Co-existence of Service Classes with RPR Conservative Mode Fairness Mechanism

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

AT&T Labs. Research, NJ

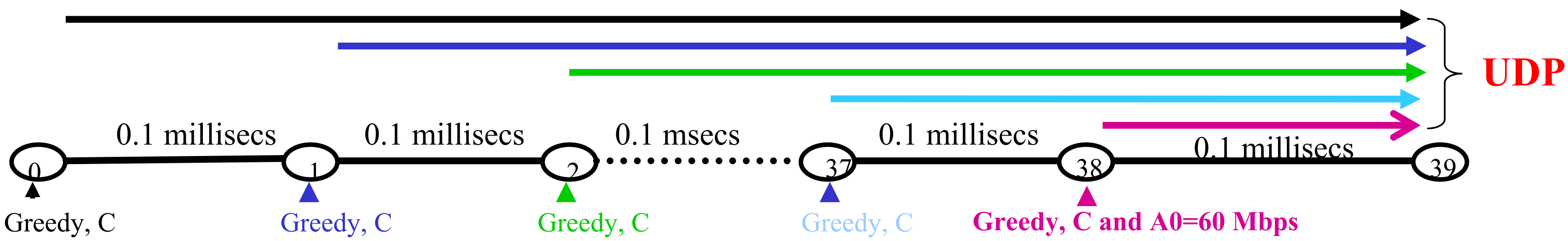
Introduction

- This presentation shows some simulation results with various options discussed in the FAH for dealing with the co-existence of multiple service classes along with the conservative mode fairness mechanism
  - Class A0 and Class C traffic co-existence
  - Class A0, Class A1 and Class C co-existence
- We also address some issues related to scalability of conservative mode with large numbers of stations
- Proposals made based on presentation of simulation results, for this meeeting
  - Downstream Shaper should apply to the STQ traffic
    - In dual-queue transmit selection states, in Table 6.28 (Draft 2.4), add passD to the condition for selecting a packet from STQ
      - ensures that reserved bandwidth (Class A0) is carved out
      - shaperD credit won't go negative, nor go below Low_limit, since the transmit rate for non-classA0 traffic is no greater than unreserved rate
  - Update equation for maximum Class A1 traffic that can be supported
    - Based on formulas presented at Montreal meeting
  - Split the "rampCoef" parameter for conservative mode to two parameters
    - rampUpCoef for increase (Row 6 of Conservative state machine)
    - rampDownCoef for decrease (Row 5 of Conservative state machine)

Downstream Shaper Issues: Co-existence of Class A0 and Class C traffic

Scenario 1

- 40 nodes, with only the last hop sending Class A0 (reserved) traffic = 60 Mbps plus Class C traffic
- Class A0 traffic starts at time T = 0, along with all class C traffic.



• This scenario is used to demonstrate the impact of Class C FE traffic on Class A0, because the STQ at Node 38 reaches Full Threshold.

Motivation: If the congestion control mechanism does not react before the "Full Threshold" occurs, priority inversion may occur

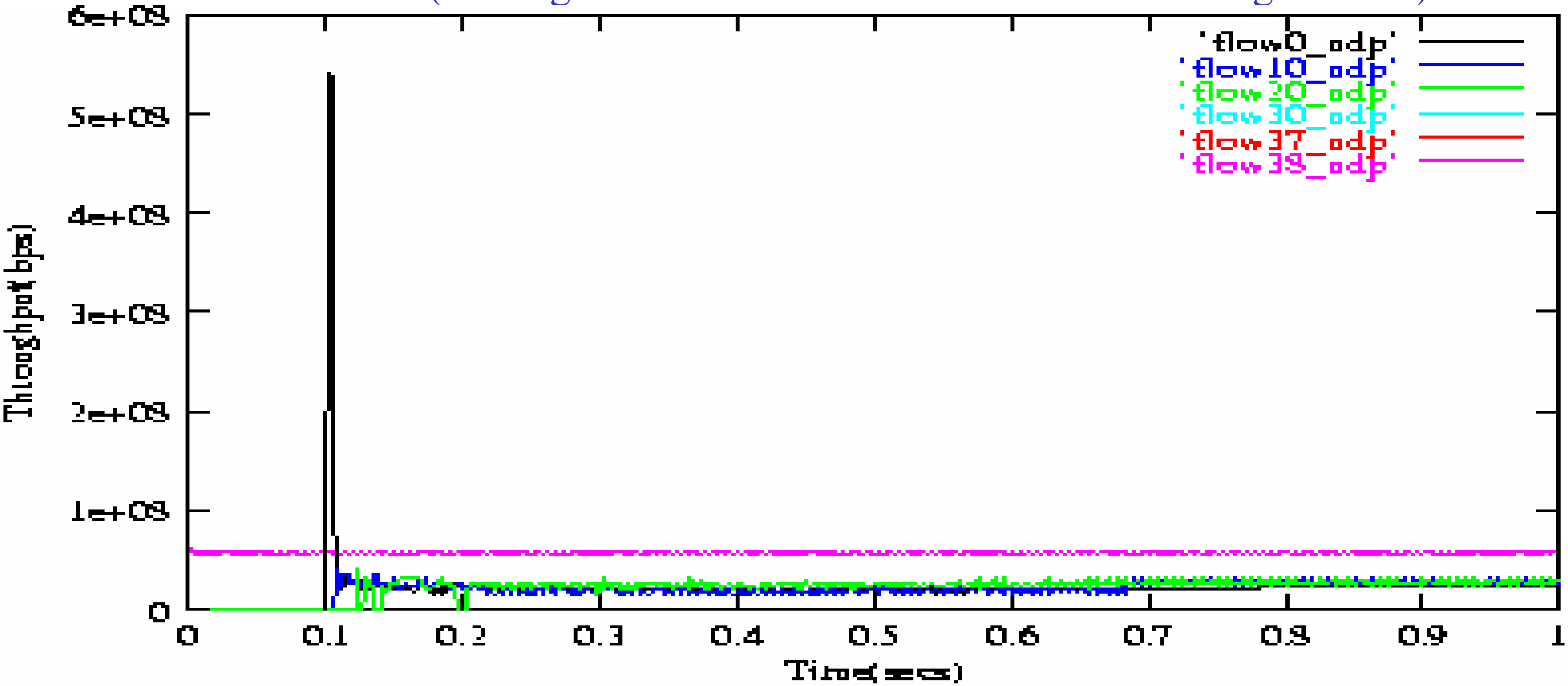
Does the conservative mode react fast enough, to not impact Class A0 traffic?

Simulation Parameters

- Parameters:
  - STQsize = 256 Kbytes
  - Advertisement interval = 0.1 milliseconds
  - Aging interval = 0.1 milliseconds
  - Link Rate = 600 Mbits/sec
  - Low_Threshold = 1/8 * STQ, Medium_Threshold = 3/16 * STQ, High_Threshold = ¼ * STQ
  - Shaper parameters
    - Low_limit = 1 MTU
    - High_limit = 2 MTU

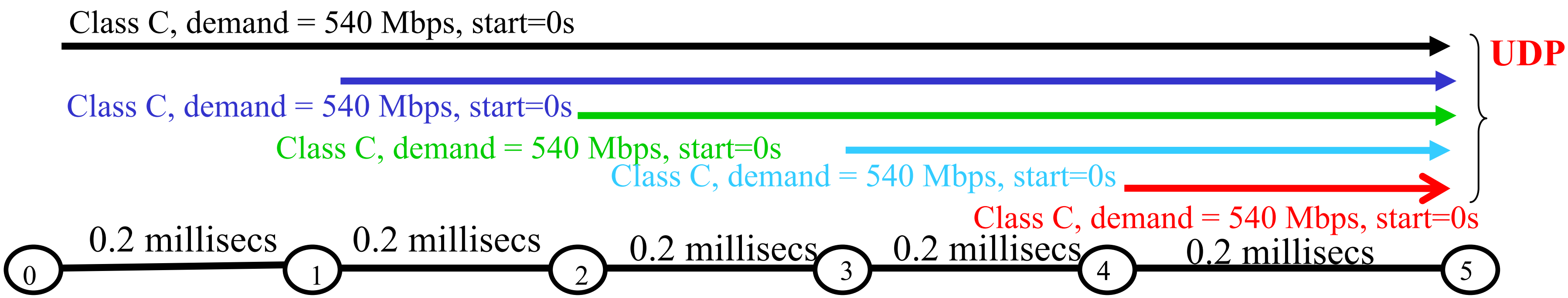
Impact on Class A0 traffic in scenario with large # nodes

- 40 nodes, with only the last hop sending Class A0 (reserved) traffic of 60 Mbps
- isCongested determined by (NrXmitRate>unreservedRate) || (STQ > lowThreshold)
- Class A0 is not impacted, and remains at the targeted rate of 60 Mbps
  - Note: the setting of the initial ShaperD parameters is critical to ensure there is no impact on Class A0 traffic (too large an initial "Low_limit" results in too large a burst)



Scenario 2: Justification for STQ Shaping

- Reserved rate of 60 Mbps (no reserved traffic is being sent)



This scenario is used to demonstrate starvation of FE traffic because ShaperD credits go below Low_limit or even negative.

