
SYSC 3303 Real-Time Concurrent Systems

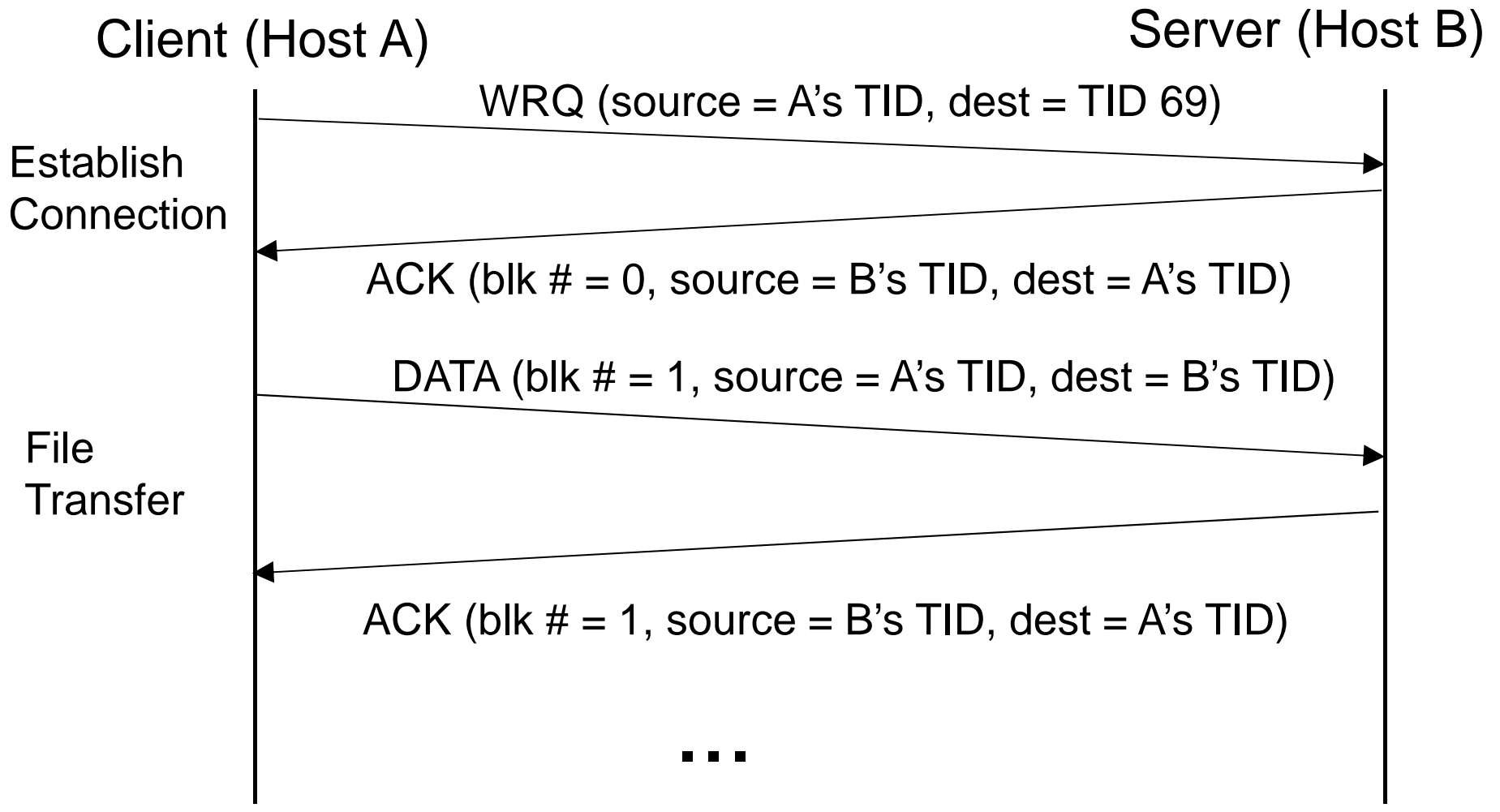
The TFTP Protocol (RFC 1350) Part 2

- Copyright © 2000-2004 D.L. Bailey, 2007-2014 L.S. Marshall, Systems and Computer Engineering, Carleton University
- revised May 25th, 2014

What's In This Set of Slides?

- We'll review error-free file transfers between a TFTP client and server
- We'll see how the TFTP protocol handles lost DATA and ACK packets
- We'll see how the original TFTP protocol handled the delayed reception of packets (Sorcerer's Apprentice bug)
- We'll see how RFC 1350 corrects this bug
- We'll see how ERROR packets allow TFTP hosts to deal with errors

