<u>Diagrams for Iteration 4</u>

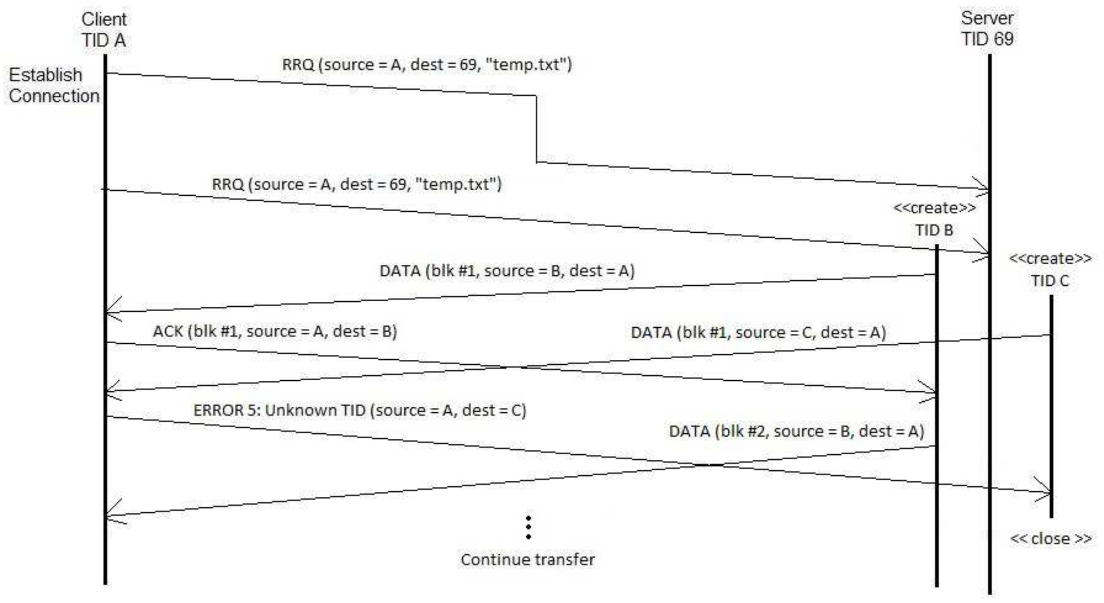
Team 4

June 2, 2015

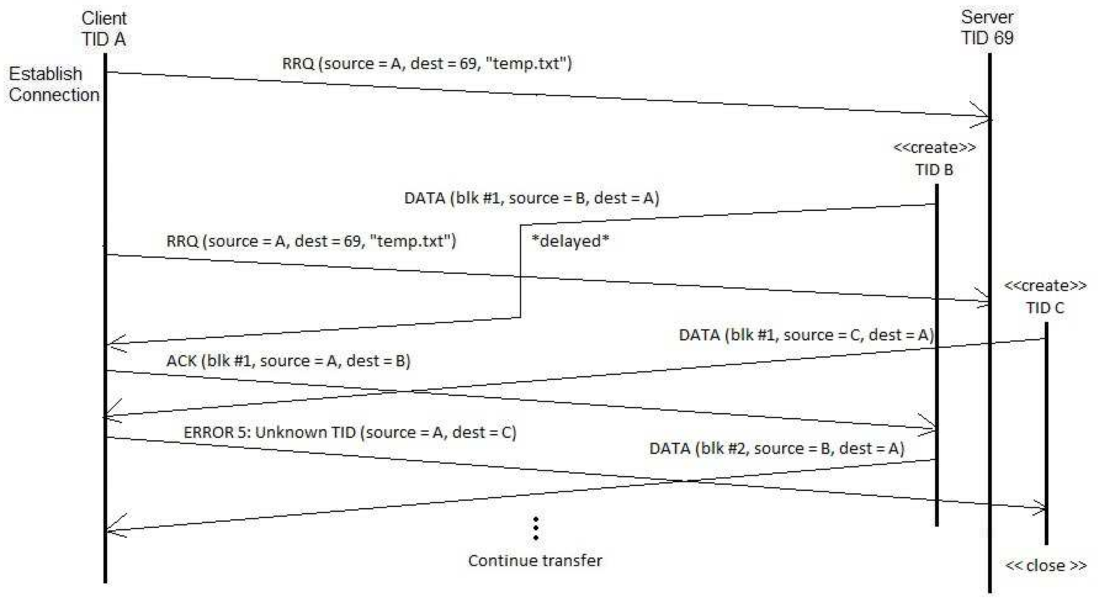
Timing diagrams for iteration #4

[Delay Errors]

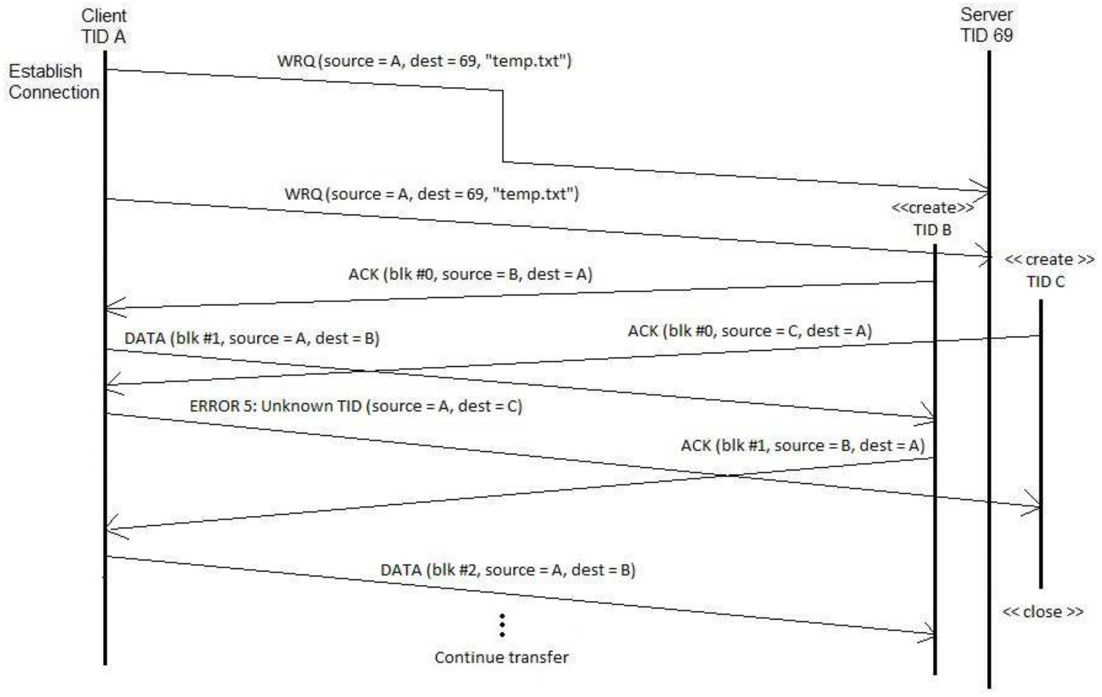
Scenario 1 – Client RRQ Delay:


Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69, "temp.txt")
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
<<create>>
TID C
ACK (blk #1, source = A, dest = B)
DATA (blk #1, source = C, dest = A)
ERROR 5: Unknown TID (source = A, dest = C)
DATA (blk #2, source = B, dest = A)
Continue transfer
<< close >>

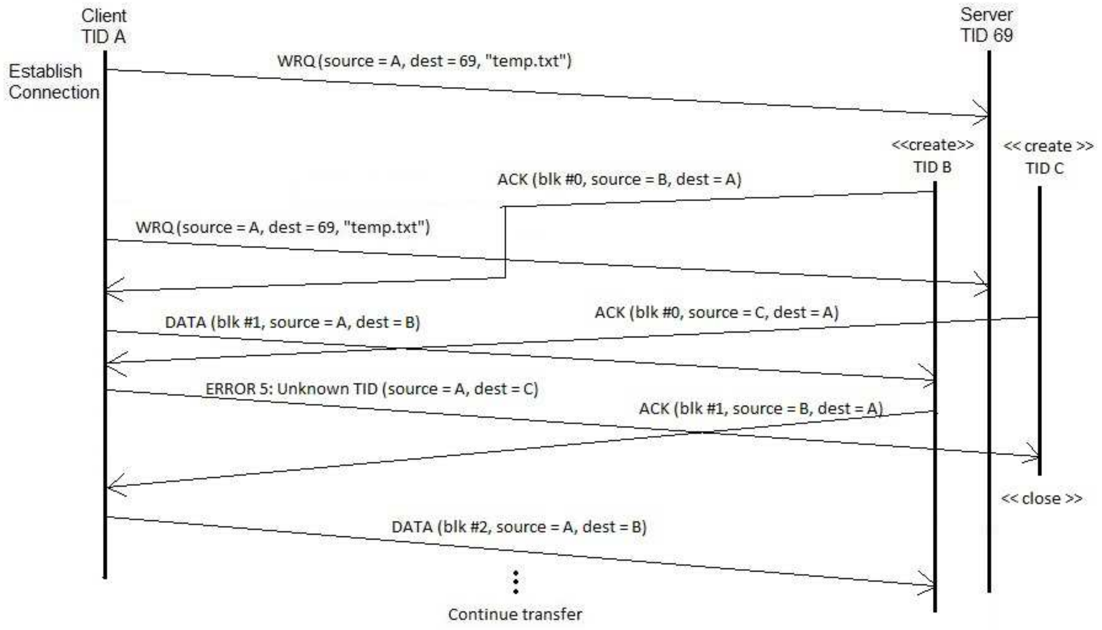
Scenario 2 – Server Response to RRQ Delay:


Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
RRQ (source = A, dest = 69, "temp.txt")
delayed
<<create>>
TID C
DATA (blk #1, source = C, dest = A)
ACK (blk #1, source = A, dest = B)
ERROR 5: Unknown TID (source = A, dest = C)
DATA (blk #2, source = B, dest = A)
Continue transfer
<< close >>

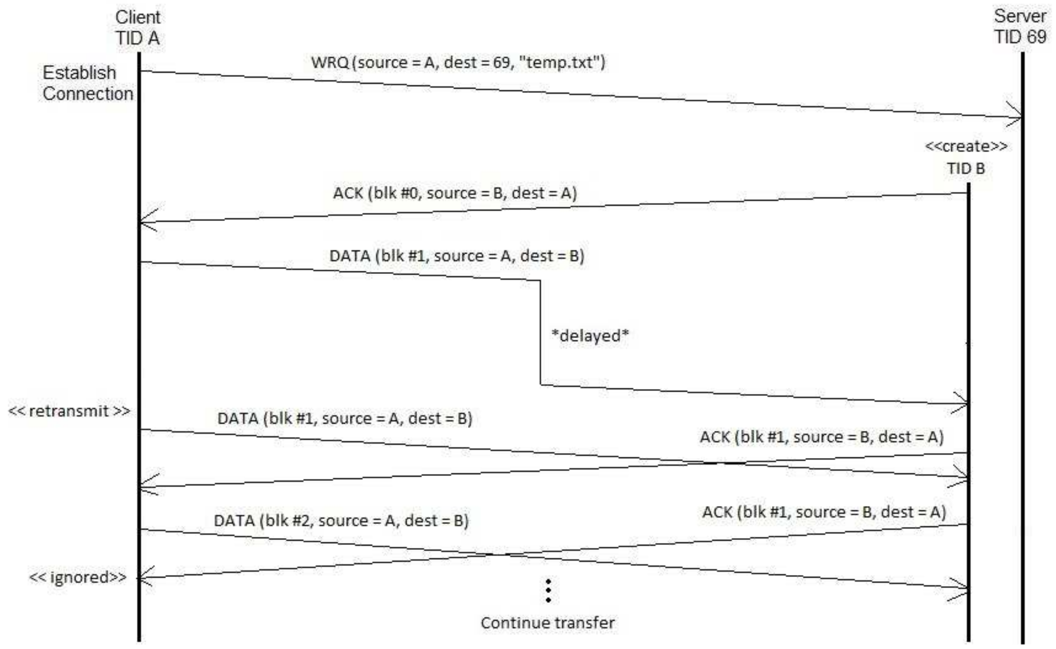
Scenario 3 – Client WRQ Delay:


Client
TID A
Server
TID 69
Establish Connection
WRQ (source = A, dest = 69, "temp.txt")
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
<< create >>
TID C
ACK (blk #0, source = B, dest = A)
DATA (blk #1, source = A, dest = B)
ACK (blk #0, source = C, dest = A)
ERROR 5: Unknown TID (source = A, dest = C)
ACK (blk #1, source = B, dest = A)
DATA (blk #2, source = A, dest = B)
<< close >>
Continue transfer

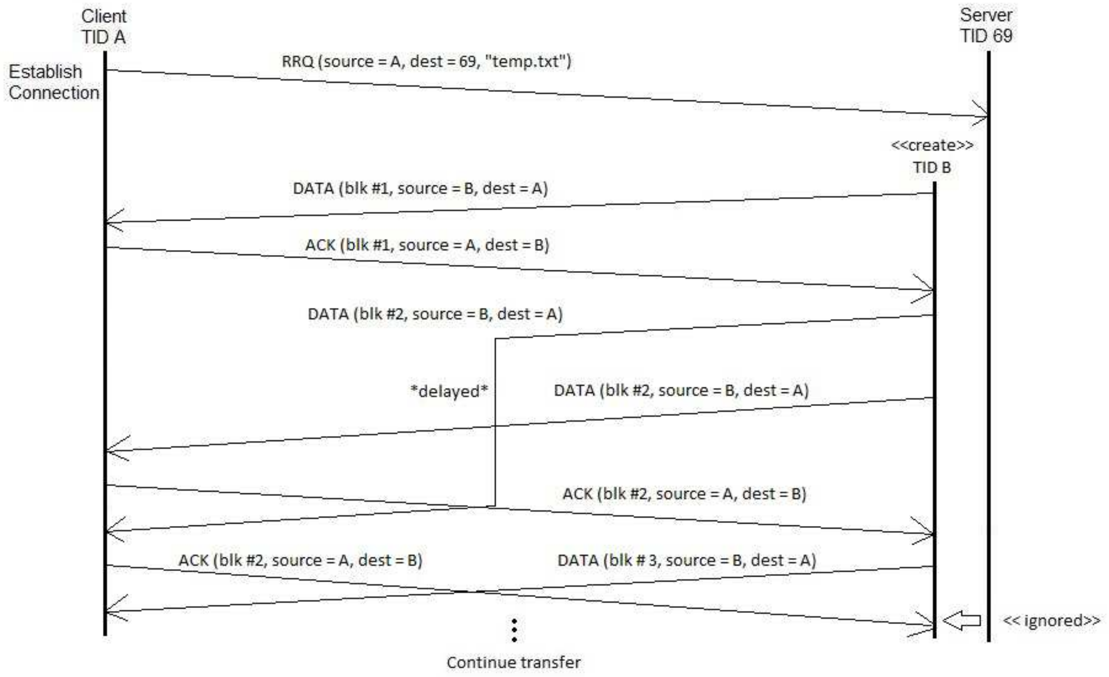
Scenario 4 – Server Response to WRQ Delay:


Client
TID A
Server
TID 69
Establish Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
<< create >>
TID C
ACK (blk #0, source = B, dest = A)
WRQ (source = A, dest = 69, "temp.txt")
DATA (blk #1, source = A, dest = B)
ACK (blk #0, source = C, dest = A)
ERROR 5: Unknown TID (source = A, dest = C)
ACK (blk #1, source = B, dest = A)
<< close >>
DATA (blk #2, source = A, dest = B)
Continue transfer

