

Fig 1: Acks of Acks trigger False fast rexmit, resulting in unnecessary rexmits

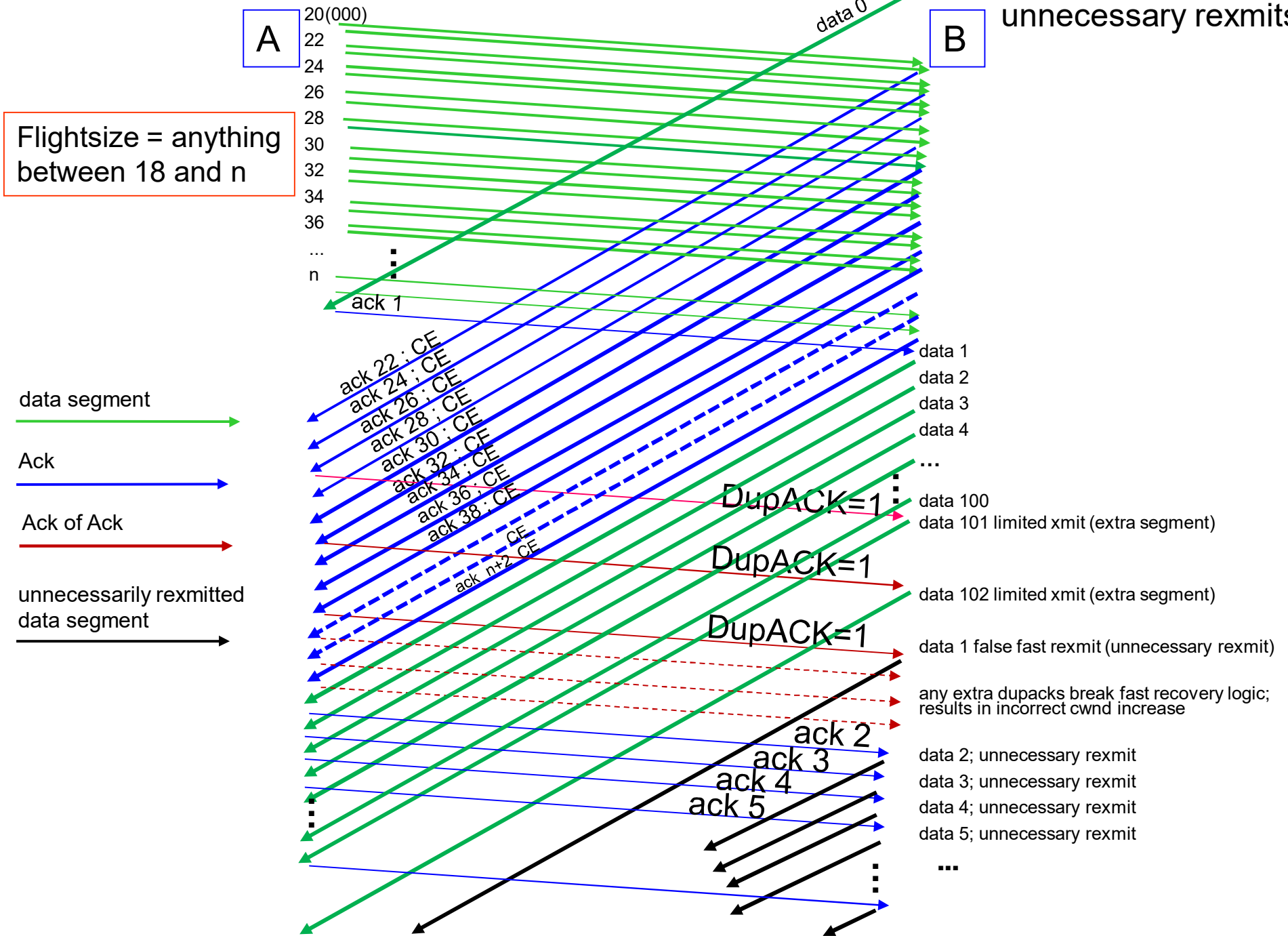


Fig 2. Acks of acks create unnecessary packets and add to the network load for no actual reason

