Introduction to Computer Networks

COSC 4377

Lecture 18

Spring 2012

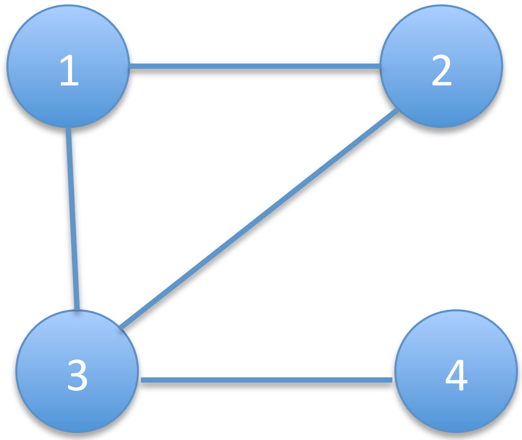
March 28, 2012

Announcements

- HW8 due this week
- HW9 out
- Student presentations

HW8

- Distance Vector Routing
- Count-to-infinity
- Split-horizon



HW9

- Capture packets using Wireshark
- Plot CDF

- Packet size: 2, 4, 4, 5, 6, 6, 6, 3

CDF

- Packet size: 2, 4, 4, 4, 4, 5, 4, 6, 3

Packet Size	Num. Pkts	Cum. Pkts
2		
3		
4		
5		
6		

CDF

- Packet size: 2, 4, 4, 4, 4, 5, 4, 6, 3

Packet Size	Num. Pkts	Cum. Pkts
2	1	
3	1	
4	5	
5	1	
6	1	

CDF

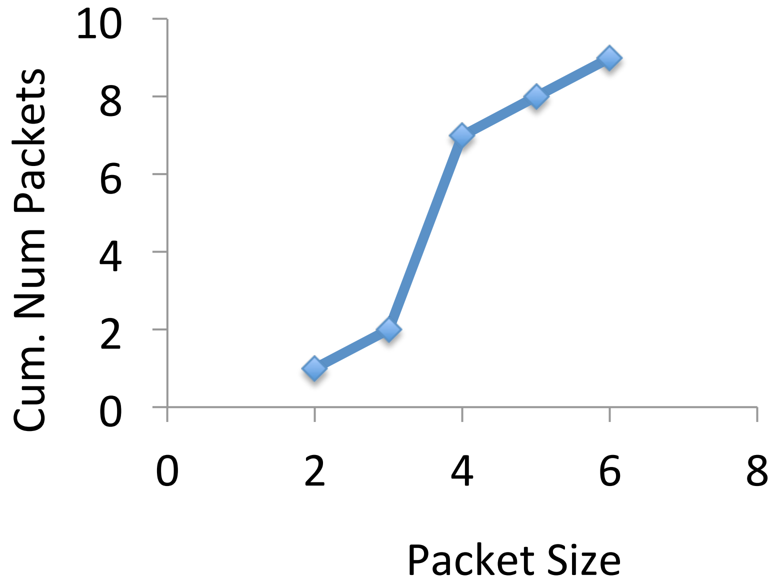
- Packet size: 2, 4, 4, 4, 4, 5, 4, 6, 3

Packet Size	Num. Pkts	Cum. Pkts
2	1	1
3	1	2
4	5	7
5	1	8
6	1	9

CDF

- Packet size: 2, 4, 4, 4, 4, 5, 4, 6, 3

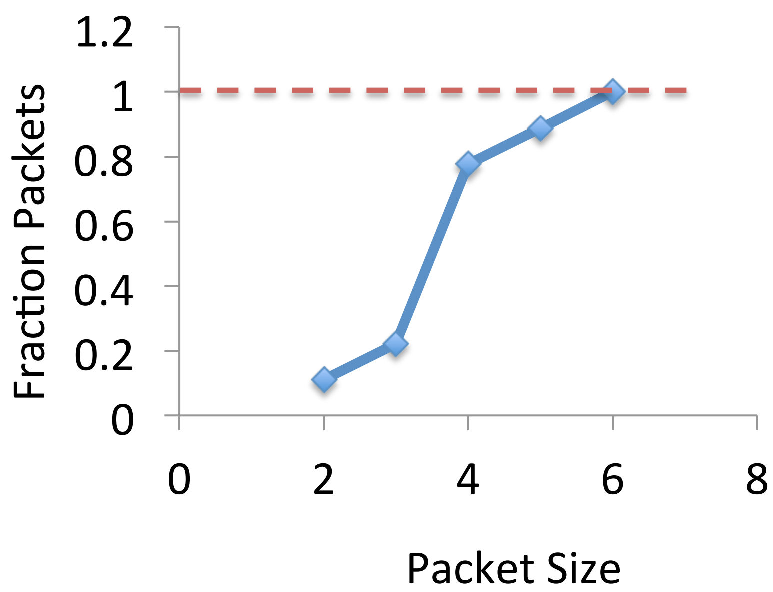
Packet Size	Num. Pkts	Cum. Pkts
2	1	1
3	1	2
4	5	7
5	1	8
6	1	9



