# Performance Results for Enhanced Conservative Single Choke Fairness Mechanism

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

AT&T Labs. Research, NJ

March 10, 2003

# Introduction

- Discussions in the FAH have been very productive
  - Large amount of additional insight obtained for all participants
  - Spirit of cooperation
- Significant improvement in the protocol
  - Simplification
  - Substantial improvement in overall performance
  - Much more implementation friendly
    - option for lower cost implementation w/ some performance penalty
  - Dramatically improved understanding of performance over wide range of scenarios
- Rough Consensus reached
  - Table 9.4 specification
  - Frequency and accuracy of F-RTT estimation
  - Frequency of active weights estimation

# Protocol changes Accepted

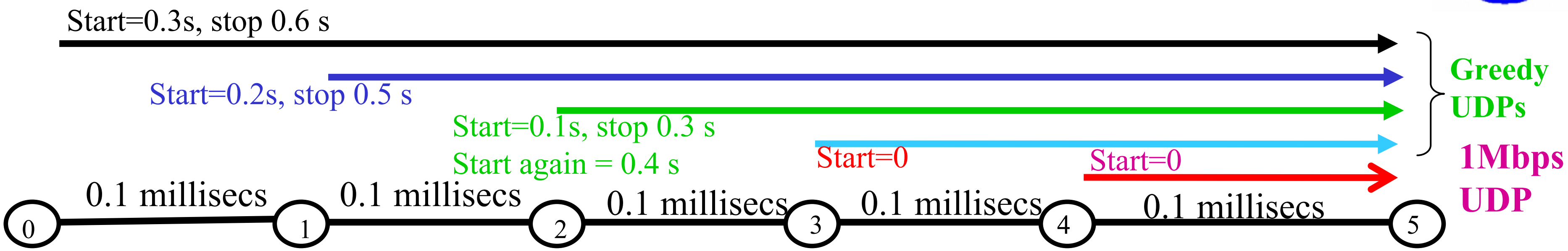
- Table 9.4 modifications:
  - Row 4 condition: localFairRate/localWeight ≥ unreservedRate
  - Row 5 condition: STQDepth > STQMediumThreshold && RTTWorthofIntervalsHavePassed
  - Row 5 action: lower_bound = (lpaddRate+lpFwRate/activeWeights)*Weight; localFairRate = max(lower_bound, localFairRate - localFairRate / RAMPCOEF);
  - Row 6 condition: (STQDepth<STQLowThreshold) && (RTTWorthofIntervalsHavePassed)
  - Row 6 action: LocalFairRate = min(unreservedRate, localFairRate+(localWeight*(unreservedRate-lpAddRate-lpFwRate))/RAMPCOEF)
  - Row 7 action: Compute LocalFairRate as:

    if ( (STQDepth > STQHighThreshold) && (lpAddRate <lpFwRate/activeWeights) ) then

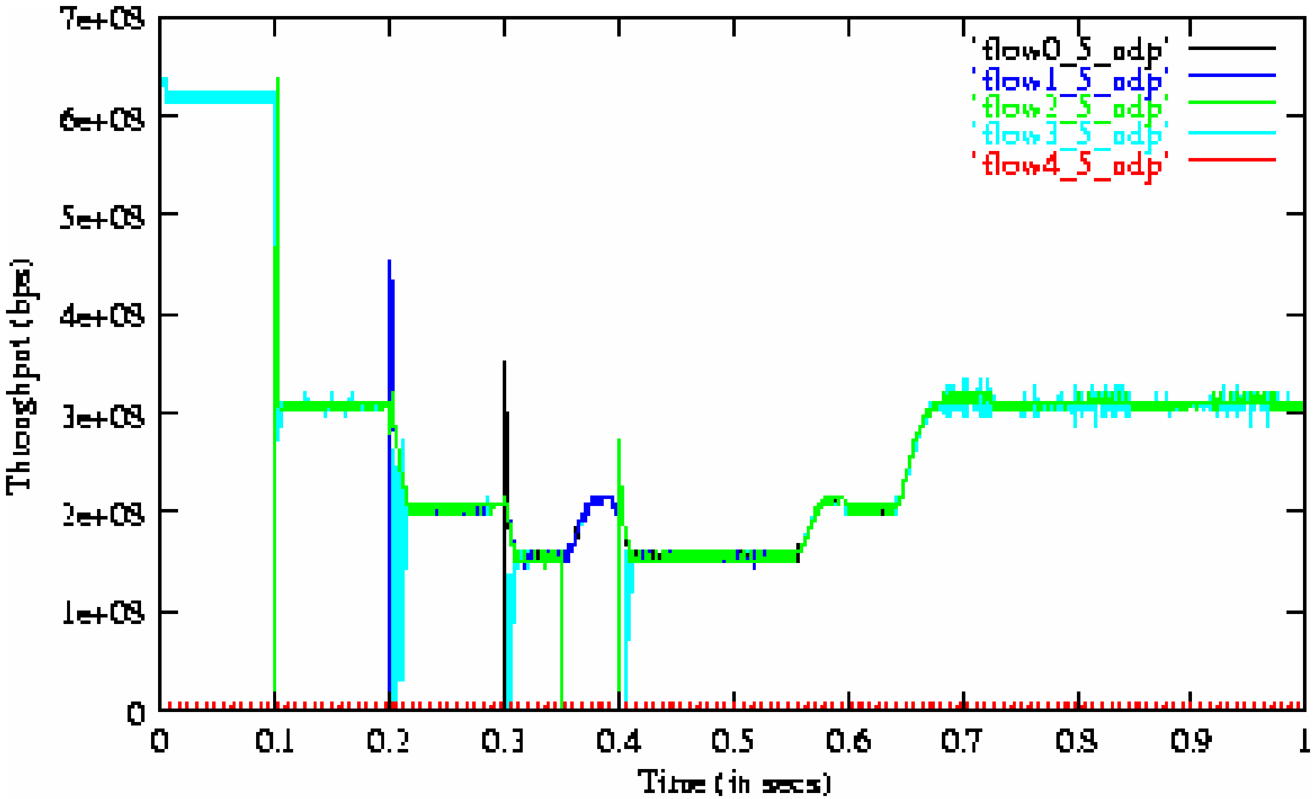
    localFairRate =min (localFairRate, **lpaddRate+lpFwRate/activeWeights*Weight)**

- Throughout, set: allowed_rate = min(unreservedRate, localFairRate);