data
←

ack
←

(Original Acks for data 100, 101, etc. are likely not sent as pure Acks but w/ data 2000, 2000+n, etc. from B to A)

ack&CE
←

ack of ack
←

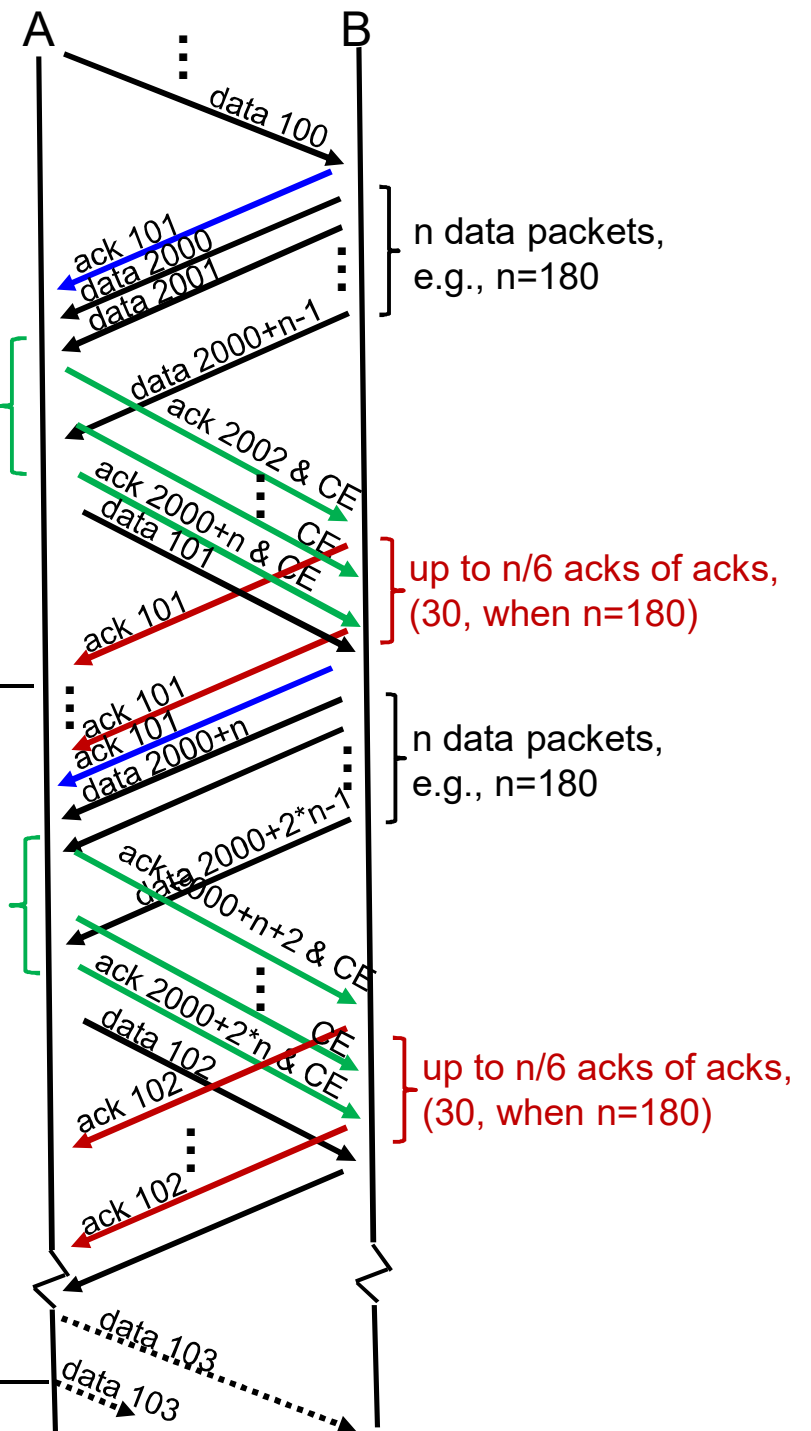
Figure is missing the subsequent rounds of ACKs of ACKs after each 1st round of ACKs of ACKs

