

Redes de Computadores

Computer Networks

labs

Stop & Wait Protocol

Lab class 2

Summary

- Brief overview of the Stop and Wait Protocol
- The SW FTP Protocol
- Study files: SwFtpPacket.java & SwFtpServer.java
- Write and test: SwFtpClient.java

Stop & Wait Protocol

- Reliable, general purpose data transfer protocol
- Breaks the file contents into **DATA** packets
- Each data packet has a sequence number
- The receiver uses **ACK**nowledge packets to tell the sender what it has received correctly up to now
- Very simple — the sender only advances to the next data packet when it is sure the receiver received everything previously sent

Stop & Wait Protocol

Sender:

- Assigns to each data packet a (growing) sequence number
- Sends the packet and installs an alarm. Then, if it receives:
 - An ACK packet with the expected sequence number, it progresses to the next data packet
 - An ACK packet with a wrong sequence number is ignored
 - If the timer expires, it retransmits the last sent data packet

Receiver:

- Whenever it receives a data packet:
 - If the packet has the expected sequence number, keeps it, otherwise it is ignored
 - **Always** sends an ACK packet with the sequence number of last correctly received packet

Stop & Wait File Transfer Protocol

- This protocol transfers a file over UDP using the stop & wait protocol
- It is a client / server application, where all initiative is with the client
- It uses several different messages:
 - To start the transfer (read file, write file, op codes 1 and 2)
 - To send data packets (op code 3)
 - To send ACKs (op code 4)
 - To send errors (op code 5)
 - To signal the file is finished (op codes 6 and 7)
- All control messages (read, write, data, finalized) are ACKed
- Sequence numbers of initial packets start at 0, the first data packet at 1

Stop & Wait File Transfer Protocol Messages

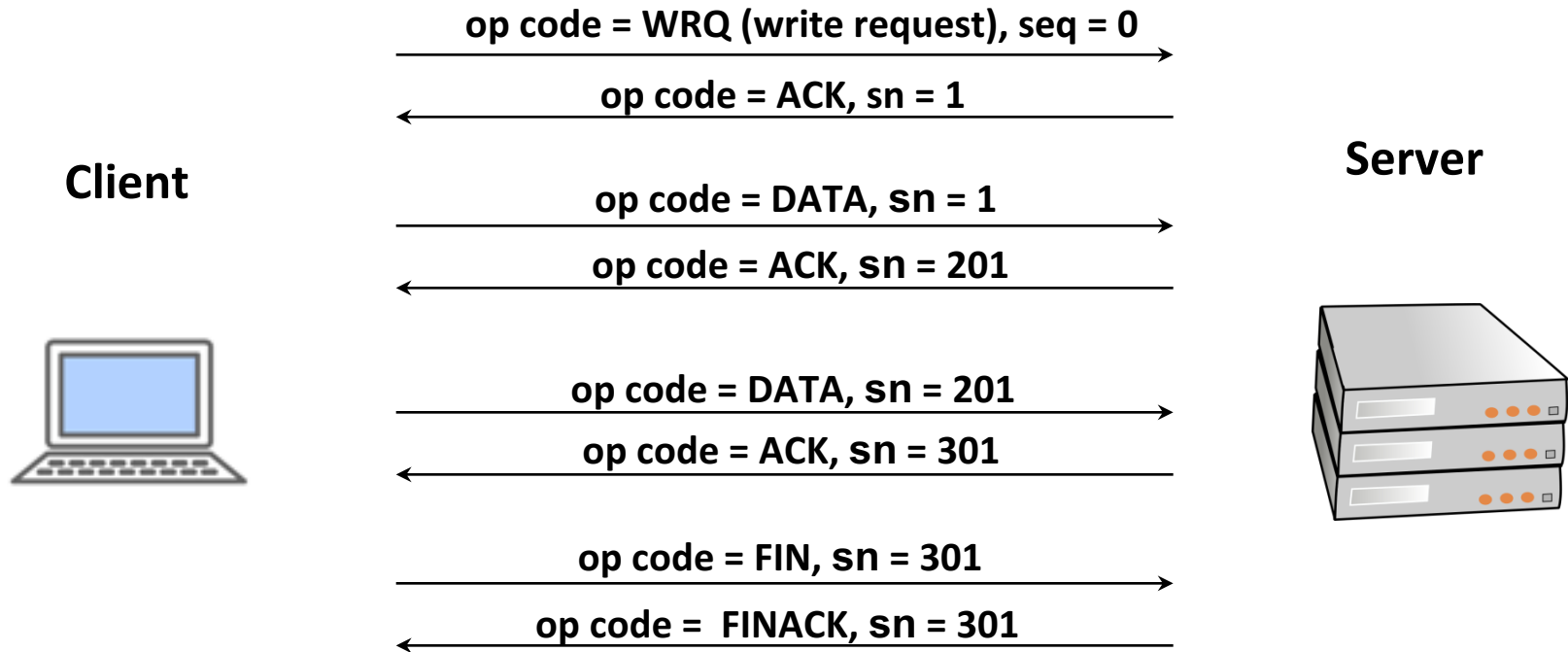
ip header	udp header	op code	File name	0
		op code=3	seq number	data: 0 up to Block Size bytes
		op code=4	seq number	
		op code=5	error number	Error message 0
		op code=6	seq number	
		op code=7	seq number	