Note that as per current draft 2.4:

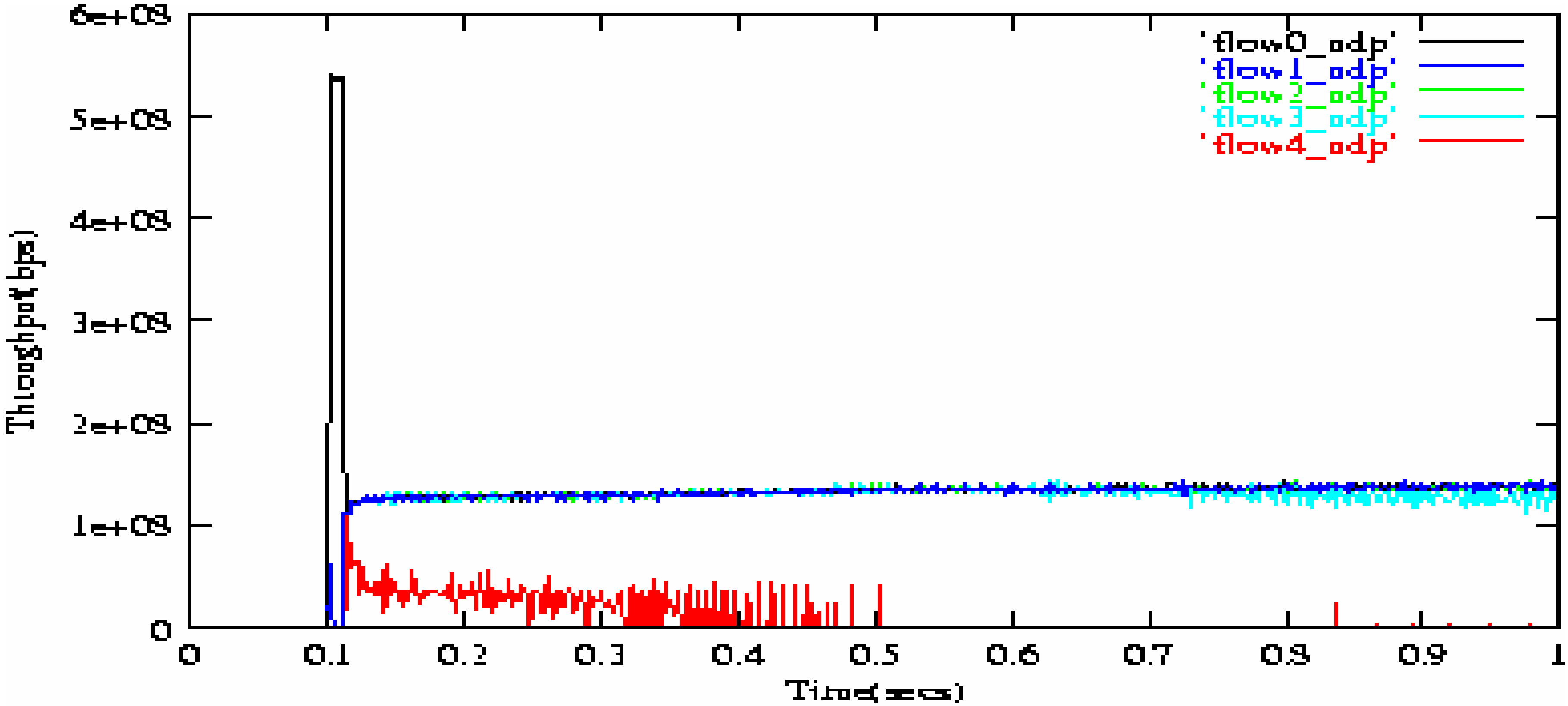
- ShaperD credits at a node are incremented at **unreserved rate**
- When packets from the STQ are forwarded at **link rate**, the shaperD credits are decremented at **link rate**

Thus, if shaperD credits fall below Low_limit, and there is continuous traffic forwarded through the STQ, then shaperD credits are not able to catch up.

This causes starvation at a node

Performance with Draft 2.4

- Scenario 2: Reserved rate of 60 Mbps (no reserved traffic is being sent)
- isCongested determined by (NrXmitRate>unreservedRate) || (STQ > lowThreshold)
- Downstream station 4 is starved after some time
  - Cause: ShaperD credits go below Low_limit

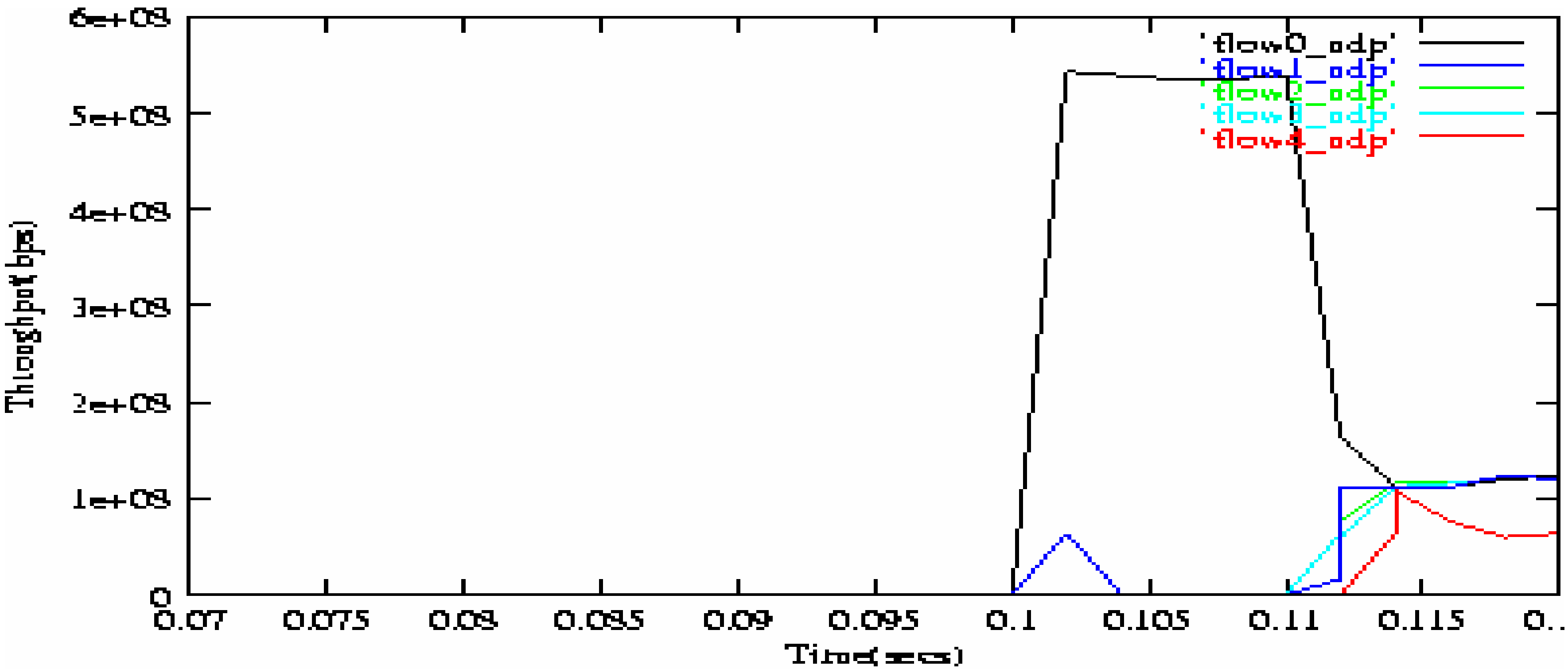


Observations

- Over the long time scale, node 4 (last source node) is starved for a long period of time, and receives drastically unfair service
- Observing the detailed operation of the congestion control mechanism, it appears to behave correctly, as we understand it
  - Number of active stations = 5
  - Initial advertised fair rate is correct
- Initially node 0 transmits at unreserved rate = 540 Mbps
  - Causes the shaperD credits at nodes downstream of node 0 to go below Low_limit of 1 MTU
  - Node 4 becomes congested and advertises a local fair rate of 540/5 Mbps.
- Node 0 drops its rate upon receiving FCM, and nodes 1, 2 and 3 are now able to start sending again (see next slide)
  - This causes node 4 to drop its add rate when its shaperD credits fall below Low_limit (around 0.115 seconds)