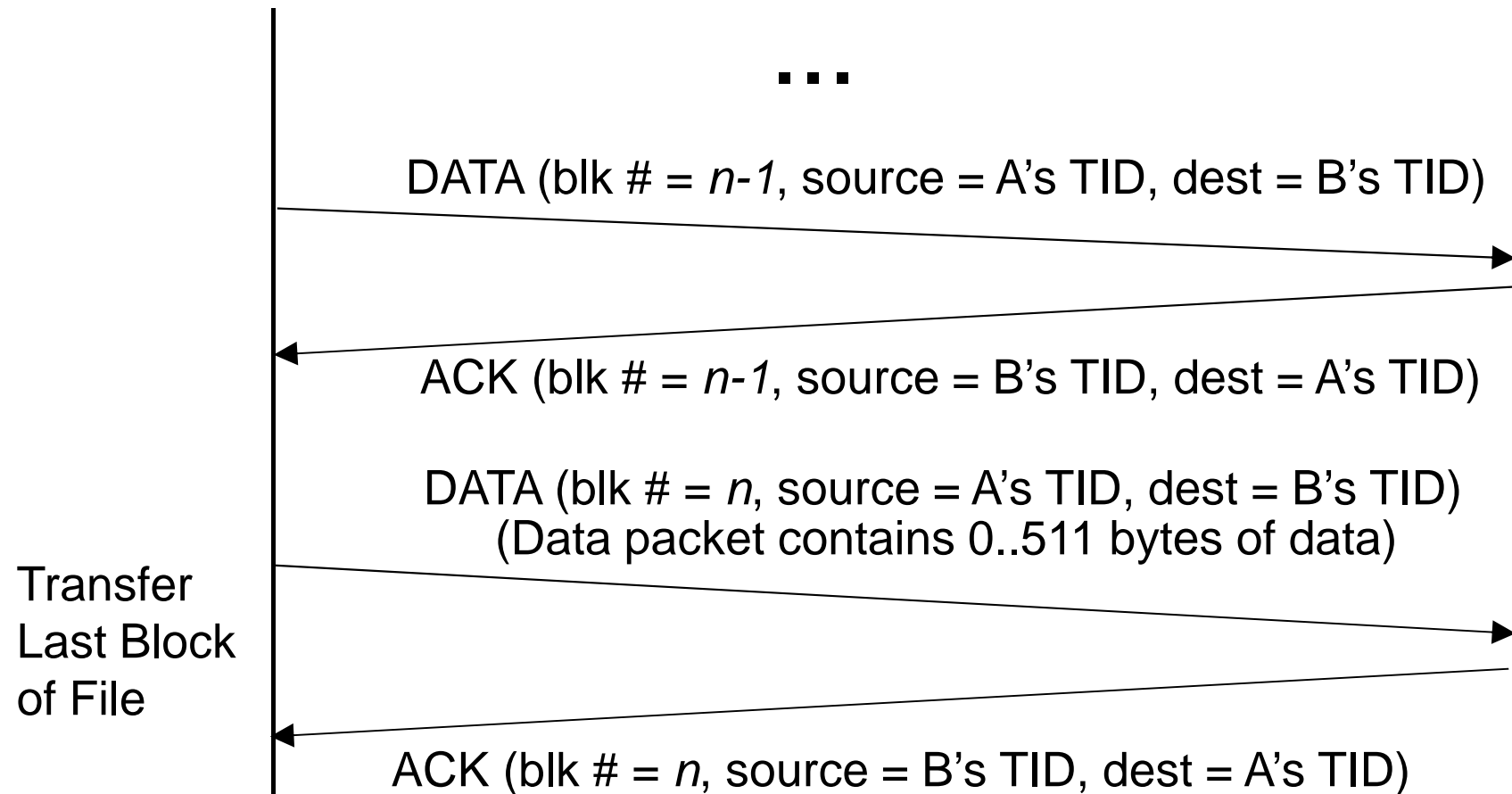
Write File from Client to Server



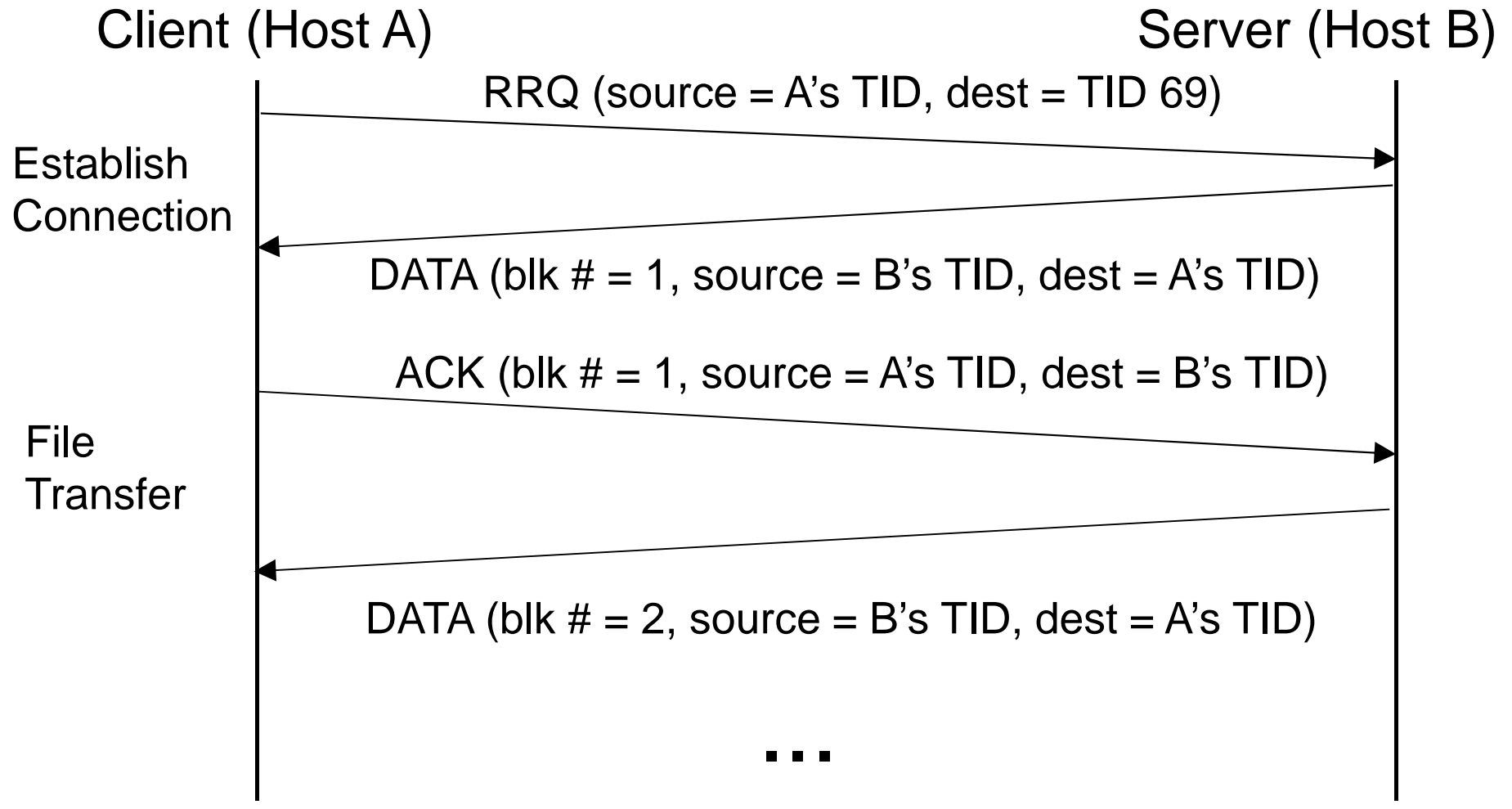
Write File from Client to Server (cont'd)

Client (Host A)

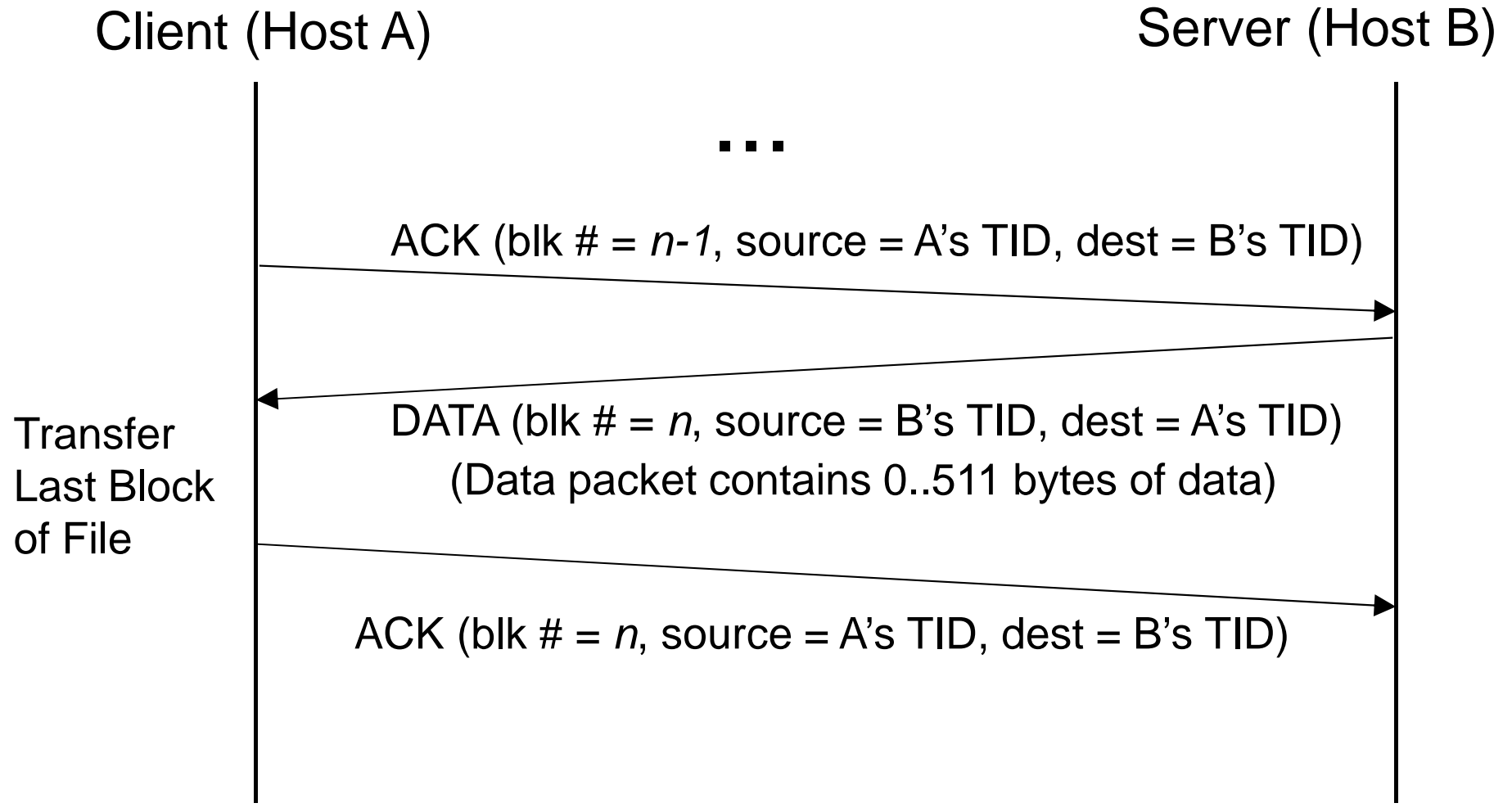
Server (Host B)