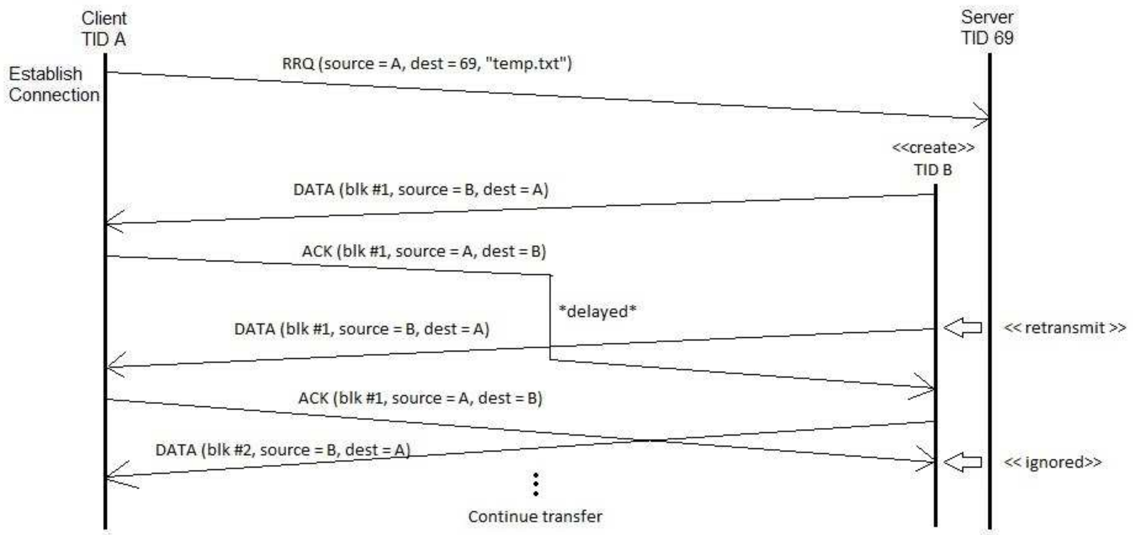
Scenario 5 – Client Data Delay:


Client
TID A
Server
TID 69
Establish Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
DATA (blk #1, source = A, dest = B)
delayed
<< retransmit >>
DATA (blk #1, source = A, dest = B)
ACK (blk #1, source = B, dest = A)
DATA (blk #2, source = A, dest = B)
ACK (blk #1, source = B, dest = A)
<< ignored>>
Continue transfer

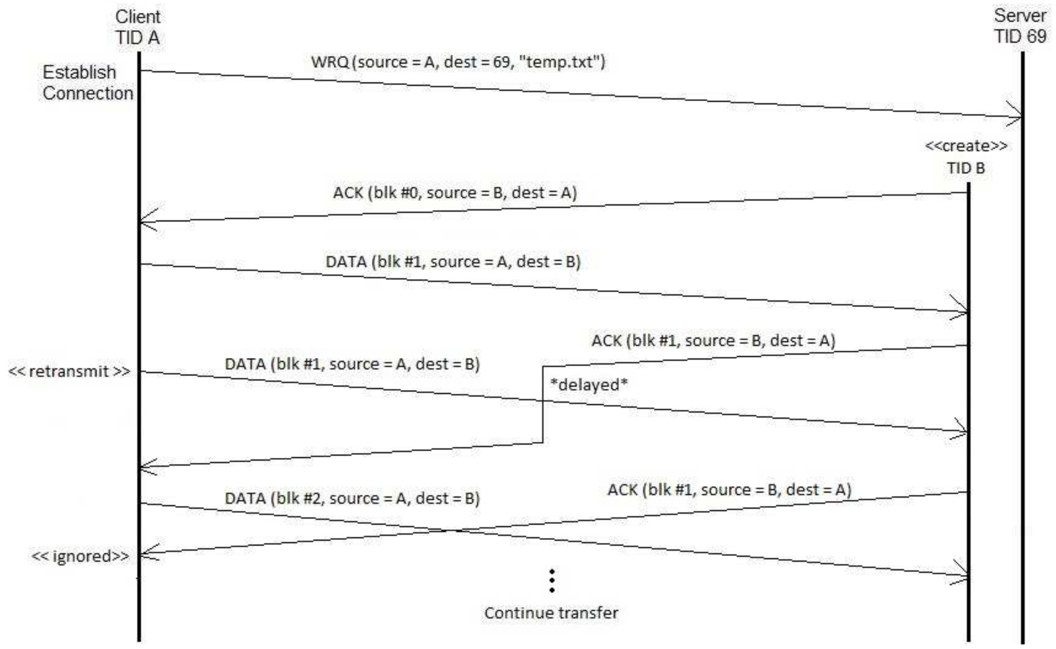
Scenario 6 – Server Data Delay:


Client
TID A
Server
TID 69
Establish Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
DATA (blk #2, source = B, dest = A)
delayed
DATA (blk #2, source = B, dest = A)
ACK (blk #2, source = A, dest = B)
ACK (blk #2, source = A, dest = B)
DATA (blk # 3, source = B, dest = A)
<< ignored>>
Continue transfer

Scenario 7 – Client Ack Delay:

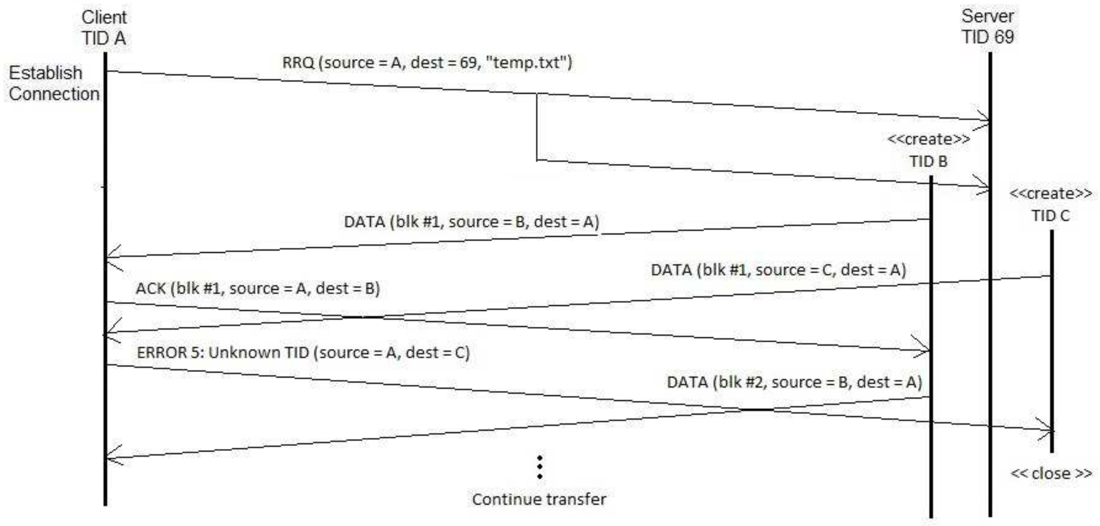

Client
TID A
Server
TID 69
Establish Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
delayed
DATA (blk #1, source = B, dest = A)
<< retransmit >>
ACK (blk #1, source = A, dest = B)
DATA (blk #2, source = B, dest = A)
<< ignored>>
Continue transfer

Scenario 8 – Server Ack Delay:

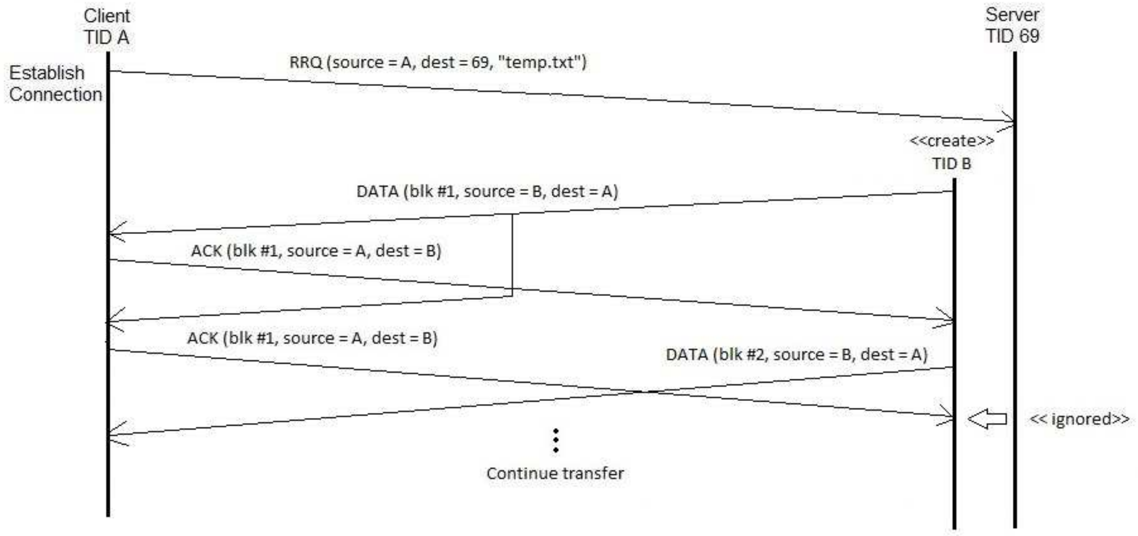

Client
TID A
Server
TID 69
Establish Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
DATA (blk #1, source = A, dest = B)
ACK (blk #1, source = B, dest = A)
<< retransmit >>
DATA (blk #1, source = A, dest = B)
delayed
DATA (blk #2, source = A, dest = B)
ACK (blk #1, source = B, dest = A)
<< ignored>>
Continue transfer

[Duplicate Errors]

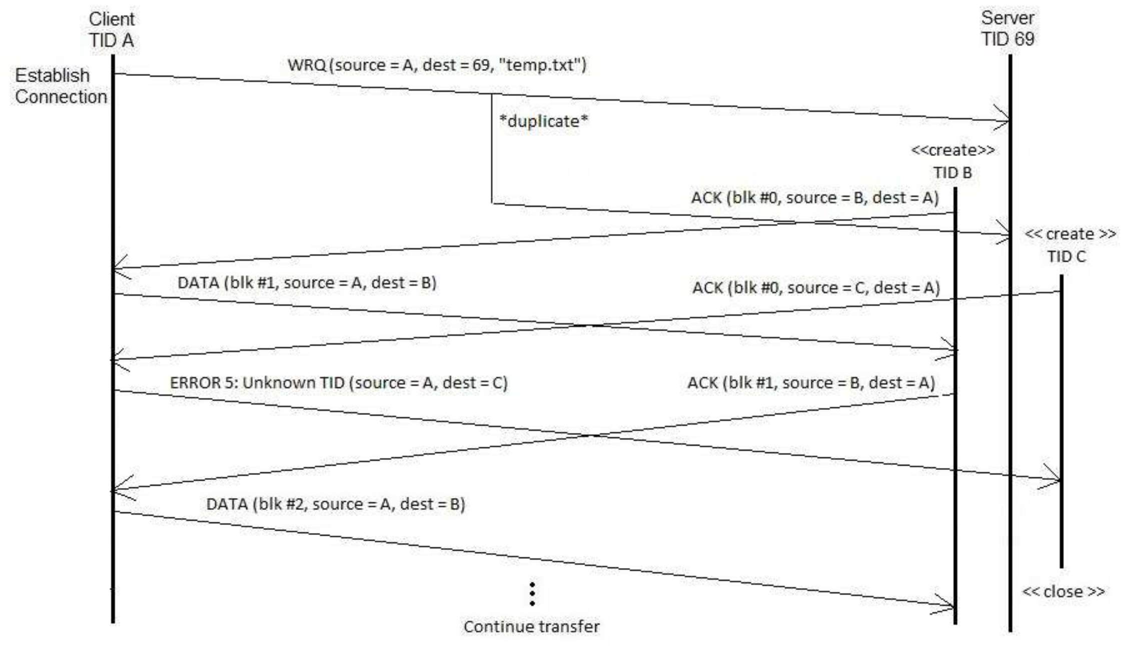
Scenario 1 – Client RRQ Duplicate:


Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
<<create>>
TID C
DATA (blk #1, source = B, dest = A)
DATA (blk #1, source = C, dest = A)
ACK (blk #1, source = A, dest = B)
ERROR 5: Unknown TID (source = A, dest = C)
DATA (blk #2, source = B, dest = A)
<< close >>
Continue transfer

Scenario 2 – Server Response to RRQ Duplicate:

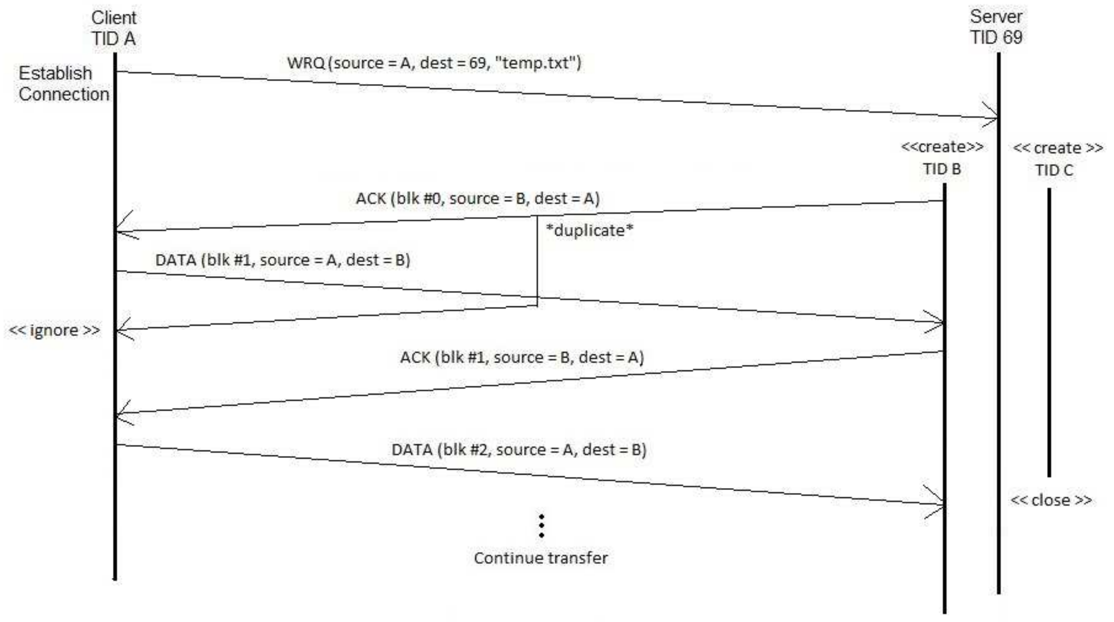

Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
ACK (blk #1, source = A, dest = B)
DATA (blk #2, source = B, dest = A)
<< ignored>>
Continue transfer

Scenario 3 – Client WRQ Duplicate:



