Interactions between Class A1, Class B and Class C Traffic with Conservative Mode

Bob Doverspike, Chuck Kalmanek, Jorge Pastor, K. K. Ramakrishnan,
Aleksandra Smiljanic, Dong-Mei Wang, John Wei

AT&T Labs. Research, NJ

Introduction

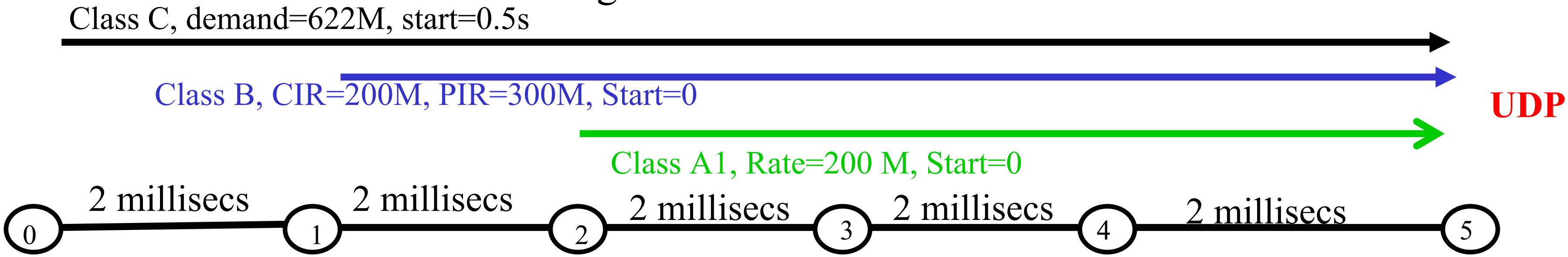
- In previous presentations, we have focused on behavior of Fairness Eligible Traffic with conservative mode
- We have now implemented Class A and Class B in our NS simulator
  - Understanding the co-existence of Fairness Eligible traffic and Class A1 and Class B-CIR traffic (whose allocations are reclaimable) is important
- Desirable that there is very little impact on Class A1 and Class B-CIR traffic because of FE traffic
  - Under steady state behavior, rate and delay guarantees are met
  - During transients, rate guarantee is minimally impacted
    - Little or no starvation of Class A1/B-CIR traffic beyond small # of round trips
  - Delay guarantee for Class A1 measured over SLA monitoring interval is met
- Also important to understand what rate A1/B-CIR can be guaranteed

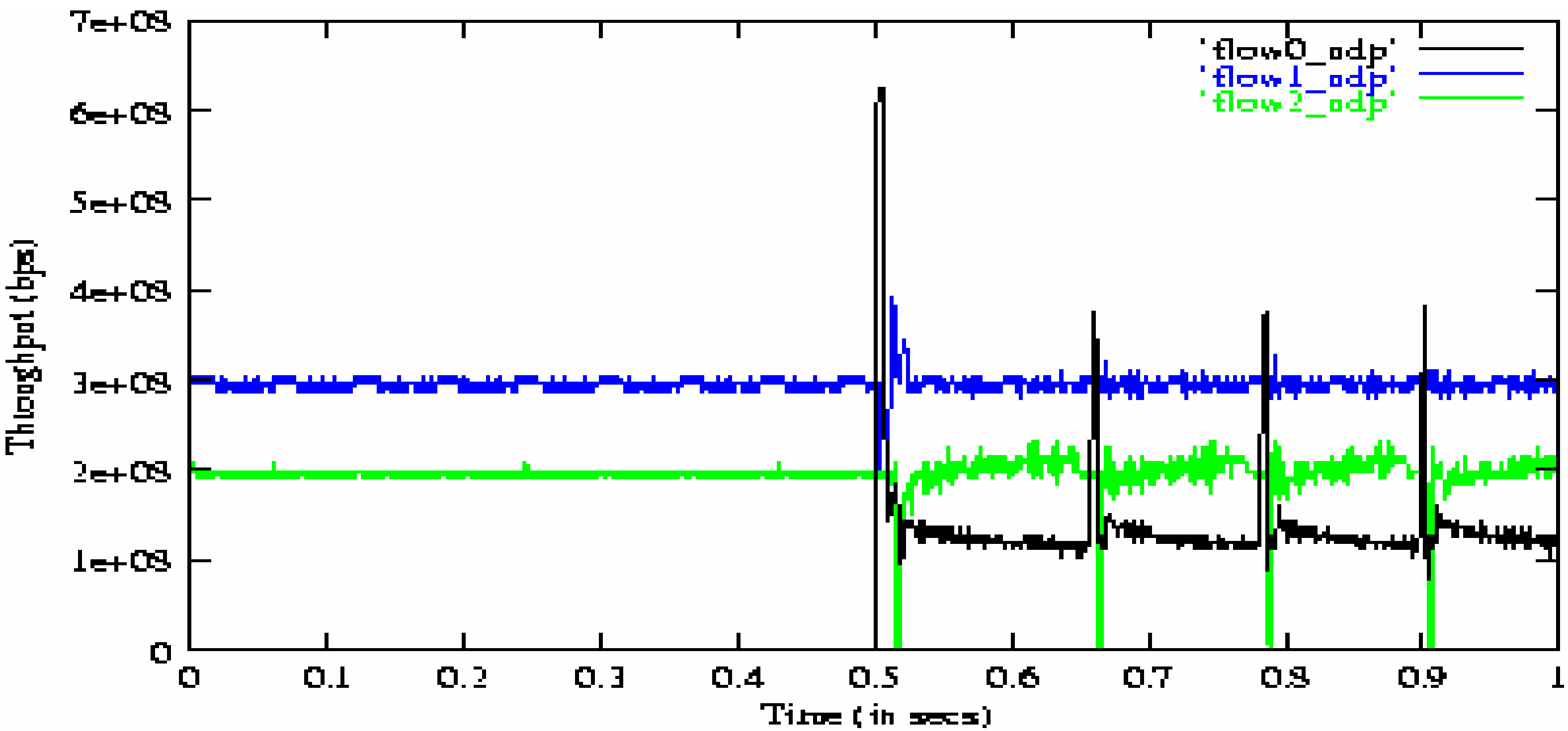
Simulation Parameters

- Our simulation includes shapers defined in the D2.2 spec for Class A, B and C
- Our inputs to the conservative scheme developed so far were primarily with FE traffic behavior in mind
  - Minimize starvation and oscillations
  - Ensure high utilization, retain “fast start” capability of FE traffic sources
- Conservative mode Table 9.24 and Table 9.25 as specified in D 2.2
- Parameters:
  - STQsize = 256 Kbytes
  - Advertisement interval = 0.1 milliseconds
  - Aging interval = 0.1 milliseconds
  - Link Rate = 622 Mbits/sec
  - Low_Threshold = 1/8 * STQ, Medium_Threshold = 3/16 * STQ, High_Threshold = ¼ * STQ