Read File from Server to Client



Read File from Server to Client (cont'd)



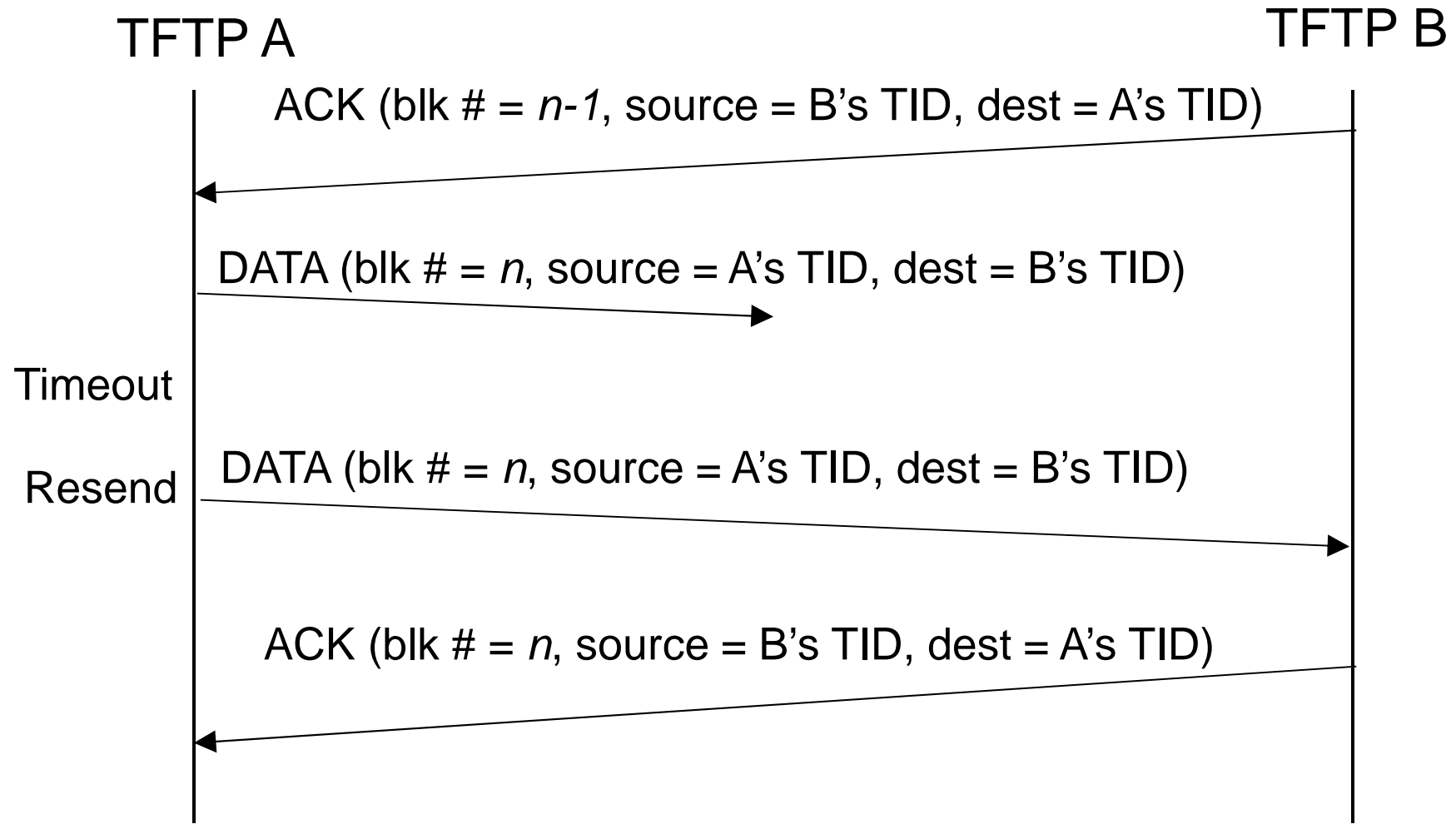
Acknowledgements

- Initial connection:
 - WRQ packets are acknowledged by ACK packets
 - RRQ packets are acknowledged by DATA packets
- File transfer:
 - DATA packets are acknowledged by ACK packets
 - ACK packets are acknowledged by DATA packets

Acknowledgements & Retransmission Timeouts

- Original spec. (RFC 783) stated “All packets other than those used for termination are acknowledged...”
 - later, we will see that this was a serious bug in the protocol specification
- Both client and server wait for acknowledgements (i.e., DATA or ACK packets) with timeouts
- Timeout/retransmit behaviour is not clearly specified?
- The following slides illustrate what happens if packets are lost or delayed during a file transfer
 - write file: TFTP A = client, TFTP B = server
 - read file: TFTP A = server, TFTP B = client

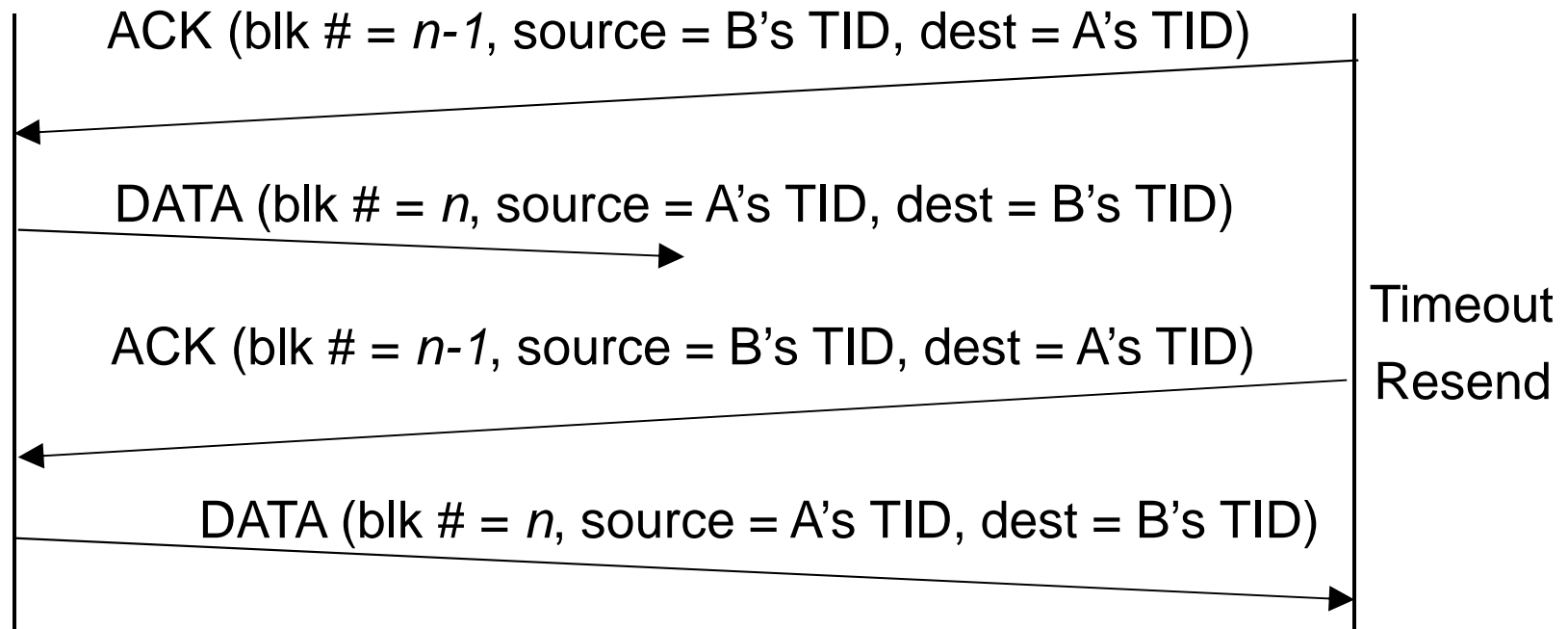
Lost Data Packets (TFTP A Times Out)