CDF

- Packet size: 2, 4, 4, 4, 4, 5, 4, 6, 3

Packet Size	Num. Pkts	Cum. Pkts
2	1	1
3	1	2
4	5	7
5	1	8
6	1	9



Today’s Topics

- BGP example
- NAT
- Link Layer

Obtaining IP Addresses

- Blocks of IP addresses allocated hierarchically
  - ISP obtains an address block, may subdivide

  ISP: 128.35.16/20 10000000 00100011 00010000 00000000

  Client 1: 128.35.16/22 10000000 00100011 00010000 00000000

  Client 2: 128.35.20/22 10000000 00100011 00010100 00000000

  Client 3: 128.35.24/21 10000000 00100011 00011000 00000000
- Global allocation: ICANN, /8's (ran out!)
- Regional registries: ARIN, RIPE, APNIC, LACNIC, AFRINIC

Obtaining Host IP Addresses - DHCP

- Networks are free to assign addresses within block to hosts
- Tedious and error-prone: e.g., laptop moving between buildings
- Solution: Dynamic Host Configuration Protocol
  - Client: DHCP Discover to 255.255.255.255 (broadcast)
  - Server(s): DHCP Offer to 255.255.255.255
  - Client: choose offer, DHCP Request
  - Server: DHCP ACK
- Result: address, gateway, netmask, DNS server

We're running out of internet addresses

Don't panic, but we're running out of internet addresses.

Not domain names -- those website names that you see at the top of this page and which always start with some semblance of "http://" and "www."

We've got plenty of those.

But, according to statements from prominent internet thinkers this week, we may run out of internet protocol -- or IP -- addresses in less than a year.

IP addresses are numbers assigned to all of the devices -- computers, phones, cars, wireless sensors, etc. -- that log on to the internet.

According to the blog ReadWriteWeb, the internet is changing and evolving so quickly -- with so many new types of devices connecting -- that we're running out of numbers to assign to all of these Web-enabled electronics.

cnn.com, 7/23/2010

The internet has (kind of) run out of space

On Thursday, the internet as we know it ran out of space.

The nonprofit group that assigns addresses to service providers announced that, on Thursday morning, it allocated the last free internet addresses available from the current pool used for most of the internet's history.

"This is an historic day in the history of the internet, and one we have been anticipating for quite some time," said Raul Echeberria, chairman of the Number Resource Organization.

But fear not. The group has seen this coming for more than a decade and is ready with a new pool of addresses that it expects to last, well, forever.

John Curran, CEO of the American Registry for Internet Numbers, said the old pool of Internet Protocol addresses had about 4.3 billion addresses.

"A billion sounds like a lot," Curran said Thursday morning. "But when you think that there's nearly 7 billion people on the planet, and you're talking about two, three, four, five addresses per person (for some Web users), obviously 4.3 billion isn't enough."