A Closer Look at Initial Startup

- Looking at the initial transient: The most upstream flow, flow 0->5 is able to send at a high rate. Flows immediately downstream of it (flow 1->5, 2->5) start up, but have their shaperD credits drop below Low_limit
  - Their shaperD credits are incremented at unreserved rate (540 Mbps)
  - Their shaperD credits are decremented at link rate (600 Mbps) because the STQ buffer has packets to send from the upstream station

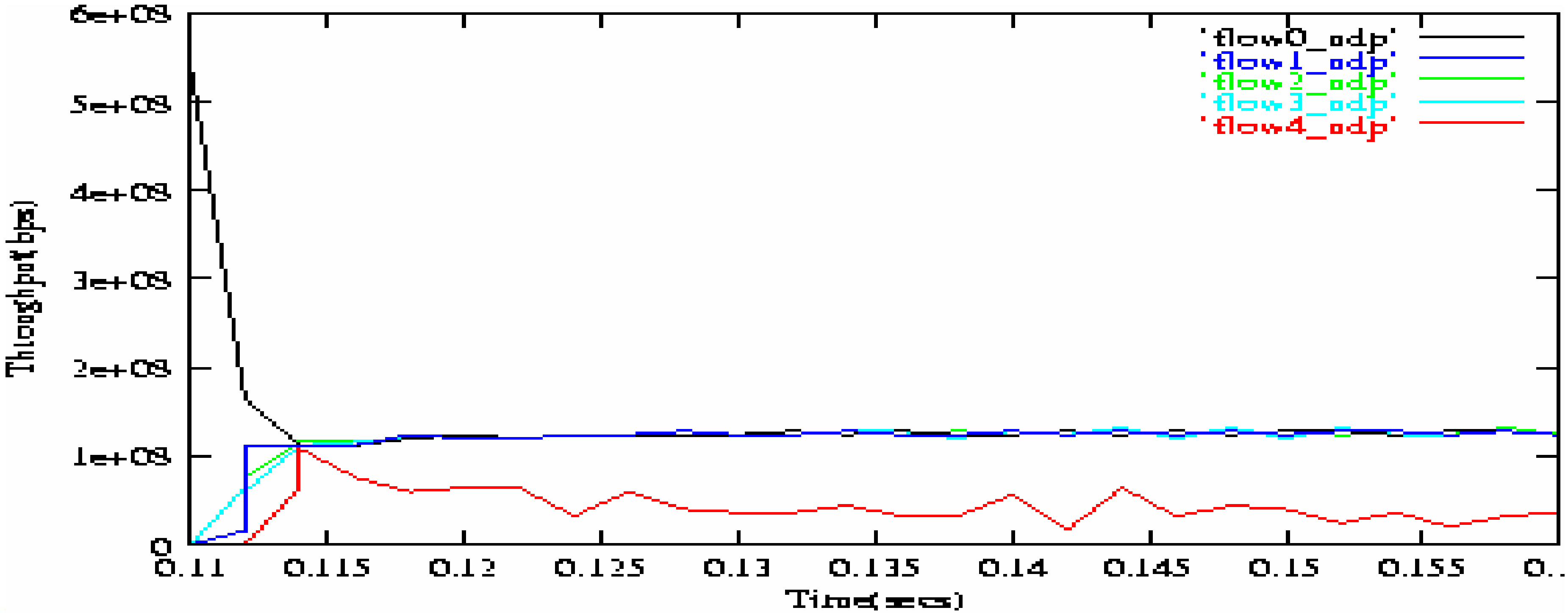


Further Observations on Startup Phase

- Node 4 detects congestion (based on "nrXmit_rate > unreserved_rate")
  - Forces upstream nodes to reduce transmit rate, relieving congestion at node 4
  - Node 4 STQ is below low_threshold, thus local fair rate allowed to ramp up
    - according to "row 6" of conservative mode scheme
  - The advertised rate allows upstream nodes to increase their add rate
    - Node 4 shaperD credits oscillates around Low_limit but eventually falls below Low_limit and the node is starved as upstream nodes ramp up their rate (see next slide for close up view of rates as time evolves)
    - Node 4's shaperD credits decremented at or above unreserved rate (up to link rate), but incremented only at unreserved rate

Further into the Initial Phase

- Node 4's local fair rate **ramps up** because STQ is below Low_threshold: allows upstream nodes to speed up
- Node 4 can send, but at a lower rate because it's shaperD credit oscillates around Low_limit
- Ultimately, node 4 is **unable to transmit** (because it is not slowing down upstream nodes, but it's shaperD credits are below Low_limit) - starvation

