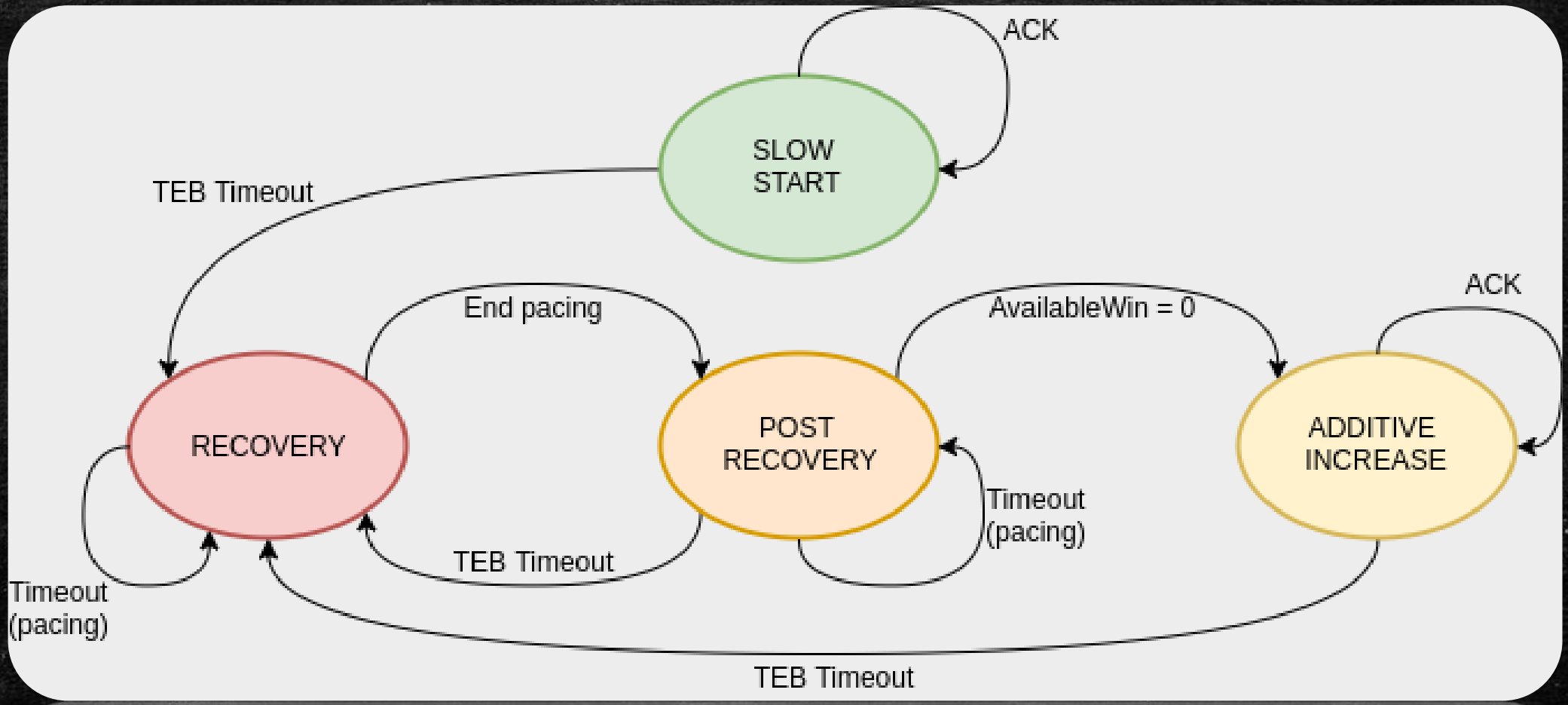
Tcp TEB on XFSM

Implementation of Time-based Exponential Backoff algorithm on NS3 XFSM executor

Brief xFSM chart



Detailed xfsm table

SS=Slow Start; AI=Additive Increase;
(*) this setTimer action schedules a «fake» timeout that triggers the next xFSM wake up, as soon as possible.

#	CONDITION	STATE	EVENT	ACTIONS
1	lastAcked<ackNo	SS	ACK	update(cwnd+=MSS), setTimer()*
2	AvailableWin > 0	SS	Timeout	setTEBTimer(2*RTT), sendPacket(), setTimer()*
3	timerEntrySeqNo<highAck && currReTXRound < timerEntryReTXRound	SS	TEBTimeout	setState(RECOVERY), update(cwnd/=2; currReTXRound++, pktsToPace=cwnd/segSize, pacingDelay=RTT/pktsToPace), sendPacket()), setTimer(pacingDelay)
4	timerEntrySeqNo<highTX	Recovery	Timeout	sendPacket(), setTimer(pacingDelay), update(timerEntryReTXRound++), setTEBTimer(2*RTT)
5	timerEntrySeqNo>=highTX	Recovery	Timeout	setState(PostRec), update(currRetxRound=0; postPacingDelay=RTT/pktsToPace++; setGenericTimer(postPacingDelay), setTEBTimer(2*RTT)
6	ackNo>=highTX	Recovery	ACK	setState(PostRec)
7	AvailableWin>0	PostRec	Timeout	update(pktsToPace++), setTimer(postPacingDelay)
8	AvailableWin==0	PostRec	Timeout	setState(AI)
9	lastAcked<ackNo	AI	ACK	sendPacket(), update(cwnd= MSS*MSS/cwnd)
10	*same as table entry 2*	AI	Timeout	*same as table entry 2*
11	*same as table entry 3*	AI	TEBTimeout	*same as table entry 3*

