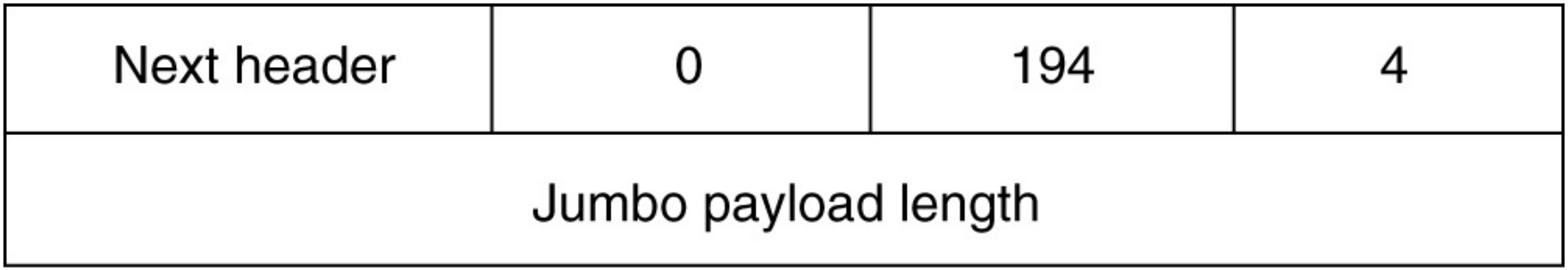
Extension Headers

Extension header	Description
Hop-by-hop options	Miscellaneous information for routers
Destination options	Additional information for the destination
Routing	Loose list of routers to visit
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender's identity
Encrypted security payload	Information about the encrypted contents

IPv6 extension headers.

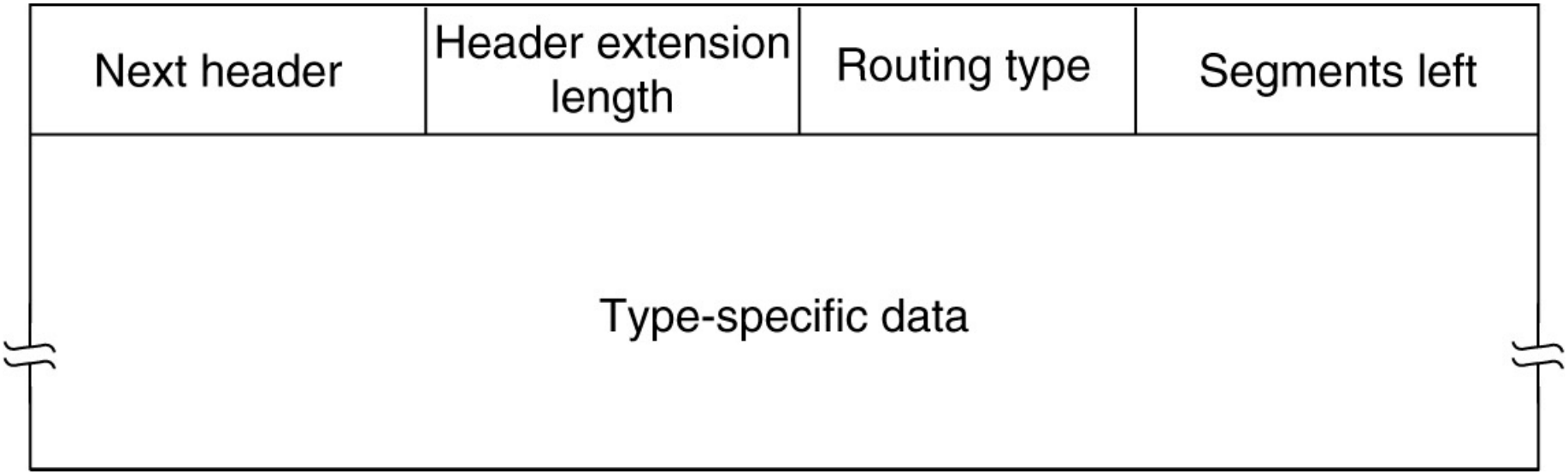
40% of all PDUs with EHs are being discarded

Extension Headers (2)



The hop-by-hop extension header for large datagrams (jumbograms).

Extension Headers (3)



The extension header for routing.