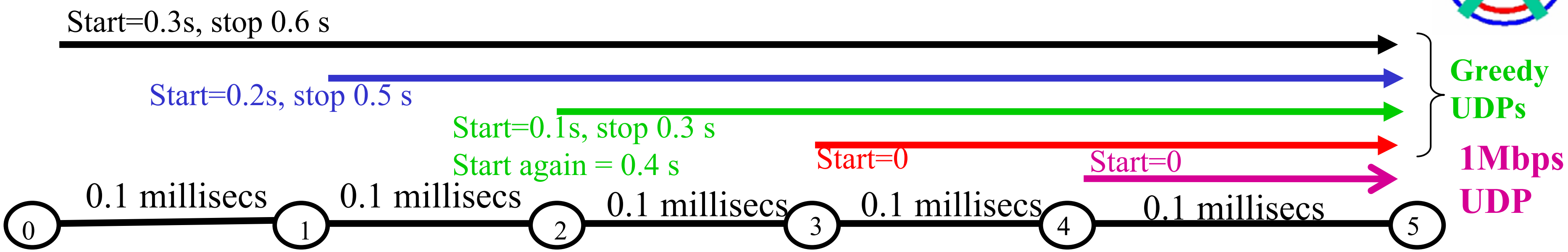
# Estimate F-RTT & # active stations per aging interval



- ❑ Both the active weights and F-RTT are estimated every aging interval of 0.1 msecs.



# Estimation of active stations every 10 milliseconds



Start=0.3s, stop 0.6 s

Start=0.2s, stop 0.5 s

Start=0.1s, stop 0.3 s

Start again = 0.4 s

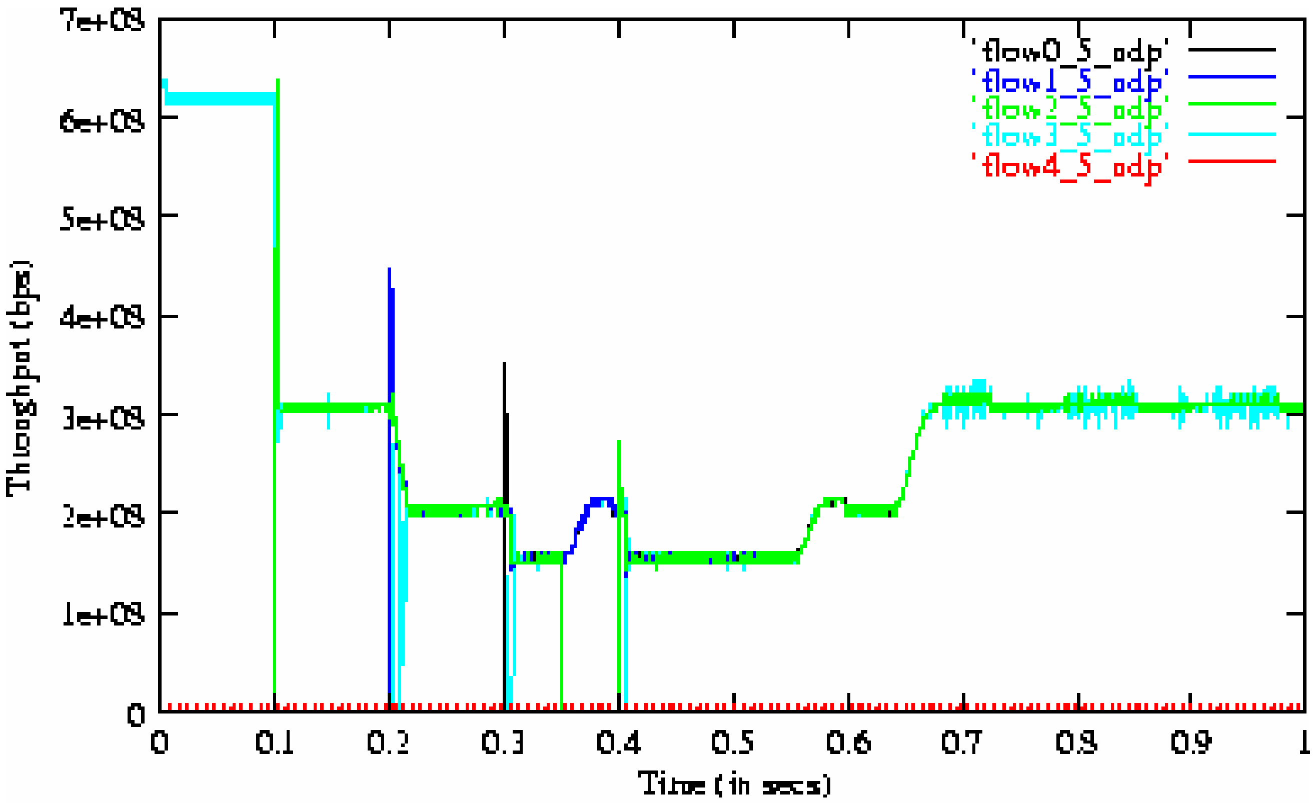
Start=0

Start=0

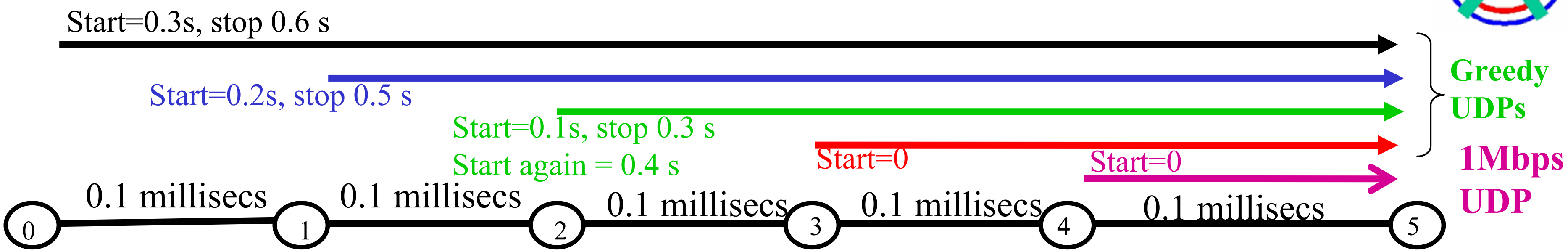
Greedy UDPs

1Mbps UDP

- Active Stations estimated every 10 millisecs, F-RTT estimated every aging interval (0.1 ms)