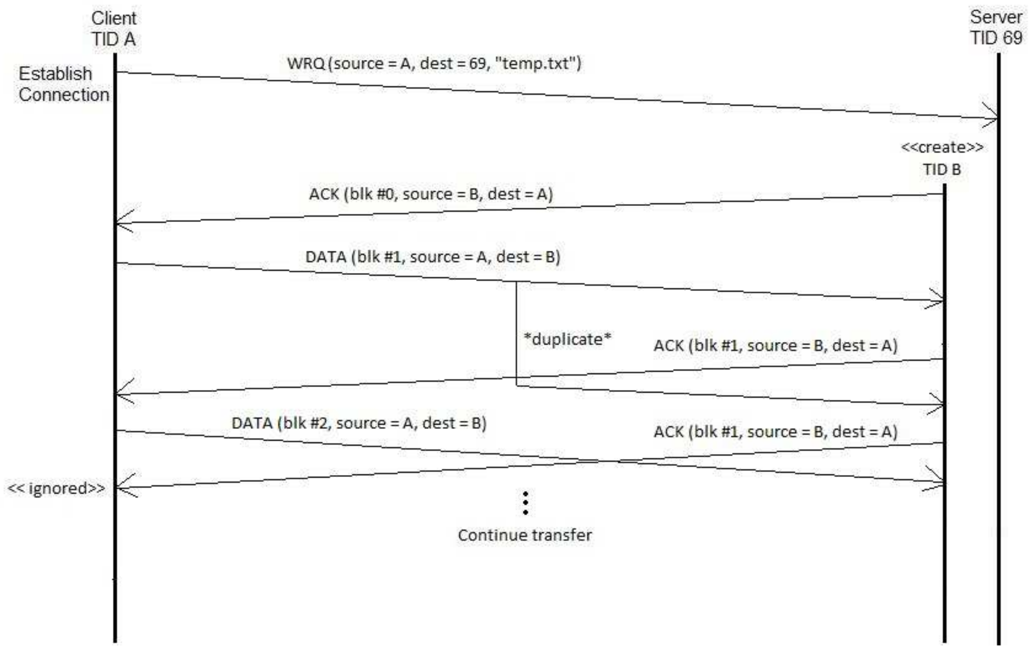
Client
TID A
Server
TID 69
Establish Connection
WRQ (source = A, dest = 69, "temp.txt")
duplicate
<<create>> TID B
ACK (blk #0, source = B, dest = A)
<< create >> TID C
DATA (blk #1, source = A, dest = B)
ACK (blk #0, source = C, dest = A)
ERROR 5: Unknown TID (source = A, dest = C)
ACK (blk #1, source = B, dest = A)
DATA (blk #2, source = A, dest = B)
<< close >>
Continue transfer

Scenario 4 – Server Response to WRQ Duplicate:

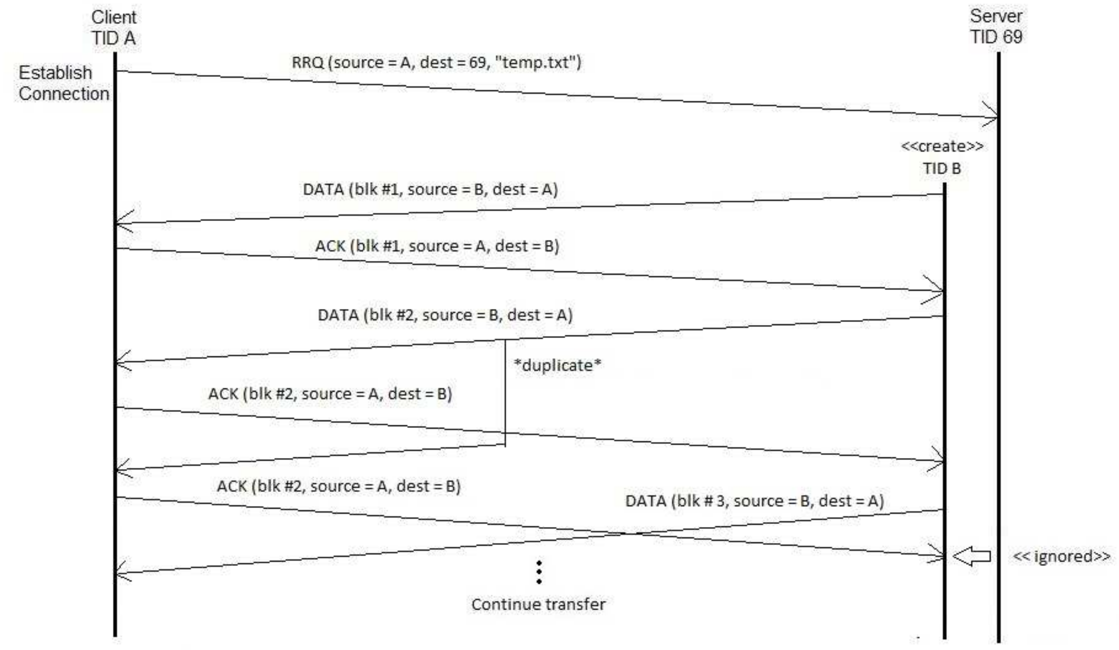


Client
TID A
Server
TID 69
Establish Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>> TID B
<< create >> TID C
ACK (blk #0, source = B, dest = A)
duplicate
DATA (blk #1, source = A, dest = B)
<< ignore >>
ACK (blk #1, source = B, dest = A)
DATA (blk #2, source = A, dest = B)
<< close >>
Continue transfer

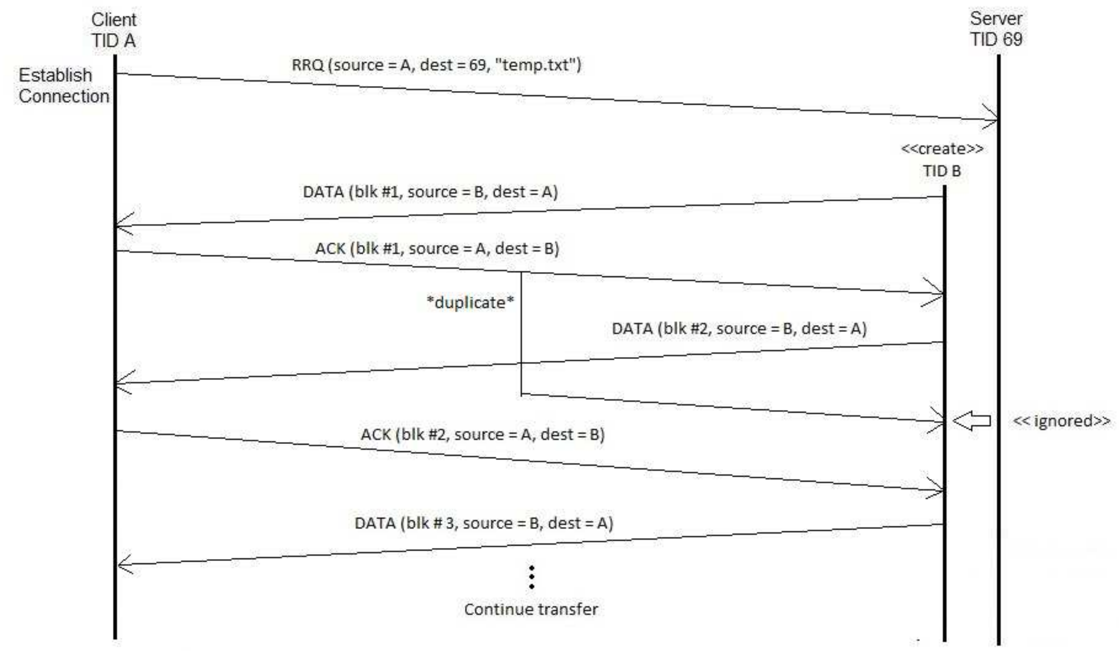
Scenario 5 – Client Data Duplicate:


Client
TID A
Server
TID 69
Establish Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
DATA (blk #1, source = A, dest = B)
duplicate
ACK (blk #1, source = B, dest = A)
DATA (blk #2, source = A, dest = B)
ACK (blk #1, source = B, dest = A)
<< ignored>>
Continue transfer

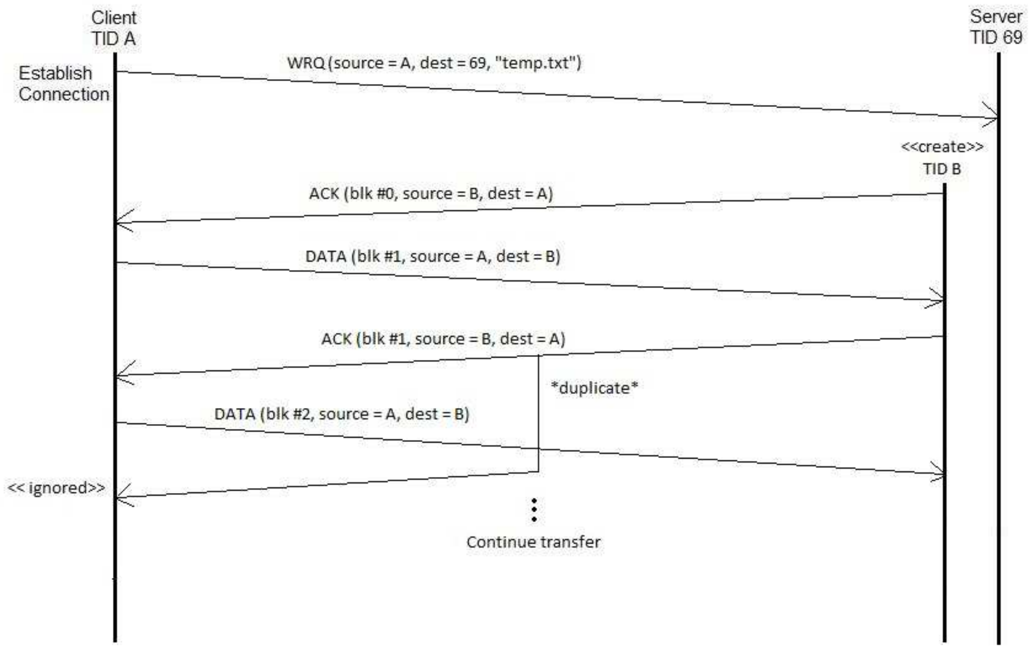
Scenario 6 – Server Data Duplicate:


Client
TID A
Server
TID 69
Establish Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
DATA (blk #2, source = B, dest = A)
duplicate
ACK (blk #2, source = A, dest = B)
ACK (blk #2, source = A, dest = B)
DATA (blk # 3, source = B, dest = A)
<< ignored>>
Continue transfer

Scenario 7 – Client Ack Duplicate:

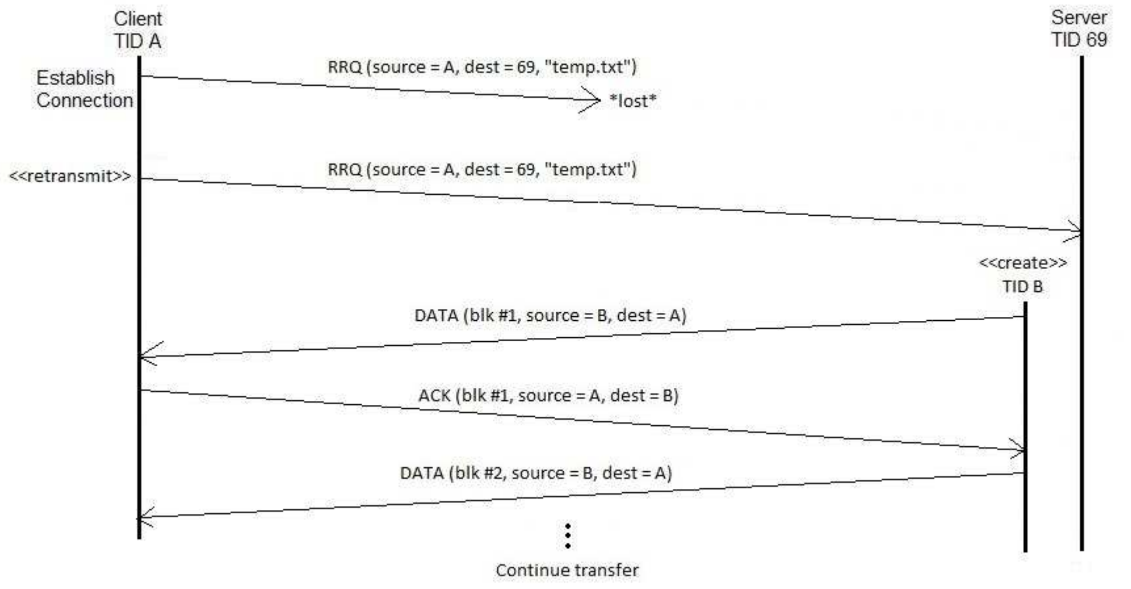

Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
duplicate
DATA (blk #2, source = B, dest = A)
<< ignored>>
ACK (blk #2, source = A, dest = B)
DATA (blk # 3, source = B, dest = A)
Continue transfer

Scenario 8 – Server Ack Duplicate:

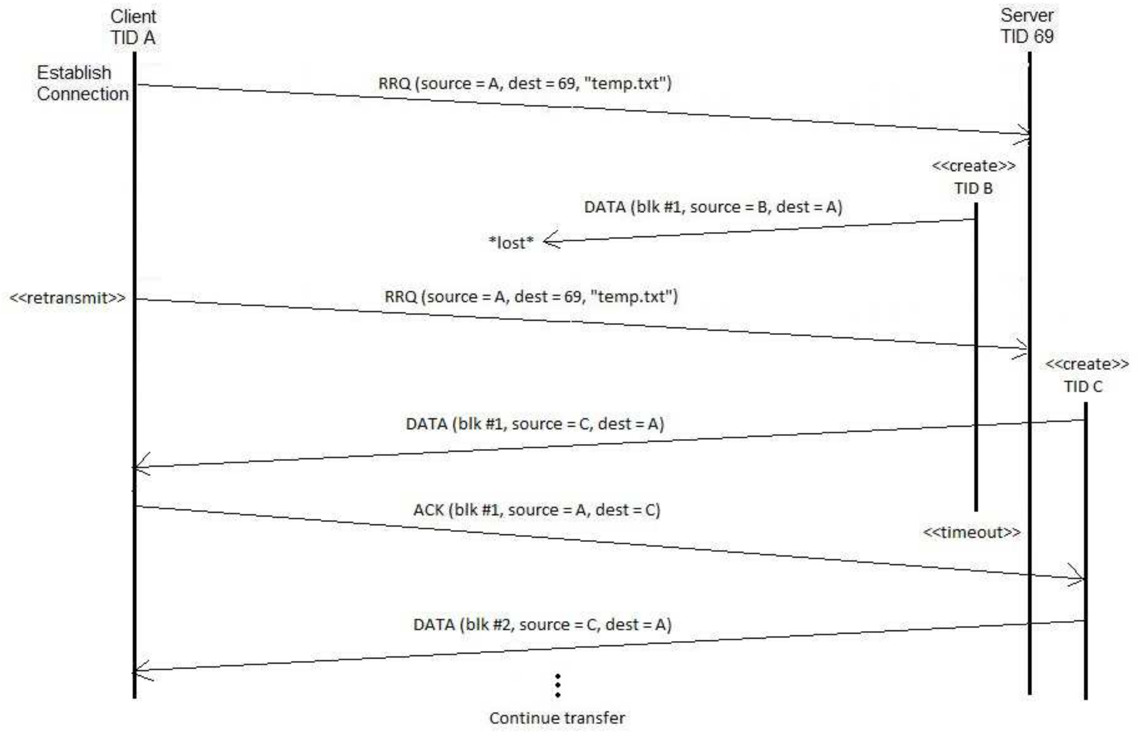

Client
TID A
Server
TID 69
Establish
Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
DATA (blk #1, source = A, dest = B)
ACK (blk #1, source = B, dest = A)
duplicate
DATA (blk #2, source = A, dest = B)
<< ignored>>
Continue transfer

[Duplicate Errors]