IPv6 Problems

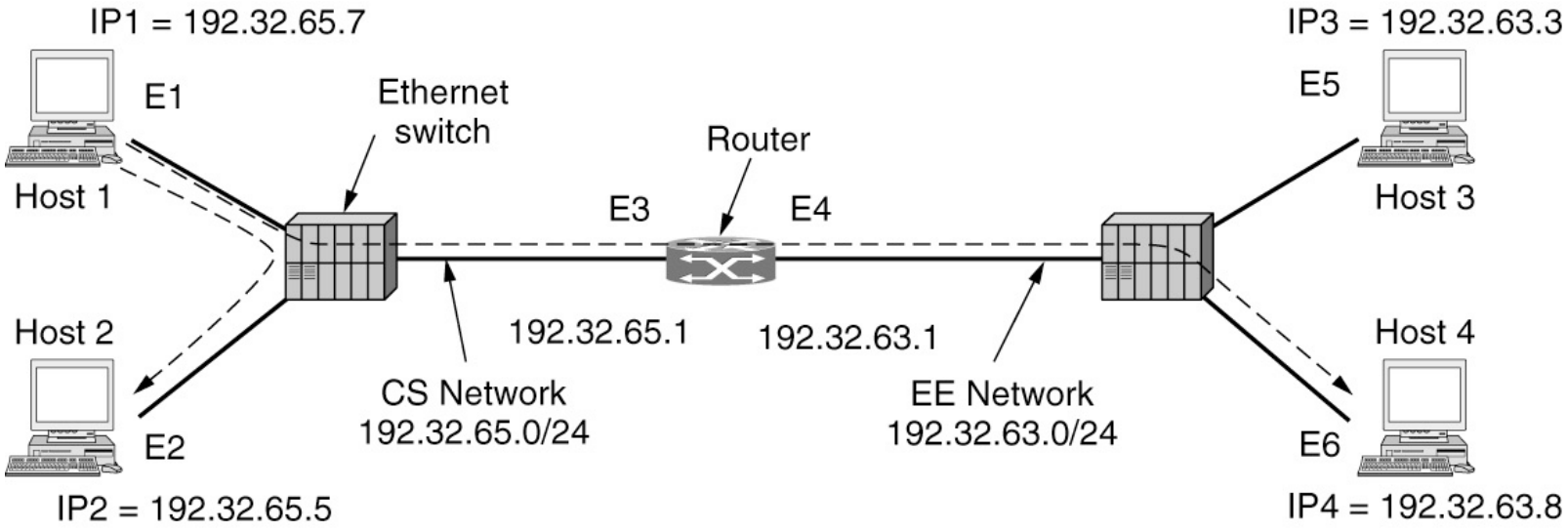
- Fragmentation Doesn't Work
  - Well, we knew that.
- Header Extension Options Don't Work
  - Two identifier registries using the same field (sigh)
- Using MAC Address for lower 64 is a security hole.
  - Nothing like putting your identity in every packet! NSA must love that!
- Using MAC Address in the IPv6 address makes address route-dependent
  - Which we have known about since the early 80s, and that makes
  - Multihoming and Mobility very hard to do.
- Can't route anything greater than a /64.
  - So those other 64 bits really are wasted.
- MTU Discovery doesn't work
  - Not completely useless: Heartbleed made good use of it!
- Interaction of Options and Fragmentation
  - What happens if the options won't fit in the first fragment? (no one knows)
- Will Routing Scale to 4.3 billion IPv4 address spaces?
  - It barely works for one!
- Must have forgotten some!

Internet Control Message Protocol

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

The principal ICMP message types.

ARP—The Address Resolution Protocol



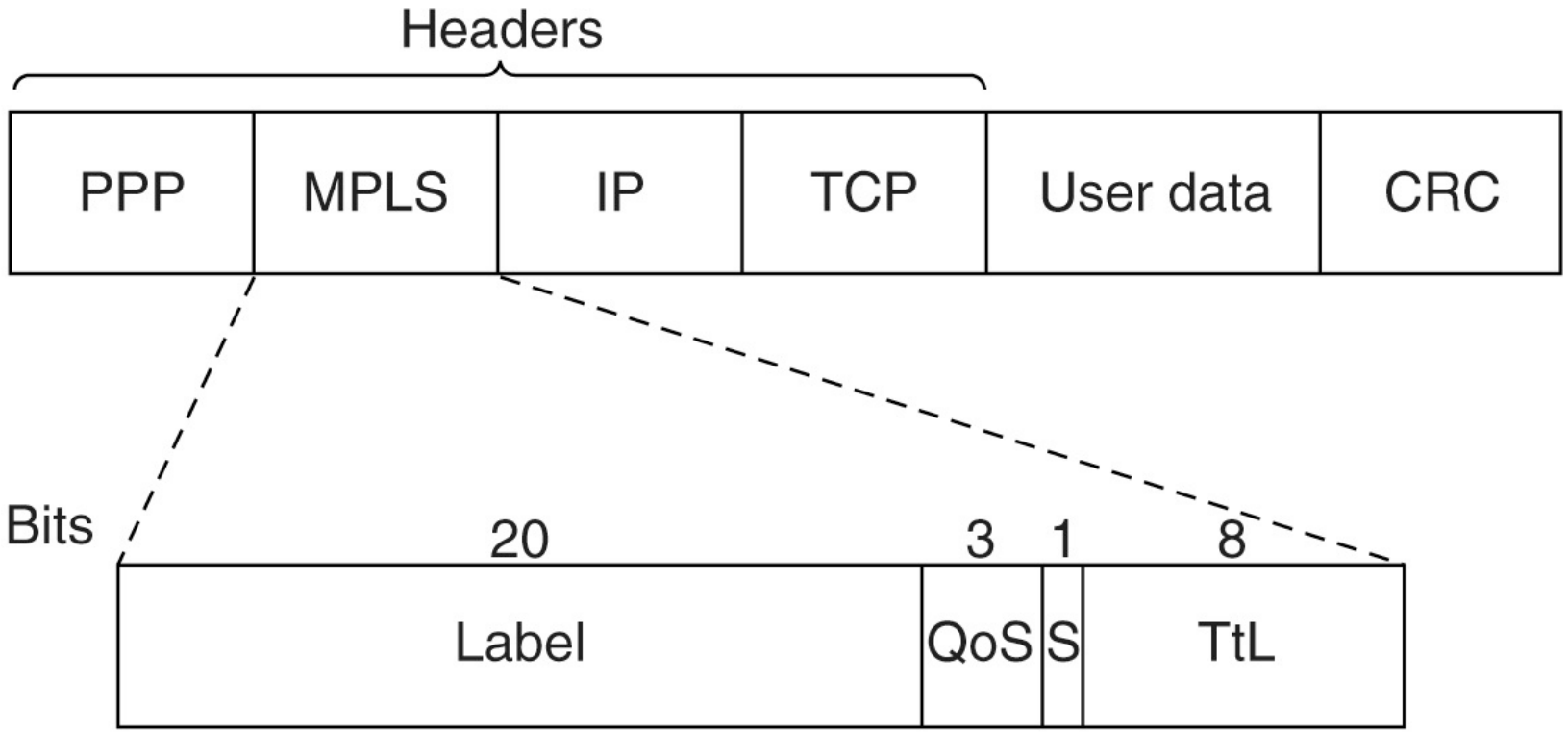
Frame	Source IP	Source Eth.	Destination IP	Destination Eth.
Host 1 to 2, on CS net	IP1	E1	IP2	E2
Host 1 to 4, on CS net	IP1	E1	IP4	E3
Host 1 to 4, on EE net	IP1	E4	IP4	E6