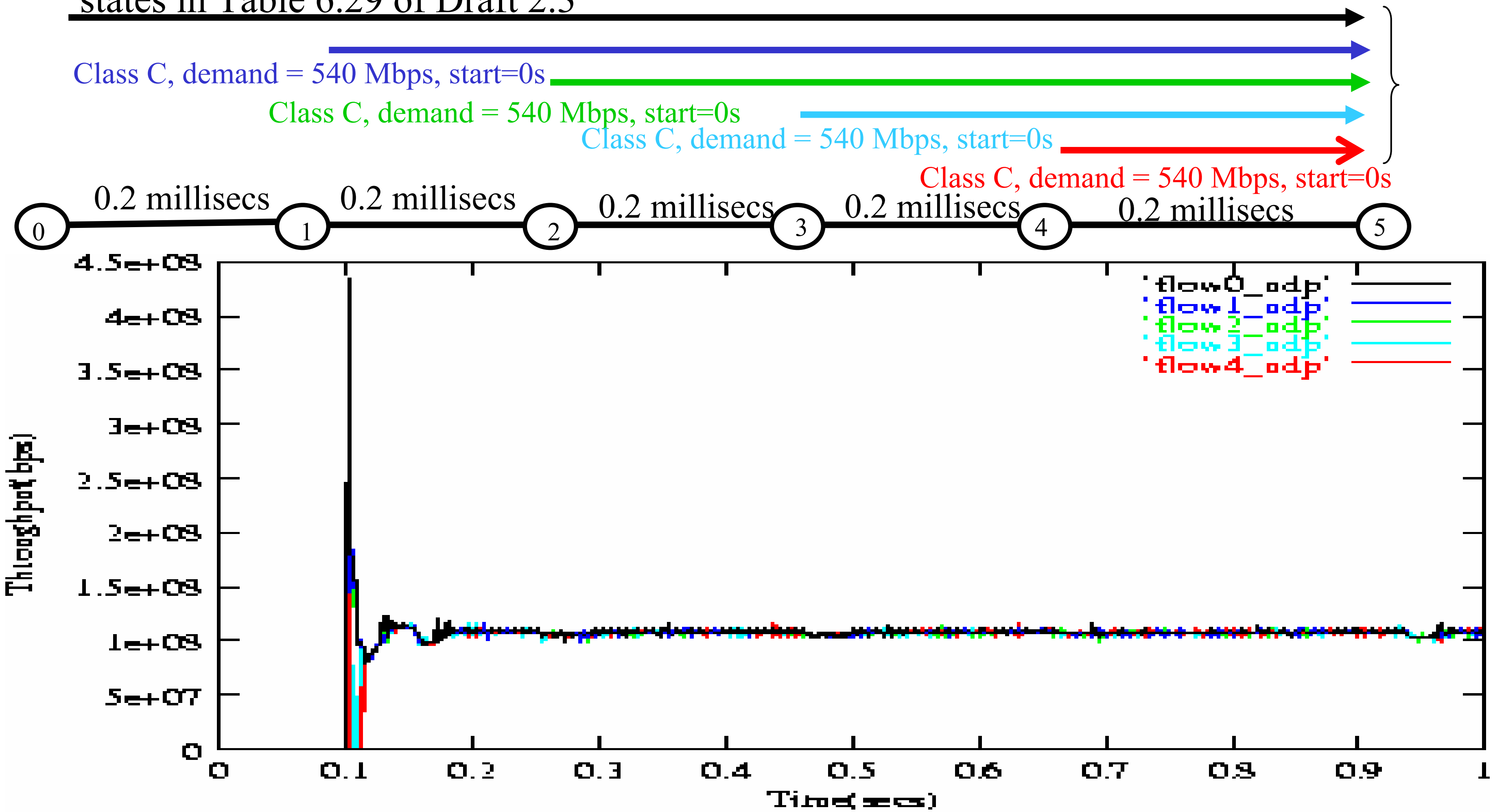


Fundamental Difficulty

- The fundamental difficulty is:
  - ShaperD credits at a station are incremented at unreserved rate
  - However, when upstream stations transmit data, and transit traffic is forwarded, ShaperD credits are decremented at link rate
- If shaperD credits are below Low_limit, then a station is not allowed to transmit
  - Continued forwarding of transit traffic prevents a station from building up credits to allow it to transmit (i.e., go above Low_limit)
- We observe a station (e.g., station 4) has an STQ buffer occupancy below Low_threshold, but is still starved
  - Does not add traffic, but does not reduce advertised rate to upstream (i.e., nrXmitrate ≤ unreserved_rate; STQ occupancy < low_threshold – remains in Row "6")
    - **Hence station allows upstream stations to ramp up their rate according to Row 6!**
- **Fundamental Need: Match the shaperD credit increment rate to credit decrement rate for correct operation**
- Alternative: push down upstream stations much more aggressively so that the aggregate upstream rate is below unreserved rate
  - The aggressive mode attempts to do this, but at the cost of fairness and oscillation

Proposed Solution: Shape STQ traffic

- Reserved rate of 60 Mbps (no reserved traffic is being sent)
- add passD to condition for selecting a packet from STQ in dual-queue transmit selection states in Table 6.29 of Draft 2.3



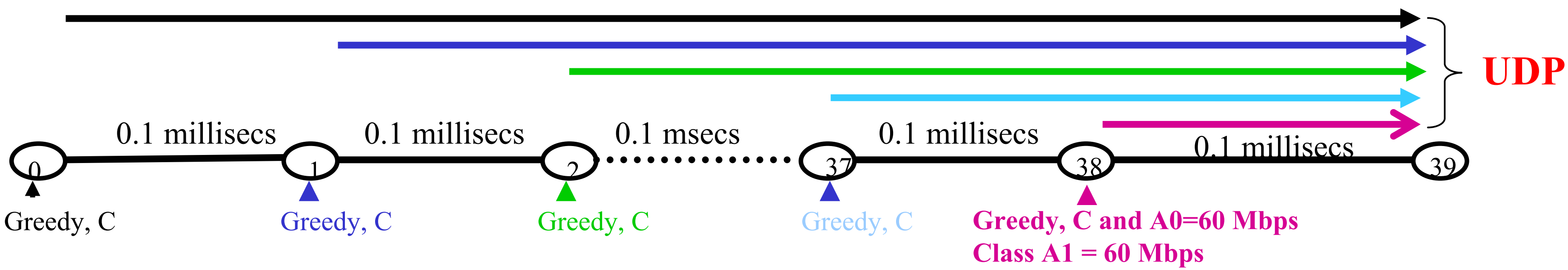
Observations on Solution Approach

- All of our previous analyses of the conservative mode (fairness, utilization) with just fairness eligible traffic will *hold* with the suggested change
  - ShaperD being applied to the transit traffic at a node ensures that at no node do we have a rate of transmission above "unreserved_rate".
  - Hence, the criteria used in prior analyses remains, isolating Class A0 reserved traffic from fairness eligible traffic
- Shaping transit traffic ensures:
  - ShaperD incremented & decremented at commensurate rates: "unreserved_rate"
  - Isolates and eliminates all interactions between FE traffic and reserved traffic
  - Completely precludes impact on Class A0 reserved traffic by fairness eligible traffic and the dynamics of the congestion control feedback mechanisms
- We believe this will have a superior overall performance and fairness compared to the alternative scheme of pushing upstream stations down based on local "add_rate" (i.e., aggressive scheme)

Co-existence of Class A0, A1 and Class C traffic

Co-existence of Class A0, A1 and C traffic

- Scenario (3a) used to demonstrate co-existence of Class A0, A1 and maintenance of their guarantees in the presence of Class C traffic
- 40 nodes, with the last hop sending Class A0 traffic, Class A1 plus Class C traffic
  - Class A0 and Class A1 traffic starts at time T = 0; all class C traffic start at T = 0.1 seconds



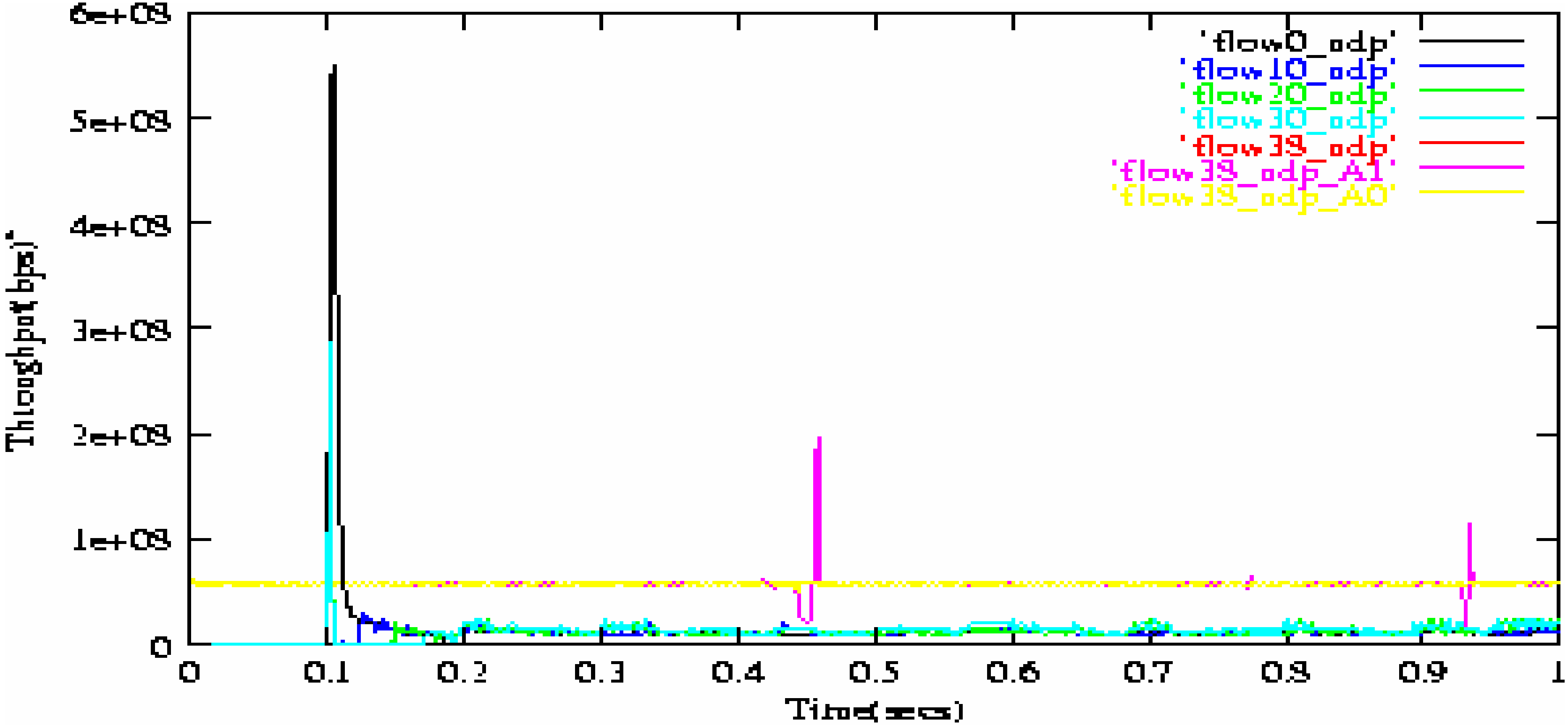
- ShaperD shapes STQ traffic as well as stops local addition of all Class B and Class C add traffic (when below Low_limit)
- Also, when ShaperD credits go below Low_limit, stop all Class A1 add traffic also (interpretation of current spec.)
- Also assumed that when Class A1 traffic is forwarded from PTQ or added, ShaperD credits can go below Low_limit. However, Class B and Class C traffic can only be added when ShaperD credits go above Low_Limit.