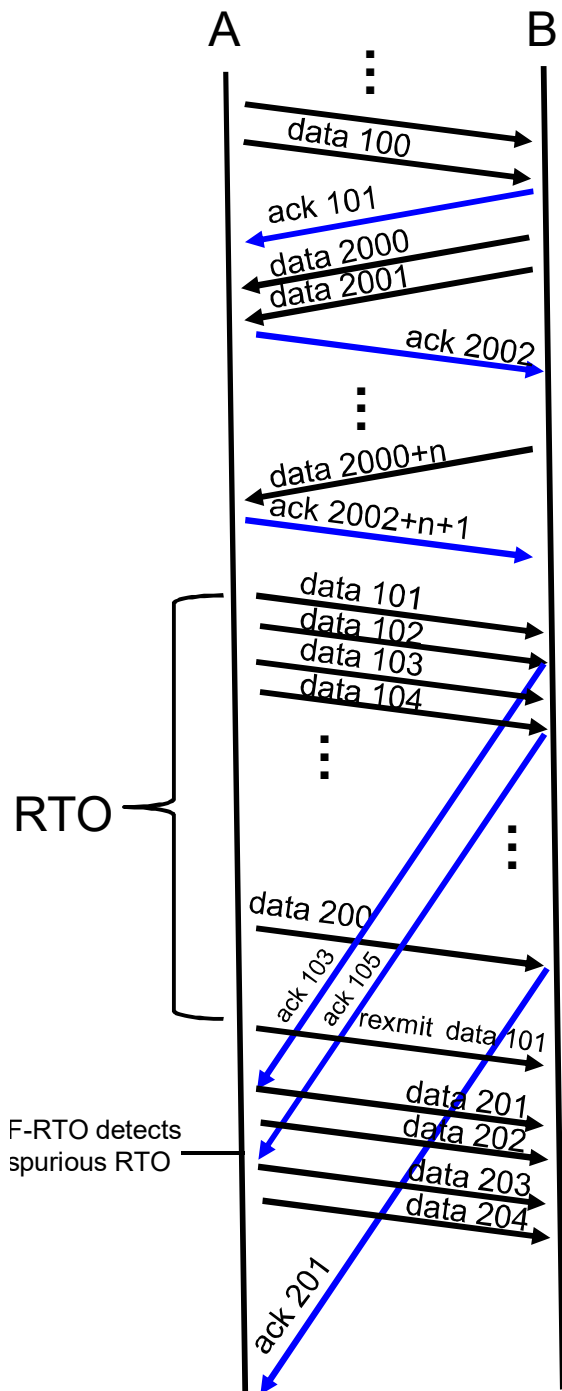
up to $n/2$ acks w/ CE (90, when $n=180$)

(acks of acks are dupacks and may trigger false fast retransmit quite unnecessarily)

up to $n/2$ acks w/ CE (90, when $n=180$)

If A w/ $cwnd=1MSS$ reacts to acks of acks reporting congestion by reducing its data rate, then next data request from A is unnecessarily delayed (while reducing ack rate from A would be the reasonable thing to do)