Performance of Existing Conservative Scheme

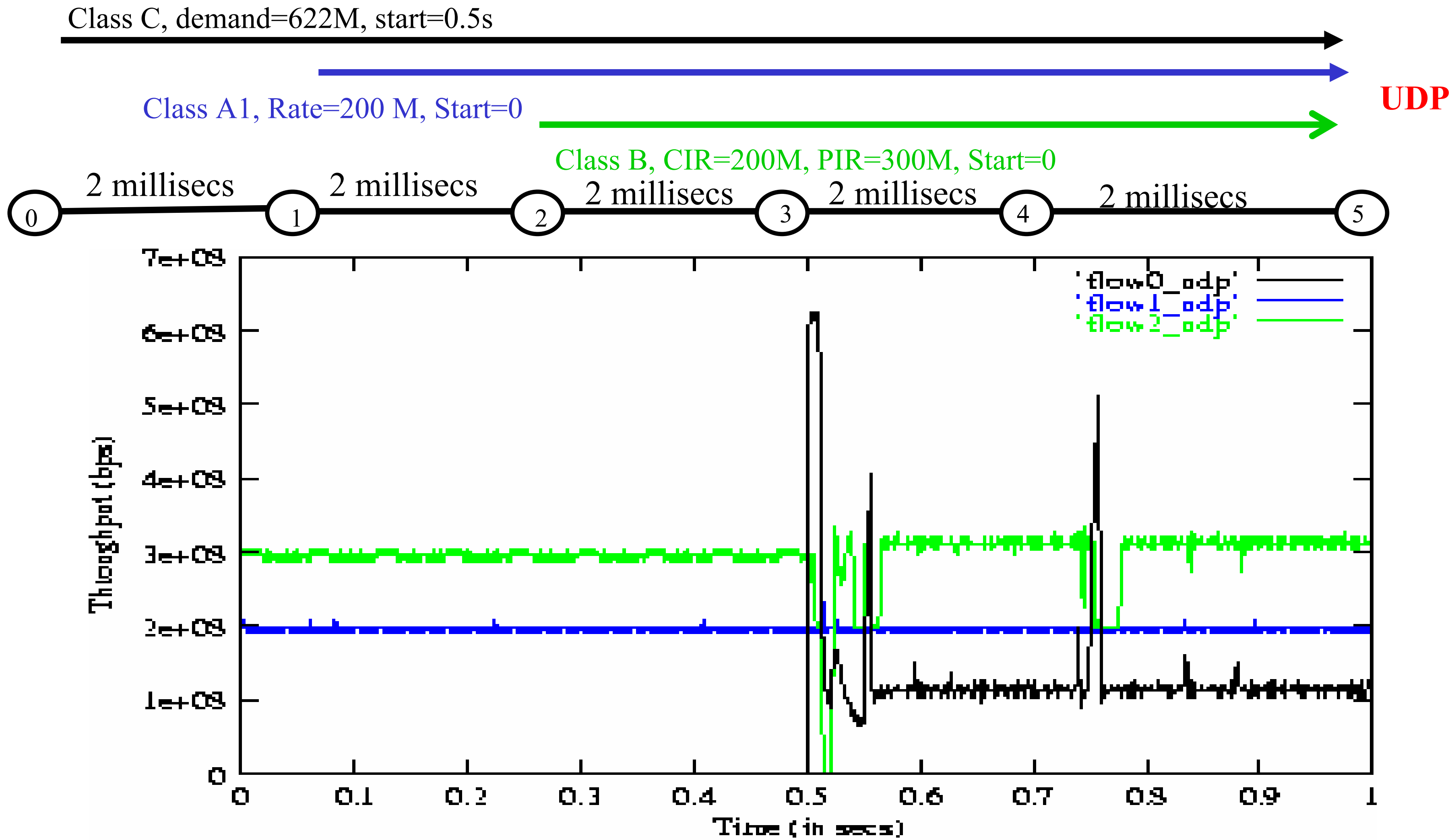
- Class A1 traffic gets impacted by start up of Class C traffic
- Periodic oscillations for traffic generated at bottleneck





Performance of Existing Conservative Scheme

- Switch nodes generating Class A1 and B-CIR
- Class B-CIR traffic at bottleneck link gets impacted by start up of a Class C traffic

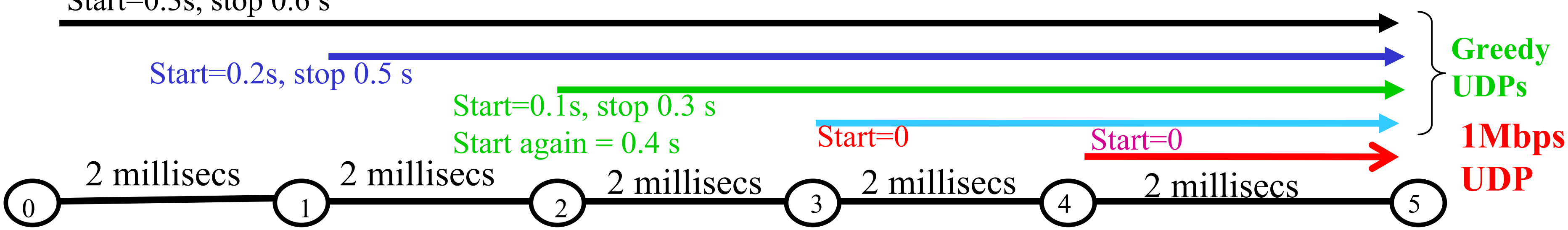


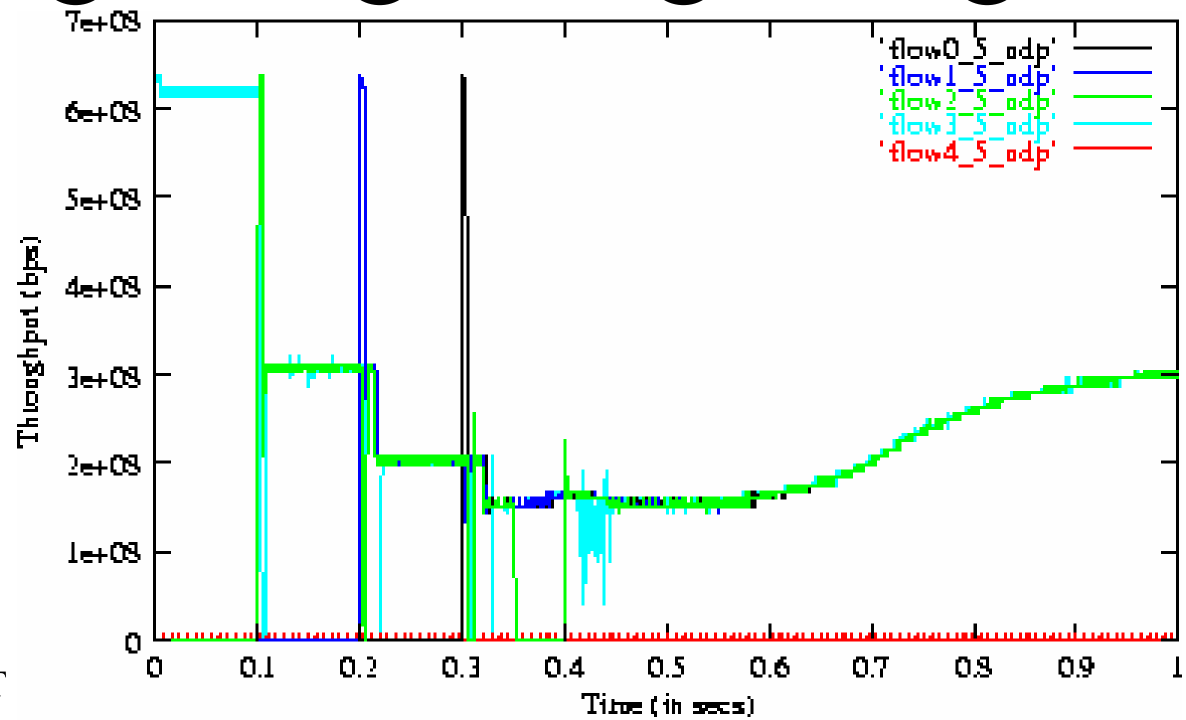
Observations

- STQ queue builds up at bottleneck
  - Initially, fast start by FE traffic is desirable. However, this causes upstream STQs to build up, and these have to be drained at link rate
  - When STQ at bottleneck becomes full, it stops insertion of local add traffic
- In our simulations, we have Class A1 or Class B-CIR traffic being locally sourced at bottleneck
  - Starvation at bottleneck node has impact on Class A1 or Class B-CIR traffic
- Lower bound on local_fair_rate calculated by Conservative mode (in Row 5, Table 9.24) limits how far the upstream nodes may be pushed down
  - Lower bound was introduced to limit oscillations for FE traffic
- Following 2 slides were used to demonstrate desirability of lower bound for FE traffic in Feb/March FAH and RPR meetings