Two switched Ethernet LANs joined by a router

Label Switching and MPLS (1 of 3)

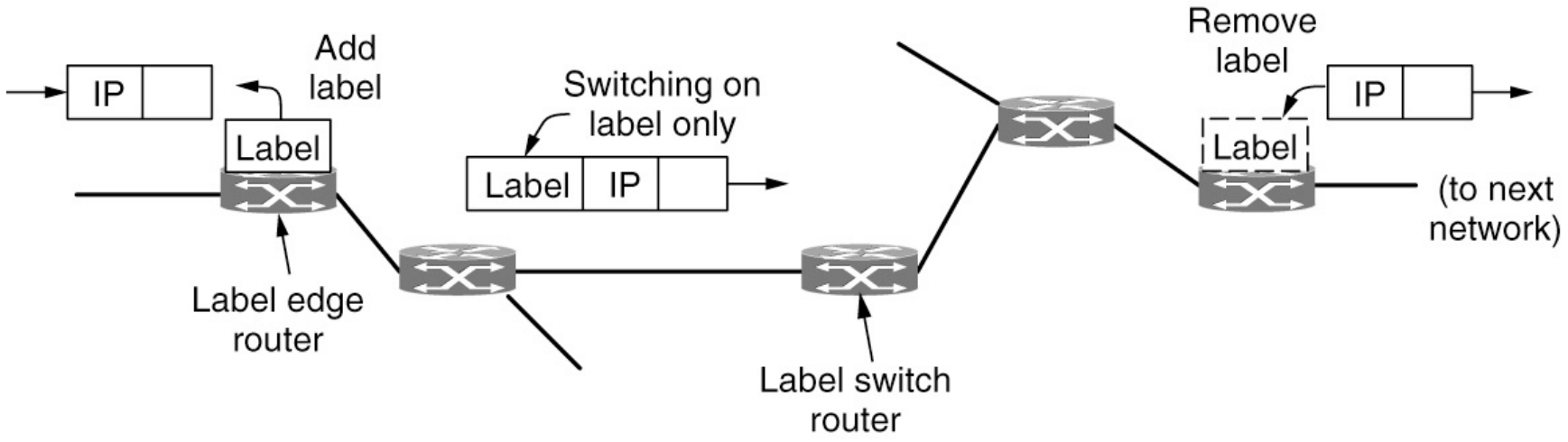
- MPLS (MultiProtocol Label Switching)
  - Perilously close to circuit switching (No, it *is* circuit-switching.)
  - Adds a label in front of each packet
  - Forwards based on the label (not the destination address)
  - Forwarding can be done very quickly
- New MPLS header is added in front of the IP header

Label Switching and MPLS (2 of 3)



Transmitting a TCP segment using IP, MPLS, and PPP

Label Switching and MPLS (3 of 3)