data
 ←
 ack
 ←
 ack&CE
 ←
 ack of ack (dupAck)
 ←

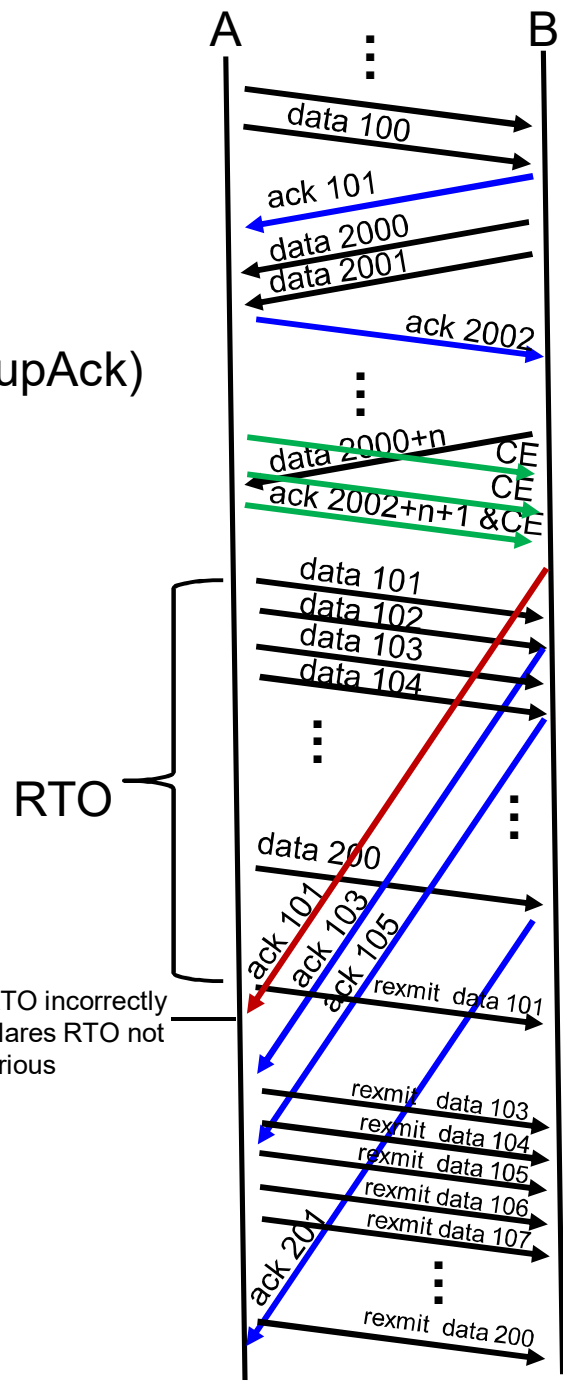


Fig 3 a) F-RTO correctly detects spurious RTO

Fig 3 b) Extra DupAck breaks F-RTO detection

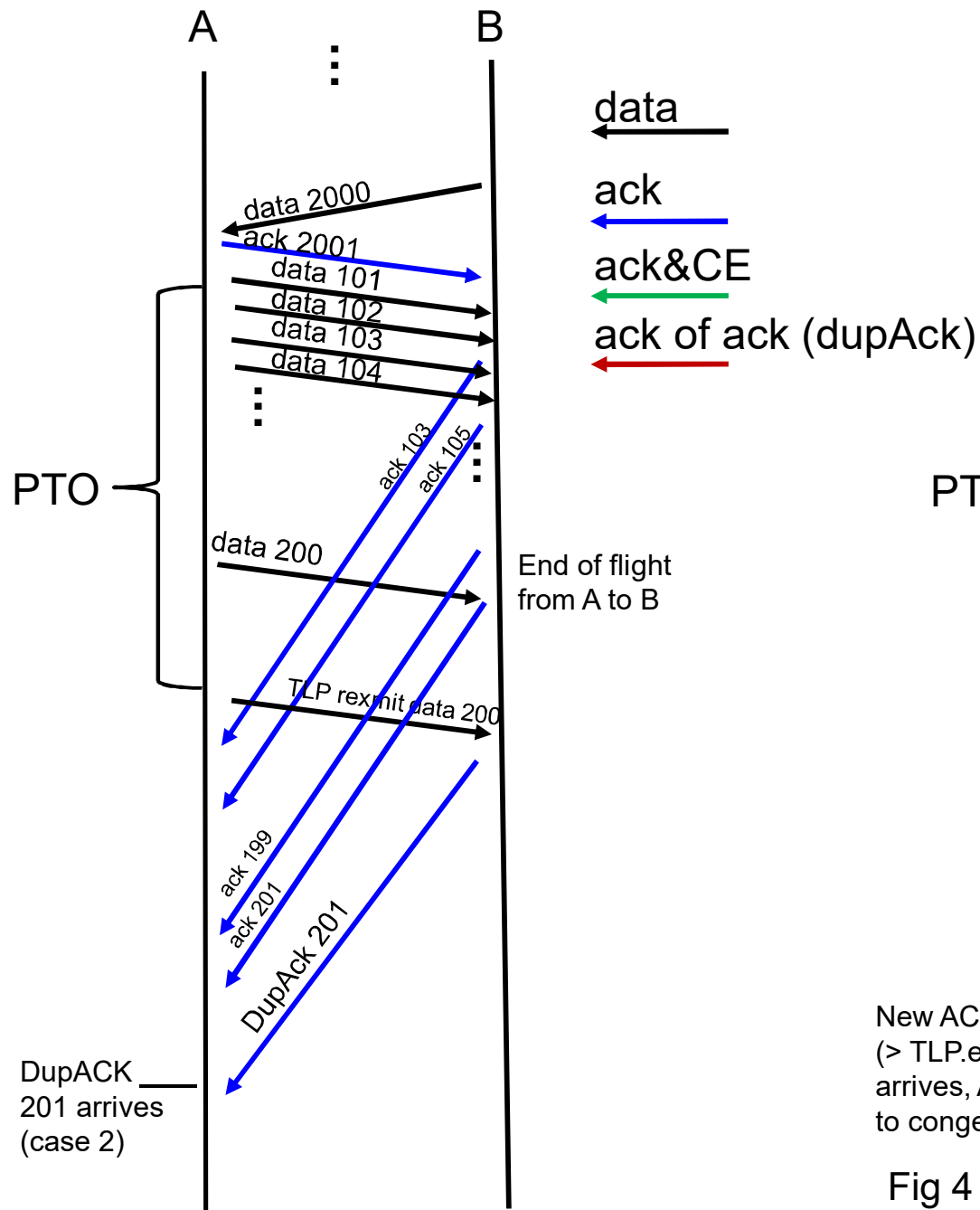