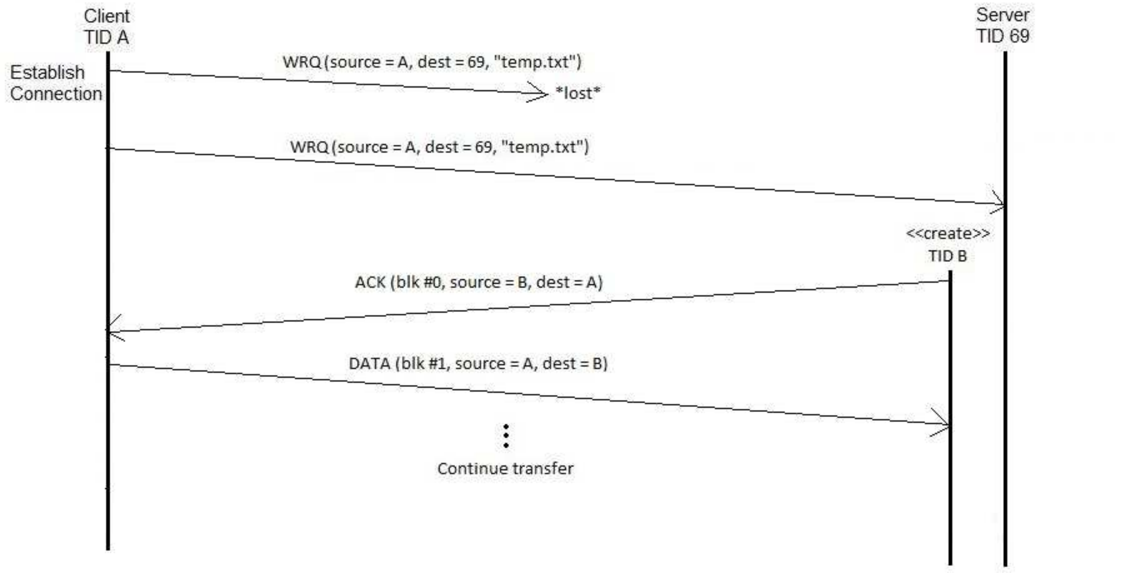
Scenario 1 – Client RRQ Lost:


Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69, "temp.txt")
lost
<<retransmit>>
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
DATA (blk #2, source = B, dest = A)
Continue transfer

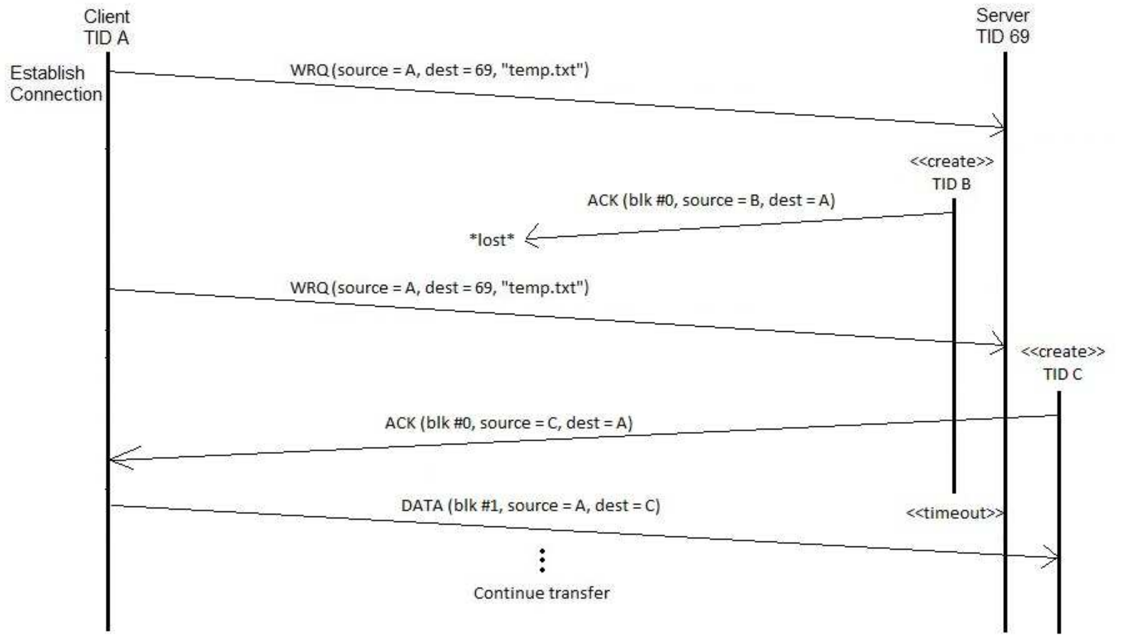
Scenario 2 – Server Response to RRQ Lost:


Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
lost
<<retransmit>>
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID C
DATA (blk #1, source = C, dest = A)
ACK (blk #1, source = A, dest = C)
<<timeout>>
DATA (blk #2, source = C, dest = A)
Continue transfer

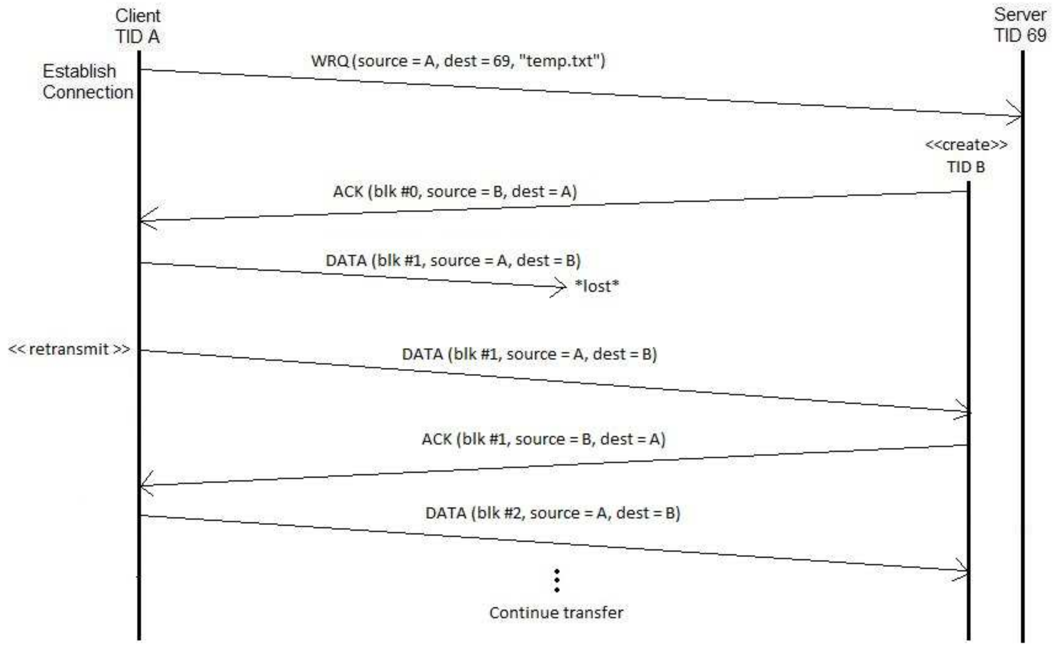
Scenario 3 – Client WRQ Lost:


Client
TID A
Server
TID 69
Establish Connection
WRQ (source = A, dest = 69, "temp.txt")
lost
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
DATA (blk #1, source = A, dest = B)
Continue transfer

Scenario 4 – Server Response to WRQ Lost:

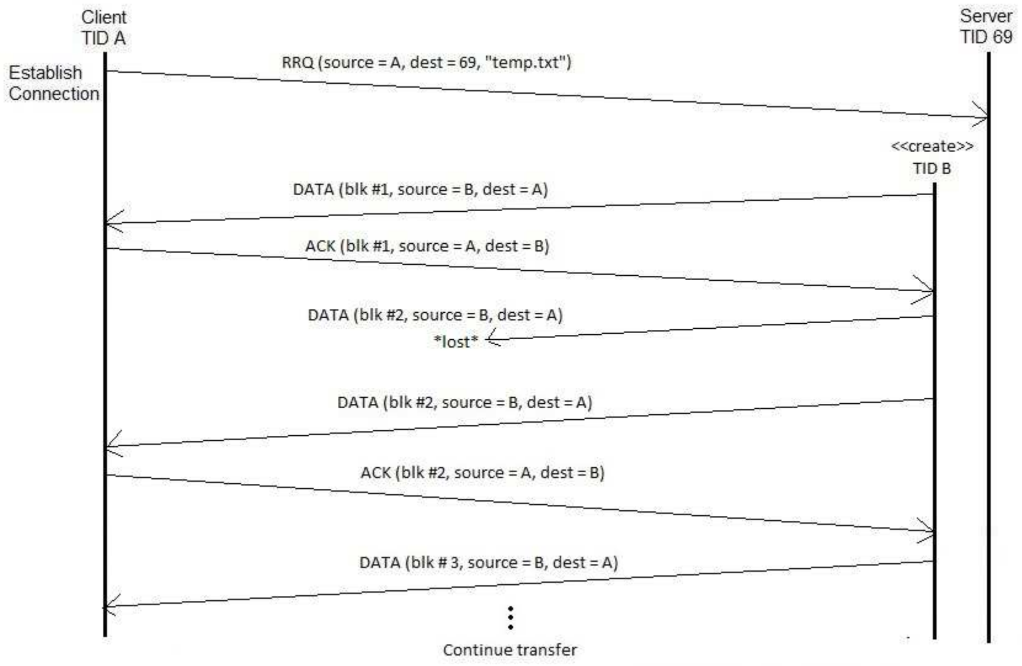

Client
TID A
Server
TID 69
Establish Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
lost
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID C
ACK (blk #0, source = C, dest = A)
DATA (blk #1, source = A, dest = C)
<<timeout>>
Continue transfer

Scenario 5 – Client Data Lost:



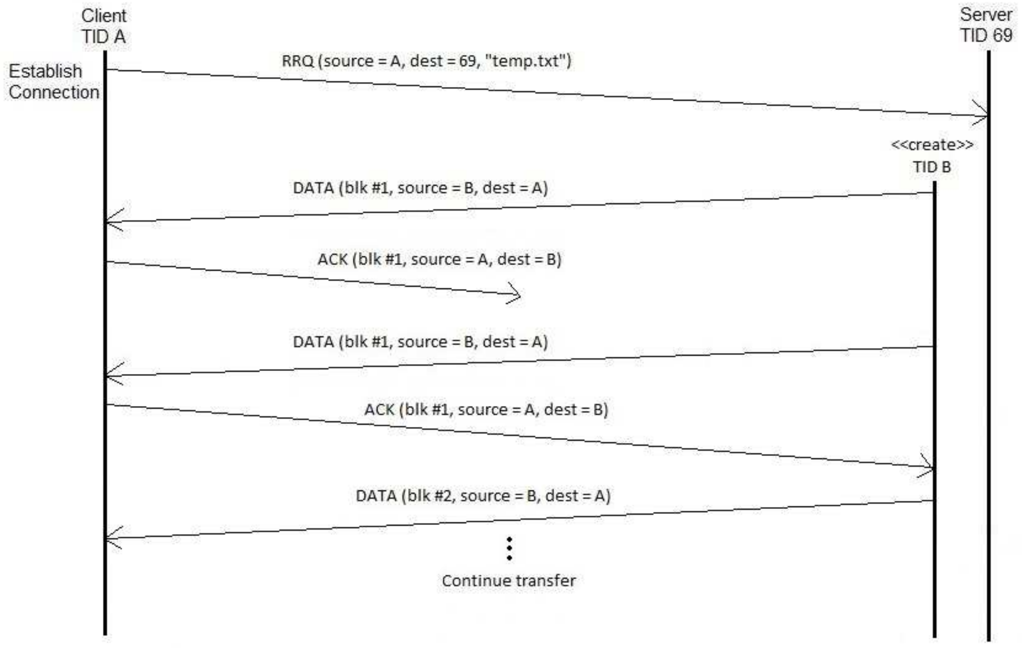
Client
TID A
Server
TID 69
Establish Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
DATA (blk #1, source = A, dest = B)
lost
<< retransmit >>
DATA (blk #1, source = A, dest = B)
ACK (blk #1, source = B, dest = A)
DATA (blk #2, source = A, dest = B)
Continue transfer

Scenario 6 – Server Data Lost:

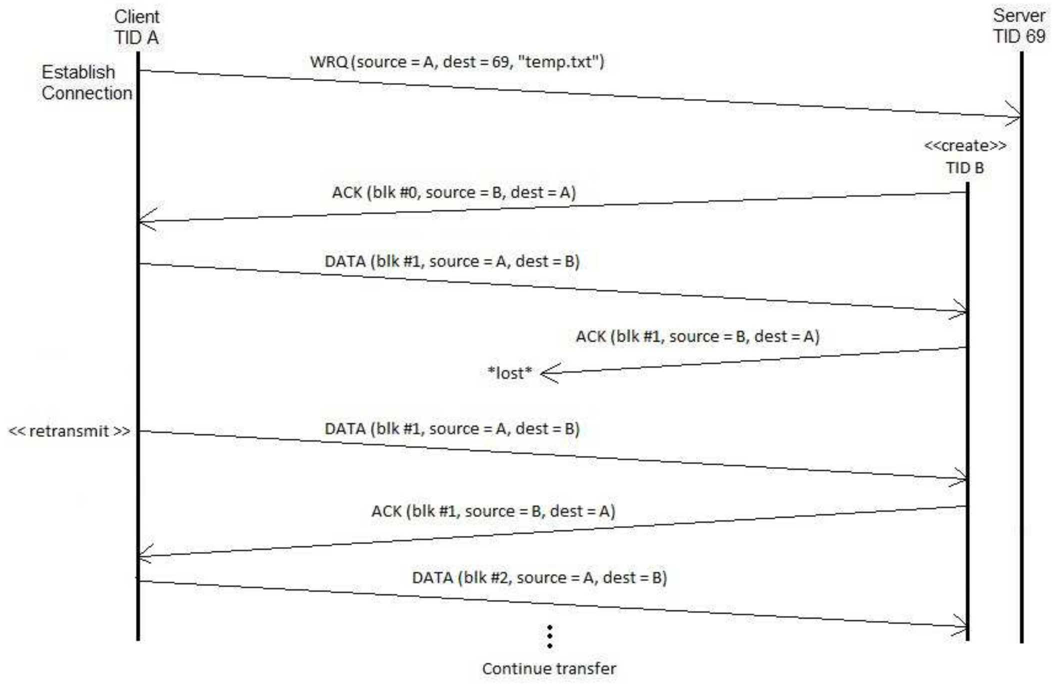


Client
TID A
Server
TID 69
Establish Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
DATA (blk #2, source = B, dest = A)
lost
DATA (blk #2, source = B, dest = A)
ACK (blk #2, source = A, dest = B)
DATA (blk # 3, source = B, dest = A)
Continue transfer

Scenario 7 – Client Ack Lost:

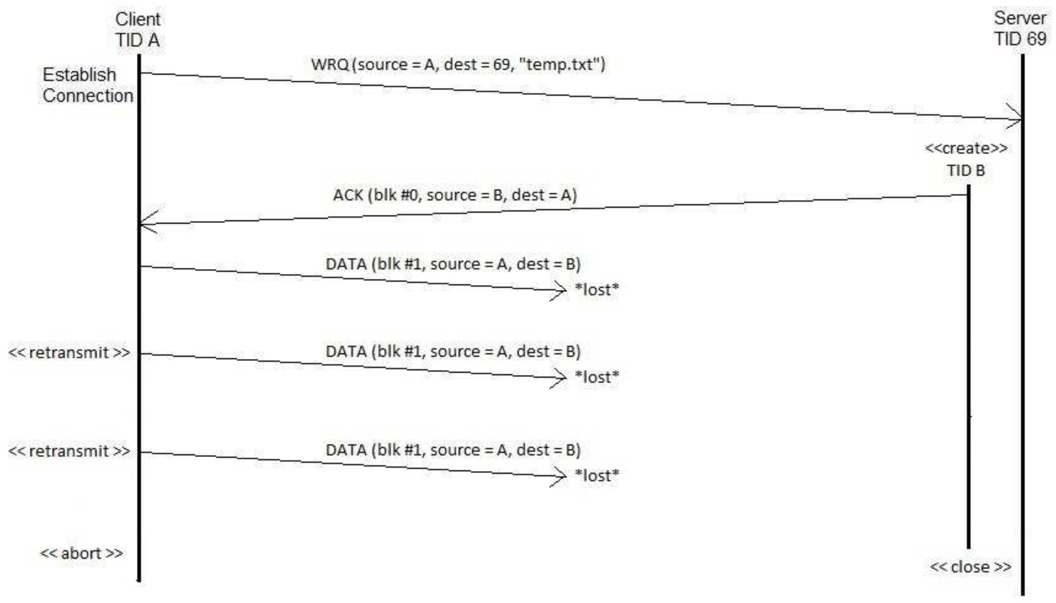

Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
DATA (blk #2, source = B, dest = A)
Continue transfer