Simulation Parameters

- Parameters:
  - STQsize = 256 Kbytes
  - **Advertisement interval = 10 microseconds (results similar for 100 µsecs)**
  - **Active station estimation interval = 10 milliseconds**
  - Aging interval = 0.1 milliseconds
  - Link Rate = 600 Mbits/sec
  - Low_Threshold = 1/8 * STQ, Medium_Threshold = 3/16 * STQ, High_Threshold = ¼ * STQ
  - rampcoef=64
  - Shaper parameters
    - Low_limit = 1 MTU
    - High_limit = 2 MTU

Performance with Scenario 3a, Draft 2.4

- ShaperD is applied to Class A1 traffic also, from node 38 (as per current spec.)
  - **Note: STQ is already being shaped**
- Class A1 traffic gets hit for a brief time, periodically
  - Due to STQ buffer occupancy goes above Full Threshold

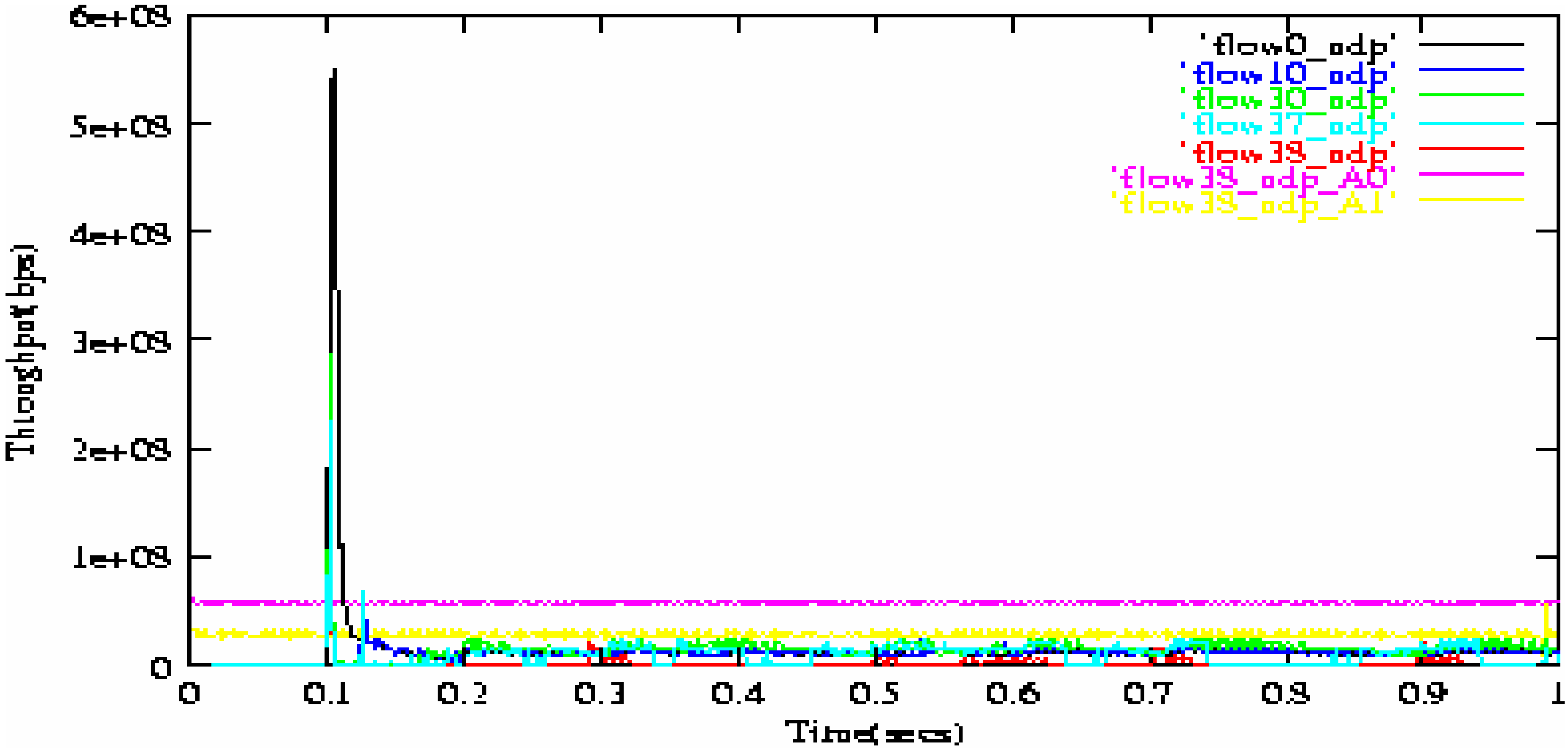


Limit (Class A1 + Class B-CIR) Rate to avoid even brief starvation

- The maximum Class A1 rate recommended in Appendix G provides a guideline for how large Class A1's rate can be:
  - Feedback is generated once STQ reaches STQLowThreshold
    - Default STQLowThreshold = 1/8 * sizeSTQ (based on default thresholds)
  - 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)/***unreservedRate***
    - FRTT' = (round_trip propagation delay + # hops * advt. delay + queueing delay + aging_filter_reaction_time*) (*: TBD)
- Estimate of max. Class A1 rate can be calculated as:
  - RateA1 <= (sizeSTQ-stqHighThreshold)/(FRTT') – ***ClassB(CIR)***

Performance with lower limit for Class A1

- Reduce Class A1 traffic = 30 Mbps, per our suggestion from Montreal
  - ShaperD **NOT** applied to Class A1 traffic also, from node 38 (doesn't matter)
- Class A1 and Class A0 traffic are NOT impacted
  - STQ occupancy remains below Full Threshold