Fig 4 a) B does not implement DSACK, TLP correctly detects spurious PTO

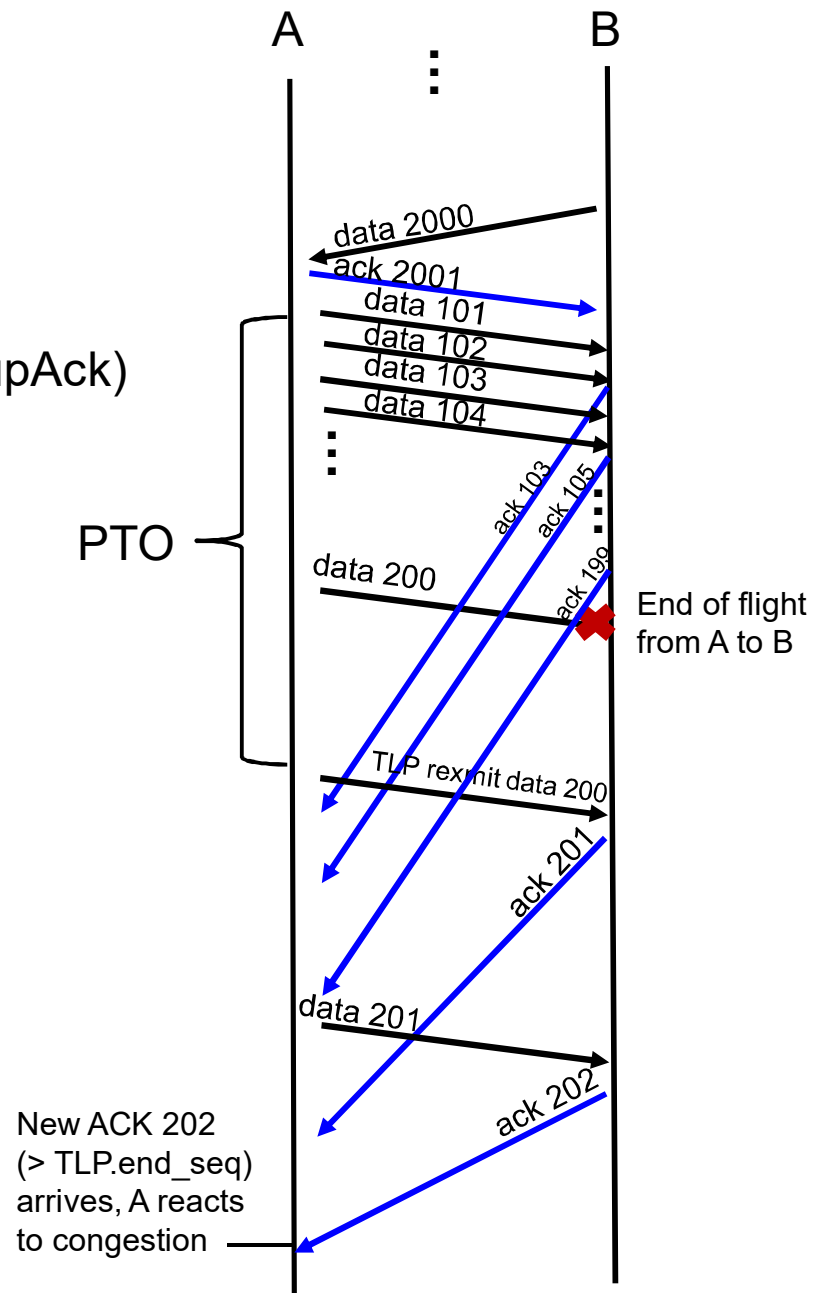


Fig 4 b)) B does not implement DSACK, TLP correctly detects loss (of data 200) and reacts to congestion