Lost Data Packets (TFTP B Times Out)

TFTP A

TFTP B

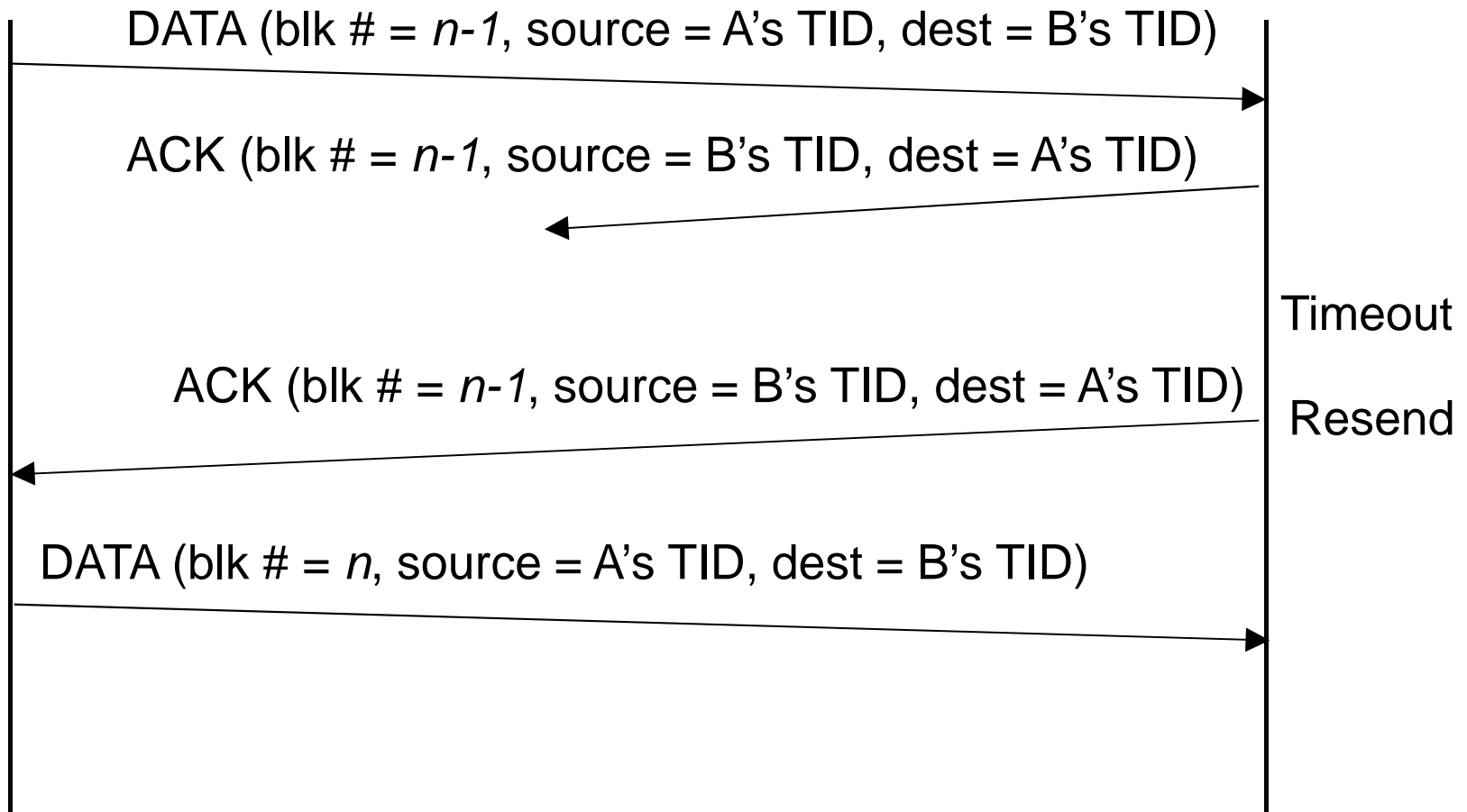


TFTP B times out while waiting for an acknowledgement for the ACK it sent (i.e., it times out while waiting for a DATA packet, so thinks the ACK it sent was lost).

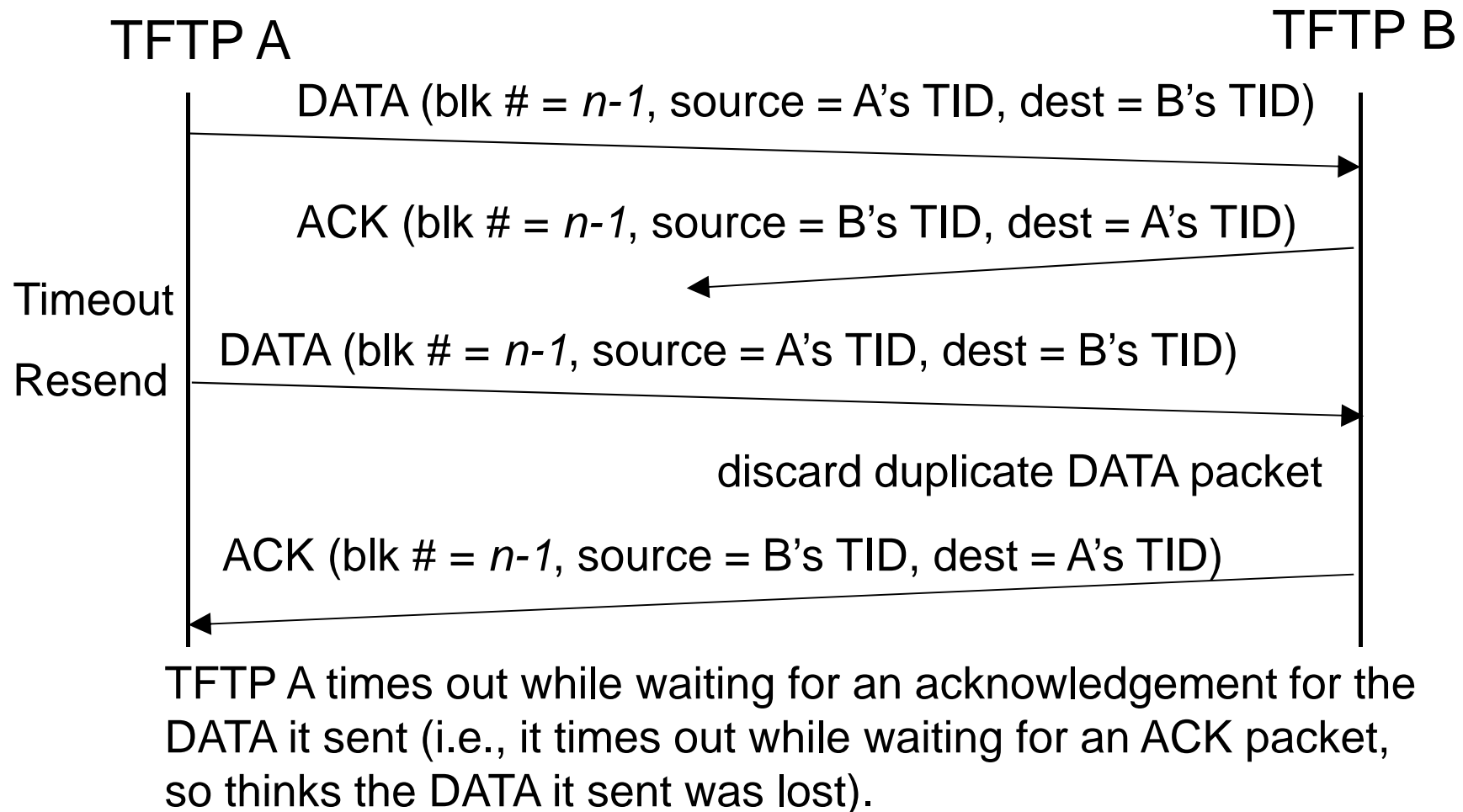
Lost Ack Packets (TFTP B Times Out)