Forwarding an IP packet through an MPLS network

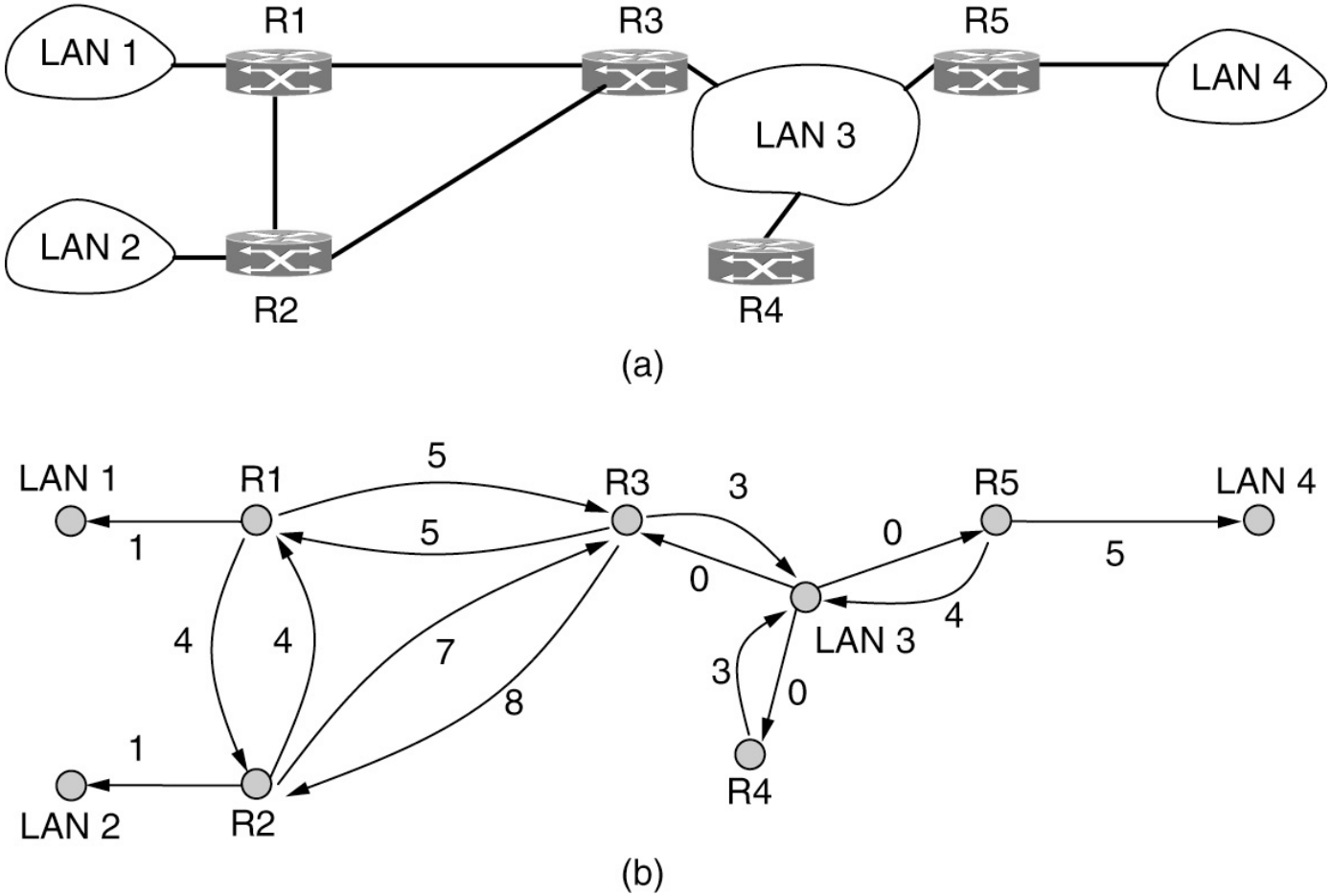
OSPF—An Interior Gateway Routing Protocol (1 of 5)

- Intradomain routing
  - IGP (Interior Gateway Protocol)
- RIP (Routing Information Protocol)
  - Works well in small systems (Distance-Vector)
- OSPF (Open Shortest Path First)
  - Widely used in company networks
  - Difficult to tune and get stable
- IS-IS (Intermediate-System to Intermediate-System)
  - Widely used in ISP networks
  - More stable, also used by IEEE 802.

OSPF—An Interior Gateway Routing Protocol (2 of 5)

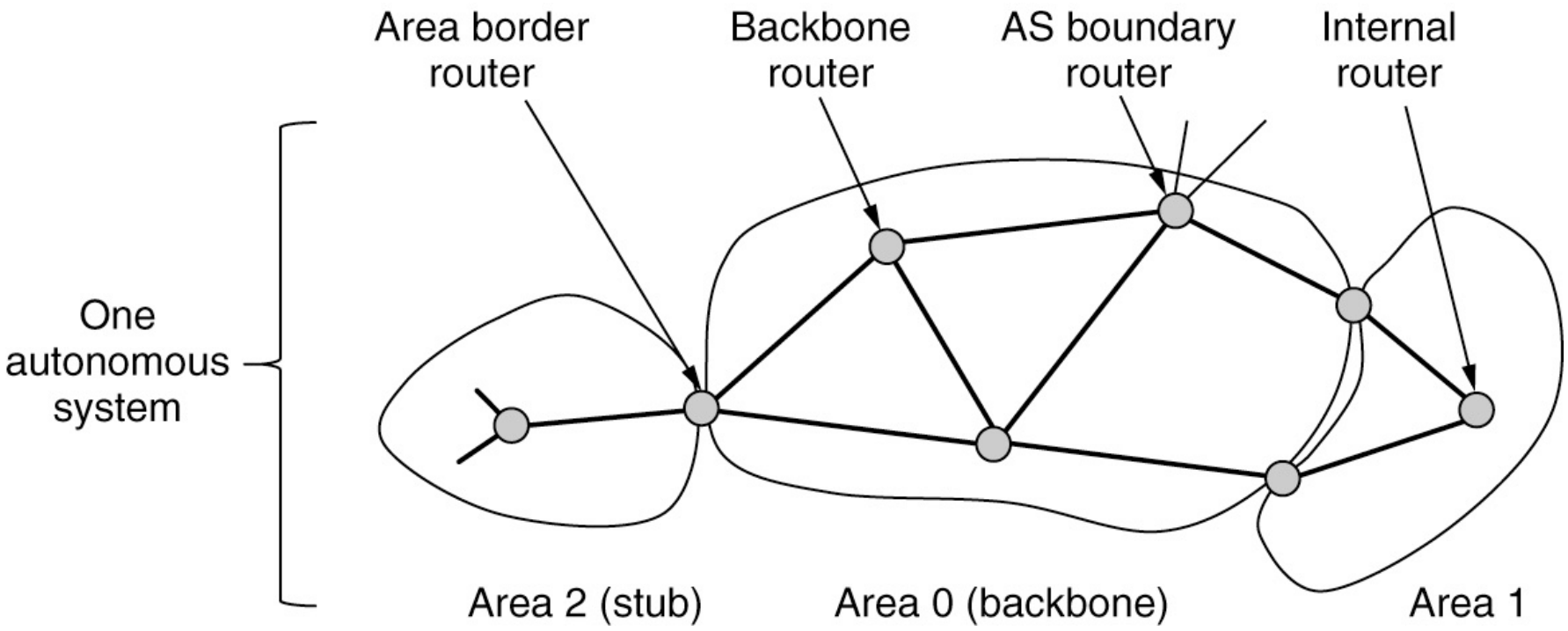
- OSPF
  - Published in the open literature
  - Supports a variety of distance metrics
  - Dynamic
  - Supports routing based on type of service
  - Performs load balancing, splitting the load over multiple lines
  - Supports hierarchical systems
  - Provides security
  - Provision for dealing with routers that were connected to the Internet via a tunnel
- OSPF supports multiaccess networks

OSPF—An Interior Gateway Routing Protocol (3 of 5)



(a) An autonomous system. (b) A graph representation of (a).