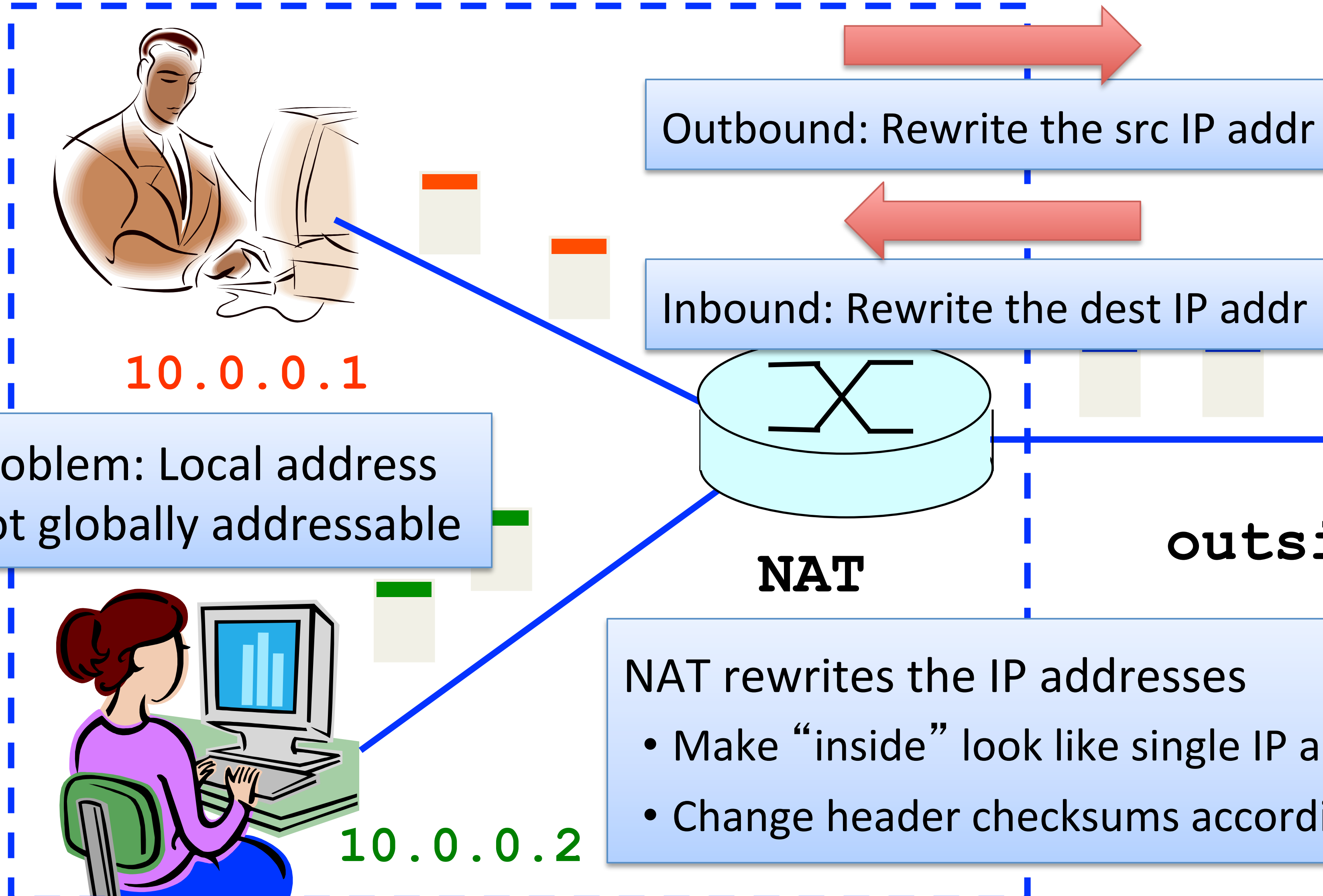
cnn.com, 2/3/2011

The Last 5 Allocations

102/8	AfriNIC	2011-02	whois.afrinic.net	ALLOCATED
103/8	APNIC	2011-02	whois.apnic.net	ALLOCATED
104/8	ARIN	2011-02	whois.arin.net	ALLOCATED
179/8	LACNIC	2011-02	whois.lacnic.net	ALLOCATED
185/8	RIPE NCC	2011-02	whois.ripe.net	ALLOCATED

- IP addresses: 2^{32} is only 4 billion
- How do we connect devices if we run out of IP addresses?
  - IPv6
  - Other solutions?