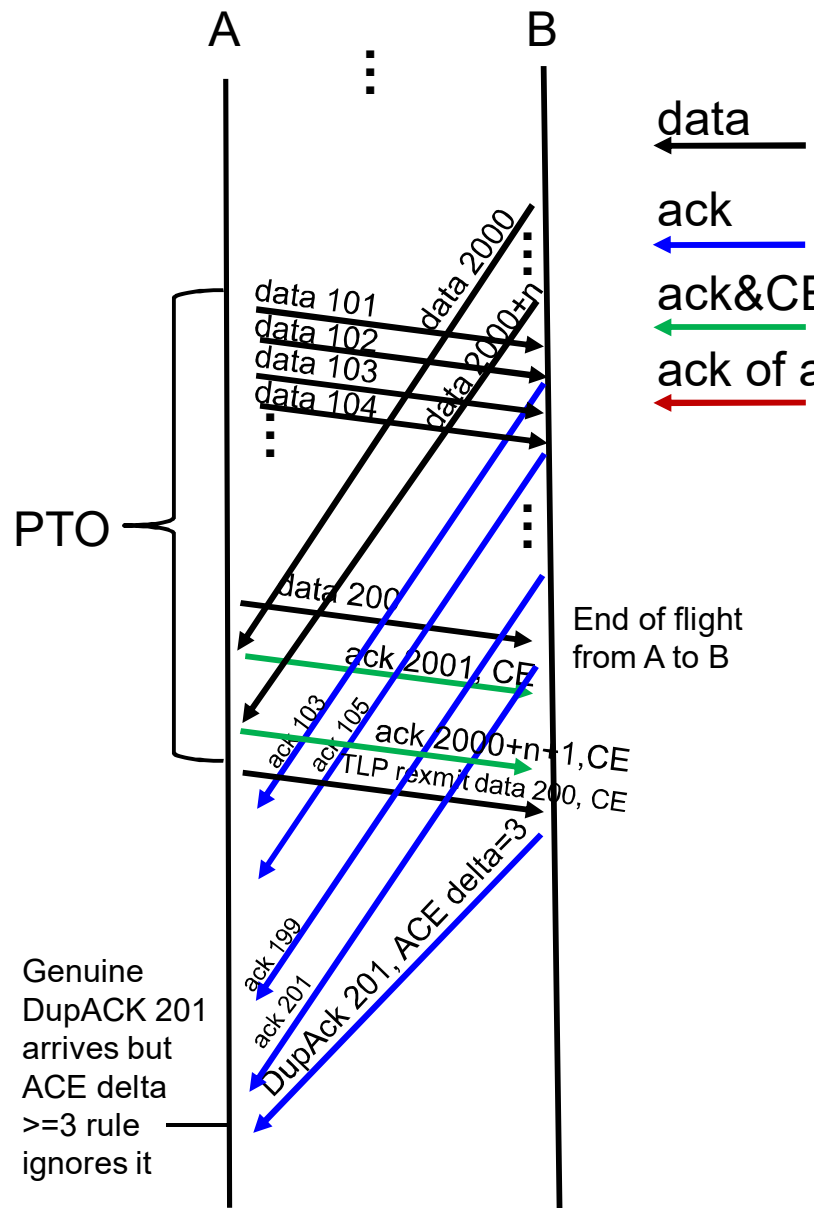


Fig 5 a) B does not implement DSACK or SACK removed, TLP incorrectly ignores genuine DupACK 201 (per ACE delta ≥ 3 rule) and does not detect spurious PTO