Lower bound in Row 5 of Conservative Mode

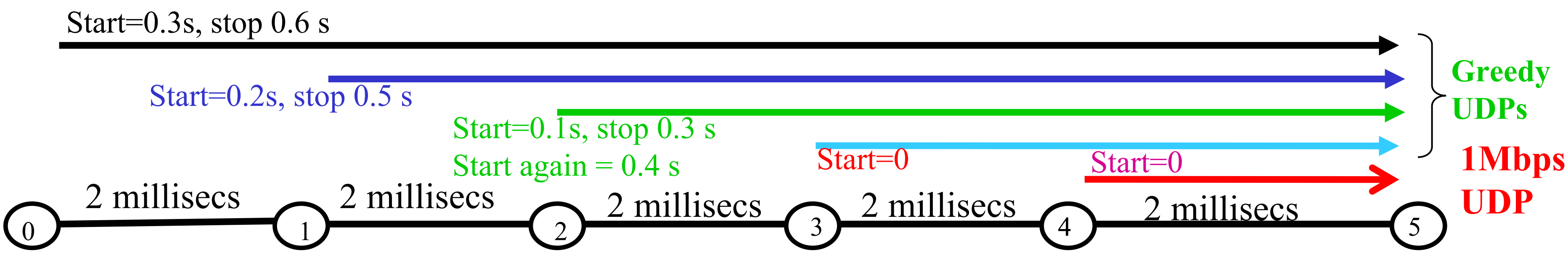
- Condition for dual queue MAC simplified as
  (STQDepth > STQMediumThreshold) && (RTTWorthofIntervalsHavePassed)
- Action computes (simplified) lower bound for LocalFairRate as:
  lower_bound = (unreservedRate/activeWeights)*Weight;   use it in setting LocalFair Rate in Row 5.

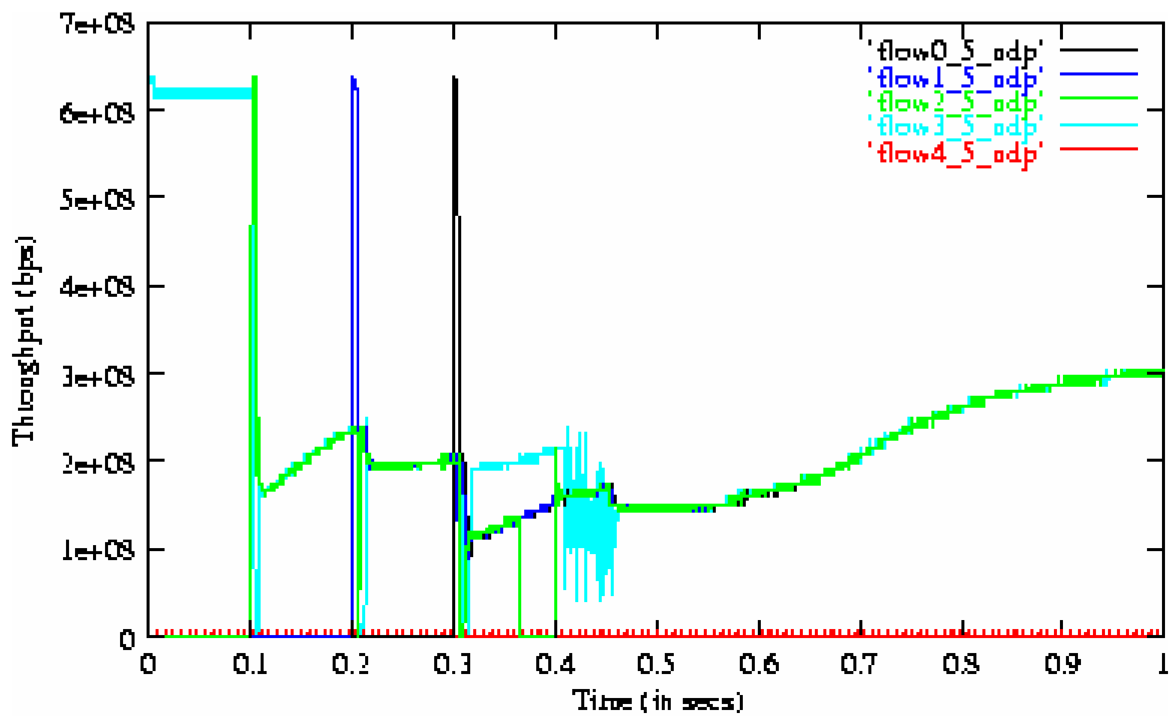




Without Lower Bound in Row 5 of Conservative Mode

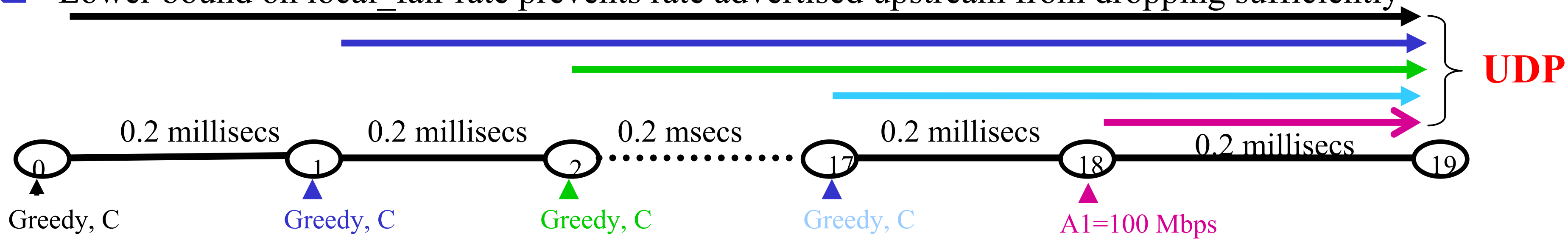
- localFairRate computation in Row 5 without lower bound
- Results in Oscillations. But, no increase in starvation period