OSPF—An Interior Gateway Routing Protocol (4 of 5)



The relation between ASes, backbones, and areas in OSPF

OSPF—An Interior Gateway Routing Protocol (5 of 5)

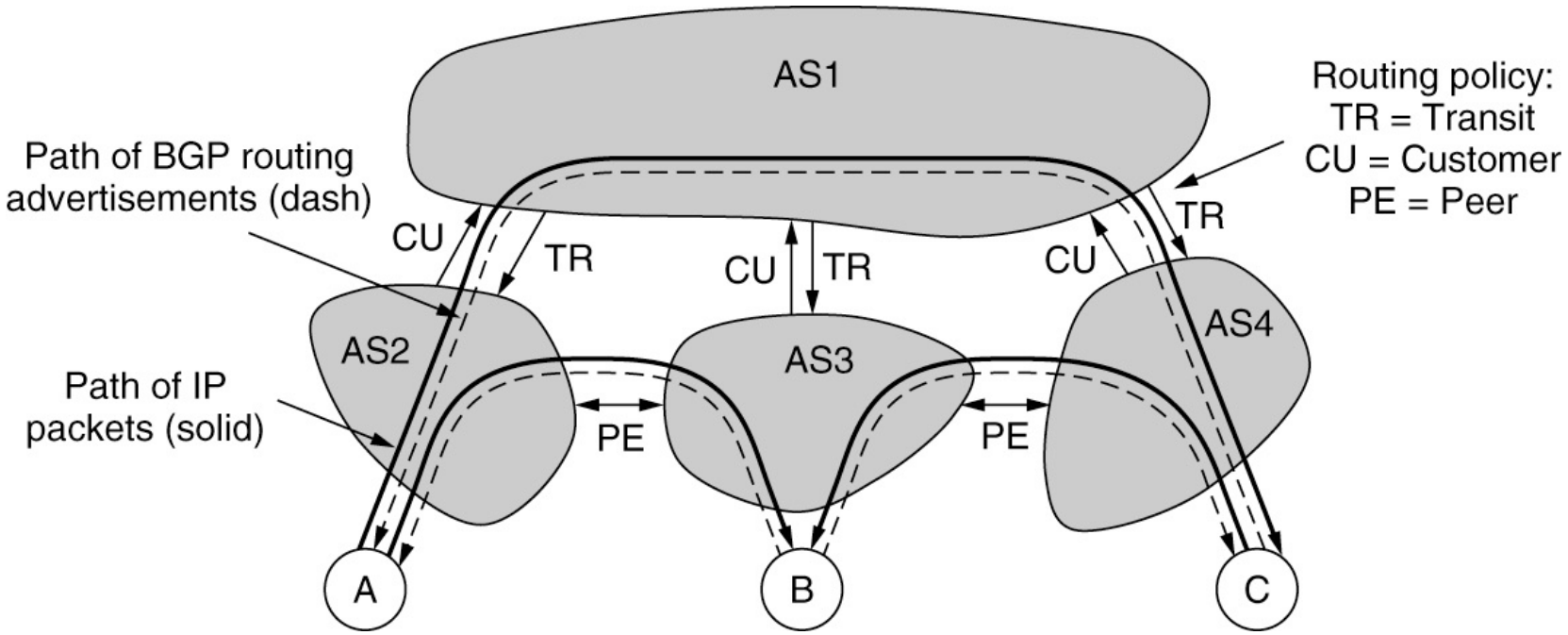
Message type	Description
Hello	Used to discover who the neighbors are
Link state update	Provides the sender's costs to its neighbors
Link state ack	Acknowledges link state update
Database description	Announces which updates the sender has
Link state request	Requests information from the partner

The five types of OSPF messages

BGP—The Exterior Gateway Routing Protocol (1 of 3)