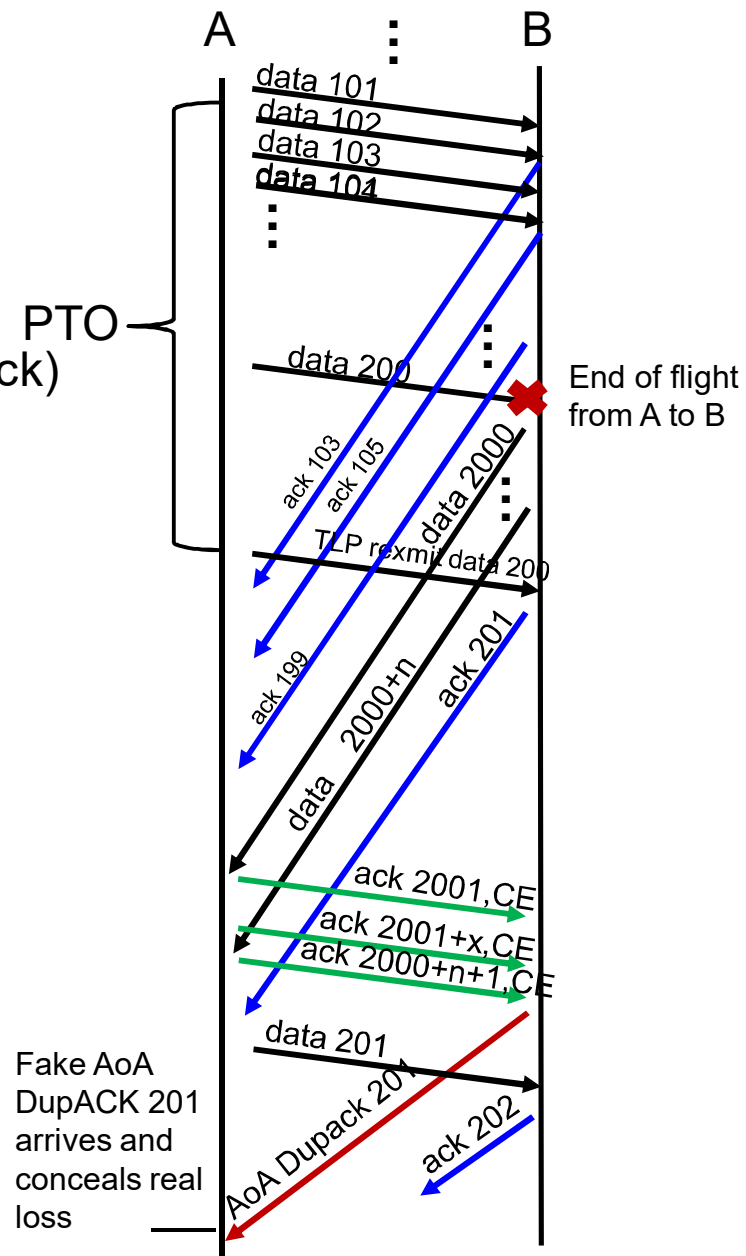


Fig 5 b) B does not implement DSACK, TLP incorrectly does not detect loss (of data 200) and does not react to congestion