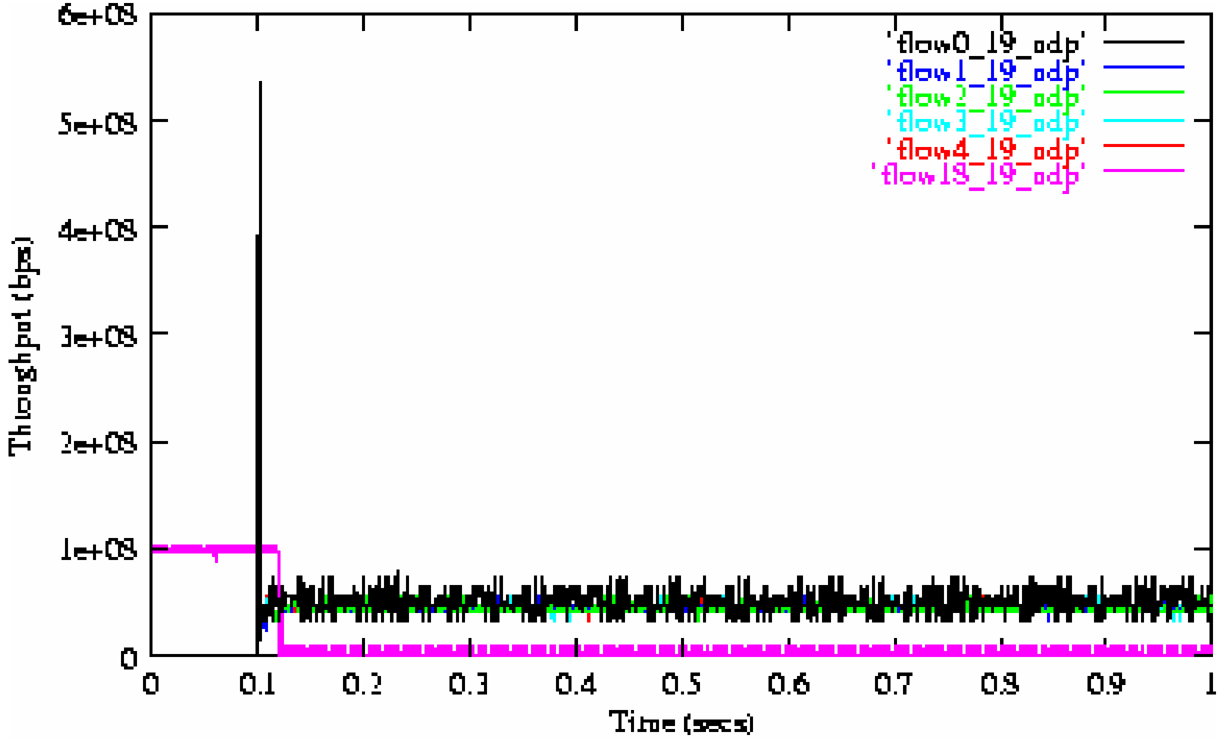




Interaction between FE and Class A1 traffic: Behavior with larger # nodes

- Examine behavior with STQ=256K, Rate A1= 100 Mbps; (180 Mbps as per current spec. Draft 2.2)
- Class A1 traffic is starved once the other FE nodes start up
- Lower bound on local_fair rate prevents rate advertised upstream from dropping sufficiently

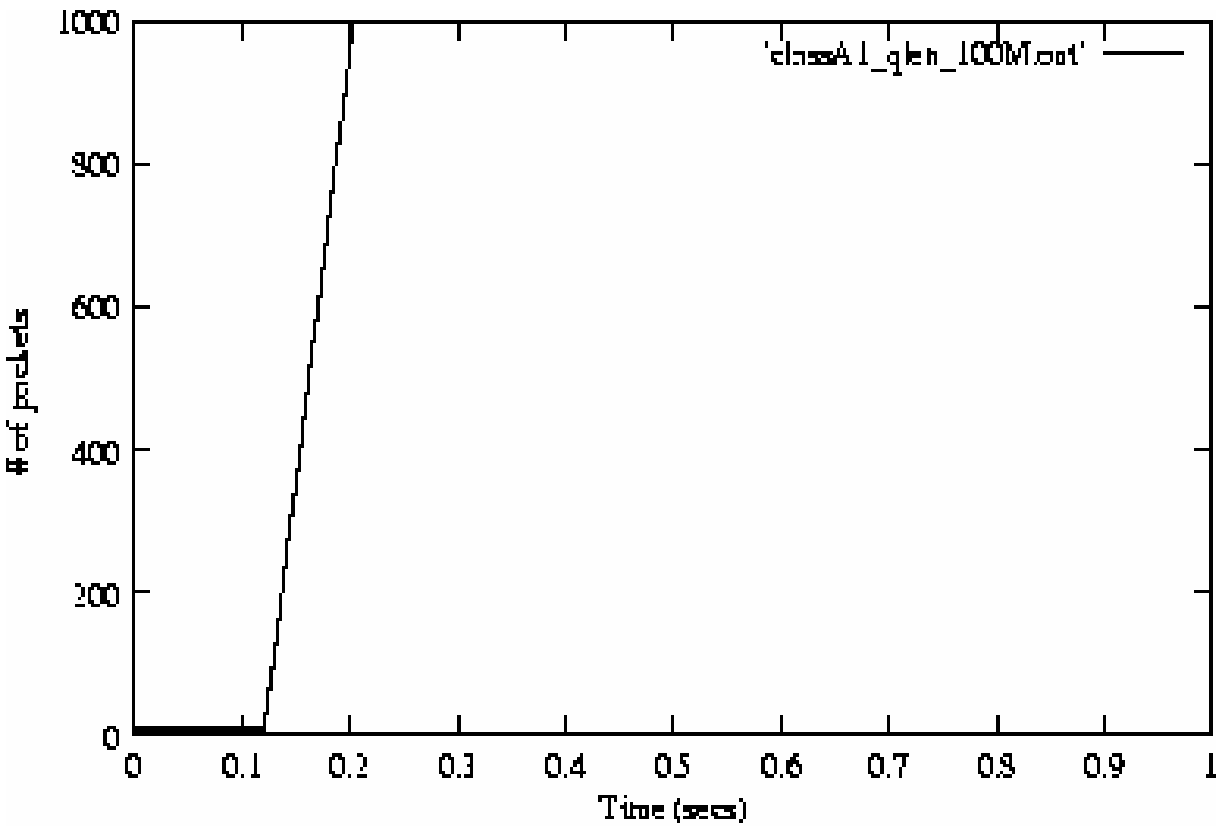




- Class A1 traffic starved once FE traffic starts up
- Lower bound on local_fair rate plays a major role