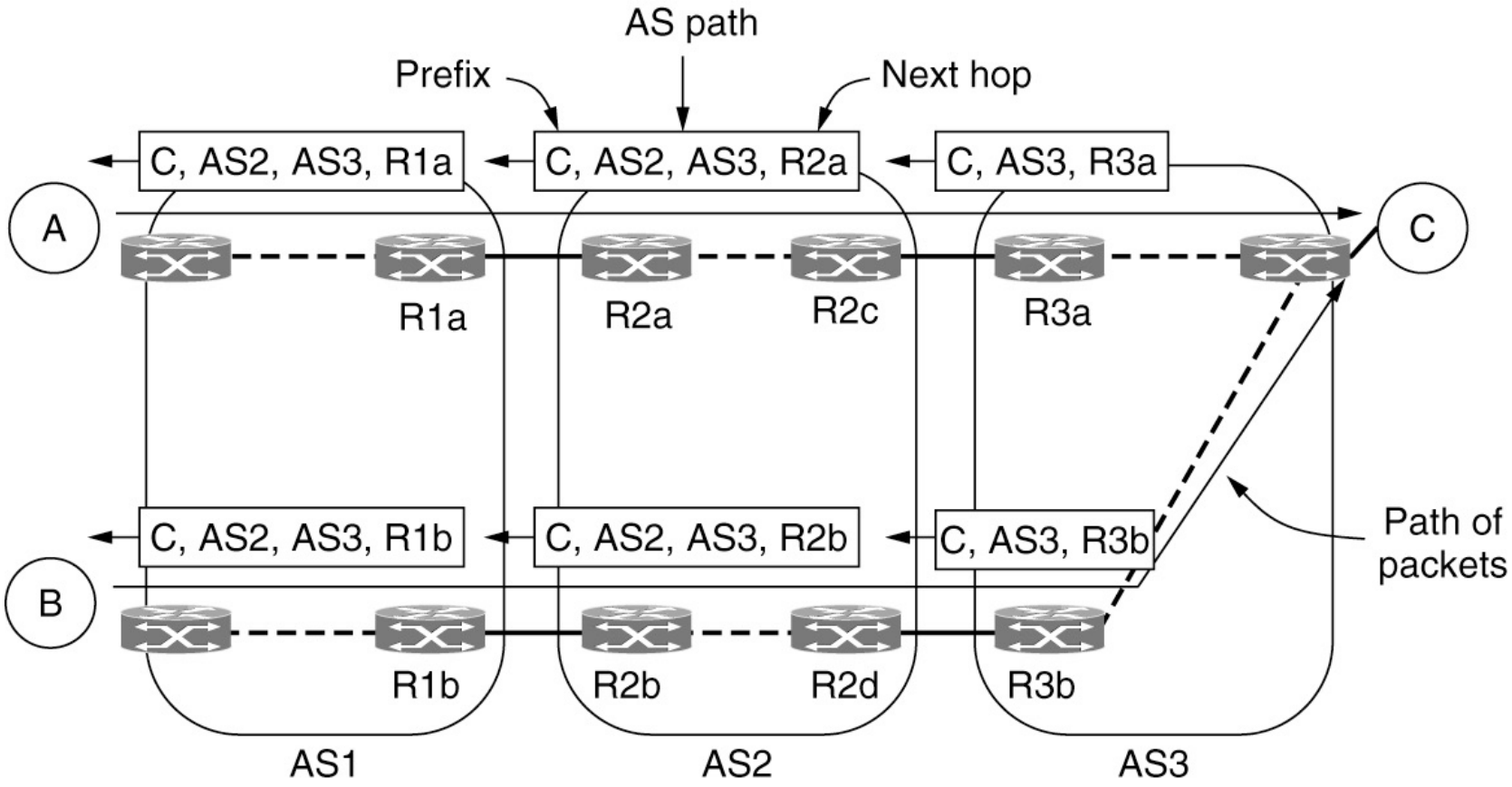
- Possible routing constraints
  - Do not carry commercial traffic on the educational network
  - Never send traffic from the Pentagon on a route through Iraq
  - Use TeliaSonera instead of Verizon because it is cheaper
  - Don’t use AT&T in Australia because performance is poor
  - Traffic starting or ending at Apple should not transit Google

BGP—The Exterior Gateway Routing Protocol (2 of 3)



Routing policies between four autonomous systems

BGP—The Exterior Gateway Routing Protocol (3 of 3)



Propagation of BGP route advertisements

Interdomain Traffic Engineering

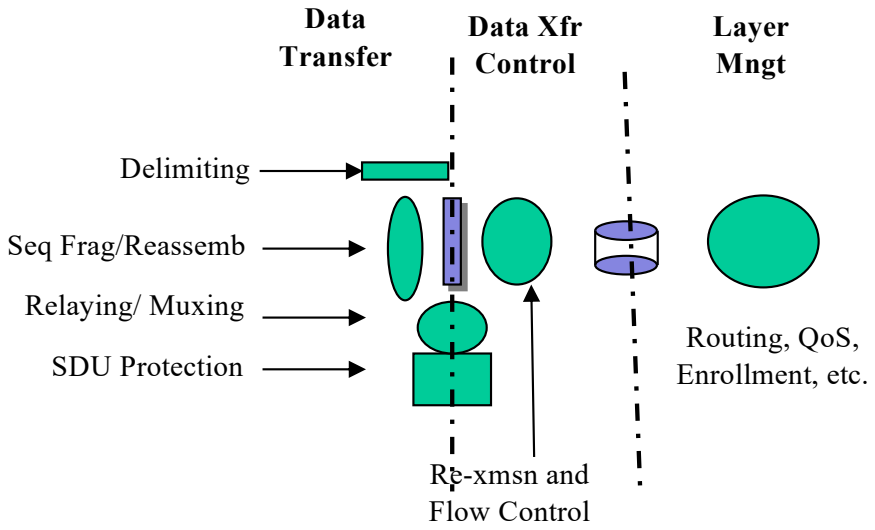
- Tune parameters and configuration network protocols to manage utilization and congestion
- Inbound traffic engineering
  - Selects routes to control how traffic enters the network
  - Set the local preference attribute for individual routes
  - Use AS path prepending
  - Leverage longest prefix match
    - Split a prefix into multiple smaller (longer) prefixes, so that upstream routers prefer the routes with longer prefixes
- Outbound traffic engineering
  - How traffic leaves the network

Internet Multicasting

- Internet multicasting
  - One-to-many communication using class D IP addresses
- Each class D address identifies a group of hosts
  - Actually, a group of IP addresses.
- Twenty-eight bits available for identifying groups
  - Over 250 million groups can exist at the same time
- Process sends a packet to a class D address
  - Best-effort attempt is made to deliver it to all the members of the group addressed, but no guarantees are given