Network Address Translation



Outbound: Rewrite the src IP addr

Inbound: Rewrite the dest IP addr

Problem: Local address not globally addressable

NAT rewrites the IP addresses

- Make "inside" look like single IP addr
- Change header checksums accordingly

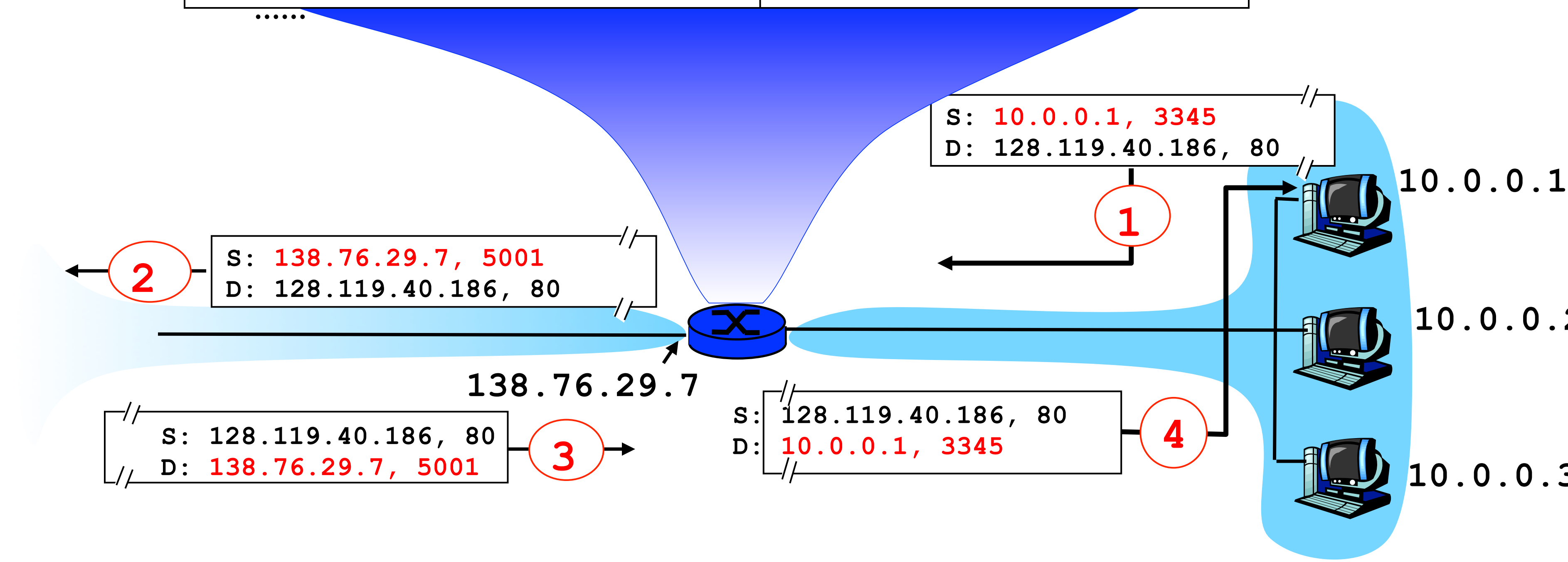
From J. Rexford slides

Port-Translating NAT

- Two hosts communicate with same destination
  - Destination needs to differentiate the two
- Map outgoing packets
  - Change source address and source port
- Maintain a translation table
  - Map of (src addr, port #) to (NAT addr, new port #)
- Map incoming packets
  - Map the destination address/port to the local host

Network Address Translation Example

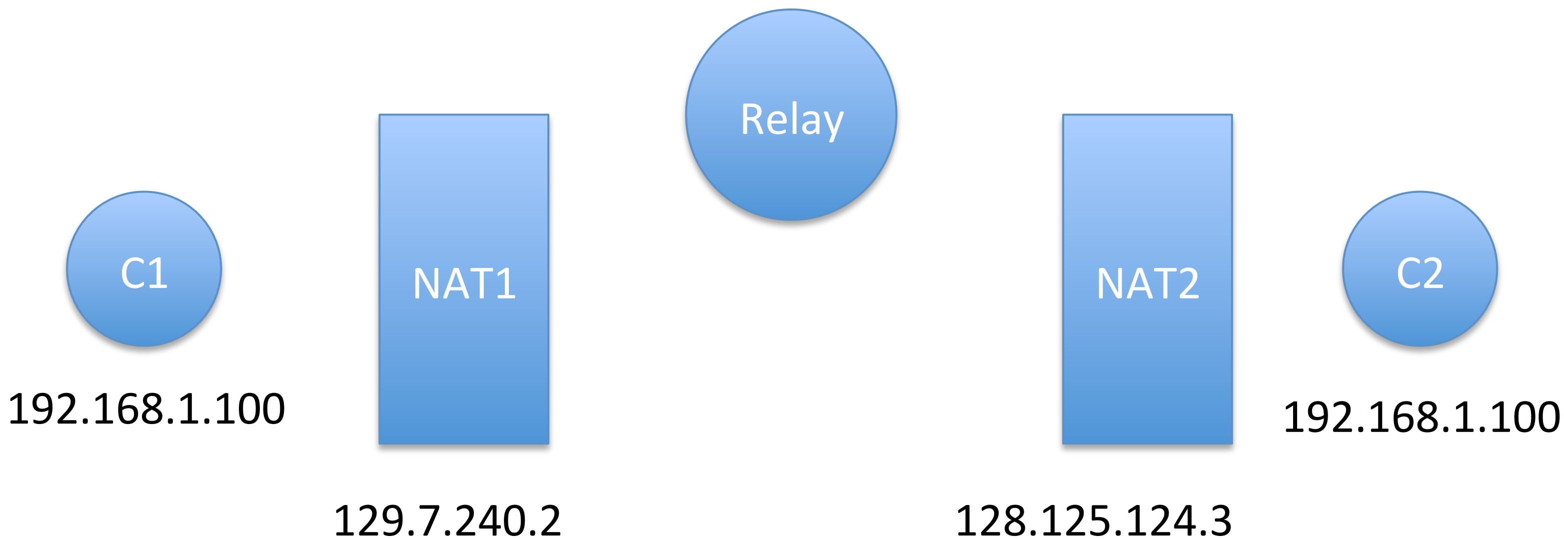
NAT translation table	
WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
……	