Scenario 8 – Server Ack Lost:

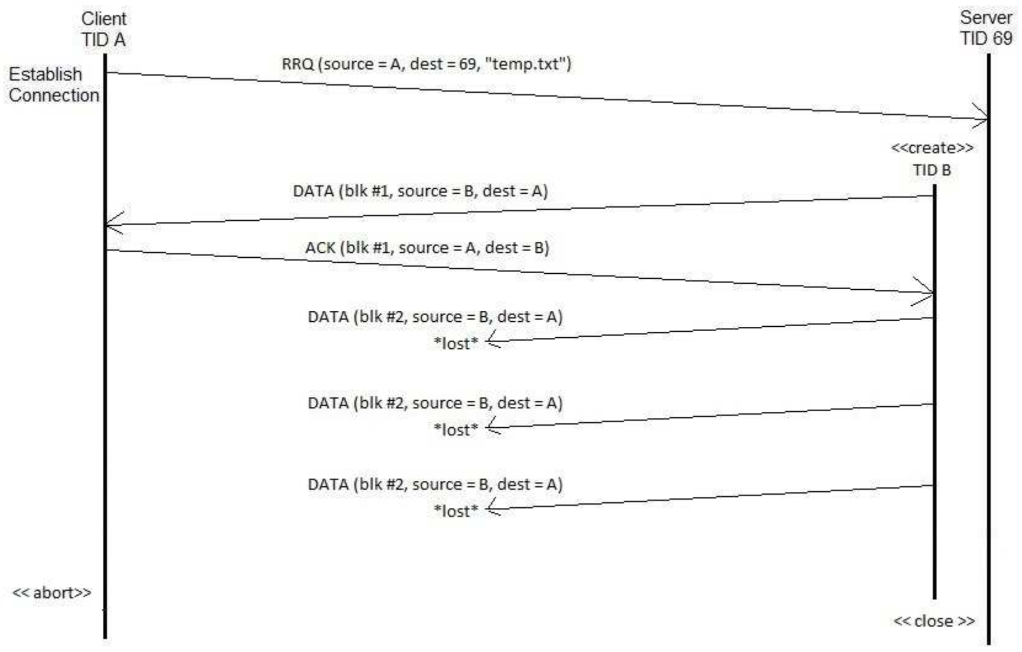

Client
TID A
Server
TID 69
Establish
Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
DATA (blk #1, source = A, dest = B)
ACK (blk #1, source = B, dest = A)
lost
<< retransmit >>
DATA (blk #1, source = A, dest = B)
ACK (blk #1, source = B, dest = A)
DATA (blk #2, source = A, dest = B)
Continue transfer

[Connection Terminated Errors]

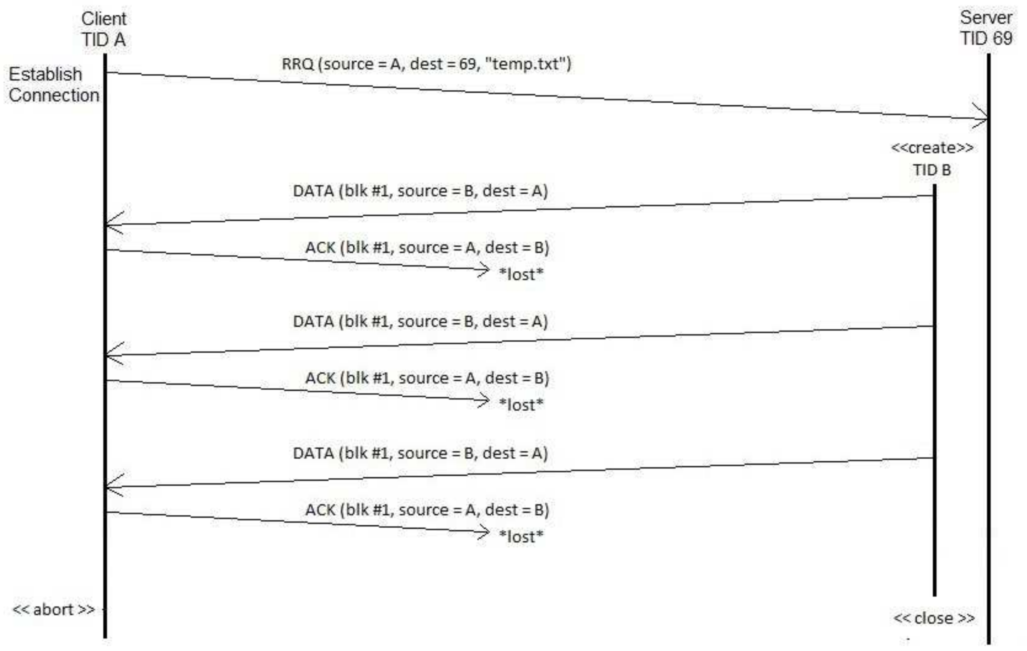
Scenario 1 – Client Data Connection Terminated:


Client
TID A
Server
TID 69
Establish Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
DATA (blk #1, source = A, dest = B)
lost
<< retransmit >>
DATA (blk #1, source = A, dest = B)
lost
<< retransmit >>
DATA (blk #1, source = A, dest = B)
lost
<< abort >>
<< close >>

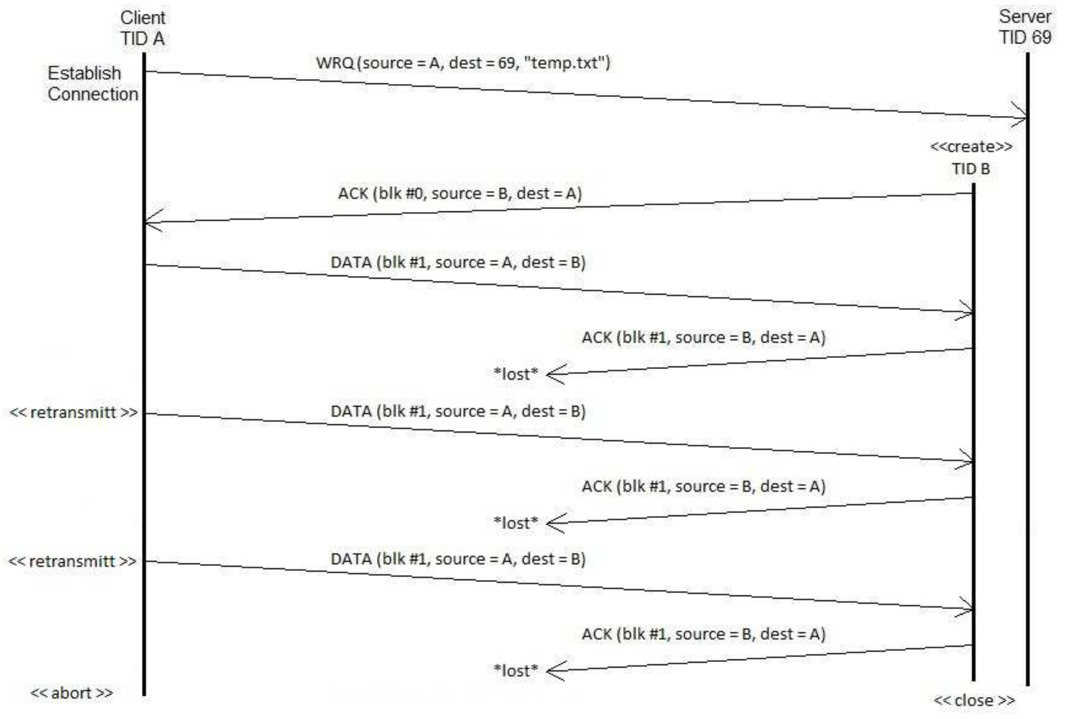
Scenario 2 – Server Data Connection Terminated:


Client
TID A
Server
TID 69
Establish Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
DATA (blk #2, source = B, dest = A)
lost
DATA (blk #2, source = B, dest = A)
lost
DATA (blk #2, source = B, dest = A)
lost
<< abort>>
<< close >>

Scenario 3 – Client Ack Connection Terminated:


Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
lost
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
lost
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
lost
<< abort >>
<< close >>

Scenario 4 – Server Ack Connection Terminated:


Client
TID A
Server
TID 69
Establish
Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
DATA (blk #1, source = A, dest = B)
ACK (blk #1, source = B, dest = A)
lost
<< retransmitt >>
DATA (blk #1, source = A, dest = B)
ACK (blk #1, source = B, dest = A)
lost
<< retransmitt >>
DATA (blk #1, source = A, dest = B)
ACK (blk #1, source = B, dest = A)
lost
<< abort >>
<< close >>

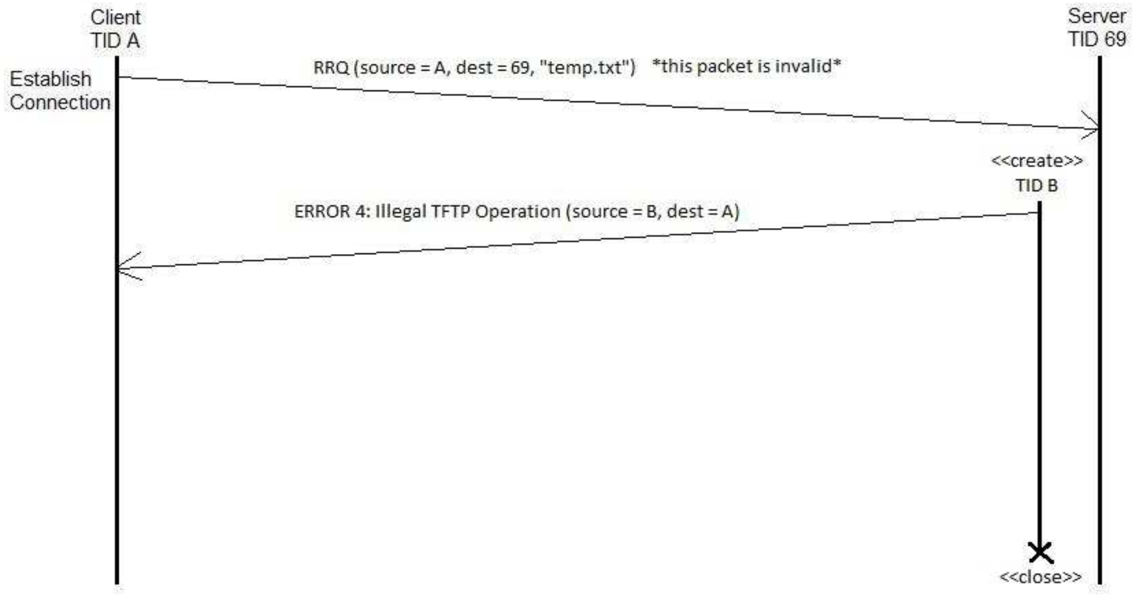
END OF DIAGRAMS FOR ITERATION 4

Timing diagrams for iteration #3

[Error Code 4]

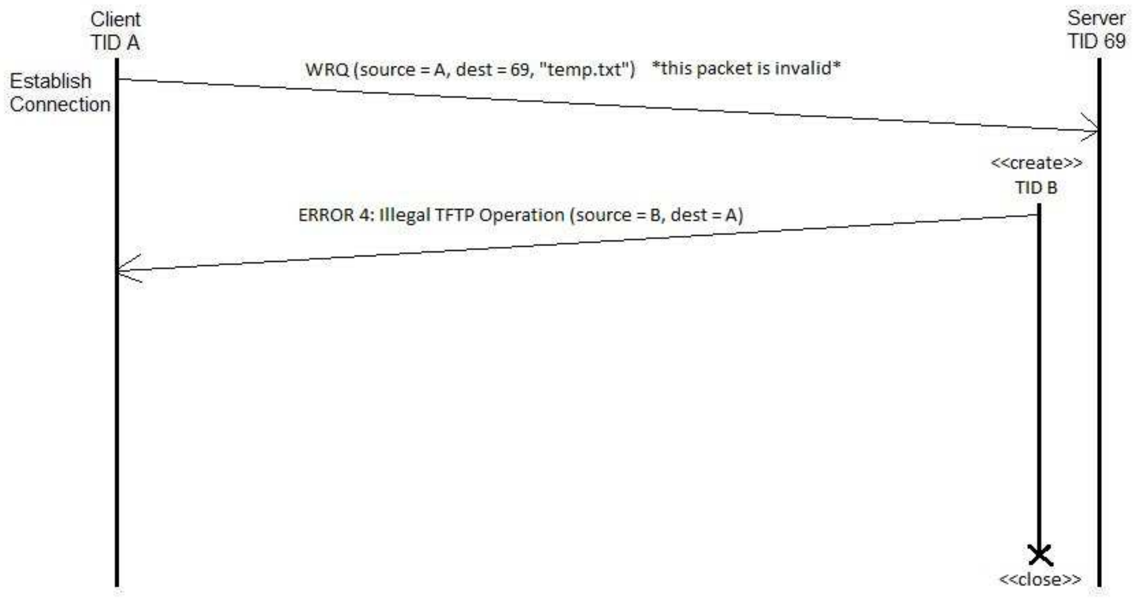
Scenario 1 – Client sent invalid RRQ:

1. Client sends invalid RRQ.
2. Server detects that RRQ packet is invalid.
3. Server forms error packet and sends back to client.
4. Server closes its socket with client.
5. Client prints an error message to screen and continues.


Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69, "temp.txt") *this packet is invalid*
<<create>>
TID B
ERROR 4: Illegal TFTP Operation (source = B, dest = A)
<<close>>

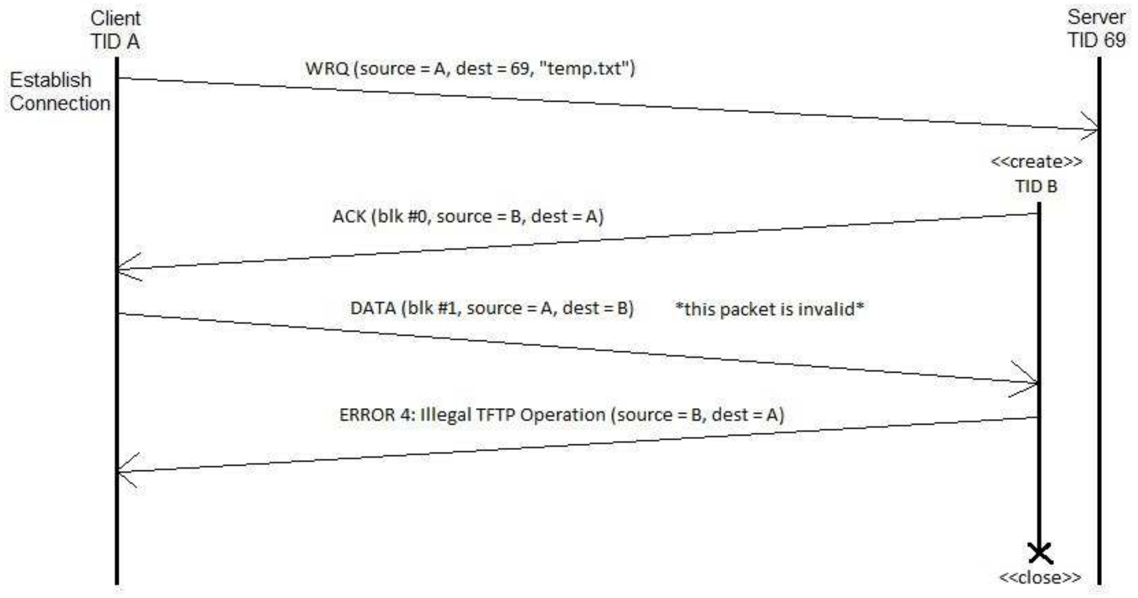
Scenario 2 – Client sent invalid WRQ:

1. Client sends invalid WRQ.
2. Server detects that WRQ packet is invalid.
3. Server forms error packet and sends back to client.
4. Server closes its socket with client.
5. Client prints an error message to screen and continues.