- Messages:
 - To start the transfer (read file, write file, op codes 1 and 2)
 - To send data packets (op code 3)
 - To send ACKs (op code 4)
 - To send errors (op code 5)
 - To signal the file is finished (op code 6) or an ack of the receipt of this message (op code 7)
- All control messages (read, write, data, finalized) are ACKed
- Sequence numbers of the sent file start at 1

Some additional details

- Op codes have 16 bits (short) and sequence numbers have 64 bits (long)
- The sequence number corresponds to the position of the first byte of the data block in the file
- The first byte of the file has sequence number 1
- The sequence number of the ACK message corresponds to the position of the next byte to receive (the last byte well received + 1)

Send File Example (Write)



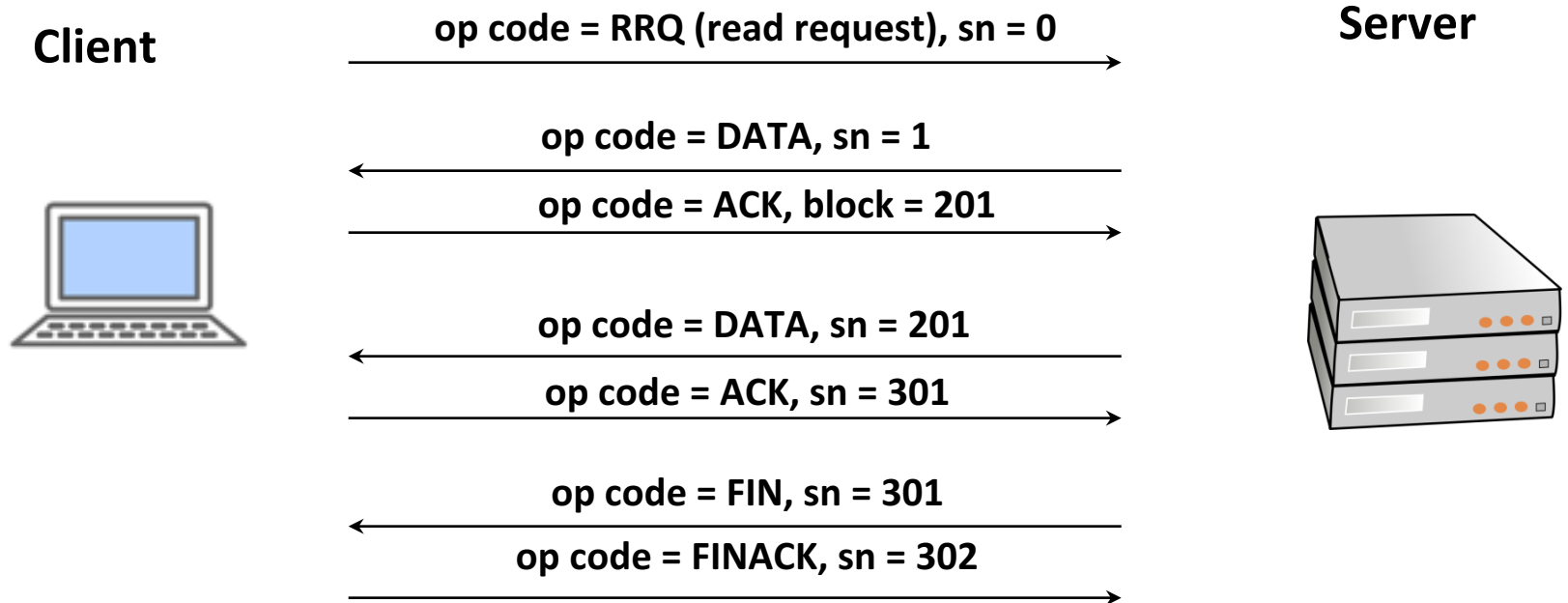
The initial request has sn = 0 and its ACK sn = 1