# Estimation of F-RTT every 10 milliseconds



Start=0.3s, stop 0.6 s

Start=0.2s, stop 0.5 s

Start=0.1s, stop 0.3 s

Start again = 0.4 s

Start=0

Start=0

Greedy UDPs

1Mbps UDP

0.1 millisecs

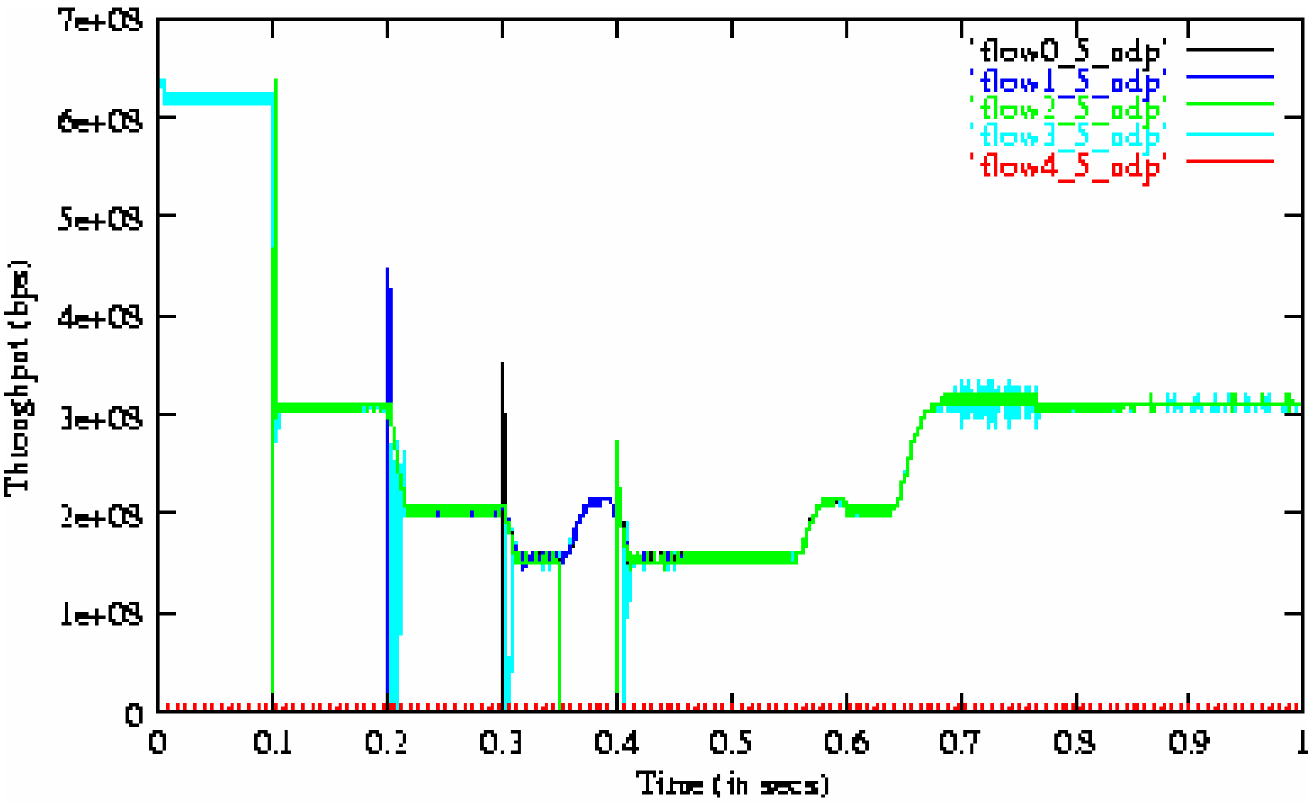
0.1 millisecs

0.1 millisecs

0.1 millisecs

0.1 millisecs

- ❑ F-RTT estimated every 10 millisecs; Active stations estimated every aging interval (0.1 ms)

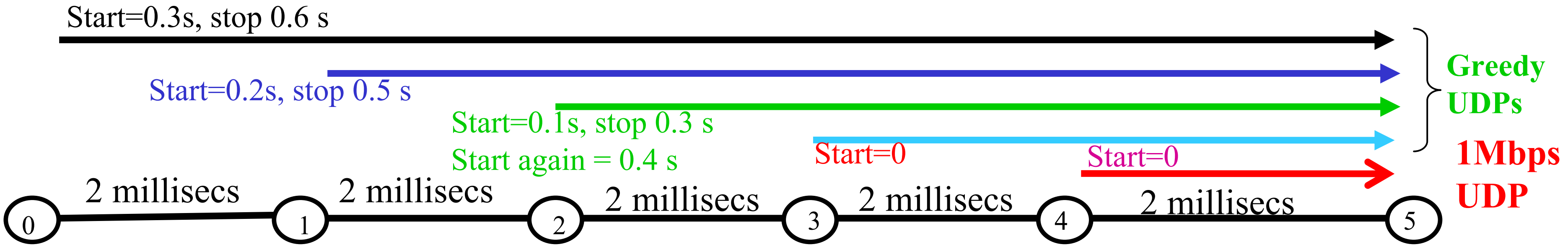


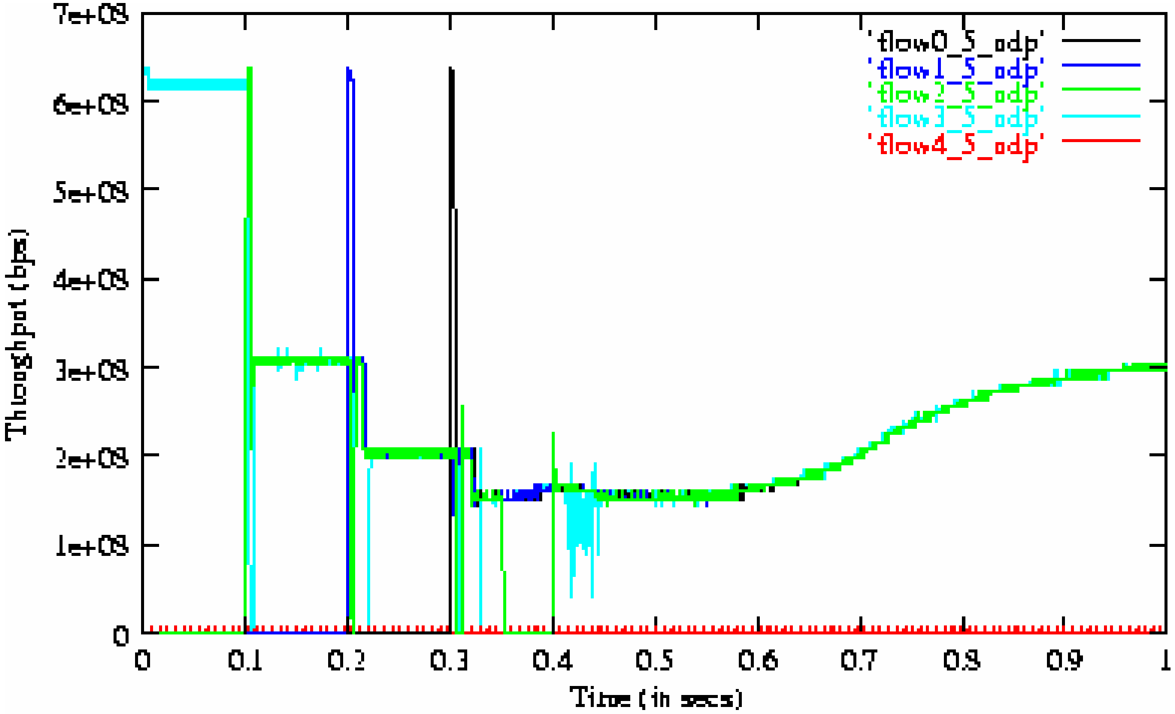
# Frequency & Precision for Measurement of F-RTT