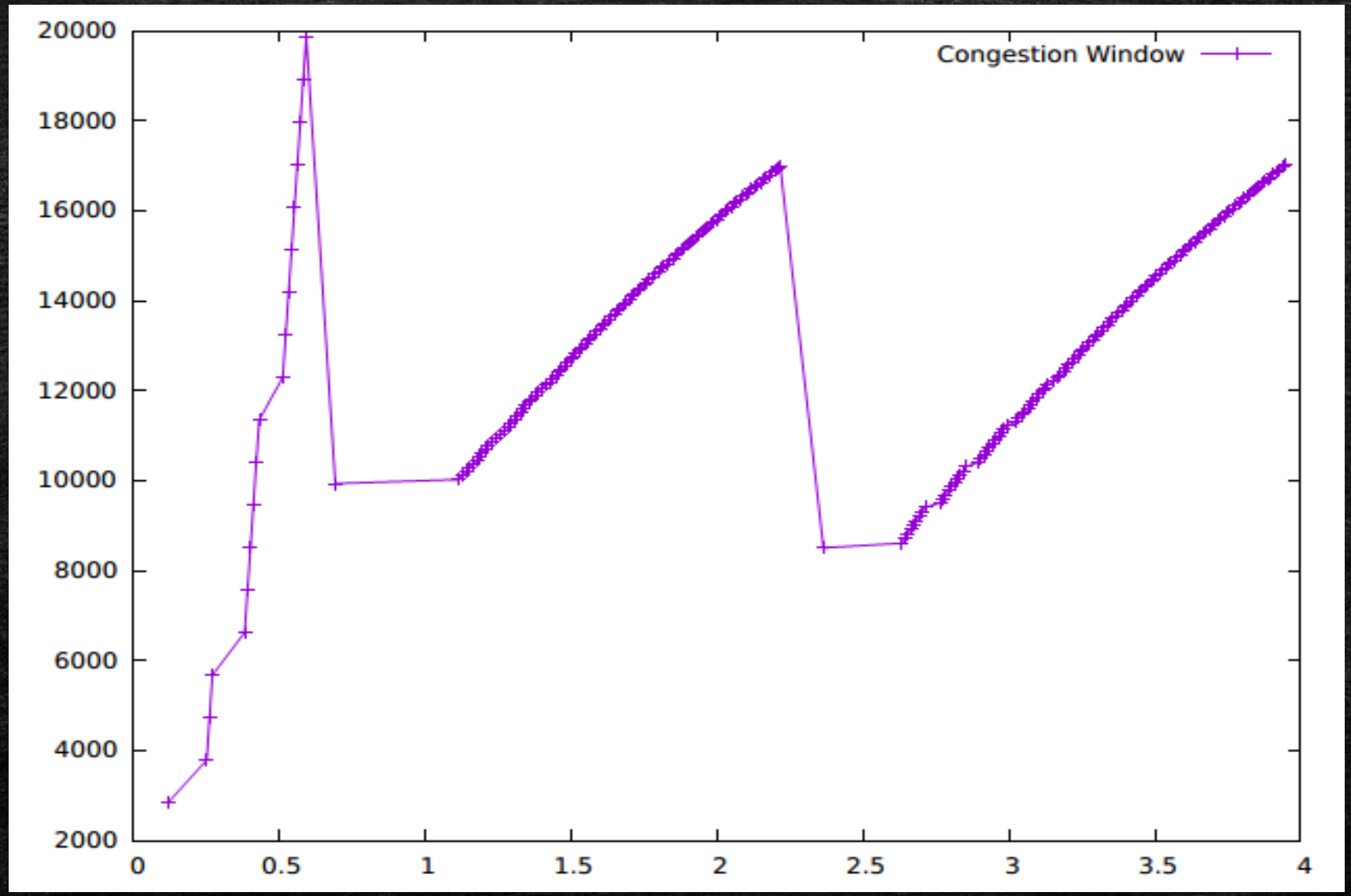
What's missing w.r.t. Michael's code (*TODO*)

- Spurious retransmission handling (Eifel);
- SACKs handling;
- RTT estimation is different (in xfsm's case it's few tens of milliseconds less)

How to read the table

- After the three-way handshake the ns3 standard TCP sends a number of packets equal to the initial window (in this case 3 MSS);
- The first xFSM wake-up occurs when the first ACK arrives (table entry #1). From that point on, all the processing follows the xFSM table description.

Results: Congestion window



Results: Sequence plot