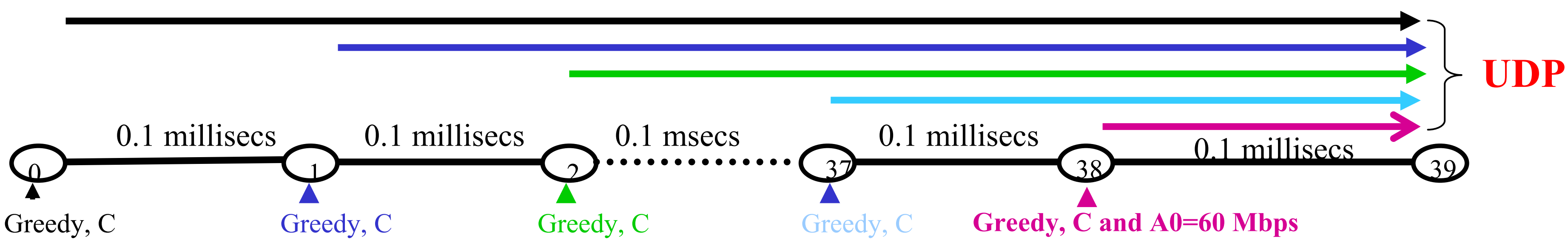


Scalability Issues for the Conservative Mode

Scalability of Conservative Mode Fairness

Scenario is used to demonstrate the scalability of the conservative mode

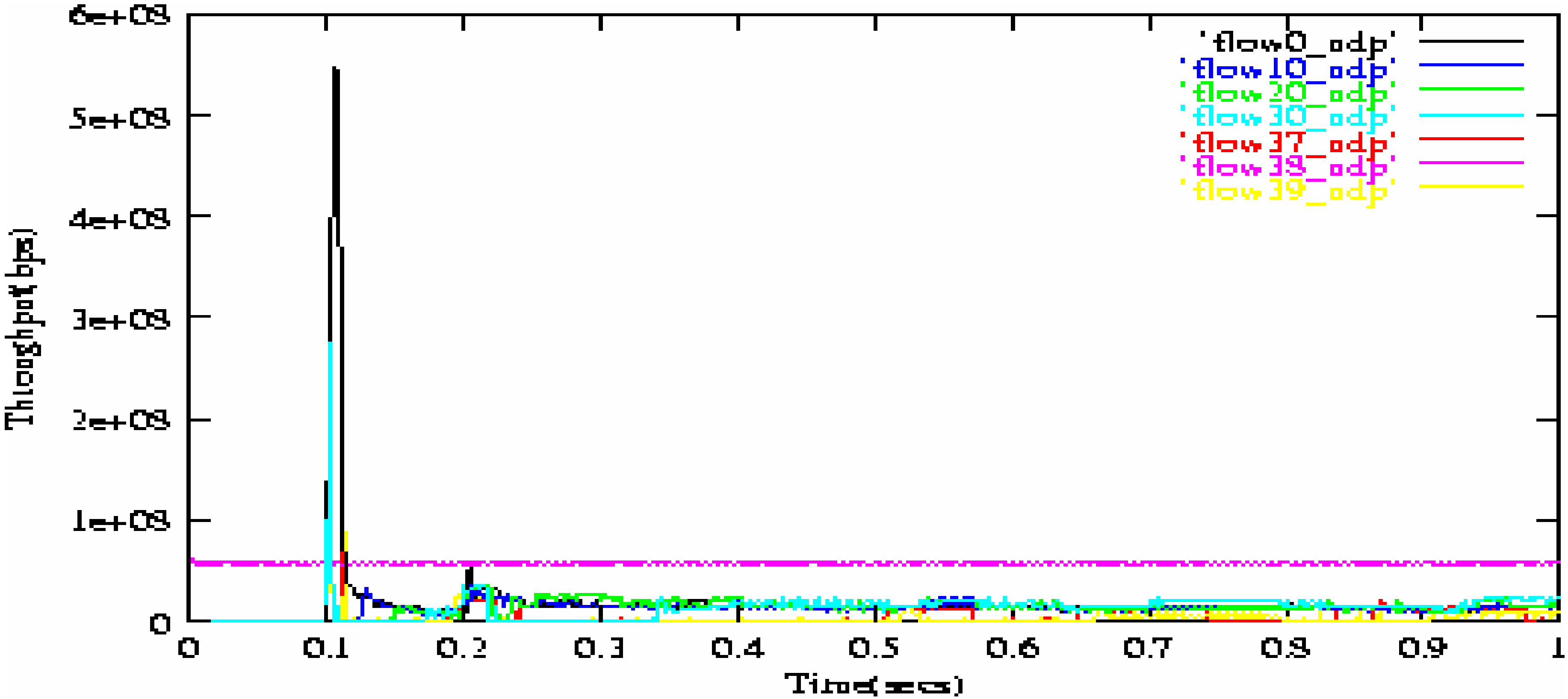
- Large # of active stations - 40 nodes, with only the last hop sending Class A0 (reserved) traffic = 60 Mbps plus Class C traffic
- Class A0 traffic starts at time T = 0; all class C traffic start at T = 0.1 seconds



- Performance with different values for rampCoef in Row 6 (increase) = 1/256; rampCoef in Row 5 – left unchanged at 1/64
- Estimation interval for active stations = 10 milliseconds (100 aging intervals)
  - Concerns about implementation: rounding/truncation of rate
    - (we had implemented the local_fair_rate calculation using integer arithmetic)

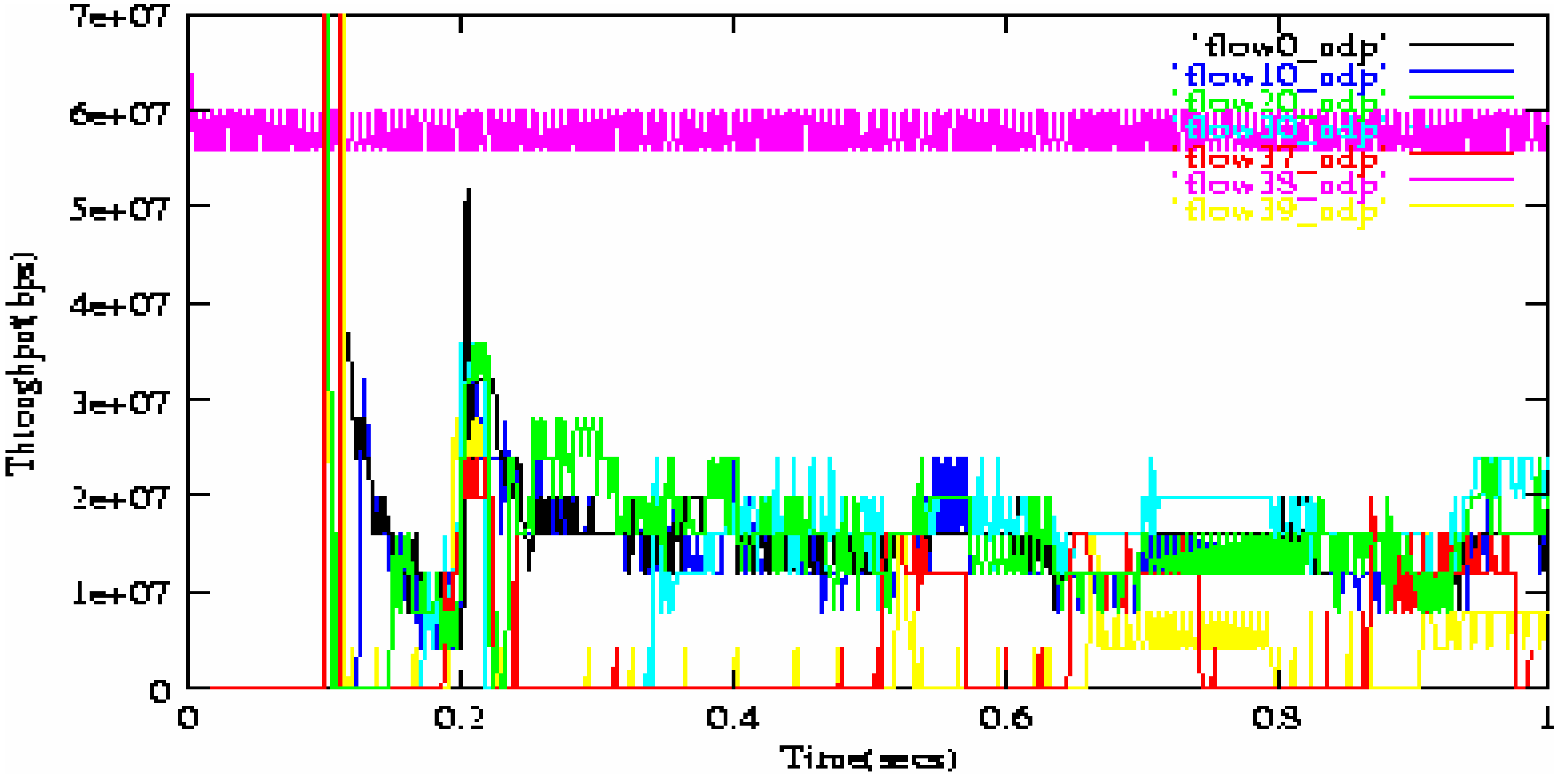
Scenario 1 with rampcoef = 64 for increase and decrease

- 40 nodes, with only the last hop sending Class A0 (reserved) traffic of 60 Mbps
  - Shaping STQ; credits reset when station has NO packets to send (doesn't matter)
- Rampcoef for increase (Row 6) = 1/64 = rampcoef for decrease (Row 5)
- Considerable periods of starvation for FE traffic from nodes 37 and 38