- We determined: acceptable to measure the F-RTT approximately once every 10 msecs.
  - The results (for the experiments we had tested, including ON-OFF scenarios) did not appear to be too sensitive to **frequency** of measurement, as long as it was not too infrequent.
- Question raised: can it be really infrequent: say once every 1 sec?
  - Main reason for dynamically measuring F-RTT: to track changes in the size of the control loop, which can change more frequently
    - We found that measuring less frequently than 10 msecs. impacts utilization in certain scenarios.
- Question raised: how precise would F-RTT have to be measured?
  - Can we be inaccurate to the order of +/- 1 millisecond?
  - Simulations indicated that an inaccuracy of up to 2 milliseconds was acceptable even for small rings with 10 Km span distance

# Modification of Row 5 of Table 9.4

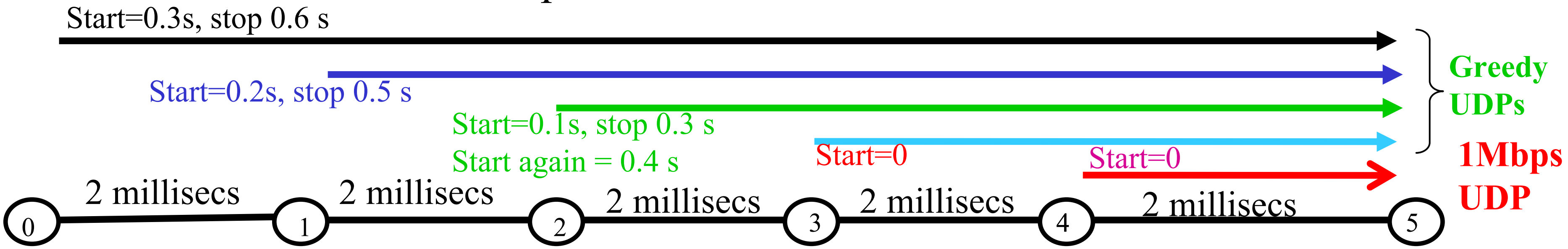
- Condition for dual queue MAC simplified as:
  (STQDepth > STQMediumThreshold) && (RTTWorthofIntervalsHavePassed)
- Action computes a lower bound for LocalFairRate as:
  lower_bound = (**unreservedRate/activeWeights)*Weight**; use it in setting LocalFair Rate in Row 5.





# Without using Lower Bound in Row 5 of Table 9.4

- ❑ Our modification improves convergence and avoids starvation
- ❑ What should “lower bound” be? **(lpaddRate + lpFwRate)/activeWeights)** or **lpaddRate** ?
- ❑ Below: localFairRate computation in Row 5 without our modification