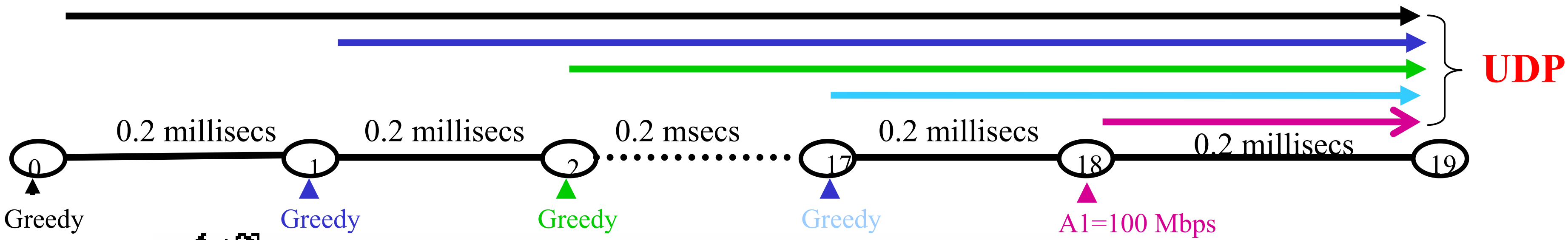
Client Queue Length

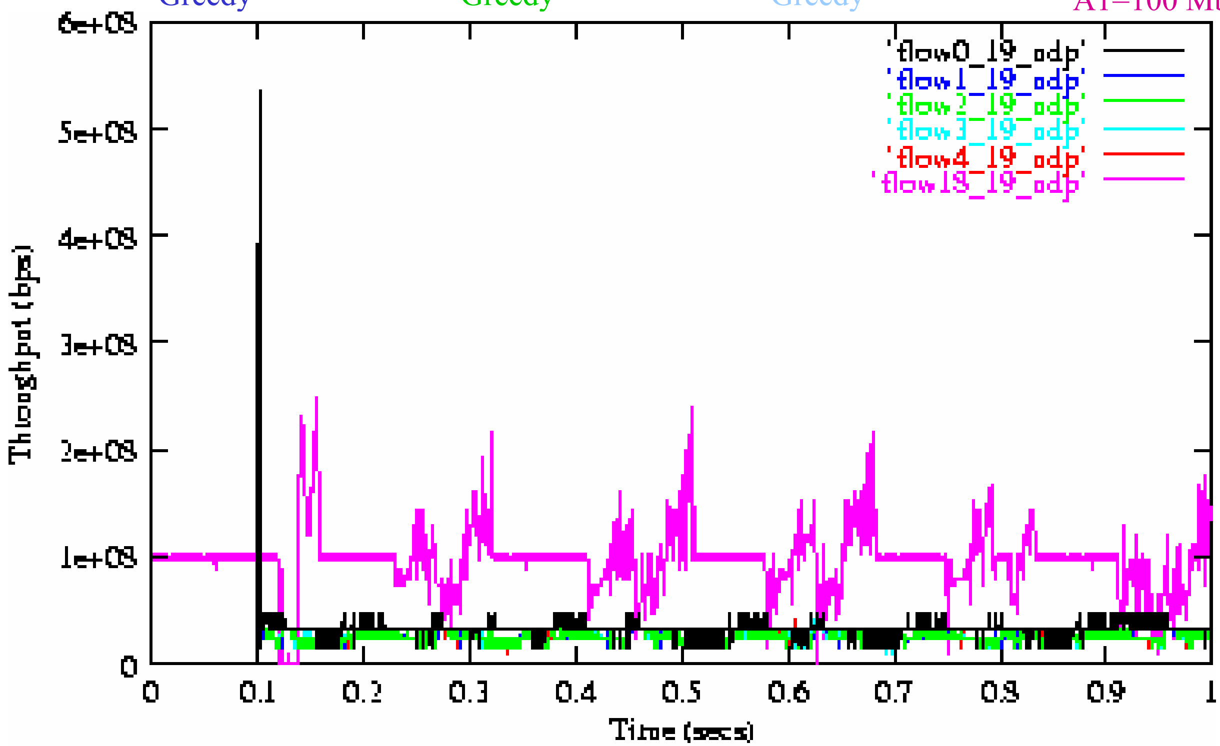
- Queue length at the client of node 18 which generates Class A1 traffic.
- Queue builds up when FE traffic starts up at upstream nodes
  - Node 18 starved after 0.1 seconds – resulting in packets being queued at client buffer
- Queue reaches maximum limit of 1000 packets in client buffer, and stays there.



Behavior with 20 nodes, remove lower bound

- Behavior with STQ=256K, Rate A1= **100 Mbps** (~180 Mbps allowed as per current spec.)
- No Lower bound on local_fair rate in Row 5 of Conservative mode (Table 9.24)

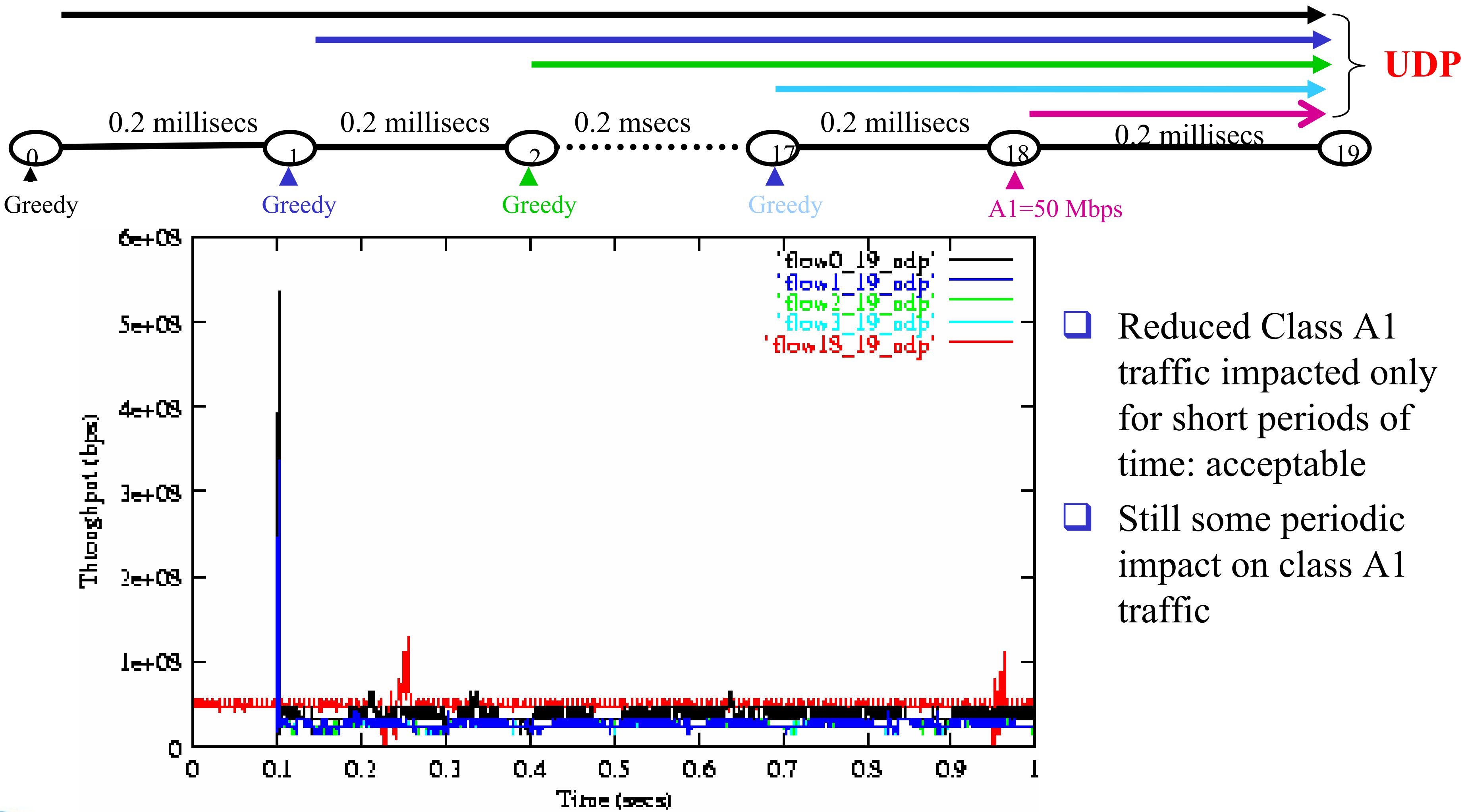




- Class A1 traffic still starved for short periods of time: is class A1 rate too high?
- Oscillations of Class A1 traffic periodically when upstream sources increase load briefly.