Client
TID A
Server
TID 69
Establish
Connection
WRQ (source = A, dest = 69, "temp.txt") *this packet is invalid*
<<create>>
TID B
ERROR 4: Illegal TFTP Operation (source = B, dest = A)
<<close>>

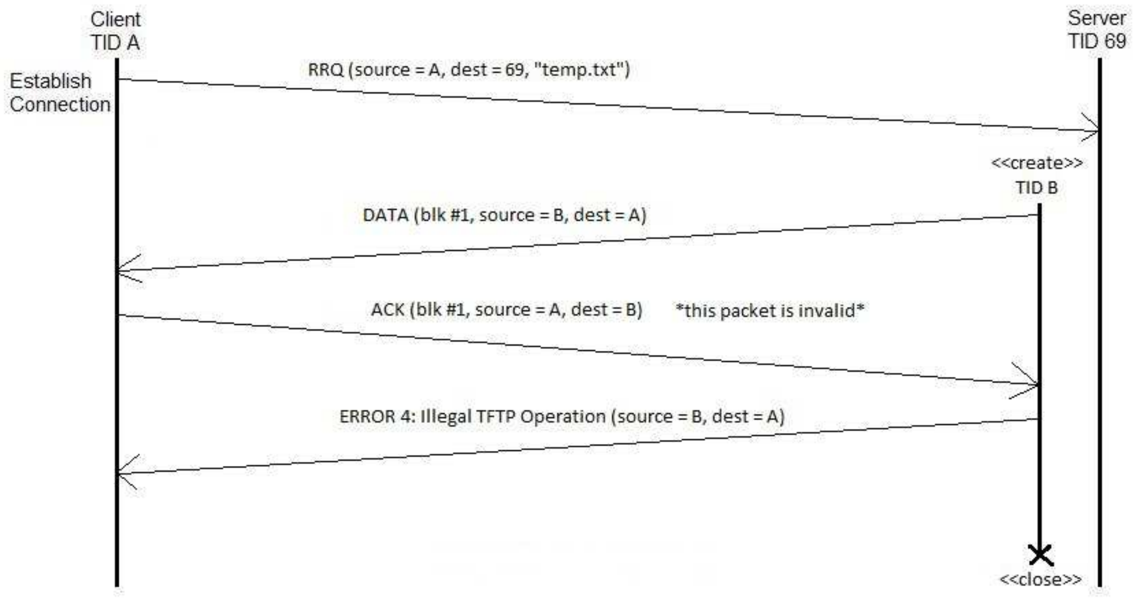
Scenario 3 – Client sent invalid DATA:

1. Client sends WRQ.
2. Server receives WRQ
3. Server forms ACK packet 0 and sends back to client.
4. Client receives ACK packet 0
5. Client sends invalid DATA packet 1
6. Server detects that DATA packet 1 is invalid.
7. Server forms error packet and sends back to client.
8. Server closes its socket with client.
9. Server prints an error message to screen and continues.
10. Client prints an error message to screen and continues.


Client
TID A
Server
TID 69
Establish
Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
DATA (blk #1, source = A, dest = B)
this packet is invalid
ERROR 4: Illegal TFTP Operation (source = B, dest = A)
<<close>>

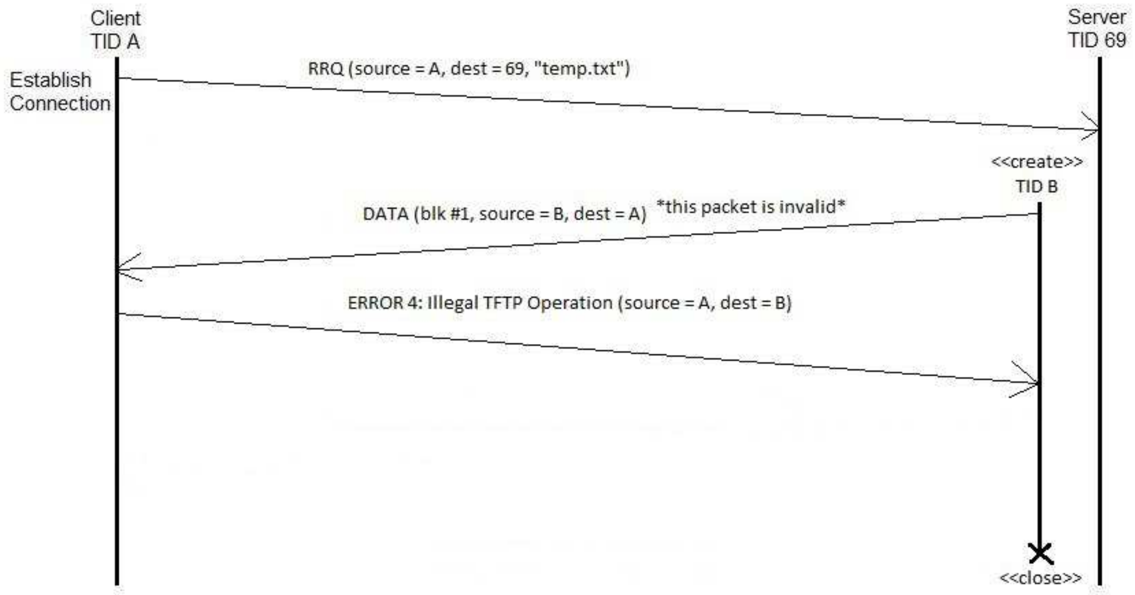
Scenario 4 – Client sent invalid ACK:

1. Client sends RRQ.
2. Server receives RRQ
3. Server forms DATA packet 1 and sends back to client.
4. Client receives DATA packet 1
5. Client sends invalid ACK packet 1
6. Server detects that ACK packet 1 is invalid.
7. Server forms error packet and sends back to client.
8. Server closes its socket with client.
9. Server prints an error message to screen and continues.
10. Client prints an error message to screen and continues.


Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = A, dest = B)
this packet is invalid
ERROR 4: Illegal TFTP Operation (source = B, dest = A)
<<close>>

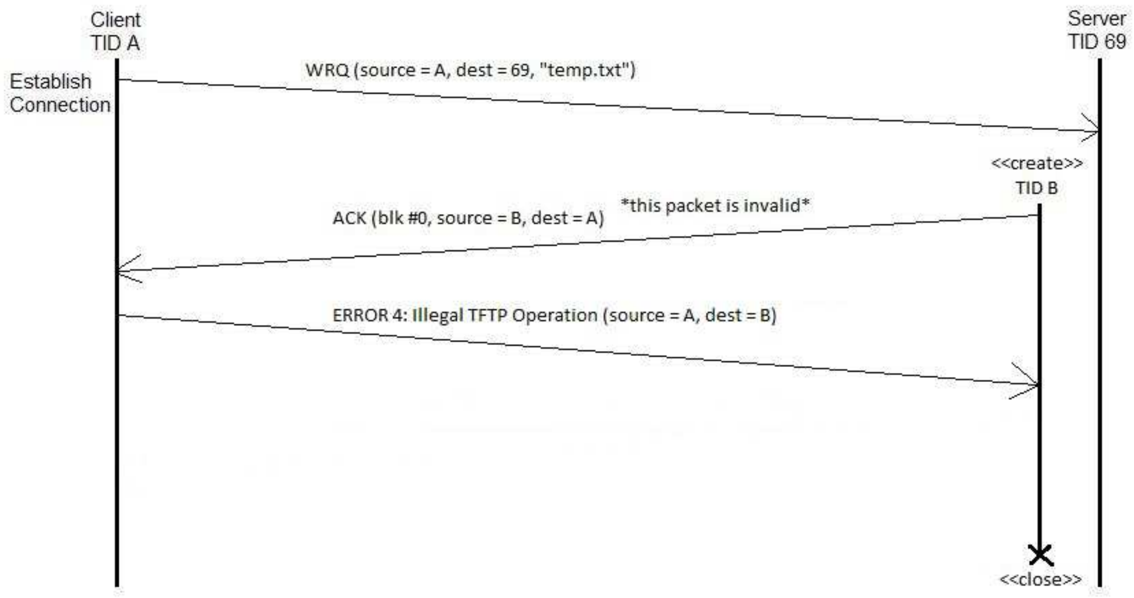
__Scenario 5 – Server sent invalid DATA:__

1. Client sends RRQ.
2. Server receives RRQ
3. Server forms invalid DATA packet 1 and sends back to client.
4. Client receives DATA packet 1
5. Client detects that DATA packet 1 is invalid.
6. Client forms error packet and sends back to server.
7. Server closes its socket with client.
8. Client prints an error message to screen and continues


Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
this packet is invalid
ERROR 4: Illegal TFTP Operation (source = A, dest = B)
<<close>>

Scenario 6 – Server sent invalid ACK:

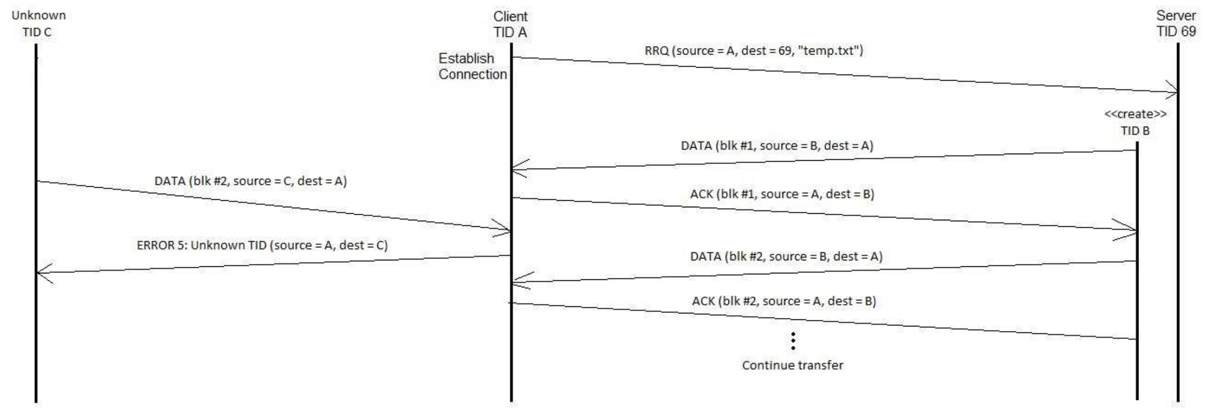
1. Client sends WRQ.
2. Server receives WRQ
3. Server forms invalid ACK packet 0 and sends back to client.
4. Client receives ACK packet 0
5. Client detects that ACK packet 0 is invalid.
6. Client forms error packet and sends back to server.
7. Server closes its socket with client.
8. Client prints an error message to screen and continues


Client
TID A
Server
TID 69
Establish
Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
this packet is invalid
ACK (blk #0, source = B, dest = A)
ERROR 4: Illegal TFTP Operation (source = A, dest = B)
<<close>>

[Error Code 5]

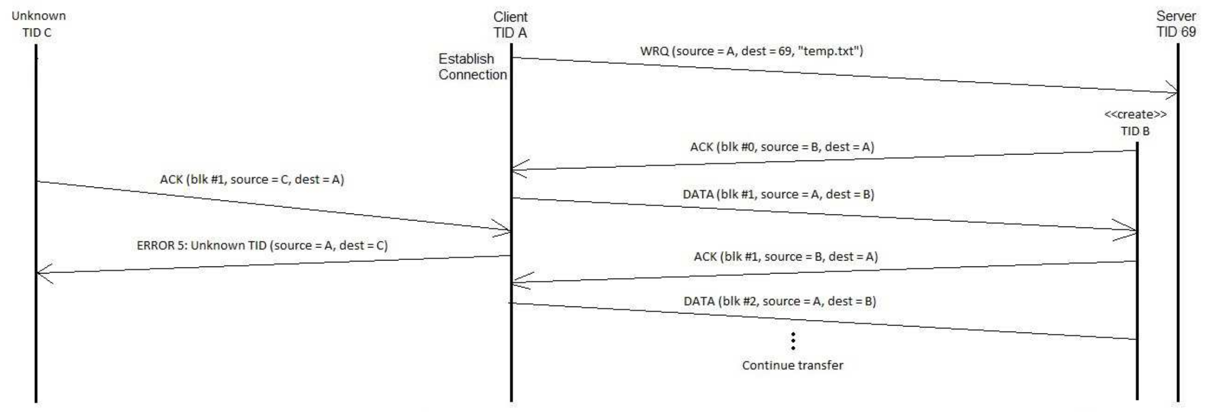
Scenario 1 – Client receives unknown packet on RRQ:

1. Client sends RRQ.
2. Server receives RRQ
3. Server forms DATA packet 1 and sends back to client.
4. Client receives DATA packet 1
5. Client forms ACK packet 1 and sends back to server
6. Unknown host sends DATA packet 2 to client
7. Client receives packet and detects that packet came from unknown TID and sends back an ERROR code 4 packet to unknown TID
8. Server receives ACK packet 1
9. Server forms DATA packet 2 and sends back to client
10. Client receives DATA packet 2
11. Client forms ACK packet 2 and sends back to server
12. Transfer continues as expected


Unknown TID C
Client TID A
Server TID 69
Establish Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>> TID B
DATA (blk #1, source = B, dest = A)
DATA (blk #2, source = C, dest = A)
ACK (blk #1, source = A, dest = B)
ERROR 5: Unknown TID (source = A, dest = C)
DATA (blk #2, source = B, dest = A)
ACK (blk #2, source = A, dest = B)
Continue transfer

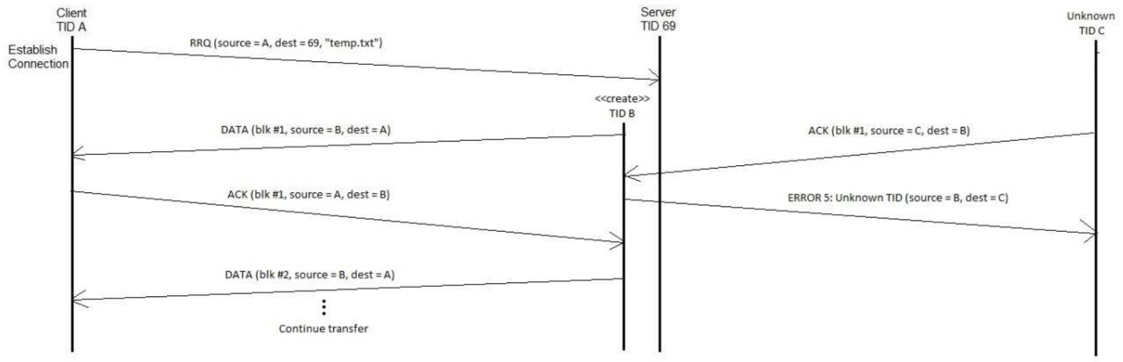
Scenario 2 – Client receives unknown packet on WRQ:

1. Client sends WRQ.
2. Server receives WRQ
3. Server forms ACK packet 0 and sends back to client.
4. Client receives ACK packet 0
5. Client forms DATA packet 1 and sends back to server
6. Unknown host sends ACK packet 1 to client
7. Client receives packet and detects that packet came from unknown TID and sends back an ERROR code 4 packet to unknown TID
8. Server receives DATA packet 1
9. Server forms ACK packet 1 and sends back to client
10. Client receives ACK packet 1
11. Client forms DATA packet 2 and sends back to server
12. Transfer continues as expected