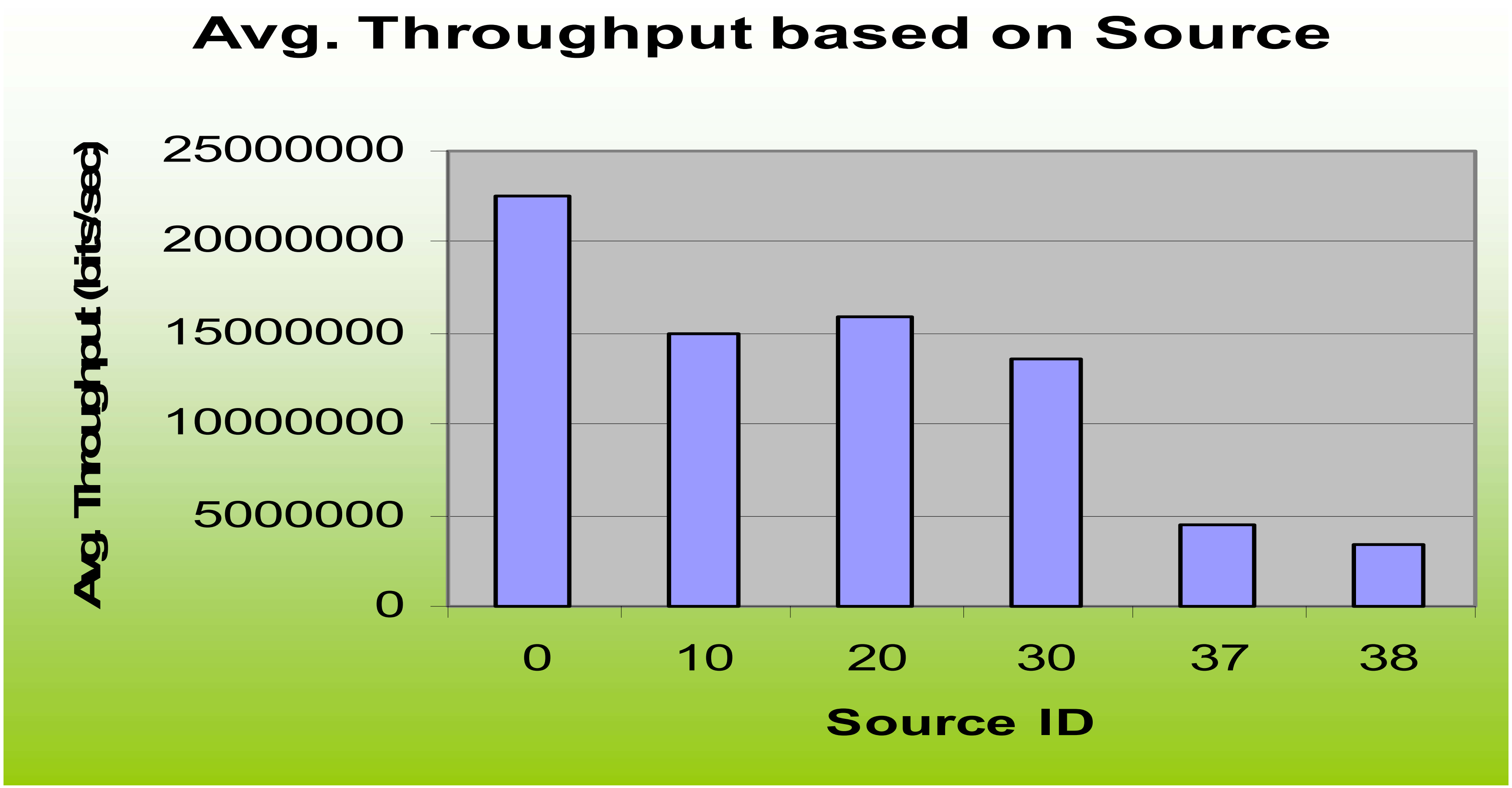
Scenario 1 with rampcoef = 64 for increase and decrease

- A closer look by examining the behavior of the FE traffic without the initial transient
  - Considerable periods of starvation for FE traffic from nodes 37 and 38 (flow 39)



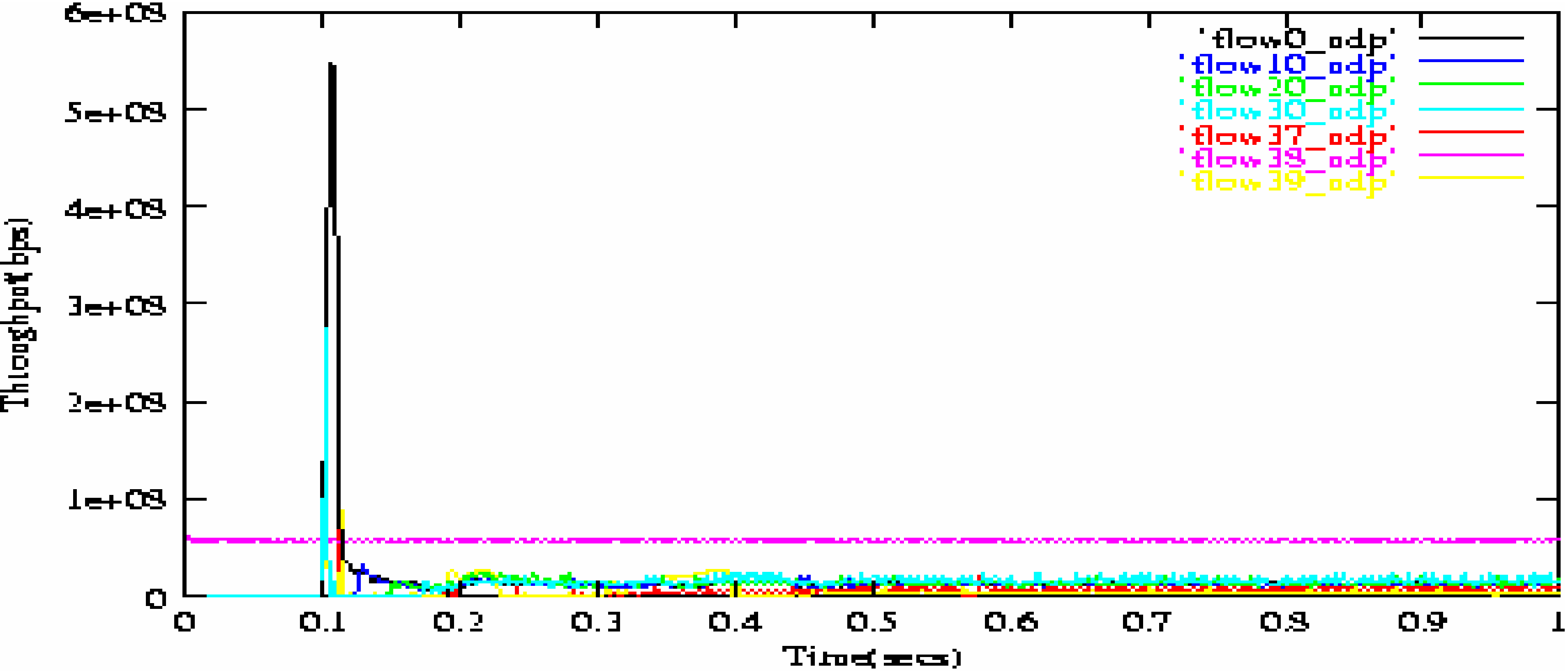
Fairness: Avg. Station Throughput

- Examine average station throughput, measured over the total simulation time, for selected stations.