Start=0.3s, stop 0.6 s

Start=0.2s, stop 0.5 s

Start=0.1s, stop 0.3 s

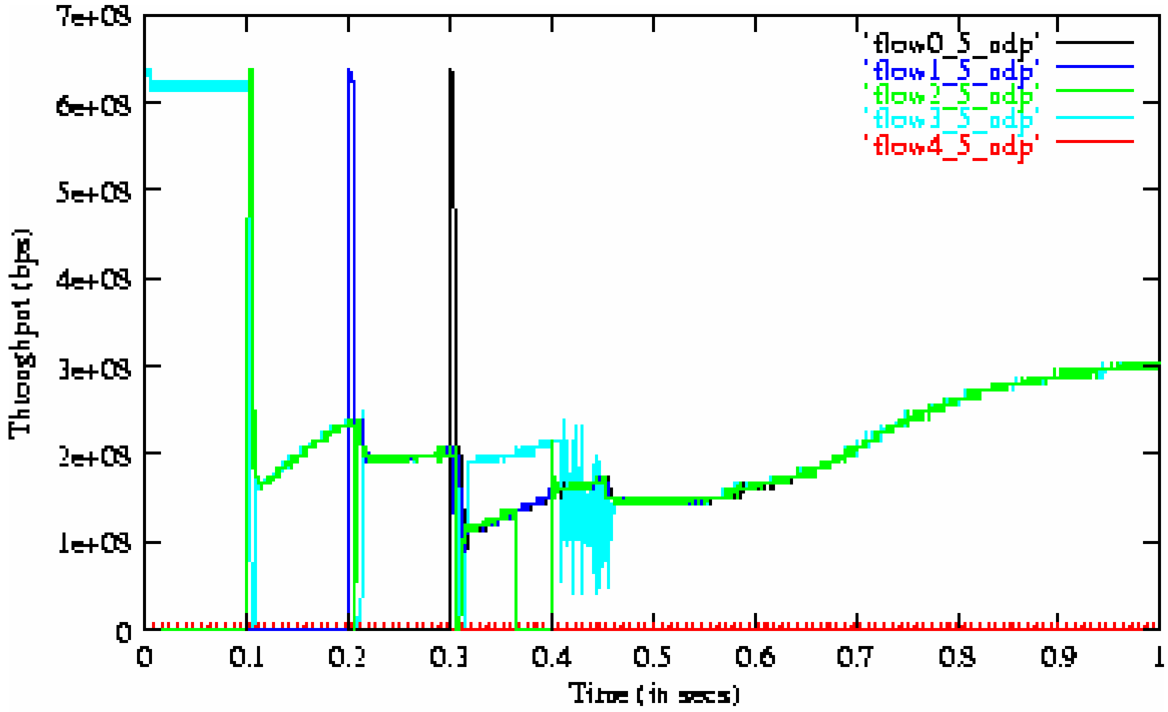
Start again = 0.4 s

Start=0

Start=0

**Greedy UDPs**

**1Mbps UDP**

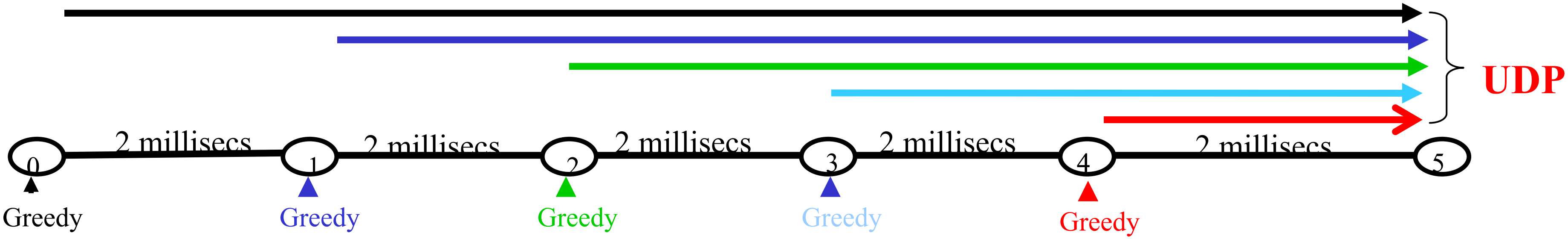


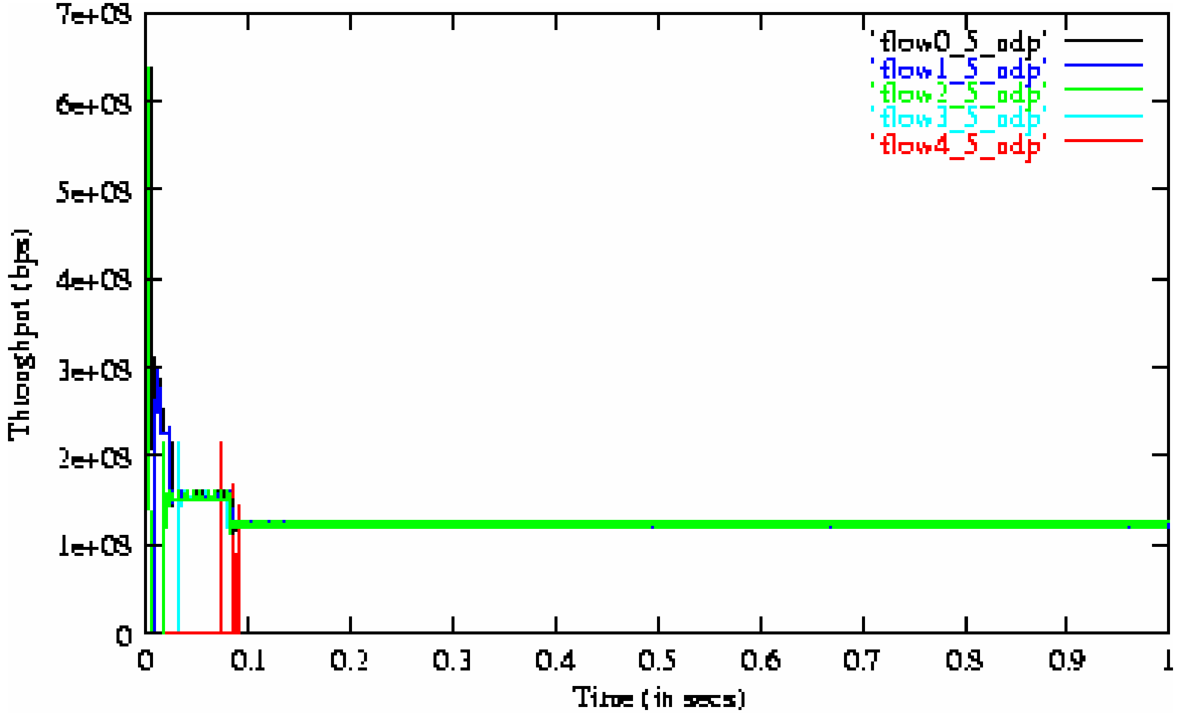
# Modifications to Row 7 of Table 9.4

- Action computes LocalFairRate as (to overcome onset of congestion):

  if ( (STQDepth > STQHighThreshold) && (lpAddRate <lpFwRate/activeWeights) ) then

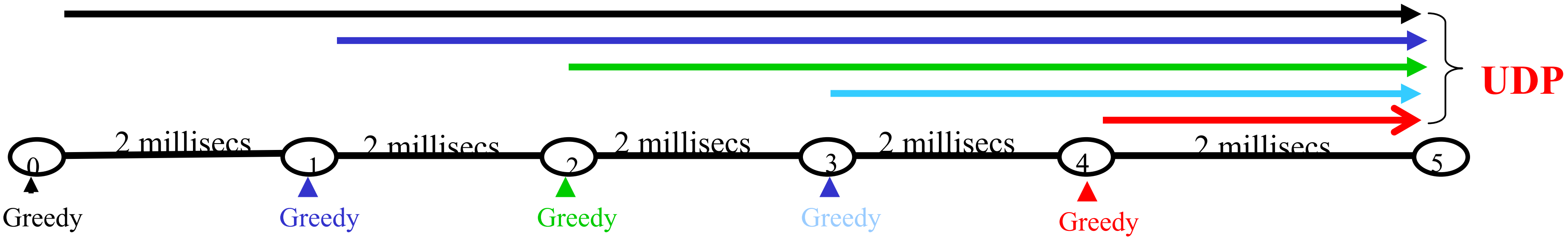
  localFairRate = min ( localFairRate, **unreservedRate/activeWeights*Weight**)