Behavior w/20 nodes, no LB, reduced A1

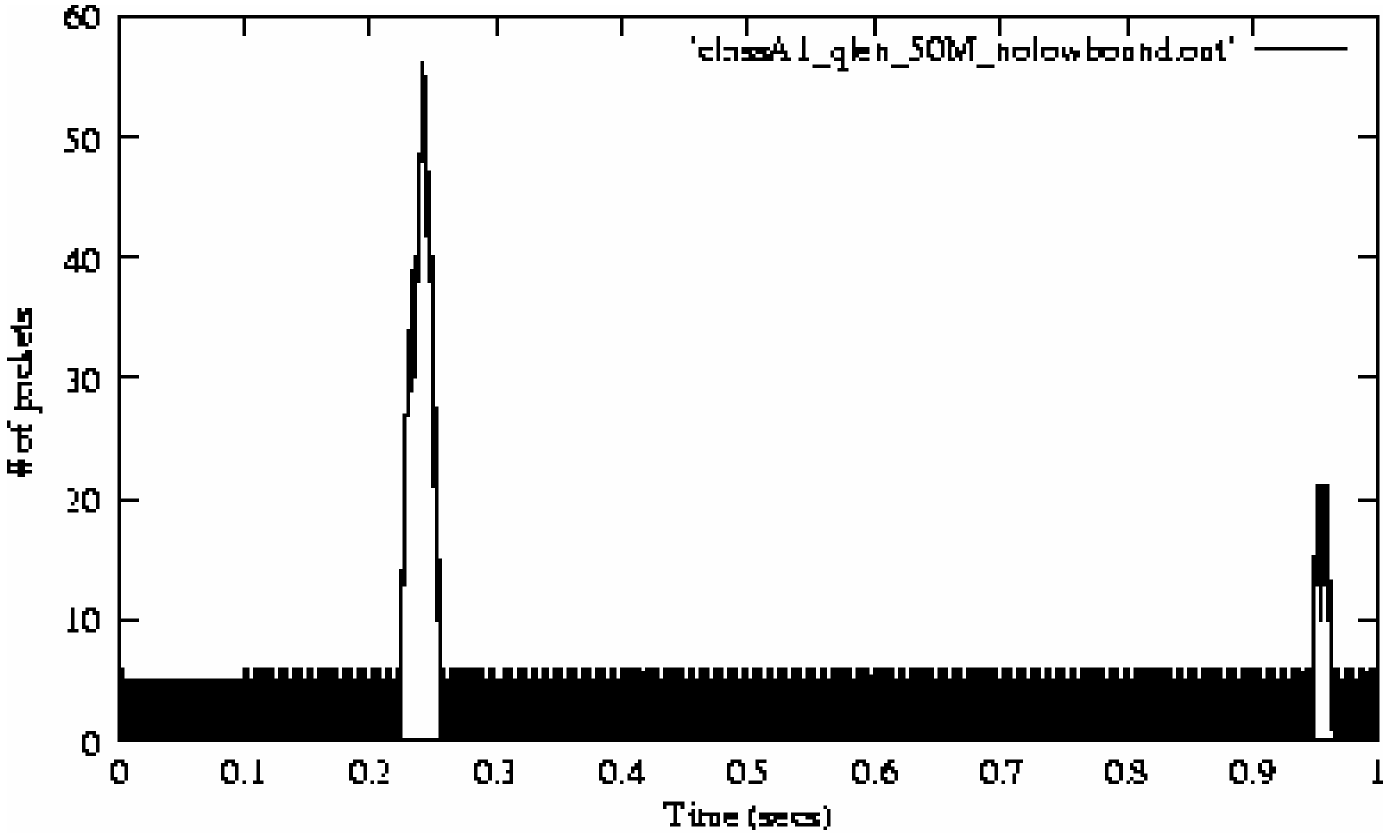
- Behavior with STQ=256K, Rate A1= **50 Mbps** (~180 Mbps allowed rate in spec.)
- No Lower bound on local_fair rate.



- Reduced Class A1 traffic impacted only for short periods of time: acceptable
- Still some periodic impact on class A1 traffic

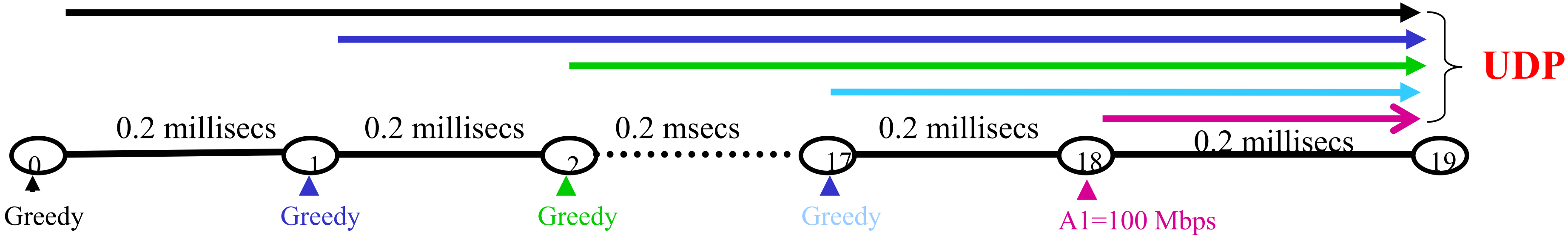
Client Buffer Queue Length

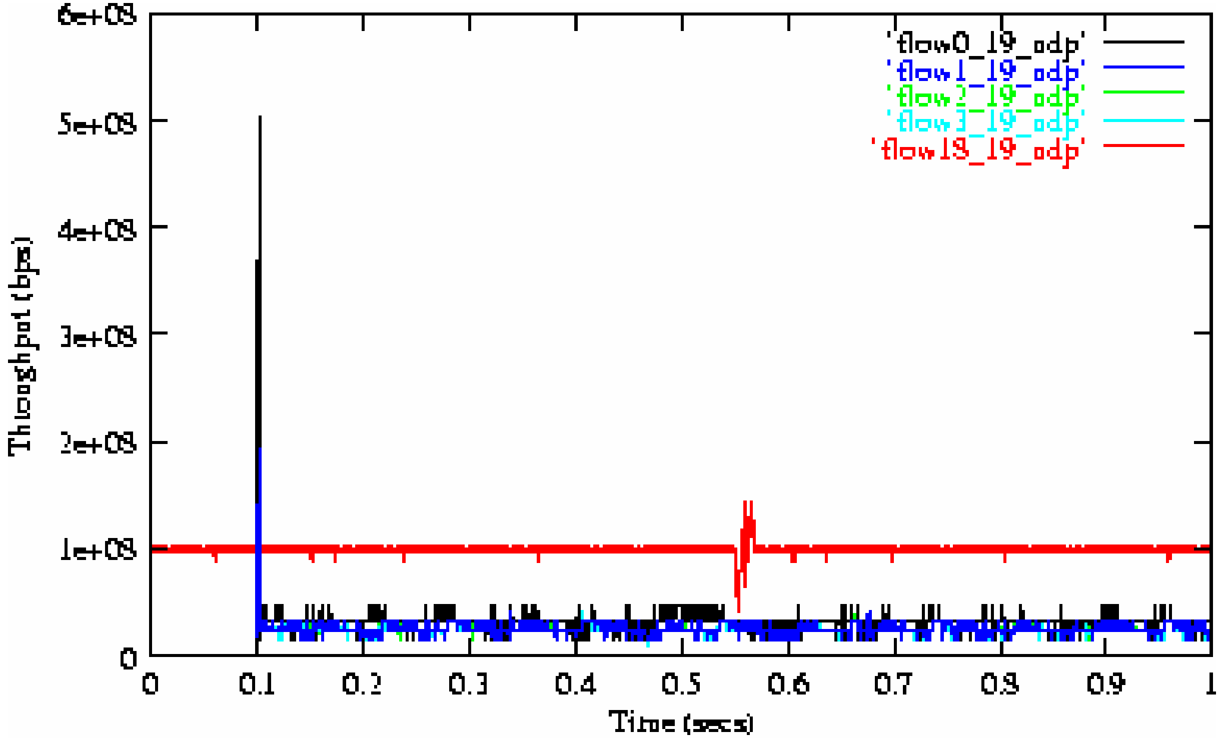
- Queue behavior with STQ=256K, Rate A1= **50 Mbps**
- Queue length at the client of node 18 which generates Class A1 traffic
- Initial queue buildup at 0.25 seconds when STQ briefly hits Full Threshold
  - Starvation of Class A1 traffic is brief and short-lived to drain out STQ, and results in limited # packets queued in the client



What if we increase STQ, but keep a fixed Low_threshold?

