COMP202-08B
Computer Communications

Lecture 14

TCP part 2

11 September 2008

TCP

- Transmission control protocol (TCP)
- TCP is known as a *transport protocol* that is:
  - reliable,
  - connection-oriented,
  - stream-based

- When you use Socket / ServerSocket in Java, you use TCP as your transport protocol.
- TCP is implemented in the operating system kernel, not the application software.


TCP features

- Connection-oriented
  - Applications establish connections between computers using TCP (port numbers)
  - No broadcast or multicast mechanism
- Full duplex
  - Both client and server can send information to each other
- Reliable
  - Lost messages get retransmitted
  - Messages are protected using a (weak) checksum
- Network friendly
  - Avoids overwhelming the network's ability to forward information
  - Tries to be fair to others using the network at the same time
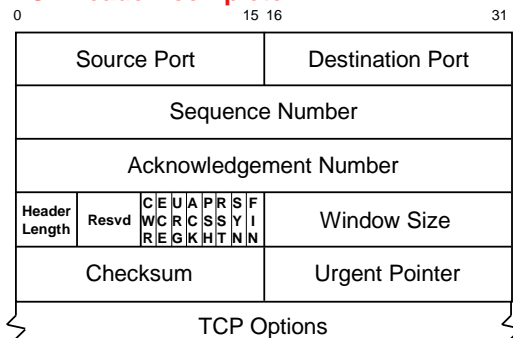

TCP
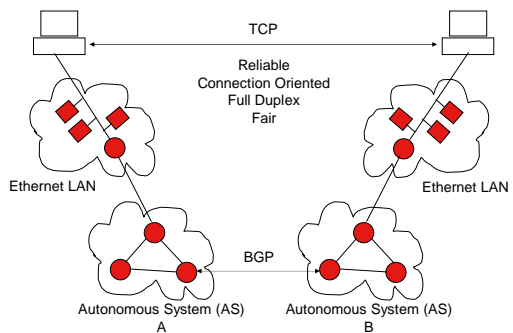
Encapsulation: TCP + Internet model
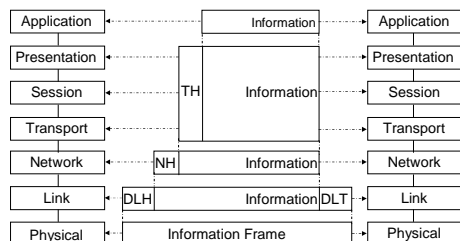
- TCP + Sockets cover the presentation, session, and transport layers
- Difficult to build an efficient network stack using the 7-layer model

---

TCP header: complete

1

TCP header: needed for 202

0		15 16							31
Source Port			Destination Port						
Sequence Number									
Acknowledgement Number									
			A C K	R S T	S Y N	F I N	Window Size		
Checksum									
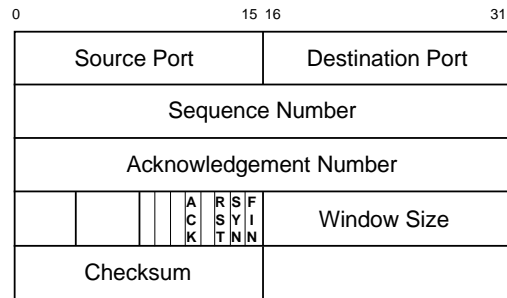
TCP header: destination port
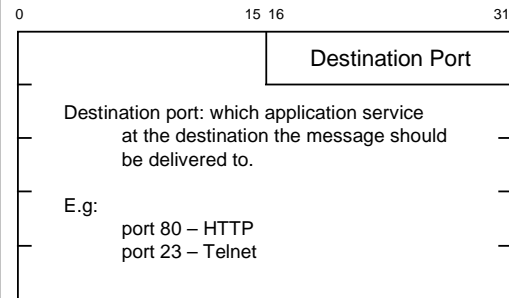
0	15 16	31
		Destination Port

Destination port: which application service
at the destination the message should
be delivered to.