TFTP A

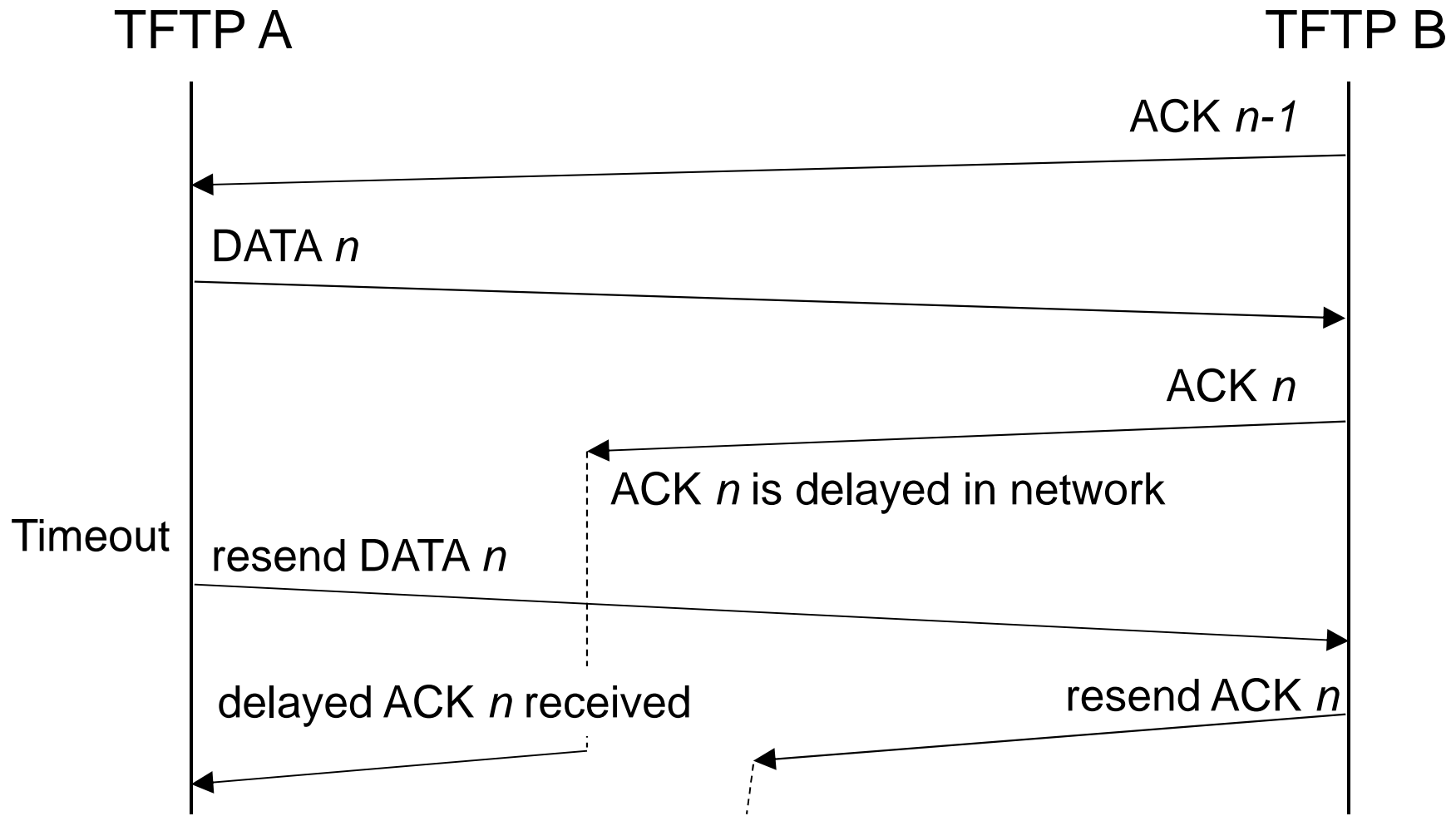
TFTP B



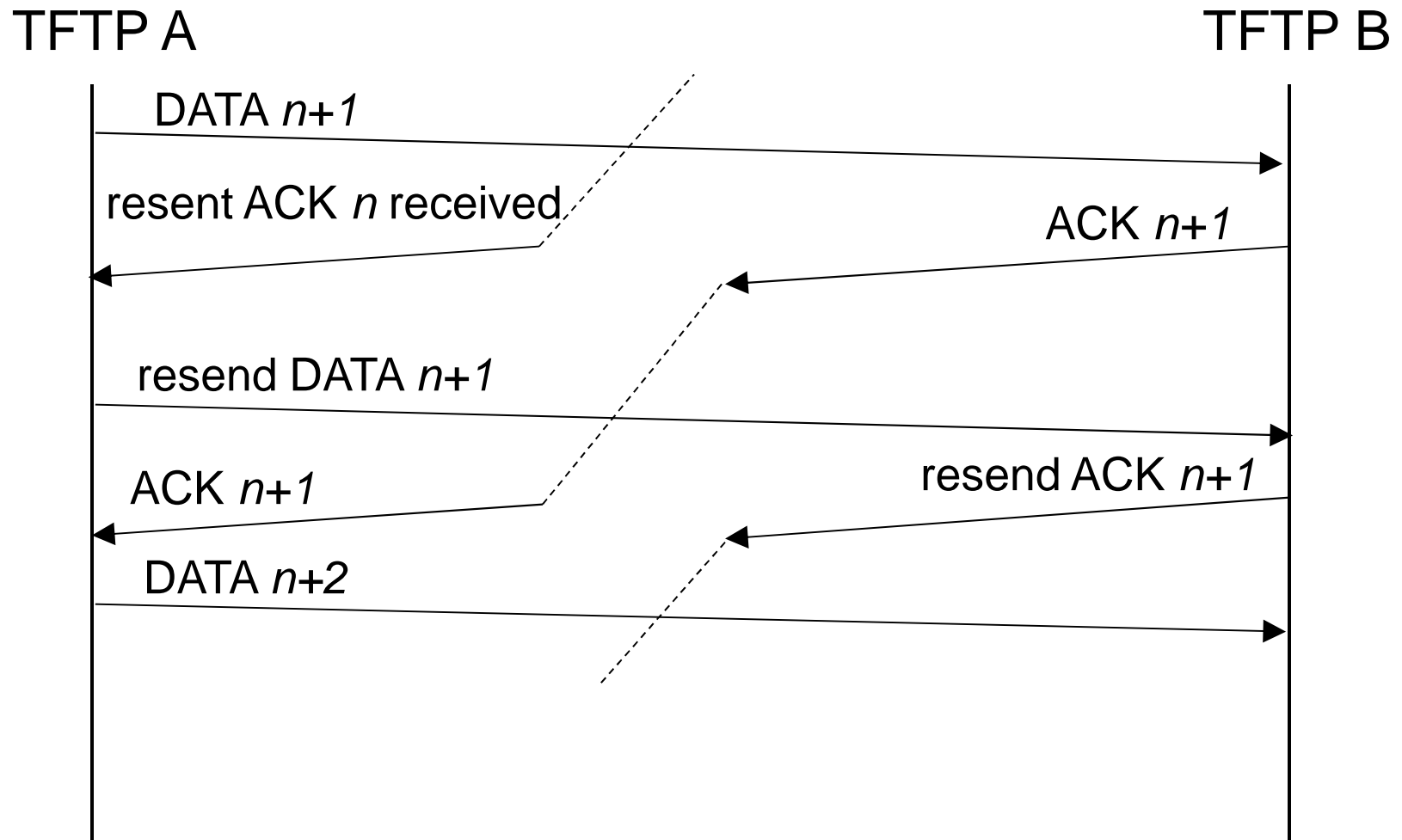
Lost Ack Packets (TFTP A Times Out)