- Examine behavior with **STQ=2560K**, Rate A1= 100 Mbps (~ 180 Mbps allowed by spec.)
- **BUT: keep low_threshold = 256K/8**; No Lower bound on local_fair rate





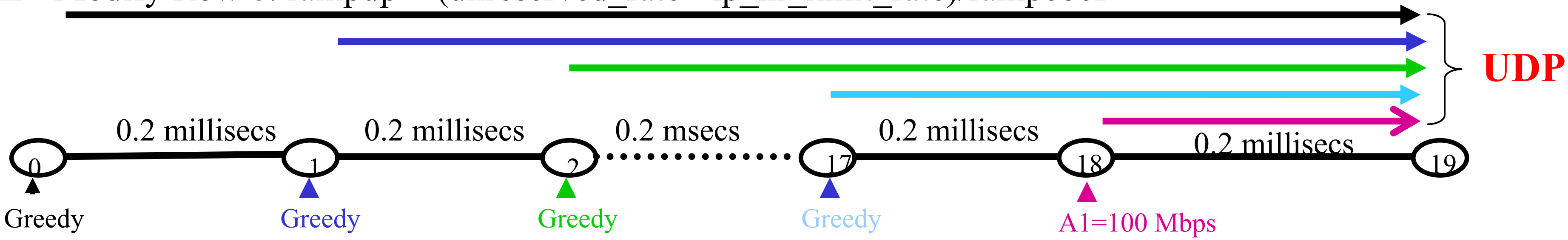
- Class A1 traffic minimally impacted; some periodic impact due to increase by FE
- Having a larger STQ, but keeping the low_threshold to be a smaller fixed value (rather than STQ/8) allows a larger rate guarantee

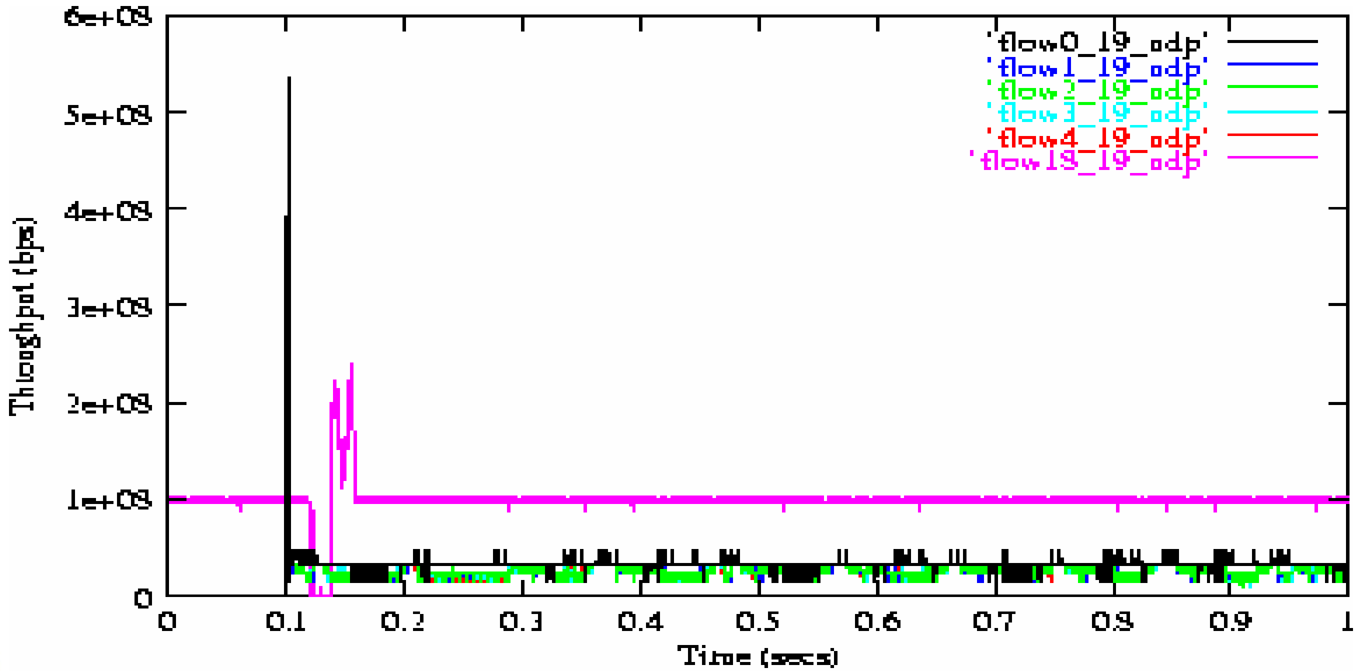
Overcoming the “periodic” effect on traffic sourced from bottleneck node

- We had previously focused on FE traffic
- Row 6 of conservative mode table allows local_fair_rate calculated to increase when STQ depth falls below low_threshold
  - Amount available for increase = (unreservedRate – (lpAddRate+lpFwRate))
  - But that did not take into account Class A1 or Class B-CIR rates
- Modify Row 6 to correctly reflect the amount local_fair_rate can be increased by
  - Amount available for increase = unreserved_rate - lp_nr_xmit_rate
  - Reduces the “over-correction” of Row 6

Behavior with 20 nodes, no lower bound, modified Row 6

- Examine behavior with STQ=256K, **Rate A1= 100 Mbps (~ 180 Mbps allowed as per current spec.)**
- No Lower bound on local_fair rate
- Better knowledge of what is the available remaining capacity: unreserved_rate - lp_nr_xmit_rate
- Modify Row 6: rampup = (unreserved_rate - lp_nr_xmit_rate)/rampcoef





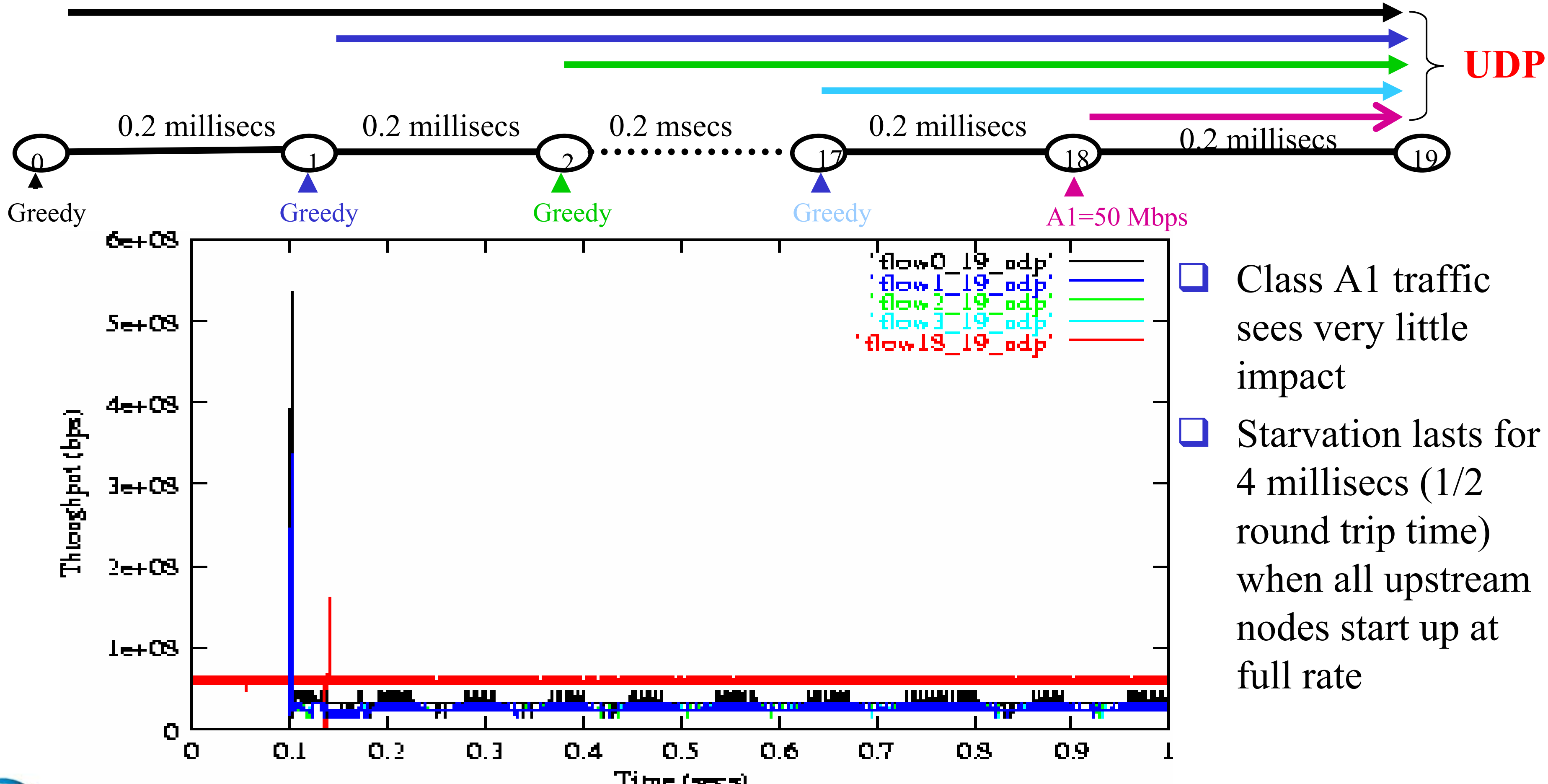
- Class A1 traffic starved for a period of time at start up (~18 milliseconds)
  - What is the correct A1 rate to avoid starvation?
  - But, periodic oscillations removed: compare with slide 11