E.g:
port 80 – HTTP
port 23 – Telnet

TCP header: source port

0	15 16	31
Source Port		

Source port: port number where the message
was sent from

Source port is often dynamically assigned when
a Socket is created
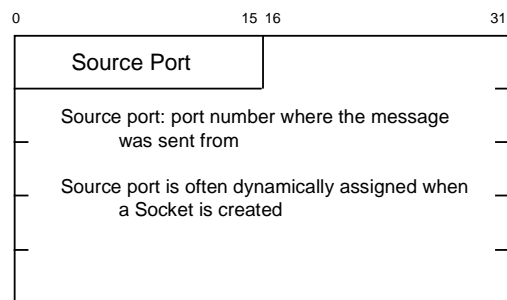
Why two port values?

- Allows a client to have multiple connections to a
destination application service simultaneously
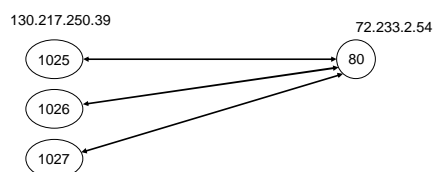 - This often occurs when web browsing – fetch multiple page
elements at the same time
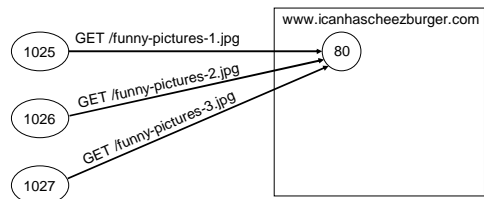
www.icanhascheezburger.com

1025 — GET /funny-pictures-1.jpg → 80
1026 — GET /funny-pictures-2.jpg
1027 — GET /funny-pictures-3.jpg

TCP connections

- A TCP connection can be uniquely identified by:
  - Source IP address
  - Destination IP address
  - Source port
  - Destination port

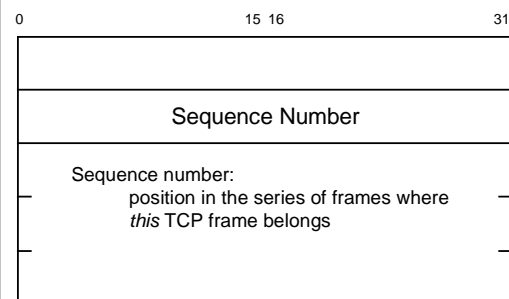130.217.250.39   72.233.2.54

1025
1026   80
1027

TCP header: sequence number

0	15 16	31
Sequence Number		

Sequence number:
position in the series of frames where
this TCP frame belongs

2

TCP sequence numbers

```
 1   2   3   4   5   6   7   8   9  10  11
 ↓   ↓   ↓   ↓   ↓   ↓   ↓   ↓   ↓   ↓   ↓
 h   e   l   l   o       w   o   r   l   d
```
↑
12
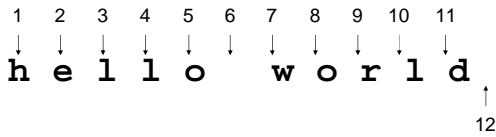
TCP sequence numbers

Message: hello world   Message: hello world

Sequence numbers allow the data in a series of frames to be ordered correctly at the receiver.
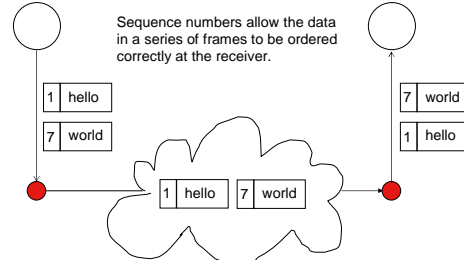
| 1 | hello |
| 7 | world |

| 7 | world |
| 1 | hello |

| 1 | hello | 7 | world |

TCP header: acknowledgement number

0   15 16   31

Acknowledgement Number

Acknowledgement number:
    the next byte the receiver expects
    cumulatively acknowledges previous data
    as successfully received
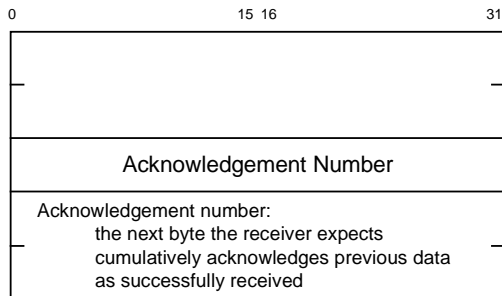
TCP acknowledgement numbers