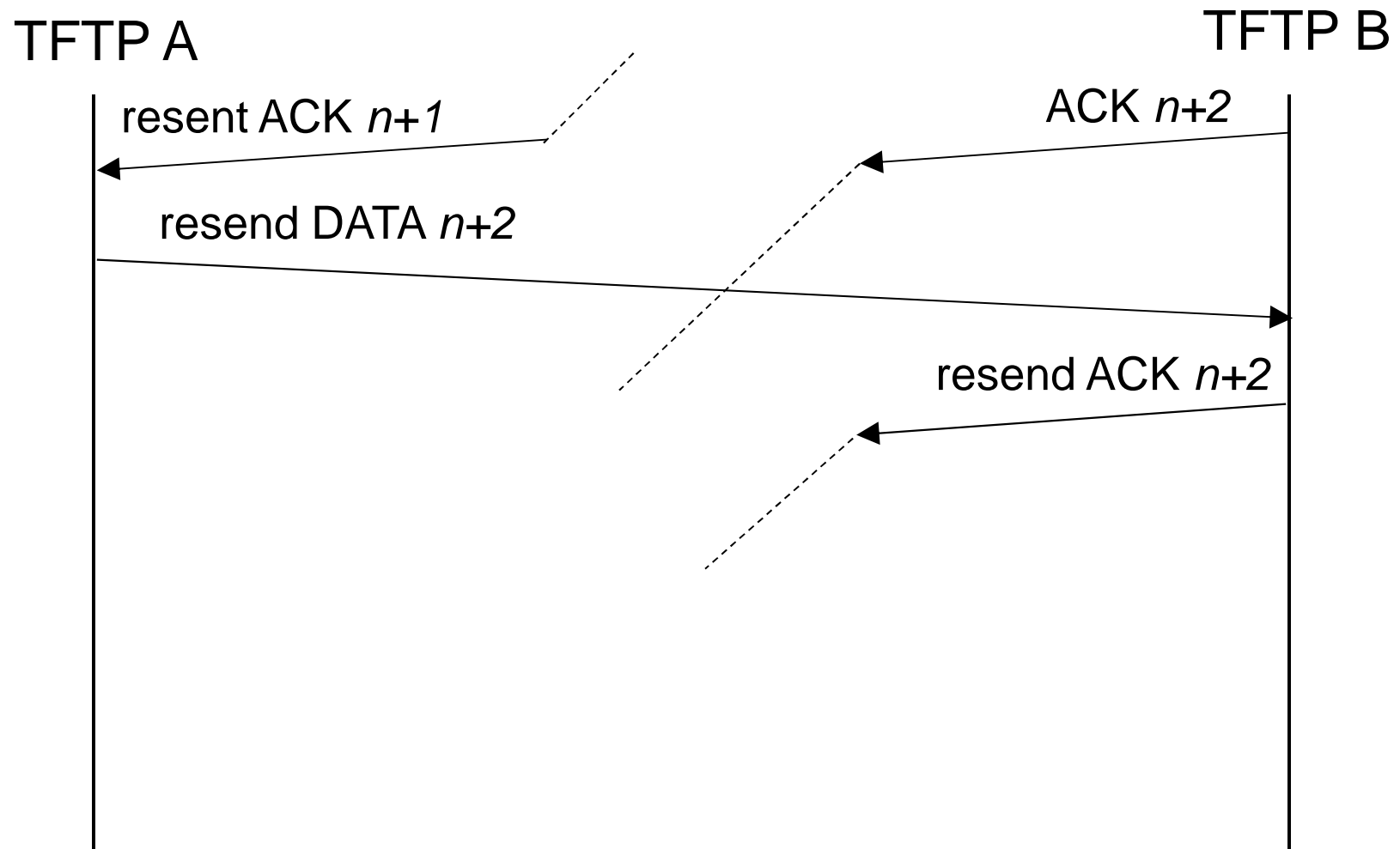
Delayed Packets: Sorcerer's Apprentice Bug



Delayed Packets: Sorcerer's Apprentice Bug



Delayed Packets: Sorcerer's Apprentice Bug



Cause of Sorcerer's Apprentice Bug

- Original spec. (RFC 783) stated “All packets other than those used for termination are acknowledged...”
- This means that either side, on receiving a duplicate of the last packet received (other than those used for termination), retransmits the last packet it sent

Cause of Sorcerer's Apprentice Bug

- “If a packet is delayed in the network but later successfully delivered after either side has timed out and retransmitted a packet, a duplicate copy of the response may be generated. If the other side responds to this duplicate with a duplicate of its own, then every datagram will be sent in duplicate for the remainder of the transfer (unless a datagram is lost, breaking the repetition). Worse yet, since the delay is often caused by congestion, this duplicate transmission will usually cause more congestion, leading to more delayed packets, etc.” -- RFC 1123

Solution to Sorcerer's Apprentice Bug

- TFTP implementations must fix the problem this way:
 - the side that sends DATA packets must never resend the current DATA packet on receipt of a duplicate ACK
- Correction in RFC 1350: “All packets other than *duplicate ACKs and* those used for termination are acknowledged...”