Unknown
TID C
Client
TID A
Server
TID 69
Establish
Connection
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
ACK (blk #1, source = C, dest = A)
DATA (blk #1, source = A, dest = B)
ERROR 5: Unknown TID (source = A, dest = C)
ACK (blk #1, source = B, dest = A)
DATA (blk #2, source = A, dest = B)
Continue transfer

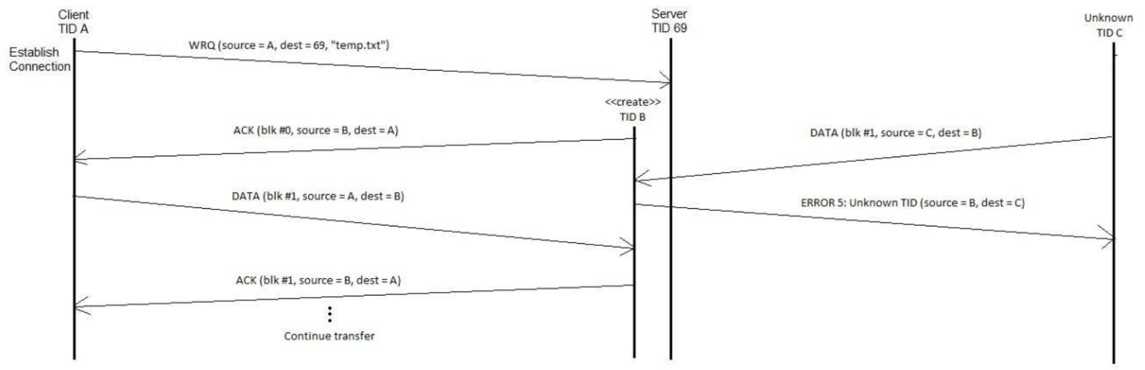
Scenario 3 – Server receives unknown packet on RRQ:

1. Client sends RRQ.
2. Server receives RRQ
3. Server forms DATA packet 1 and sends back to client.
4. Client receives DATA packet 1
5. Unknown host sends ACK packet 1 to server
6. Server receives packet and detects that packet came from unknown TID and sends back an ERROR code 4 packet to unknown TID
7. Client receives DATA packet 1
8. Client forms ACK packet 1 and sends back to server
9. Server receives ACK packet 1
10. Server forms DATA packet 2 and sends back to client
11. Transfer continues as expected


Client
TID A
Establish Connection
Server
TID 69
Unknown
TID C
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
ACK (blk #1, source = C, dest = B)
ACK (blk #1, source = A, dest = B)
ERROR 5: Unknown TID (source = B, dest = C)
DATA (blk #2, source = B, dest = A)
Continue transfer

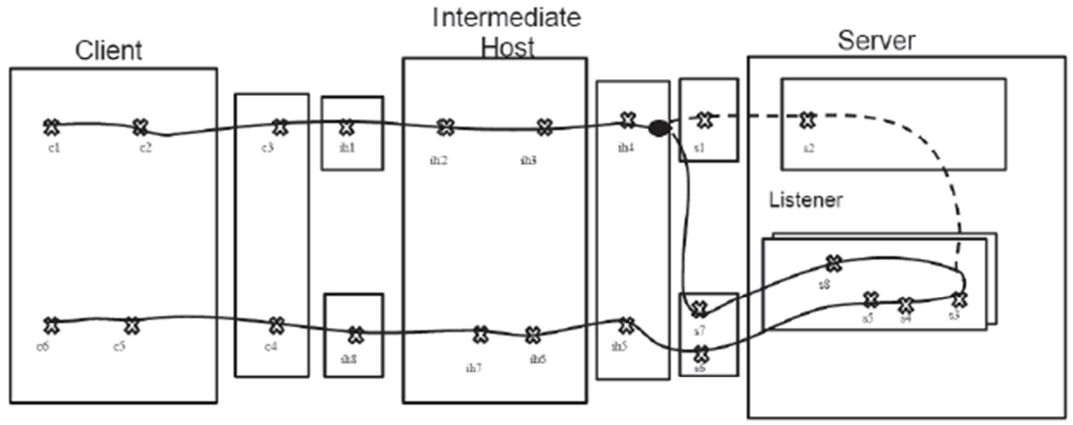
__Scenario 4 – Server receives unknown packet on WRQ:__

1. Client sends WRQ.
2. Server receives WRQ
3. Server forms ACK packet 0 and sends back to client.
4. Unknown host sends DATA packet 1 to server
5. Server receives packet and detects that packet came from unknown TID and sends back an ERROR code 4 packet to unknown TID
6. Client receives ACK packet 0
7. Client forms DATA packet 1 and sends back to server
8. Server receives DATA packet 1
9. Server forms ACK packet 1 and sends back to client
10. Transfer continues as expected


Client
TID A
Establish
Connection
Server
TID 69
Unknown
TID C
WRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
DATA (blk #1, source = C, dest = B)
DATA (blk #1, source = A, dest = B)
ERROR 5: Unknown TID (source = B, dest = C)
ACK (blk #1, source = B, dest = A)
Continue transfer

END OF DIAGRAMS FOR ITERATION 3

UCM Read Request:

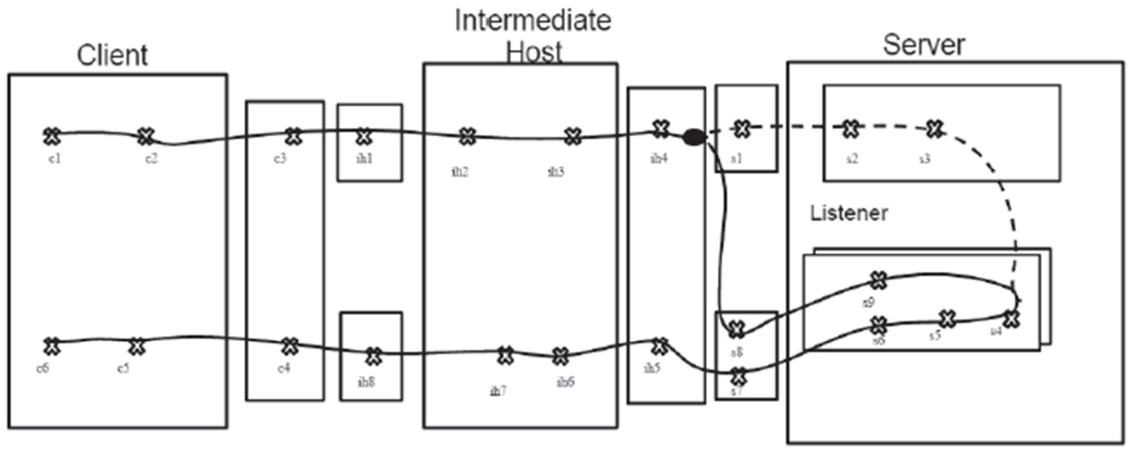


c1: form message
c2: create datagram
c3: send datagram
c4: wait and receive response
c5: extract message
c6: output message

ih1: wait and receive packet
ih2: extract message
ih3: form packet
ih4: send packet
ih5: wait and receive response
ih6: extract message
ih7: form packet
ih8: send packet

s1: receive datagram
s2: extract message
s3: verify message (quit if error)
s4: form response and data
s5: form datagram
s6: send datagram
s7: receive datagram
s8: extract message

UCM Write Request:



c1: form message
c2: create datagram
c3: send datagram
c4: wait and receive response
c5: extract message
c6: output message

ih1: wait and receive packet
ih2: extract message
ih3: form packet
ih4: send packet
ih5: wait and receive response
ih6: extract message
ih7: form packet
ih8: send packet

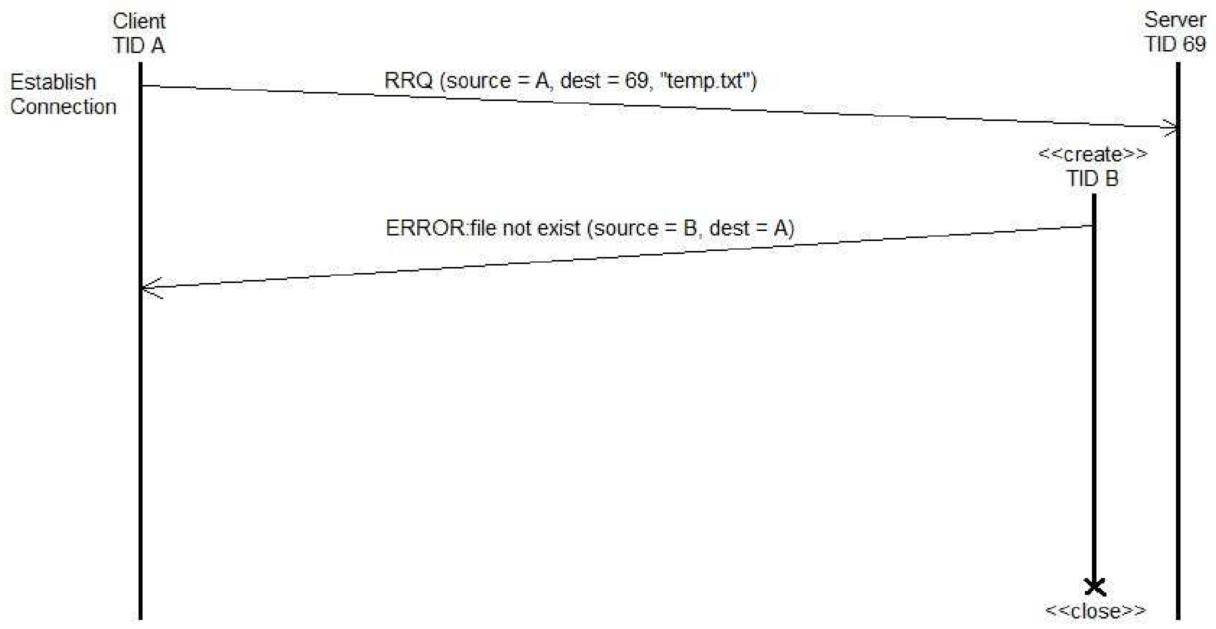
s1: receive datagram
s2: extract message
s3: create socket
s4: verify message (quit if error)
s5: form response and Ack
s6: form datagram
s7: send datagram
s8: receive datagram
s9: extract message

Timing diagrams for iteration #2

[Error Code 1]

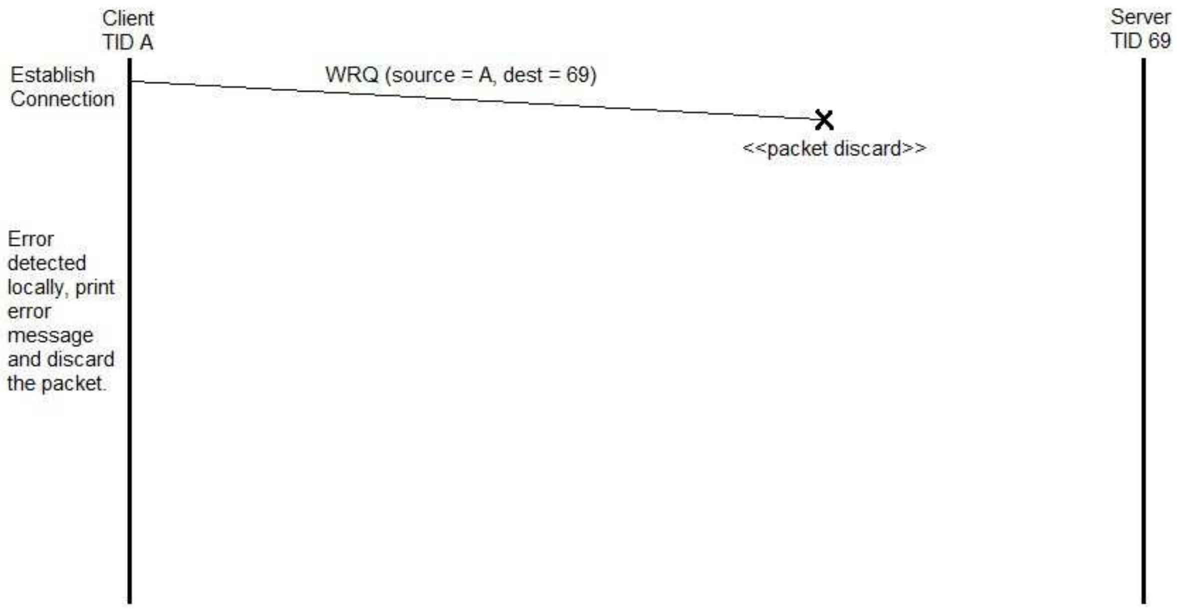
Scenario 1 - RRQ File not found on server:

1. Client sends RRQ.
2. Server detects that file does not exist.
3. Server forms error packet and sends back to client.
4. Server closes its socket with client.
5. Client prints an error message to screen and continues.


Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69, "temp.txt")
<<create>>
TID B
ERROR:file not exist (source = B, dest = A)
<<close>>

Scenario 2 - WRQ File not found on Client:

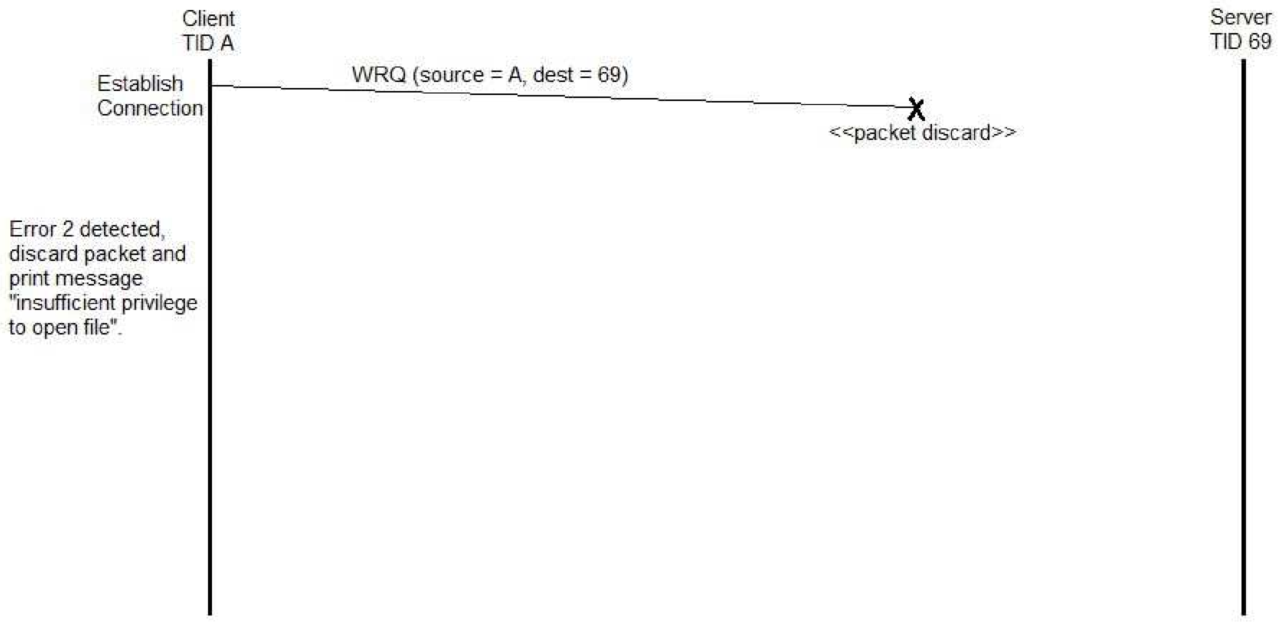
1. Client initiates WRQ.
2. File does not exist on client side.
3. Packet not sent, error message output to screen and prompts for new file name.


Client
TID A
Server
TID 69
Establish Connection
WRQ (source = A, dest = 69)
<<packet discard>>
Error detected locally, print error message and discard the packet.