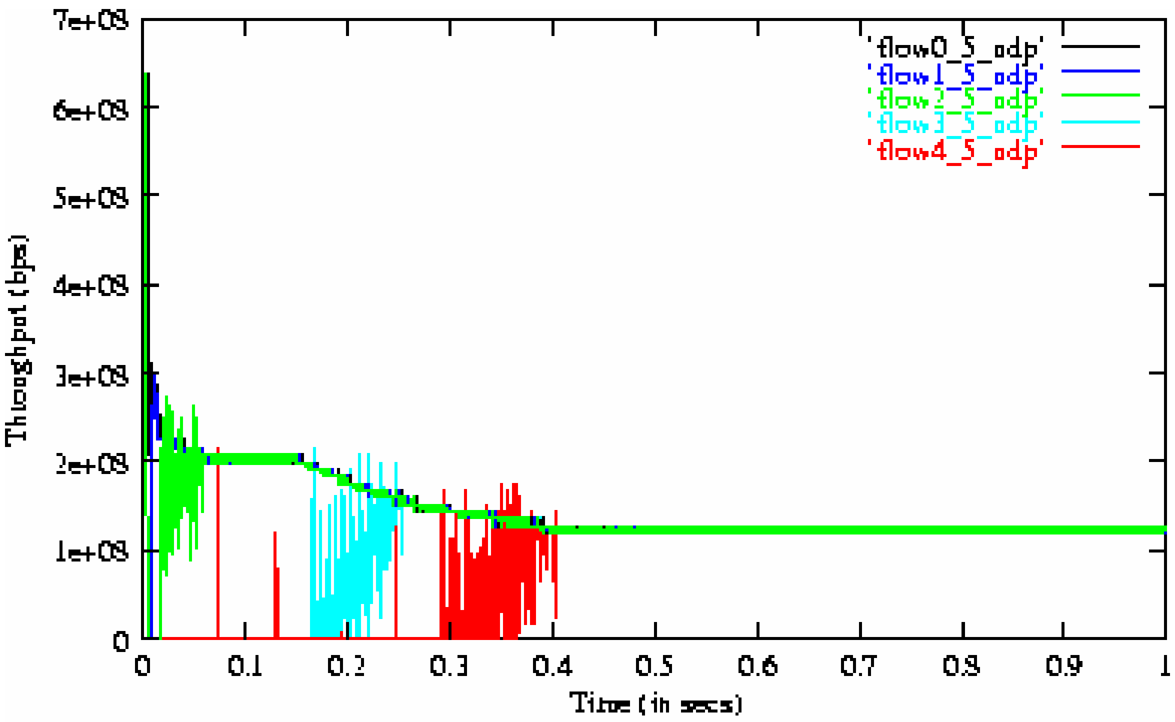




# Without Modifications to Row 7 of Table 9.4

- Below: as in Draft 2.1, do not re-compute LocalFairRate at all.

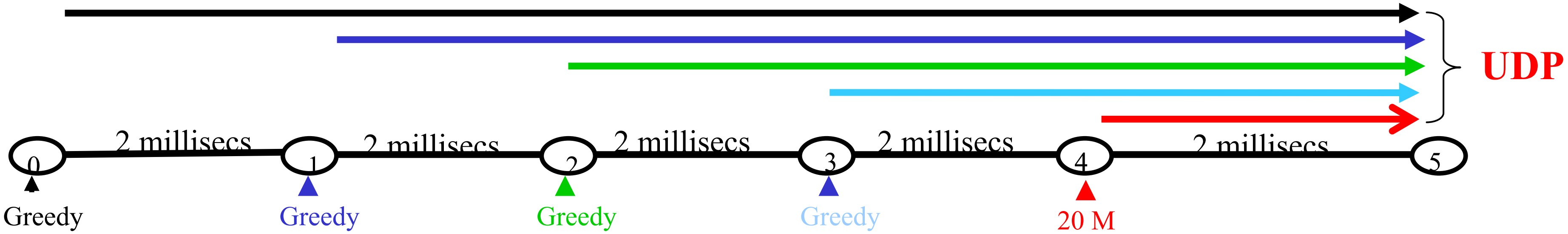


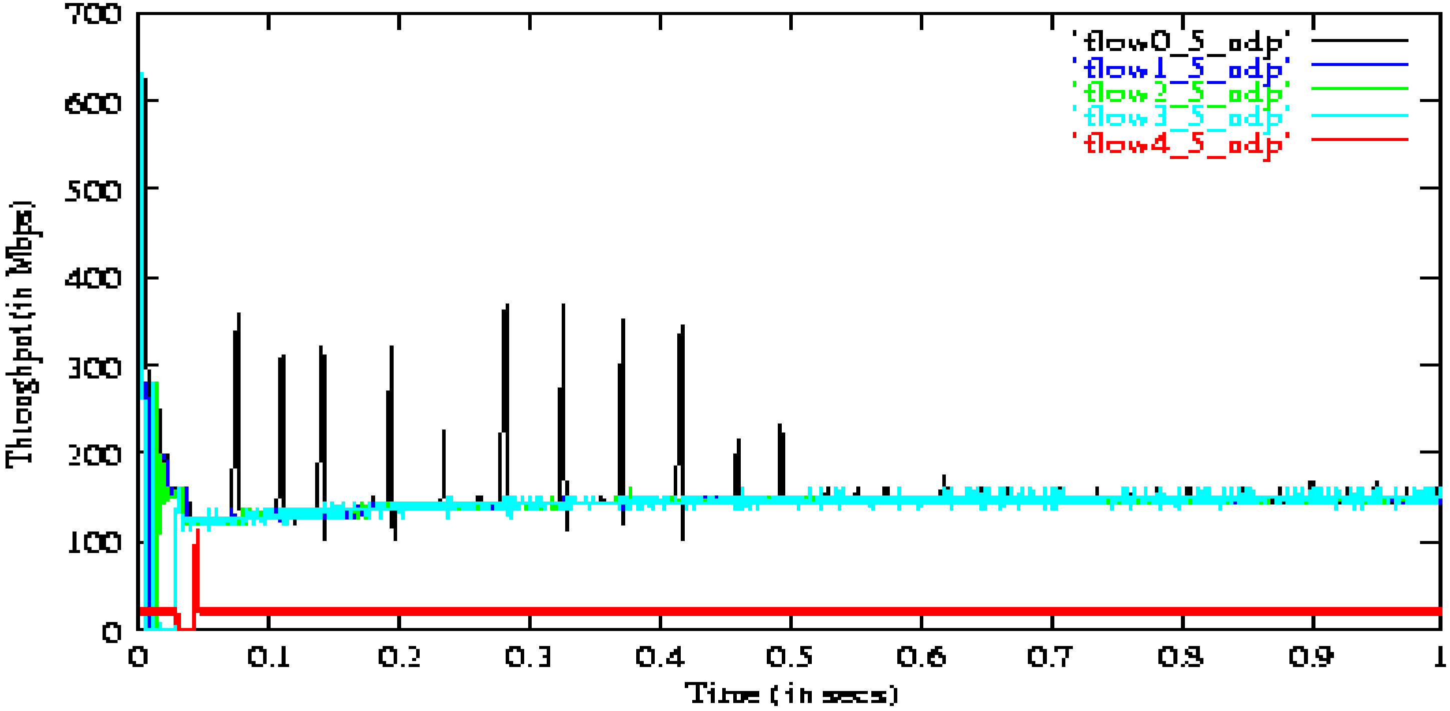


# Option1: unreservedRate ⇒ lpAddRate + lpFwRate

- Table 9.4, Row 5 Action computes lower bound for LocalFairRate as: lower_bound = ((lpAddRate+lpFwRate)/activeWeights)*Weight; use it in setting LocalFair Rate in Row 5
- Table 9.4, Row 7: if ( (STQDepth > STQHighThreshold) && (lpAddRate <lpFwRate/activeWeights) ) then