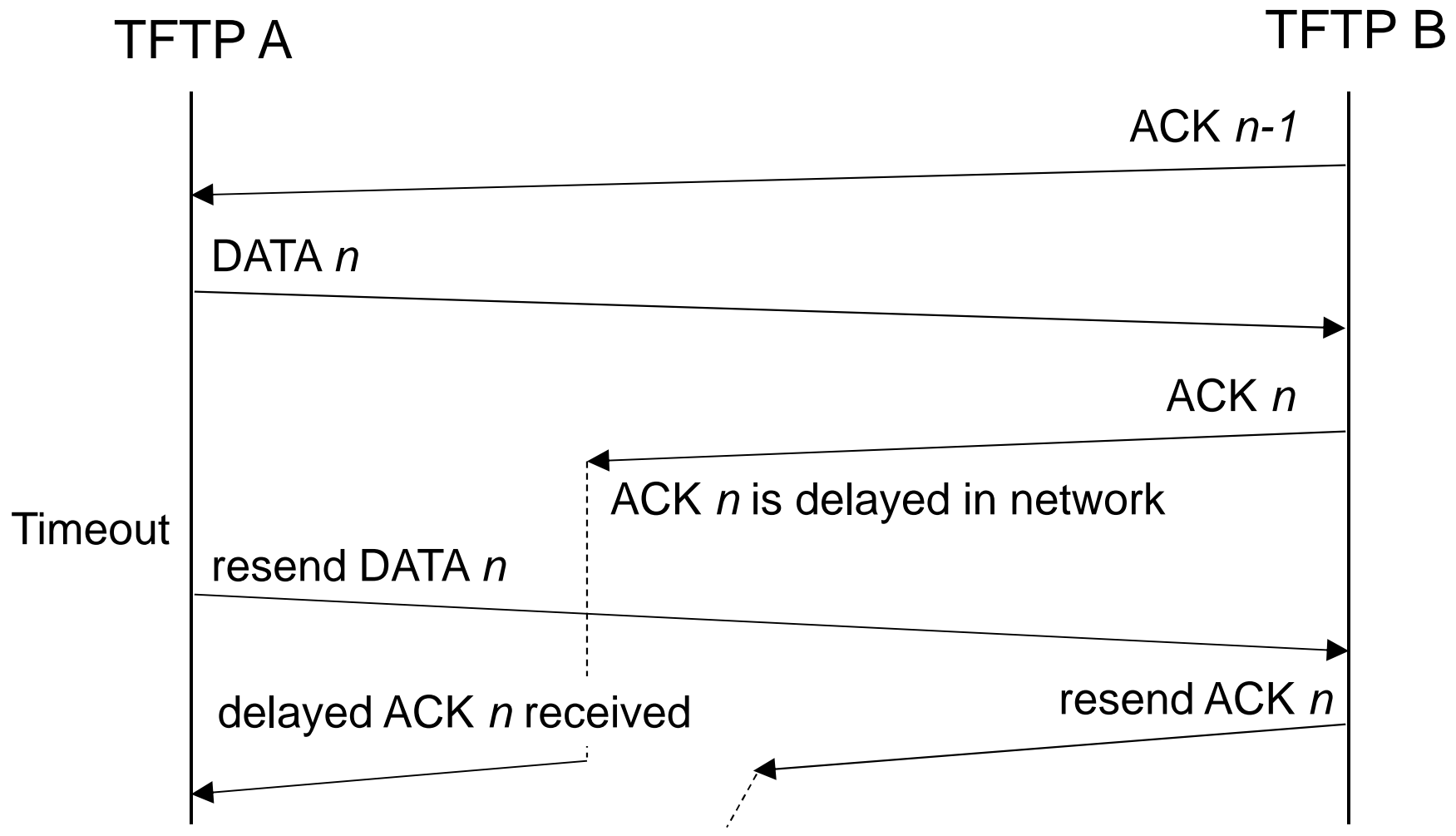
Solution to Sorcerer's Apprentice Bug

- Optionally:
 - keep the retransmission timer on the side that sends DATA packets (will timeout and retransmit if ACK not received)
 - remove the retransmission timer on the side that sends ACK packets (a retransmitted ACK is always ignored, so why retransmit it?)
- We still need the solution described on the previous slide (ACK packets may still be duplicated along the network, even if the TFTP software doesn't retransmit ACKs)

Implications for TFTP Clients and Servers

- Write file from client to server
 - client ignores duplicate ACKs
 - client requires a retransmission timer
 - server does not need a retransmission timer (but it's o.k. if it has one)
- Read file from server to client
 - server ignores duplicate ACKs
 - server requires a retransmission timer
 - client does not need a retransmission timer (but it's o.k. if it has one)

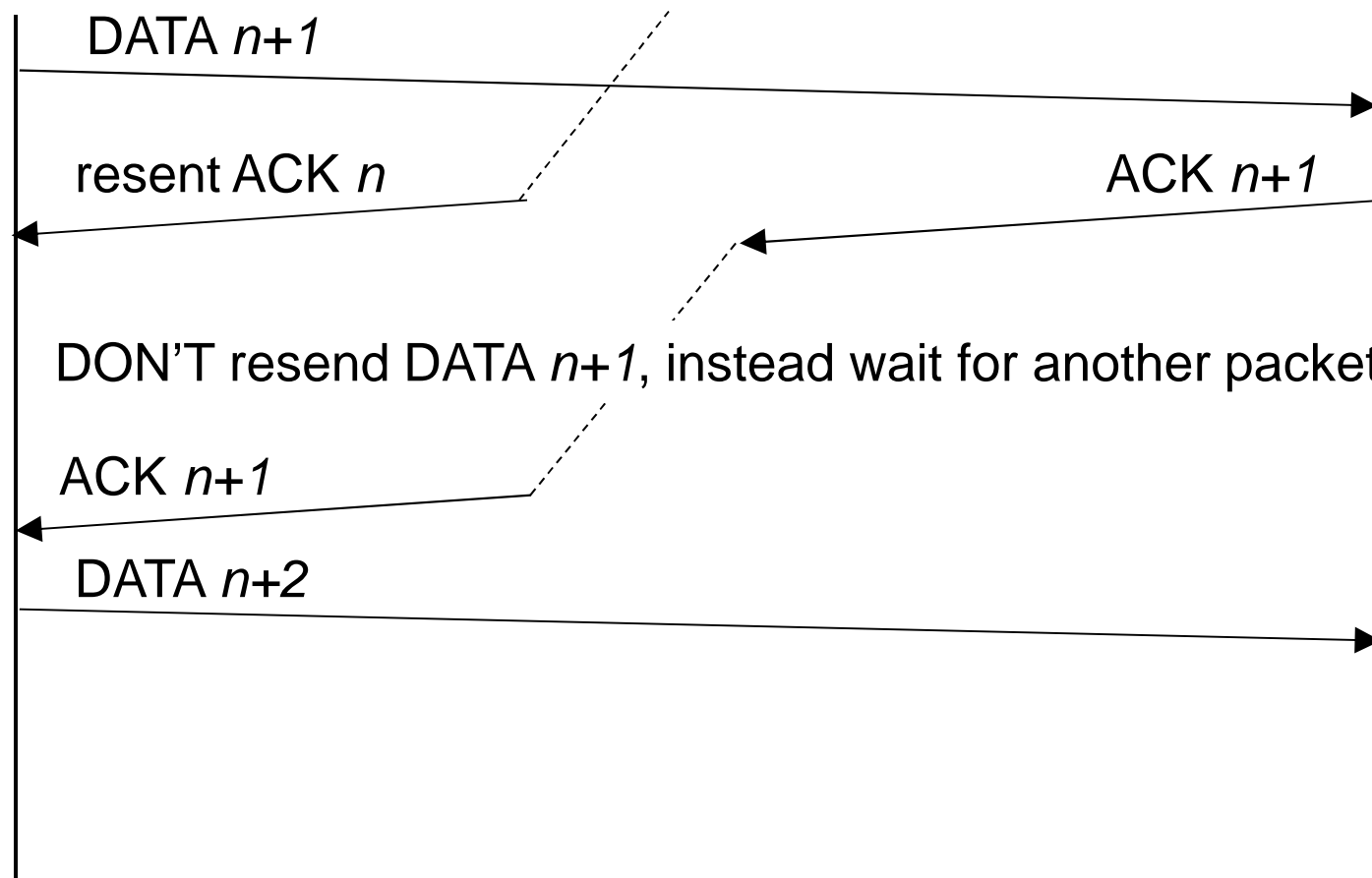
Delayed Packets: Sorcerer's Apprentice Bug Removed



Delayed Packets: Sorcerer's Apprentice Bug Removed

TFTP A

TFTP B



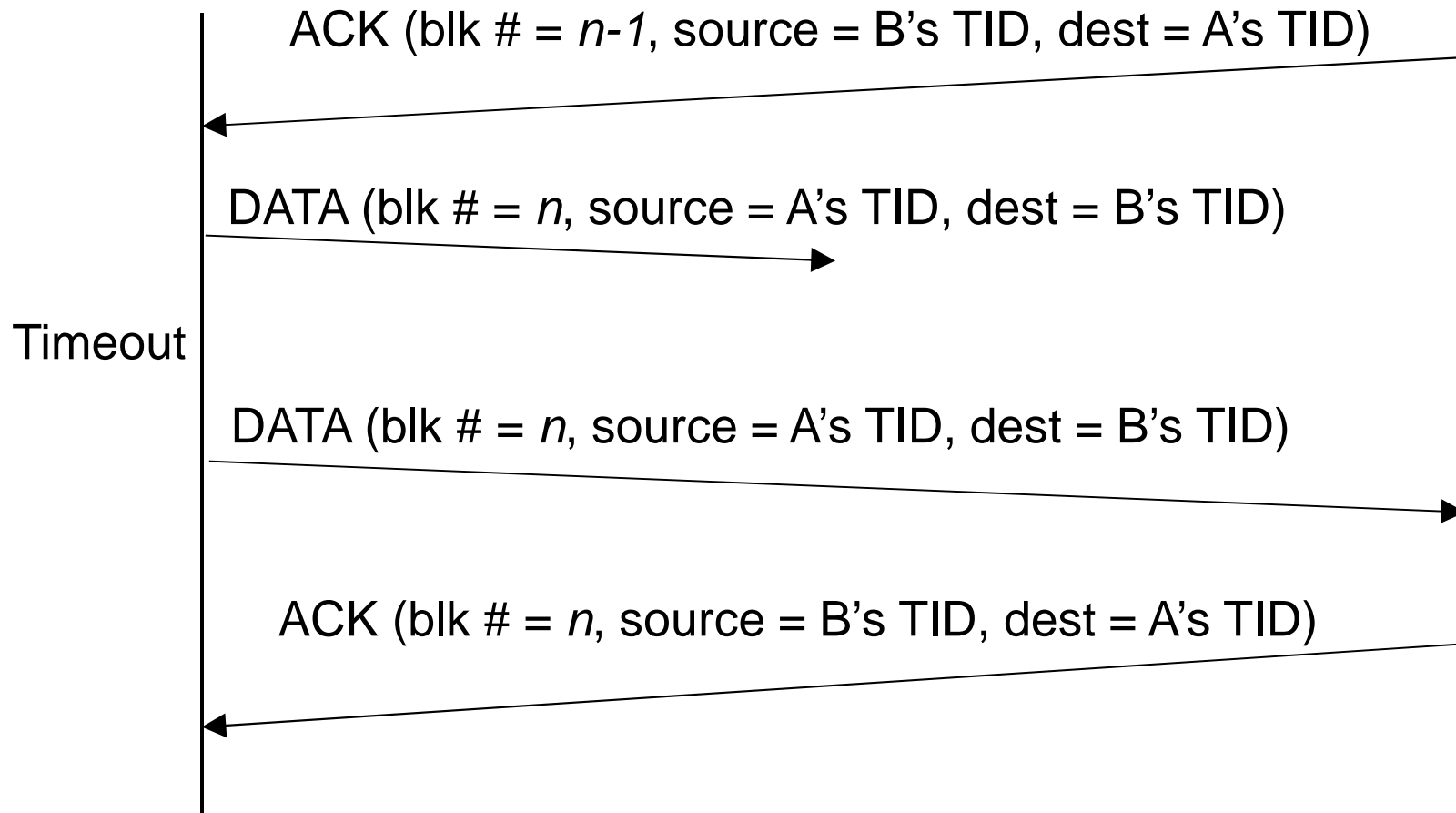
What Happens When Packets Are Lost?