Where Does Routing Go? (1 of 5)

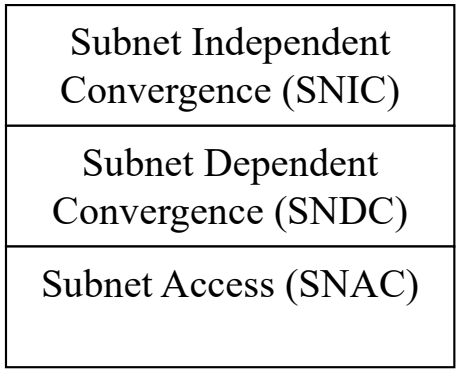
- Already seen the structure for Data Transfer and Data Transfer Control
- Some would say that Routing is an Application.
  - They are right, it *is* an application, but not in the Application Layer
  - Routing is part of the Layer.
  - It is Layer Management, part of the autonomic management.
- The Components of a layer entity, or IPC Process now look like this:

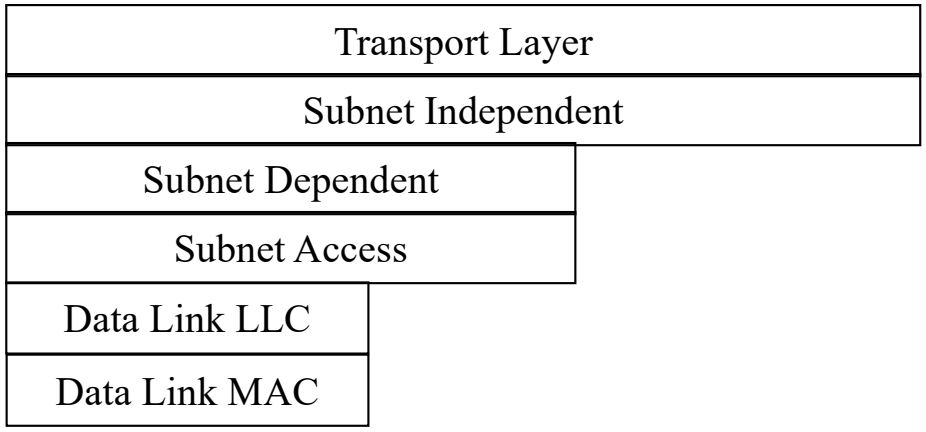


This is precisely the system structure we want to see emerge. Where fast cycle time tasks (Data Transfer) consisting of simple functions are decoupled through a state vector from more complex tasks (Data Transfer Control) with longer cycle time are decoupled through a state vector (routing database) from still more complex tasks (Layer Management).

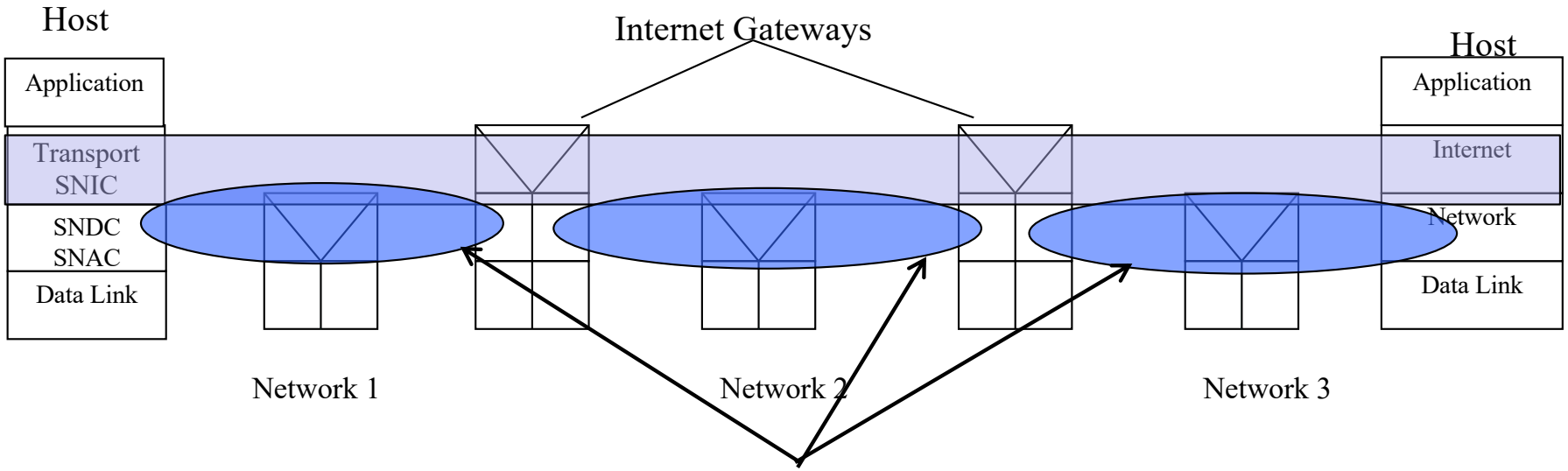
Where Does Routing Go? (2 of 5)

- But Why is it Layer Management?
- Remember OSI rediscovered the INWG model but had to Partition the Network Layer into 3 sublayers,
- The idea was to do an overlay over different networks.
- Which yields layers of different scope.