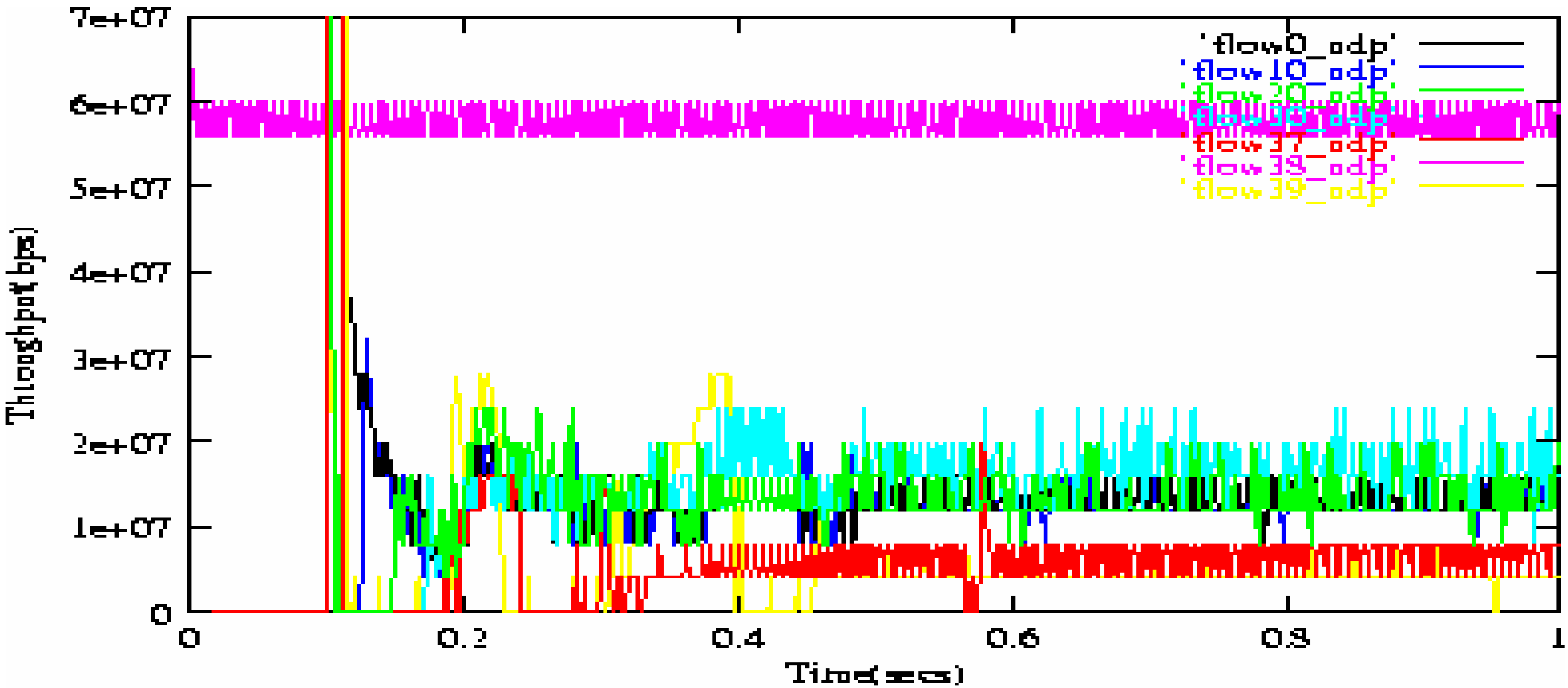
Scenario 1 with asymmetric rampcoef for increase and decrease

- 40 nodes, with only the last hop sending Class A0 (reserved) traffic of 60 Mbps
- Shaping STQ; shaperD credits reset when station has NO packets to send
- Rampcoef for decrease (Row 5) = 1/64; rampcoef for increase (Row 6) = 1/256
- Reduced starvation considerably, slightly improved fairness



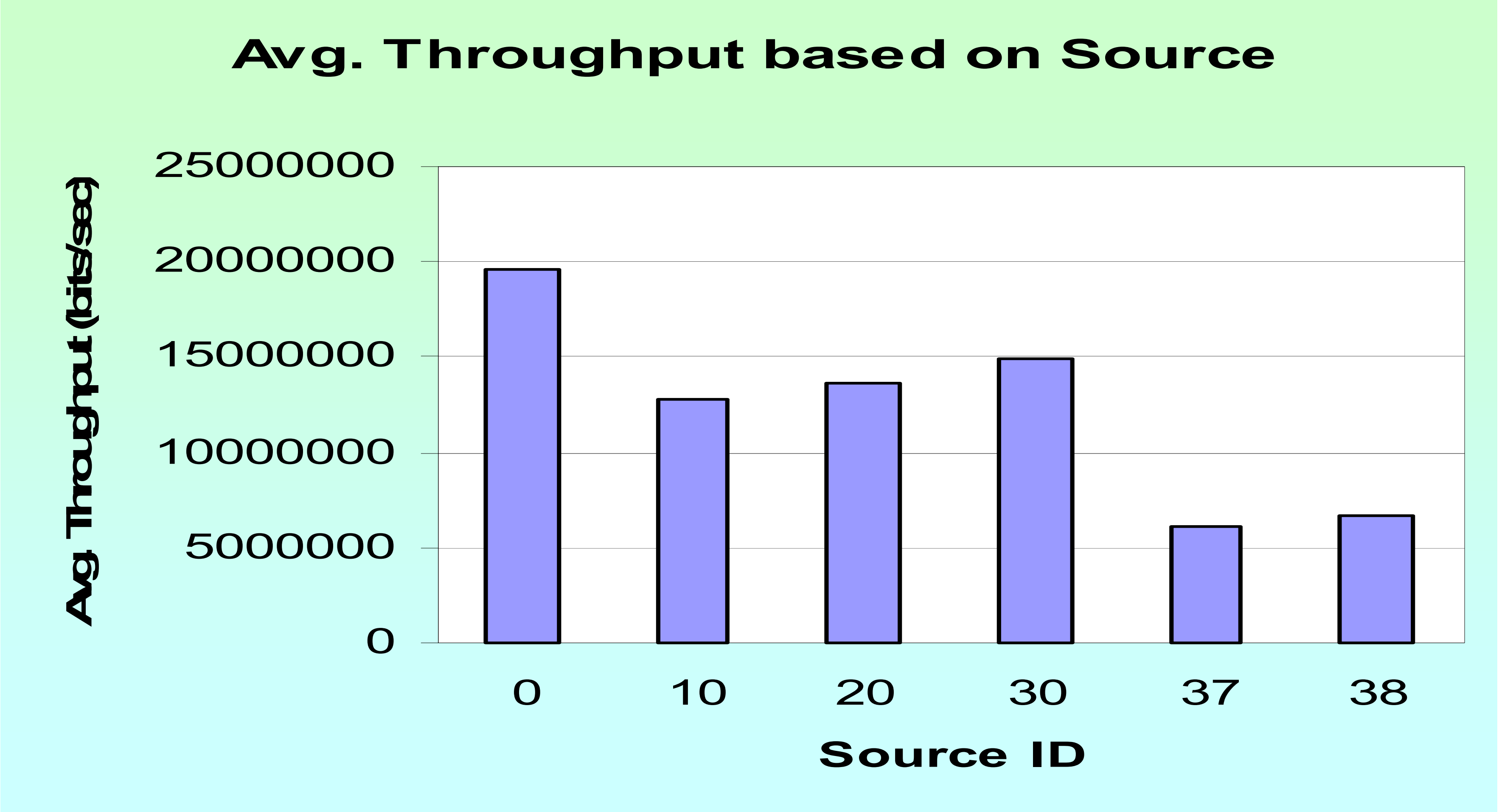
Scenario 1 with asymmetric rampcoef for increase and decrease

- Shaping STQ; shaperD credits reset when station has NO packets to send
- Rampcoef for increase (Row 6) = 1/256; rampcoef for decrease (Row 5) = 1/64
- A closer look, ignoring the initial transient (flow 37 and 39 almost overlap)
- Reduced starvation considerably, slightly improved fairness



Fairness: Avg. Station Throughput

- Examine average station throughput, measured over the total simulation time, for selected stations.



Summary

- STQ shaping enables avoid starvation of Fairness Eligible Traffic in the presence of Reserved traffic (Class A0)
- For the scenarios we have examined through simulation, we observe that Class A0 rate (and probably appropriately set delay) guarantees are met
  - With the conservative mode for FE traffic
- Class A1 rate guarantees can also be met, as long as the rate of class A1 traffic is *suitably* limited
  - Update the formula in Clause 6 (and corresponding explanations in Appendix G) to reflect the lower limit on Class A1, to continue to meet rate guarantees in the presence of Fairness eligible traffic
- To have a scalable conservative scheme for FE traffic, decouple the parameters for increase and decrease
  - Define a rampUpCoef (Row 5) and a rampDownCoef (Row 6)
    - Make them independently configurable;
    - We recommend that rampUpCoef be smaller for rings with larger # stations