- The following slides show what happens when the Sorcerer's Apprentice bug has been fixed, the retransmission timer has been removed from the side that sends ACKs, and DATA or ACK packets are lost

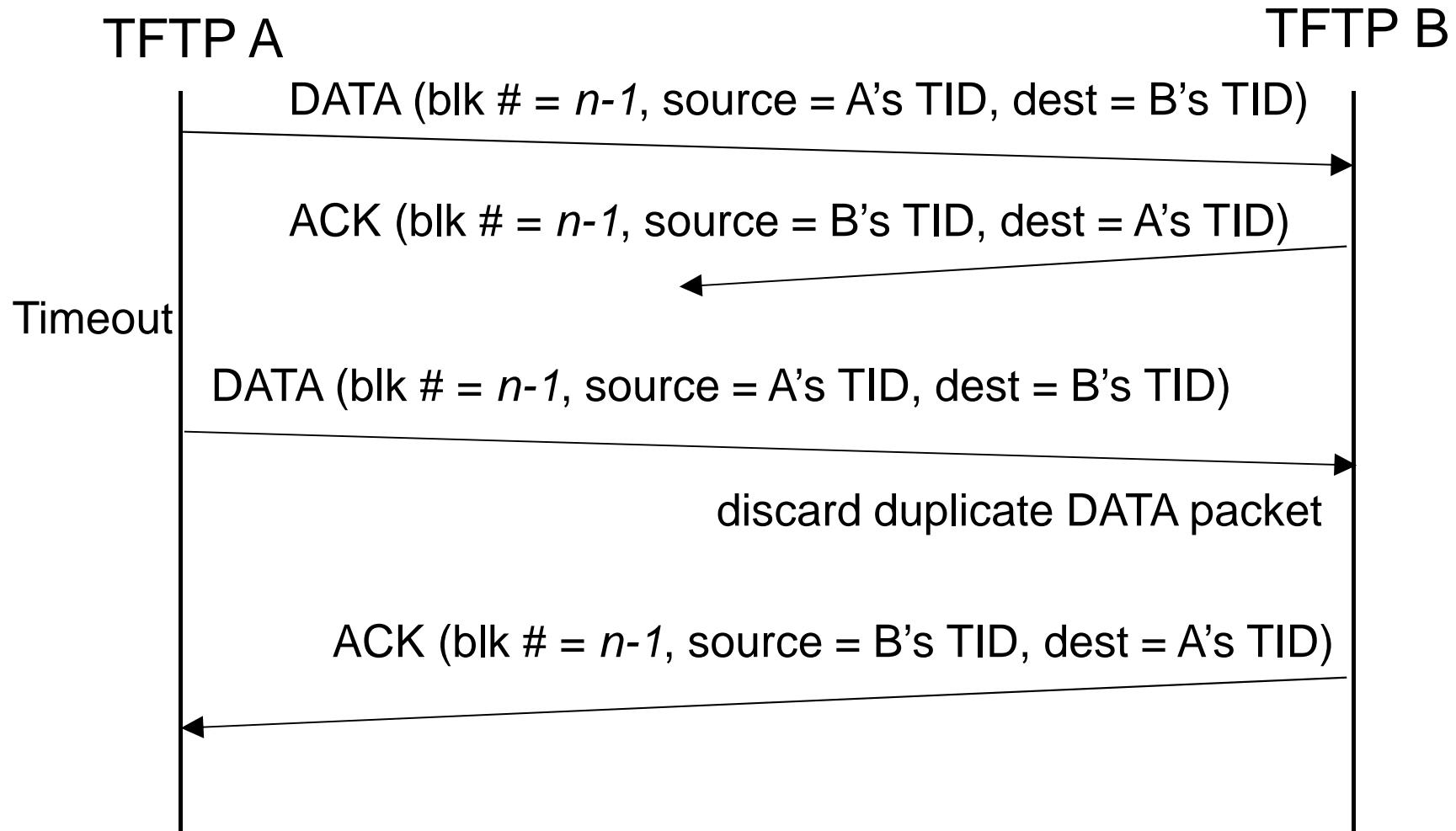
Lost Data Packets (TFTP A Times Out)

TFTP A

TFTP B



Lost Ack Packets (TFTP A Times Out)



Errors Causing Connection Termination

- 3 conditions will prevent a server from granting a request
 - file not found (server processing RRQ)
 - access violation (cause is host-dependent)
 - file already exists (maybe – why?)
 - no such user (?? - from obsolete mail mode?)
- During file transfer, access to a resource may be lost
 - disk becomes full
 - access denied

Errors Causing Connection Termination

- A host may receive an incorrect packet that cannot be explained by delay or duplication
 - packet received at TID 69 does not contain a RRQ or WRQ opcode
 - packet opcode is not a valid TFTP packet type
 - invalid filename or mode fields in RRQ/WRQ packets
 - invalid block # in DATA/ACK packets

ERROR Packet Transmission & Reception

- The host that detects the error sends an appropriate ERROR packet, then terminates the connection
 - closes the DatagramSocket associated with the connection
 - closes the file
 - releases any other resources associated with the connection (e.g., memory buffers, threads, etc.)
- The host that receives an ERROR packet terminates its end of the connection in same way

ERROR Packet Transmission & Reception

- ERROR packets are never acknowledged by the receiver
- As such, the host that sends an ERROR packet does not wait for an acknowledgement/retransmit after a timeout
 - instead, after sending an ERROR packet, the sender normally terminates the connection
- A consequence of this is that if an ERROR packet is lost, the intended recipient will never learn of the ERROR
- Repeated timeouts imply that a connection was terminated and that the ERROR packet was lost

Checking Transfer IDs

- TFTP defines one error condition that does not result in connection termination, namely, the source TID (i.e., source port) in a received packet is incorrect

Description

- When a connection is established, both hosts learn of each other's TIDs
- When a host receives a DATA or ACK packet, it should verify that the source TID in the received packet matches the value obtained when the connection was established

Checking Transfer IDs

- If the source TID does not match, the host should
 - assume that the packet was incorrectly sent from another host
 - discard the packet
 - send an ERROR packet with error code 5 ("Unknown transfer ID") to the originating host of the incorrect packet
 - continue with the file transfer
- For example, what would happen if a request was duplicated?