[Error Code 2]

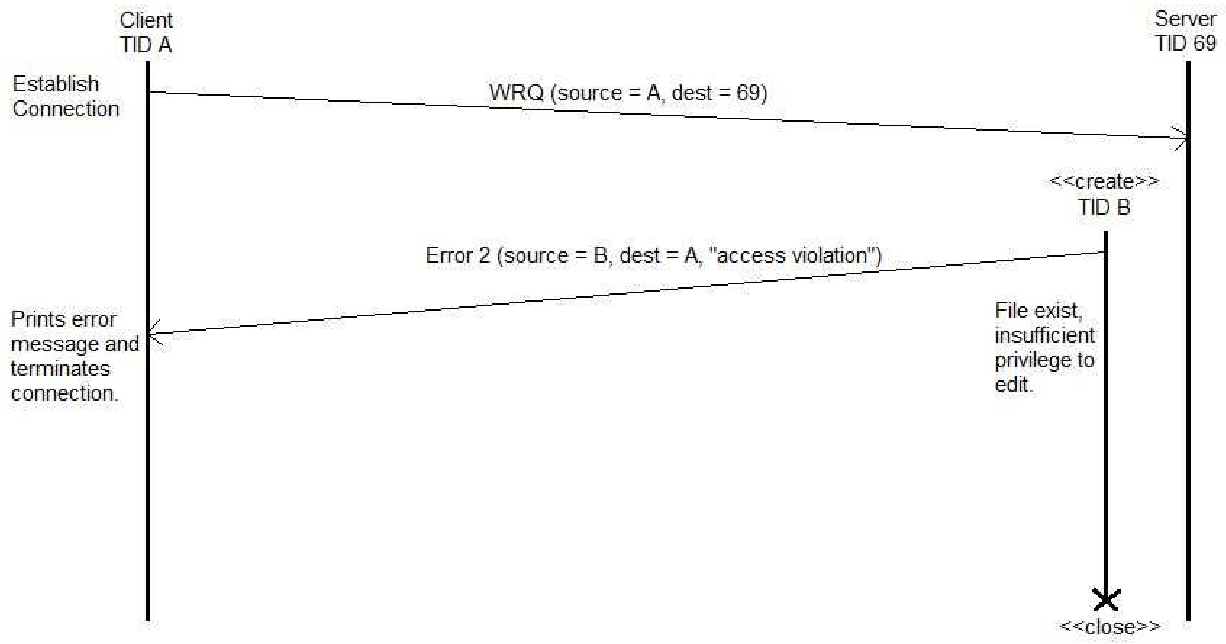
Scenario 1 - WRQ Access violation on client:

1. Client detects that it has insufficient privileges to open file to be sent.
2. Client outputs error message to screen and closes.



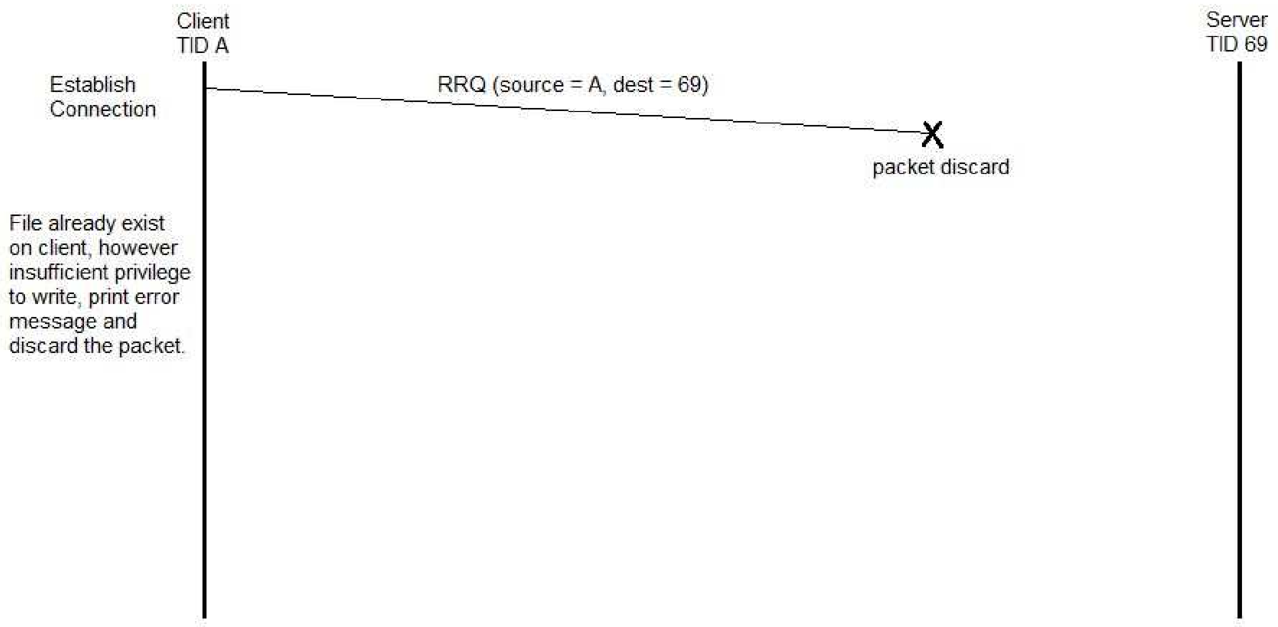
Scenario 2 - WRQ Access violation on server:

1. Client sends WRQ to server.
2. Server detects that file already exists and that client has insufficient privileges to write to file.
3. Server sends ERROR packet to client.
4. Server closes its connection with the client.
5. Client notifies user that write was unsuccessful.



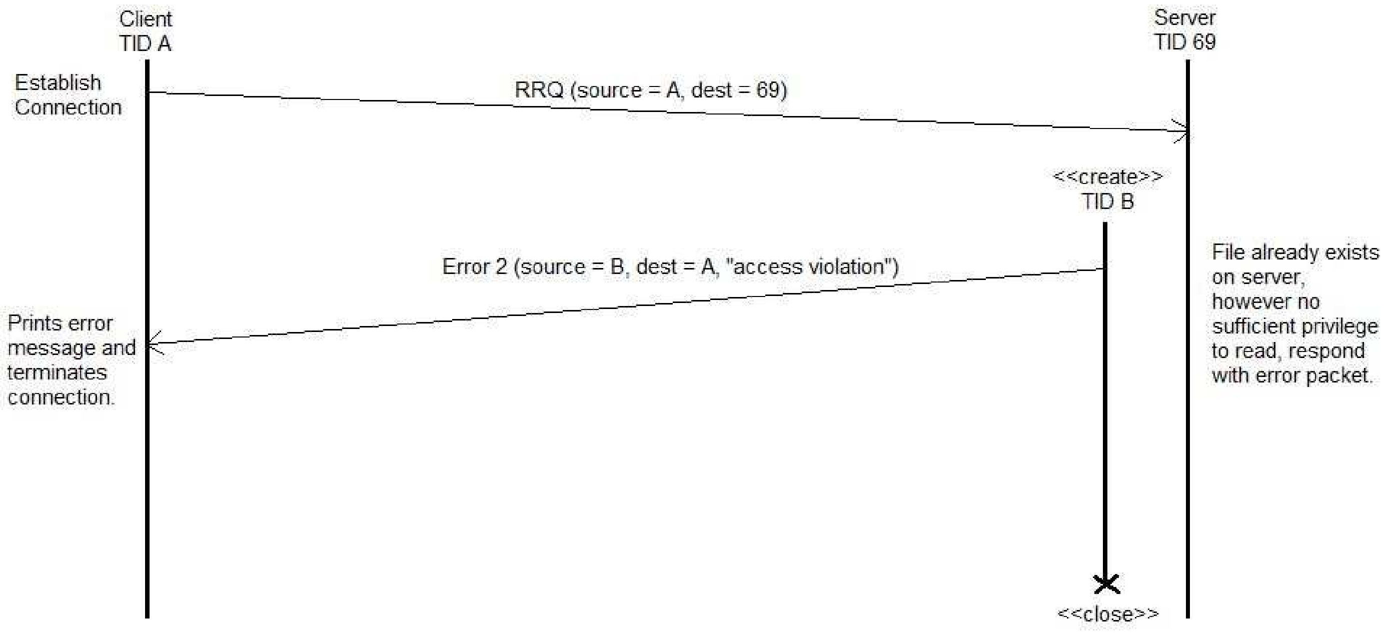
Scenario 3 - RRQ Access violation on client:

1. User initiates a RRQ.
2. Client detects that file already exists.
3. Client tries to open file but has insufficient privileges.
4. Client displays error message



Scenario 4 - RRQ Access violation on client:

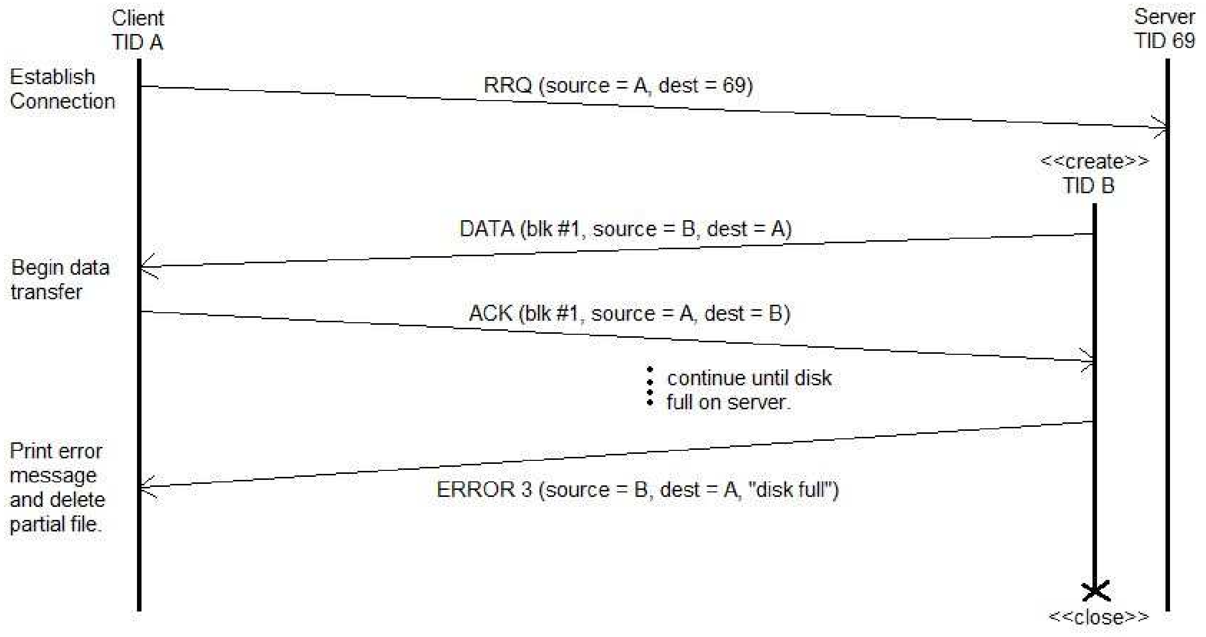
1. User initiates a RRQ
2. Client sends RRQ to Server
3. Server detects that file exists but has insufficient privileges to open
4. Server sends access violation ERROR (02) and quits
5. Error message displayed and quits


Client
TID A
Server
TID 69
Establish
Connection
RRQ (source = A, dest = 69)
<<create>>
TID B
Error 2 (source = B, dest = A, "access violation")
File already exists on server, however no sufficient privilege to read, respond with error packet.
Prints error message and terminates connection.
<<close>>

[Error Code 3]

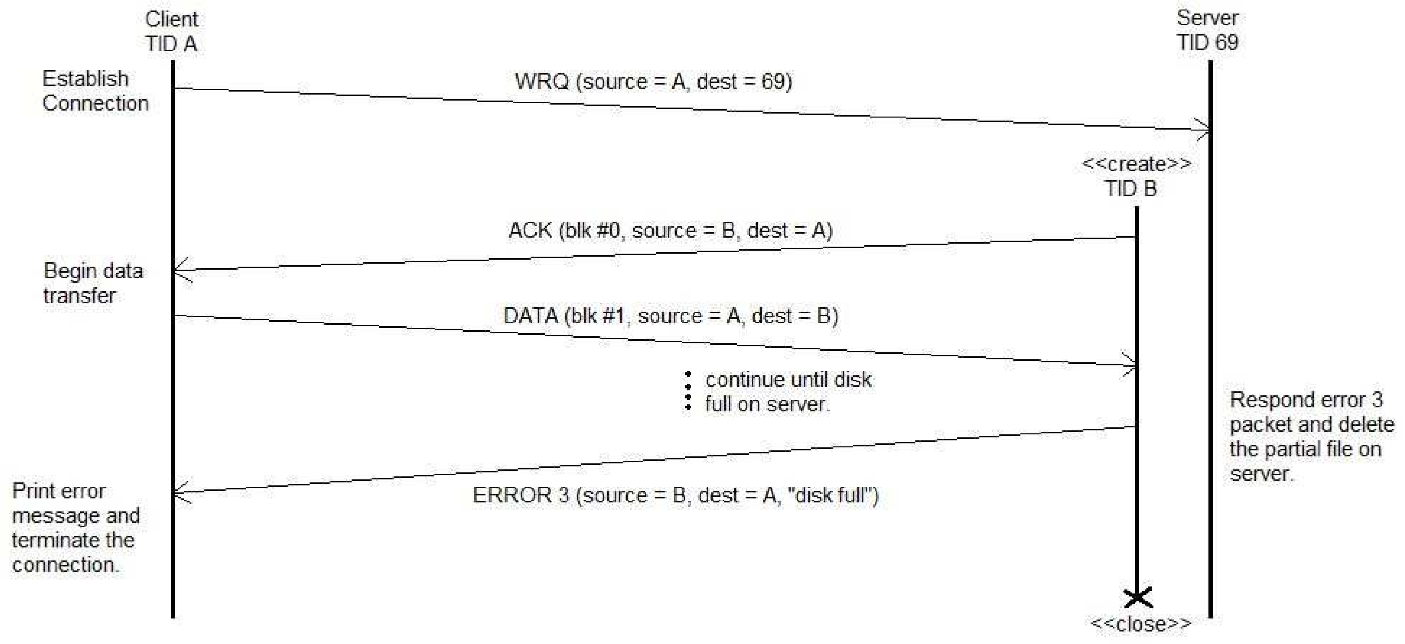
Scenario 1 - Disk full on client:

1. Client initiates RRQ
2. Server sends first data packet
3. Client acknowledges
4. Continues until client disk is full
5. Client sends Disk Full Error (03)
6. Server closes socket and closes file
7. Client deletes incomplete file and displays message


Client
TID A
Server
TID 69
Establish Connection
RRQ (source = A, dest = 69)
<<create>>
TID B
DATA (blk #1, source = B, dest = A)
Begin data transfer
ACK (blk #1, source = A, dest = B)
continue until disk full on server.
Print error message and delete partial file.
ERROR 3 (source = B, dest = A, "disk full")
<<close>>

Scenario 2 - Disk full on server:

1. Client initiates WRQ
2. Server responds with ACK
3. Client sends DATA
4. Continues until Server disk full
5. Server sends Disk Full Error (03)
6. Output message error message
7. Server closes socket and deletes incomplete file (what about while overwriting??)
8. Client closes file


Client
TID A
Server
TID 69
Establish Connection
WRQ (source = A, dest = 69)
<<create>>
TID B
ACK (blk #0, source = B, dest = A)
Begin data transfer
DATA (blk #1, source = A, dest = B)
continue until disk full on server.
Respond error 3 packet and delete the partial file on server.
Print error message and terminate the connection.
ERROR 3 (source = B, dest = A, "disk full")
<<close>>