Maintaining the Mapping Table

- Create an entry upon seeing an outgoing packet
  - Packet with new (source addr, source port) pair
- Eventually, need to delete entries to free up #'s
  - When? If no packets arrive before a timeout
  - (At risk of disrupting a temporarily idle connection)
- An example of *Soft State*
  - I.e., removing state if not refreshed for a while

P2P connection across NATs



NAT Traversal

- How do we connect to “servers” behind a NAT?

http://en.wikipedia.org/wiki/NAT_traversal

IPv6

- Address space 128 bits

- Other features
  - Multicast
  - Stateless addressing

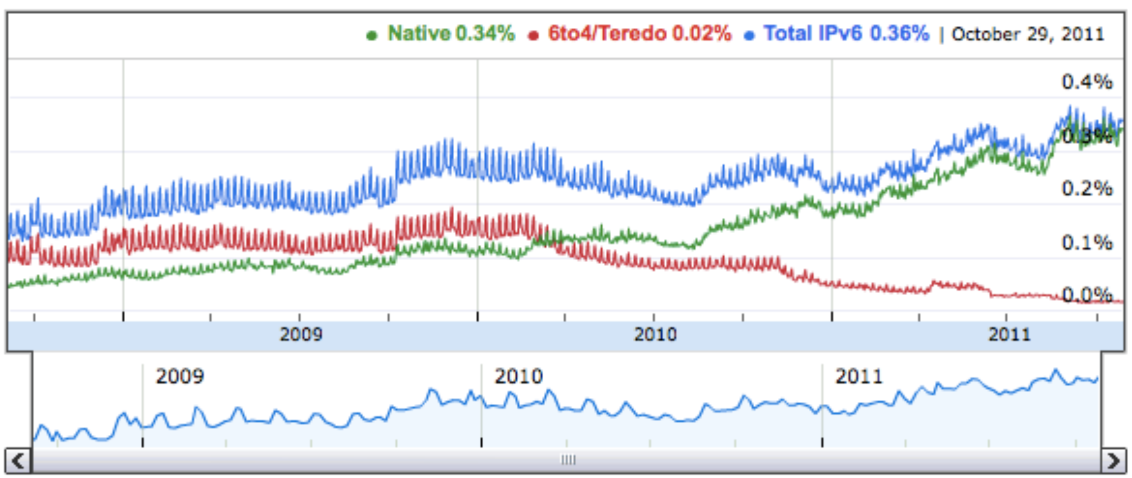
IPv6 adoption

IPv6 Deployment Growth

Of the 39,570 networks in the world running BGP, the number running IPv6 has increased to 4,830, or 12.2%. This is an increase from 7.4% just one year ago or 9.5% six months back.

The global IPv6 routing table has passed 7000 IPv6 prefixes.

Hurricane Electric, 11/13/2011



google.com

Link Layer

- Error Detection
- Reliability
- Media Access
- Ethernet

Error Detection

- Idea: add redundant information to catch errors in packet
- Used in multiple layers
- Three examples:
  - Parity
  - Internet Checksum
  - CRC