Message: hello world
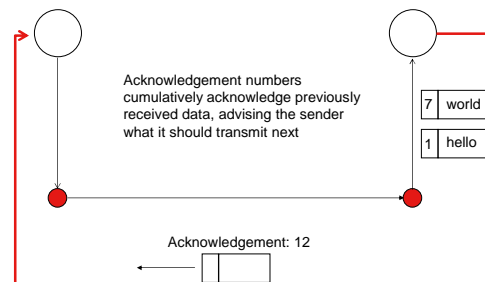
Acknowledgement numbers cumulatively acknowledge previously received data, advising the sender what it should transmit next

| 7 | world |
| 1 | hello |

Acknowledgement: 12

TCP header: flags

0   15 16   31

ACK: value in acknowledgement field is valid
RST: reset connection
SYN: synchronise sequence numbers
          used to establish new connection
FIN: no more data from sender to transmit

| A C K | R S T | S Y N | F I N |
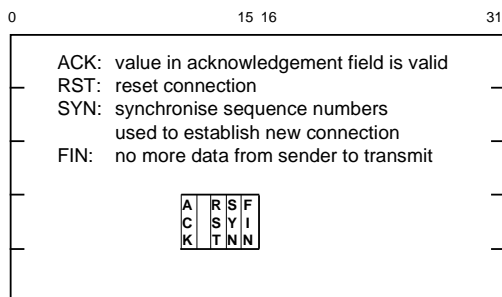
TCP header: window size

0   15 16   31

Window size:
    Receiver specifies how much space in its
    receive buffer remains.
    Sender should not transmit more than the
    receiver could cope with.
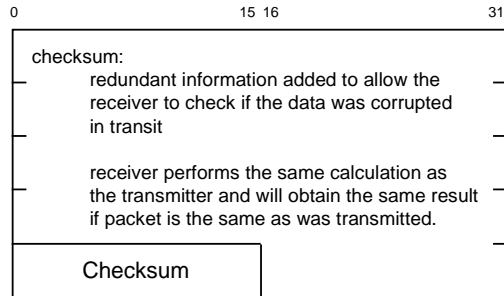
Window size

TCP header: checksum

0   15 16   31

checksum:

      redundant information added to allow the receiver to check if the data was corrupted in transit

      receiver performs the same calculation as the transmitter and will obtain the same result if packet is the same as was transmitted.

Checksum

Internet checksums

- Checksums in most Internet protocols are simple to compute but relatively weak.
- Core computation: sum of 16 bit words.
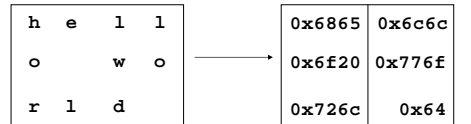
h	e	l	l
o		w	o
r	l	d	

0x6865	0x6c6c
0x6f20	0x776f
0x726c	0x64

Sum of 16 bit words =   0x6865 + 0x6c6c + 0x6f20 + 0x776f + 0x726c + 0x6400

               =   0x00022E30

Internet checksum calculation

Sum of 16 bit words =   0x6865 + 0x6c6c + 0x6f20 + 0x776f + 0x726c + 0x6400

               =   0x00022e30

To obtain 16-bit Internet checksum, we fold the 32-bit value into a 16-bit one as follows:

0x0002 + 0x2e30   =   0x2e32

Then negate:

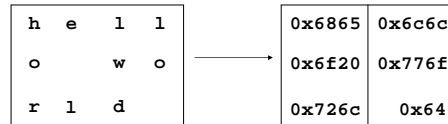0x2e16   = 0010 1110 0011 0010
Negated   = 1101 0001 1100 1101 => 0xd1cd