Class A1 Rate to avoid even brief starvation?

- The maximum Class A1 rate recommended in Appendix G (Section G.1.2) provides a guideline for how large Class A1's rate can be:
  - Feedback is generated once STQ reaches STQLowThreshold
    - Default STQLowThreshold = 1/8 * sizeSTQ
  - We have up to 7/8 of the STQ buffer to accommodate arriving traffic already admitted into ring, before STQ is full and local traffic has to be "shut off"
- With conservative mode, initial estimate of "active_stations"/ "active_weights" in Row 2 (when STQLowThreshold is reached) may not yet be accurate
  - Row 7 re-calculates local_fair_rate, when STQDepth >= STQHighThreshold
  - Remaining buffer available is ¾ sizeSTQ before local add traffic blocked
  - upstream nodes' STQbuffer also filled to STQHighThreshold in worst case
    - Queueing delay = (# hops*STQHighThreshold)/link_rate
    - FRTT' = (round_trip propagation delay + # hops * advt. delay + queueing delay)
- Estimate of max. Class A1 rate can be calculated as:
  - RateA1 <= (3/4*sizeSTQ)/(FRTT')

Behavior with 20 nodes, no lower bound, modified Row 6, new A1 rate

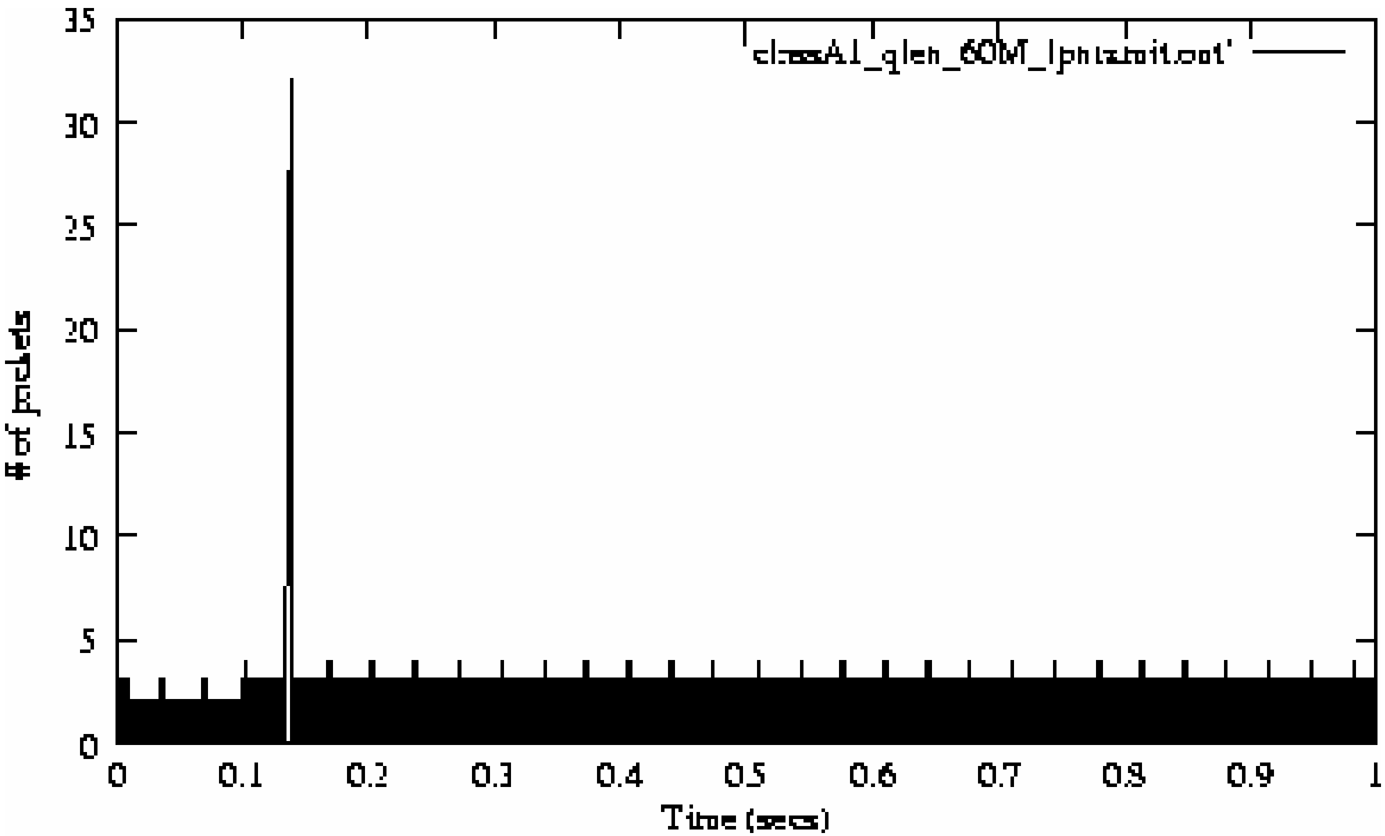
- Examine behavior with STQ=256K, **Rate A1= 60 Mbps** (using updated formula)
- No Lower bound on local_fair rate
- Modify Row 6: rampup = (unreserved_rate - lp_nr_xmit_rate)/rampcoef



- Class A1 traffic sees very little impact
- Starvation lasts for 4 millisecs (1/2 round trip time) when all upstream nodes start up at full rate

Client Buffer Queue Length

- Client queue length with STQ=256K, **Rate A1= 60 Mbps** (updated formula)
- No Lower bound on local_fair rate
- Modify Row 6: rampup = (unreserved_rate - lp_nr_xmit_rate)/rampcoef



- Compare with slide 13: smaller buffer occupancy
- Removed periodic effect
- Jitter to Class A1 traffic is well controlled

Summary

- Conservative mode, with two small changes to current draft (D2.2), co-exists with Class A1 and Class B-CIR traffic for the configurations we have simulated (up to 20 stations)
  - Remove the lower bound in Row 5 which limited how far down the local_fair_rate could go
    - Need: to prevent starvation even under extreme conditions
  - Modify the formula for increase in Row 6 to correctly account for the Class A1/Class B-CIR traffic
    - Need: to avoid oscillations – thereby controlling jitter better
- Improve the estimate for allowed Class A1 rate
  - Based on "STQHighThreshold" because feedback needs to obtain accurate estimate of active stations
  - Use estimate of FRTT to include queueing delay (which contributes to feedback delay)
    - Since it is a static estimate, using STQHighThreshold for queue size is reasonable