Simplest Schemes

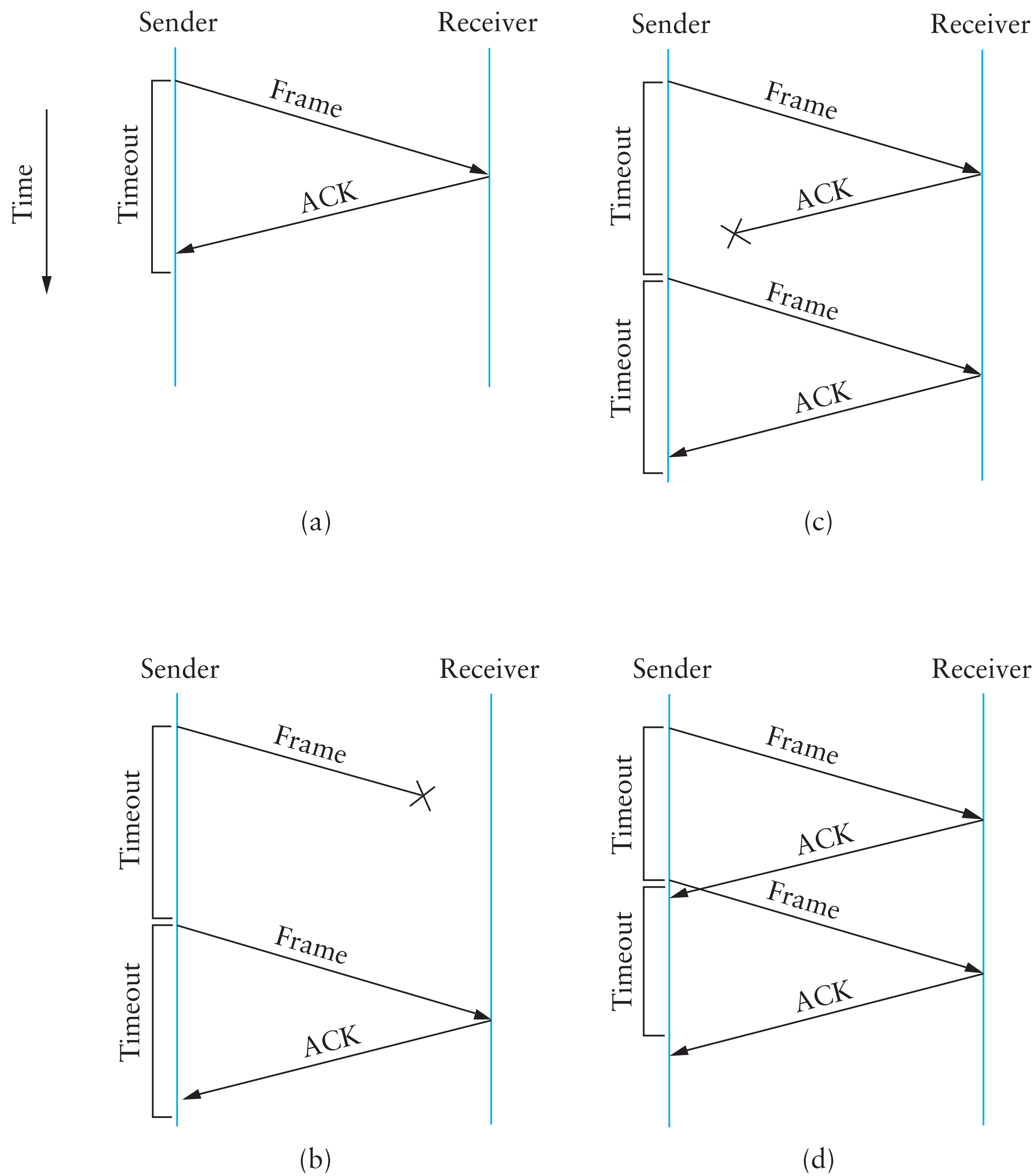
- Repeat frame
  - High overhead
  - Can’t correct error
- Parity
  - Can detect odd number of bit errors
  - No correction

Reliable Delivery

- Error detection can discard bad packets
- Problem: if bad packets are lost, how can we ensure reliable delivery?
  - Exactly-once semantics = at least once + at most once

At Least Once Semantics

- How can the sender know packet arrived *at least once*?
  - Acknowledgments + Timeout
- Stop and Wait Protocol
  - S: Send packet, wait
  - R: Receive packet, send ACK
  - S: Receive ACK, send next packet
  - S: No ACK, timeout and retransmit


Time
Sender
Receiver
Frame
ACK
Timeout
(a)
Sender
Receiver
Frame
Frame
ACK
Timeout
Timeout
(b)
Sender
Receiver
Frame
ACK
Frame
ACK
Timeout
Timeout
(c)
Sender
Receiver
Frame
ACK
Frame
ACK
Timeout
Timeout
(d)

Stop and Wait Problems

- Duplicate data
- Duplicate acks
- Can’t fill pipe (remember bandwitdh-delay product)
- Difficult to set the timeout value