Internet checksum

h	e	l	l
o		w	o
r	l	d	

0x6865	0x6c6c
0x6f20	0x776f
0x726c	0x64

Checksum: 0xd1cd

First 4 characters swapped in transit:

l	l	h	e
o		w	o
r	l	d	

0x6c6c	0x6865
0x6f20	0x776f
0x726c	0x64
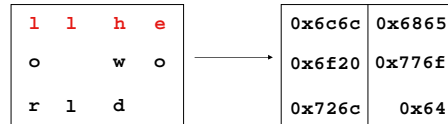
Corruption not detected!   Checksum: 0xd1cd

Establishing a TCP connection

- TCP uses a three-way handshake to agree to a new connection
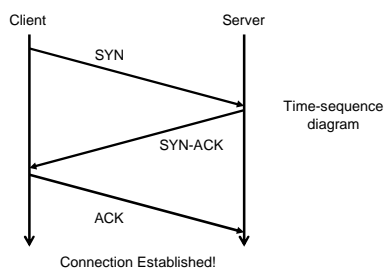
Client   Server

SYN

SYN-ACK

Time-sequence diagram

ACK

Connection Established!

Establishing a TCP connection

Client   Server
Port 1025   Port 80

SYN
1025:80

SYN-ACK
80:1025

ACK
1025:80
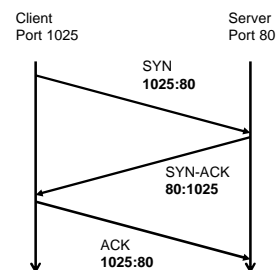
Same slide as previous, but with more detail

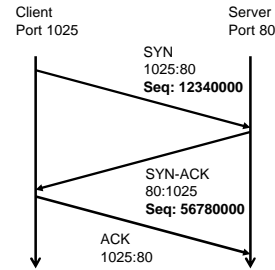## Establishing a TCP connection

- Both ends of the TCP connection need to select an initial sequence number to use
- A random initial sequence number is best
  - Reduces scope for a third-party to hijack a connection by impersonating one of the two parties involved
  - If a system reboots and then happens to choose the same ports to communicate with another system, reduces chance old packets will interfere with new connection

Establishing a TCP connection

Client
Port 1025

Server
Port 80

SYN
1025:80
Seq: 12340000
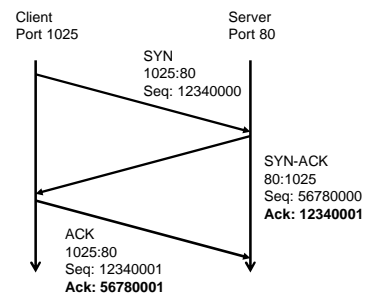
SYN-ACK
80:1025
Seq: 56780000

ACK
1025:80

Same slide as previous, but with more detail
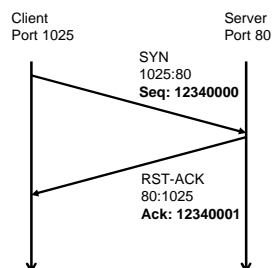
Establishing a TCP connection

- TCP SYN packets are acknowledged as being received!
- The SYN packet is defined to take up one byte in the sequence number space

Establishing a TCP connection

Client
Port 1025

Server
Port 80

SYN
1025:80
Seq: 12340000

SYN-ACK
80:1025
Seq: 56780000
Ack: 12340001

ACK
1025:80
Seq: 12340001
Ack: 56780001

Same slide as previous, but with more detail

TCP "connection refused"

Client
Port 1025

Server
Port 80

SYN
1025:80
Seq: 12340000

RST-ACK
80:1025
Ack: 12340001

Summary

- TCP is a connection-oriented, stream-based, reliable transport protocol
- This lecture:
  - TCP packet format
  - Definition and use-of each field
  - TCP three-way handshake for connection establishment
- Next lectures:
  - Using an established connection
  - Closing a connection
  - Congestion avoidance and control (fairness)