

Internet Engineering

241-461

Robert Elz

kre@munnari.OZ.AU

kre@coe.psu.ac.th

<http://fivedots.coe.psu.ac.th/~kre>

Using Sequence Number

- ◊ Needed for reliability
 - Discovering lost segments
 - Discovering reordered segments
 - Discovering duplicated segments

◊ How ?

Reordered Segments

- ◊ Sequence numbers transmitted in order
 - 100 101 102 103 ... 200 201 202 ... 300 301 302 ...
- ◊ Sequence numbers arriving out of numeric order
 - 100 101 102 103 ... 300 301 302 ... 200 201 202 ...
 - ▷ segments reordered
- ◊ Receiver can easily sort sequence numbers
 - return them to sender order
- ◊ Require sender to transmit only in numeric order

Duplicated Segments

- ◇ Sequence numbers transmitted
 - 100 101 102 103 ... 200 201 202 ... 300 301 302 ...
- ◇ Sequence numbers received
 - 100 101 102 103 ... 200 201 202 ... 100 101 102 103 ... 300 ...
- ◇ Receiver can just ignore second copy
 - remember which numbers have been received
 - not receive them again
 - ▷ until sequence number space circles around

Lost Segments

- ◇ Sequence numbers transmitted
 - 100 101 102 103 ... 200 201 202 ... 300 301 302 ...
- ◇ Sequence numbers received
 - 100 101 102 103 ... 300 301 302 ...
- ◇ Receiver can notice missing sequence numbers
- ◇ Provided sender **MUST** leave no gaps
 - must transmit every sequence number, in order
- ◇ But what to do?
 - Receiver cannot invent missing data
- ◇ And how do we know just not reordered?

Lost Segments (2)

- ◇ Receiver could tell sender
 - I did not receive 200 201 202 ...
- ◇ But consider:
 - Sequence numbers received
 - 100 101 102 103 ... 200 201 202 ...
- ◇ How does receiver know anything is missing ?
 - All numbers received
 - ▷ in correct order
 - Unless receiver knows last number expected
 - ▷ How could it know that?

Lost Segments (3)

- ◇ **Sender knows what was transmitted**
 - So make sender responsible for discovering loss
- ◇ **How does sender know what was received ?**
 - Have receiver inform sender
 - ▷ I did receive 100 101 102 ... 300 301 302 ...
- ◇ **When sender sees missing numbers**
 - Send them again
- ◇ **Requires:**
 - sender must keep a copy of transmitted data
 - until receiver has indicated arrival

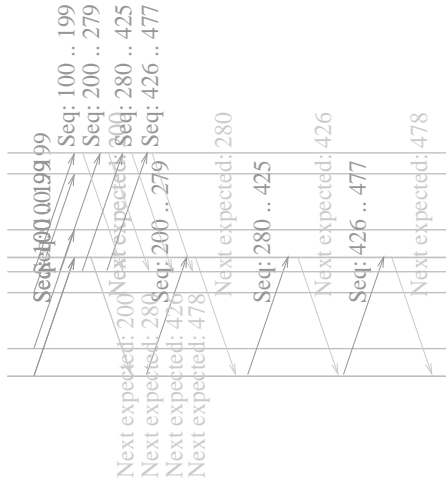
Acknowledging

- ◇ **Receiver can say**
 - I did receive 100 101 102 ... 300 301 302 ...
- ◇ **Then later**
 - I did receive 200 201 202 ... 400 401 402 ...
- ◇ **Then later**
 - I did receive 500, 501, 502, ...
- ◇ **But what if one of those messages is lost ??**
 - data receiver must send it again
 - but how does it know message was lost ??

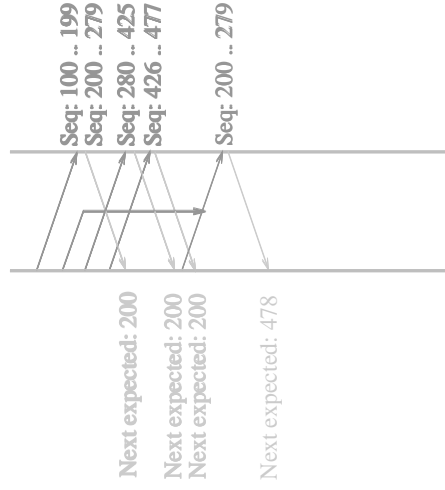
Acknowledging

- ◇ **Easier to say**
 - I did receive 100 101 102 ... 200 201 202 ...
 - 301 302 303 ... 400 401 402 ... 500 501 502 ...
- ◇ **But consider after millions of octets received**
 - acknowledgment message would be very large
- ◇ **Easier to say**
 - I received everything up to 599
- ◇ **or**
 - The first one I am waiting for is 600
- ◇ **TCP does it the last way (next expected is...)**

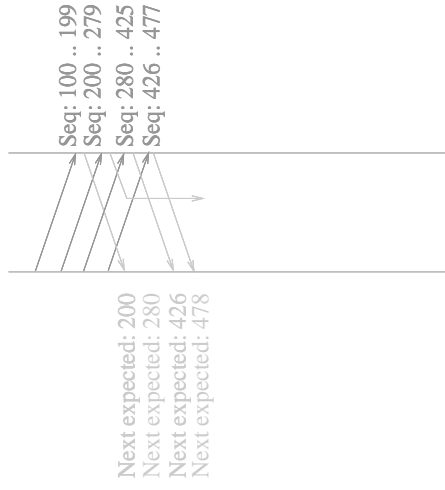
TCP Timeline



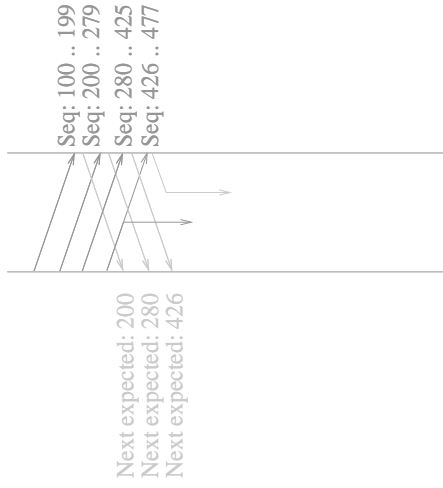
TCP Lost Packets



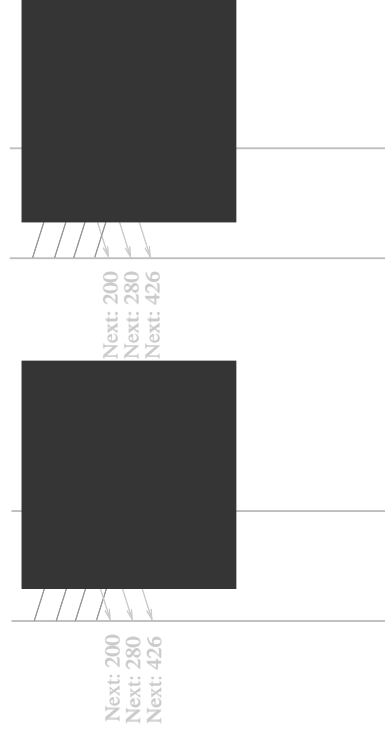
TCP ACK Lost



Losing Last Packet



Losing Last Packet (2)



Losing Last Packet (3)