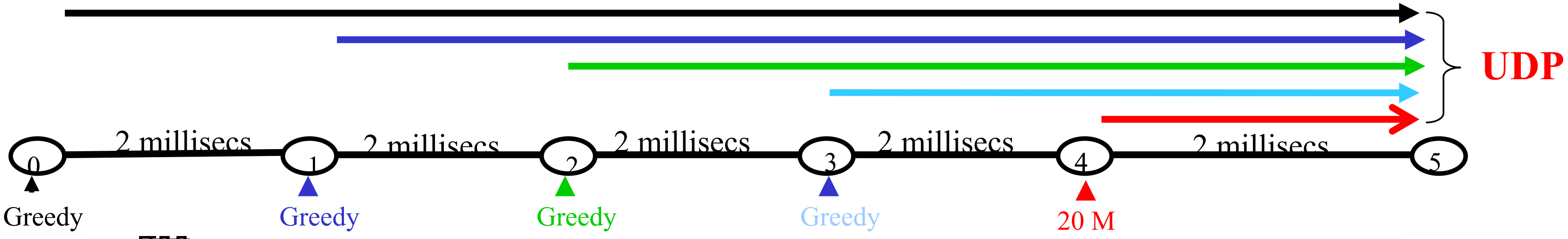
  localFairRate = min ( localFairRate, (lpAddRate+lpFwRate)/activeWeights*Weight)





# Option 2: (unreservedRate/activeWeights*Weight) ⇒ lpAddRate

- Table 9.4, Row 5 Action computes lower bound for LocalFairRate as:
  lower_bound = lpAddRate; use it in setting LocalFair Rate in Row 5
- Table 9.4, Row 7: if ( (STQDepth > STQHighThreshold) && (lpAddRate <lpFwRate/activeWeights) ) then
  localFairRate = min ( localFairRate, lpAddRate)