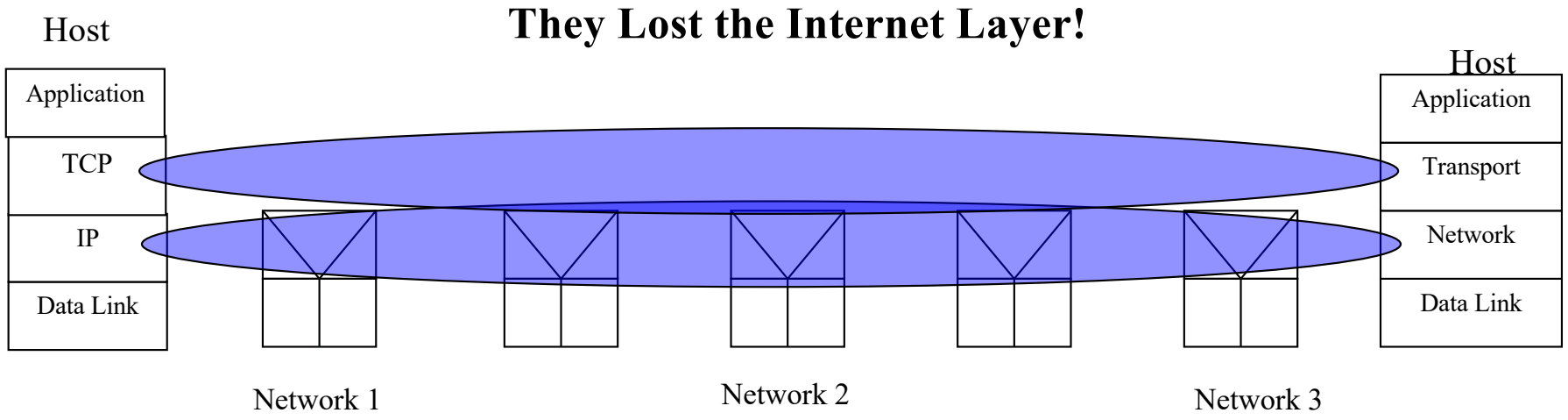
Where Does Routing Go? (3 of 5)



- What Does Intra-Domain Routing (IS-IS, OSPF) Use to Do Its Updates?
  - The Data Link Layer
  - So Intra-Domain Routing is Layer Management of the *Network Layer*.
- What Does Inter-Domain Routing (BGP) Use to Do Its Updates?
  - Network Layer, Using TCP as Subnet Dependent Convergence
  - So Inter-Domain Routing is Layer Management of the *Internet Layer*.
- Network Routing is done in the Network Layer
- Internetwork Routing is done in the Internetwork Layer
- Then Why is This So Complicated?

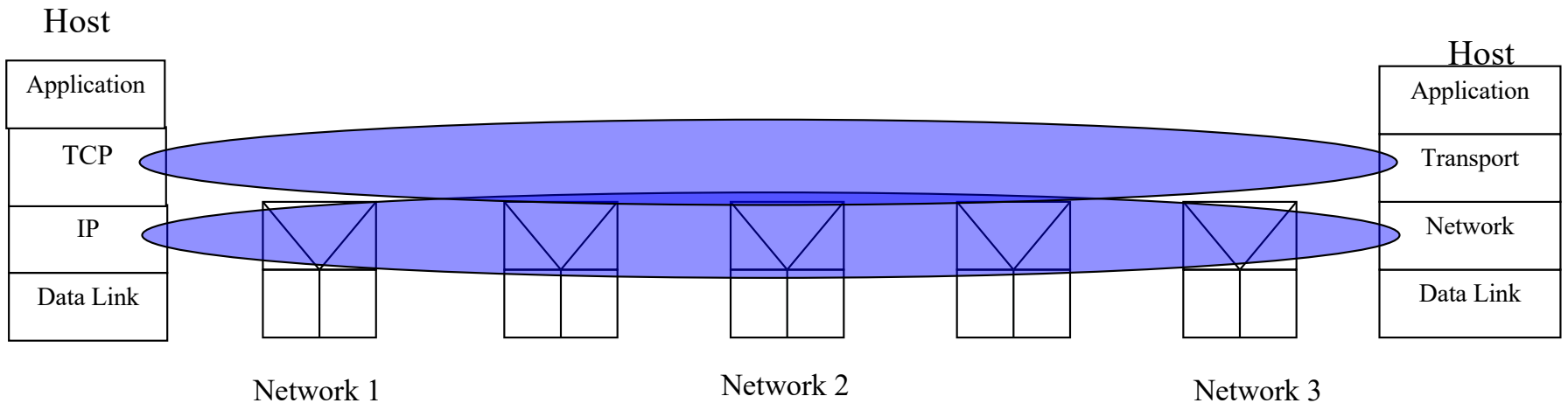
Where Does Routing Go? (4 of 5)

- Why is it so complicated?
- Long Story, But the Internet didn't know about the INWG or OSI results
  - An attitude developed that if OSI did X, the Internet wouldn't, even if it was right. "All we need is rough consensus and running code!"
  - Primarily, not doing an architecture. No plan for where they were going.
  - By 1983, the term "Internet Gateway" disappeared and was replaced by "Router."
  - They didn't really like layers but could never get away from them, so the result is:



- Different networks are not distinguished by the architecture, just administratively.
- All of this raises several questions:

Where Does Routing Go? (5 of 5)



- Why is the Internet Protocol in the Network Layer? Where are the Network Addresses? What about the two levels of routing?
- So the guys who had re-discovered the INWG structure did it anyway without the layer structure, very few people knew the details anyway.
  - Just as good, right?
  - No, this solves one problem and nothing else.
  - A solution that leverages the architecture solves this problem and several others that hadn't arisen yet at no additional cost.
  - Worse the kludge precludes the general solution.

Copyright