The first data packet has sn = 1, and has up to the byte 200 (200 bytes), its ACK has sn = 200+1

the second has sn = 201, and has up to the byte 300 (100 bytes), its ACK has sn = 300+1

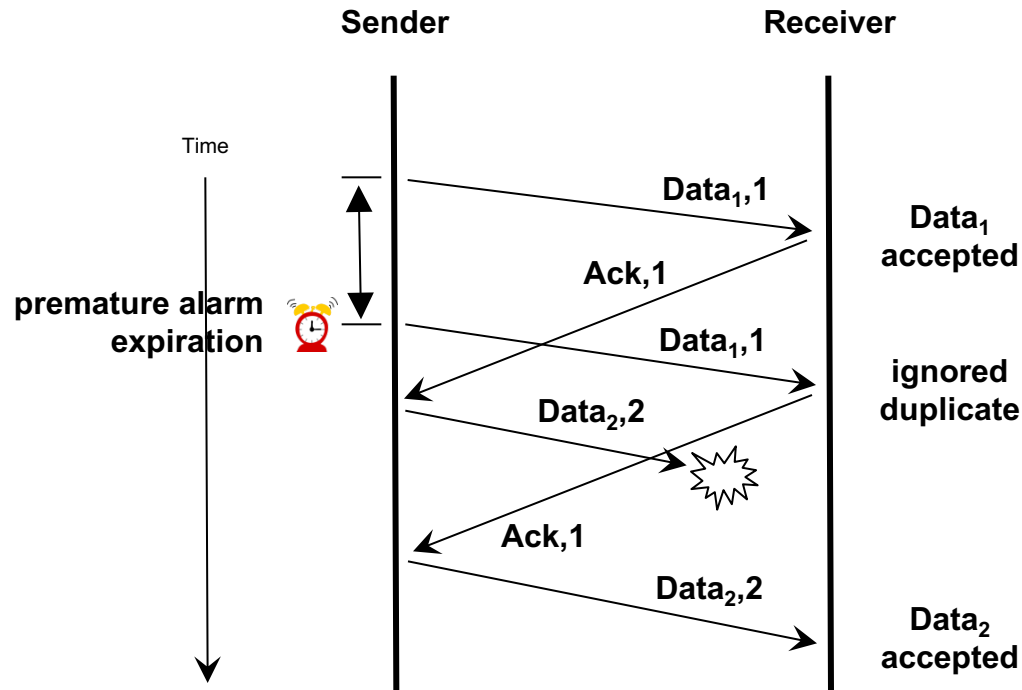
The FIN packet has sn = 301 and its ACK has sn = 301 (the last well received byte + 1)

Get File Example (Read)



The initial request has $sn = 0$ and its ACK, implicitly $sn = 1$, since it is ACKed by the first data packet
The first data packet has $sn = 1$, and has up to the byte 200 (200 bytes), its ACK has $sn = 200+1$
the second has $sn = 201$, and has to the byte 300 (100 bytes), its ACK has $sn = 300+1 = 301$
The FIN packet has $sn = 301$ and its FINACK $sn = 300 + 1 = 301$, the last well received byte

The Stop & Wait Protocol is Robust



Goals of this Lab Work

- Study file `SwFtpPacket.java`
- A simple server is available: `SwFtpServer.java`
- Build a client able to send a file to this server