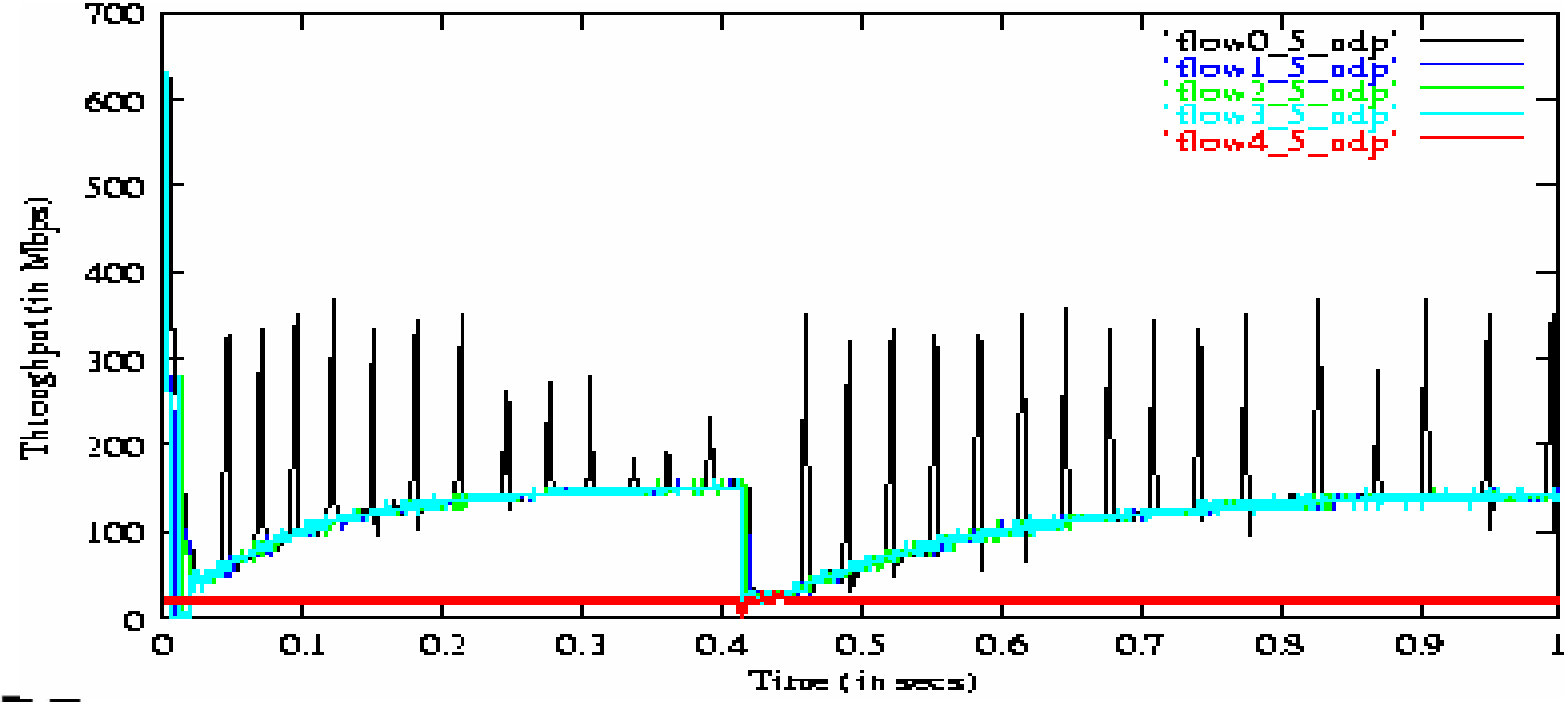


UDP

2 millisecs 2 millisecs 2 millisecs 2 millisecs 2 millisecs

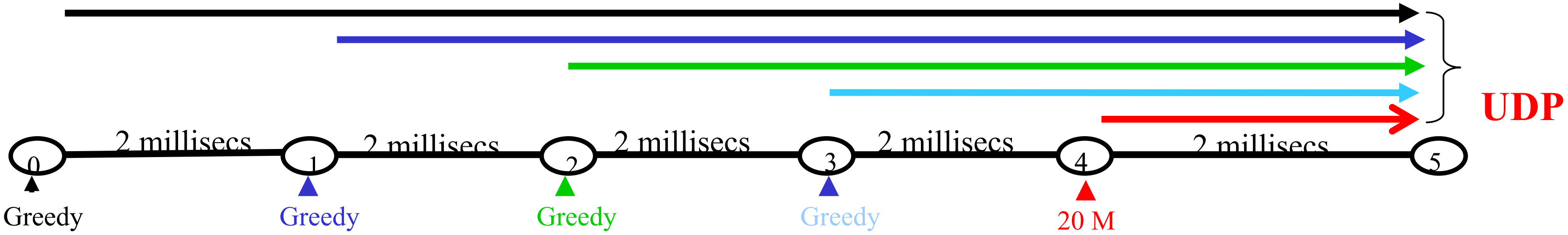
0 1 2 3 4 5

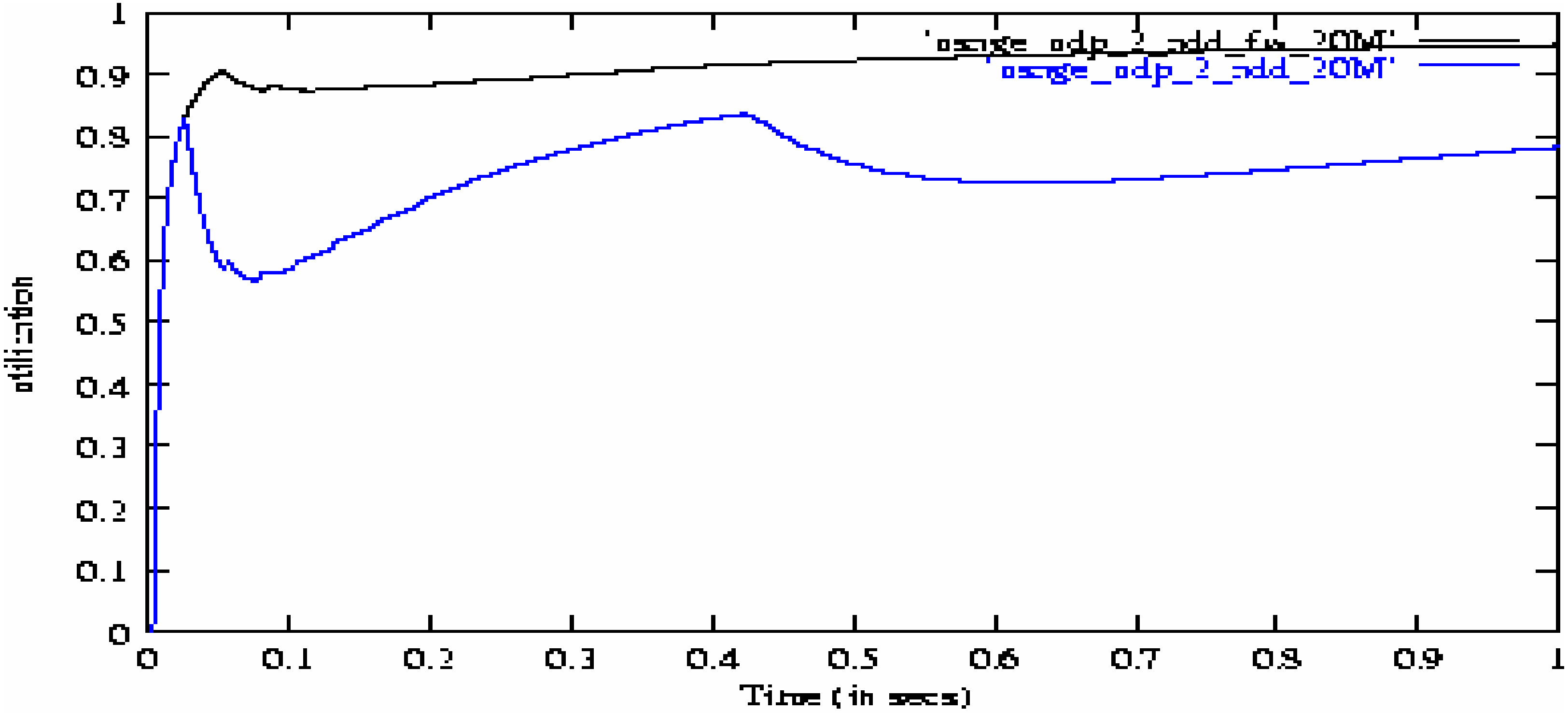
Greedy Greedy Greedy Greedy 20 M



# Link Utilizations for the two alternatives

- ❑ Option 1: unreservedRate ⇒ lpAddRate+lpFwRate
- ❑ Option 2: (unreservedRate/activeWeights*localweight) ⇒ lpAddRate





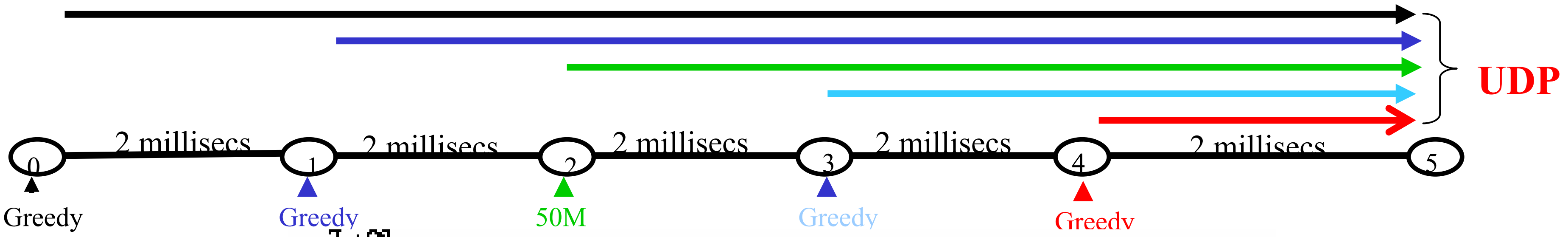
# Modifications to Row 6 of Table 9.4

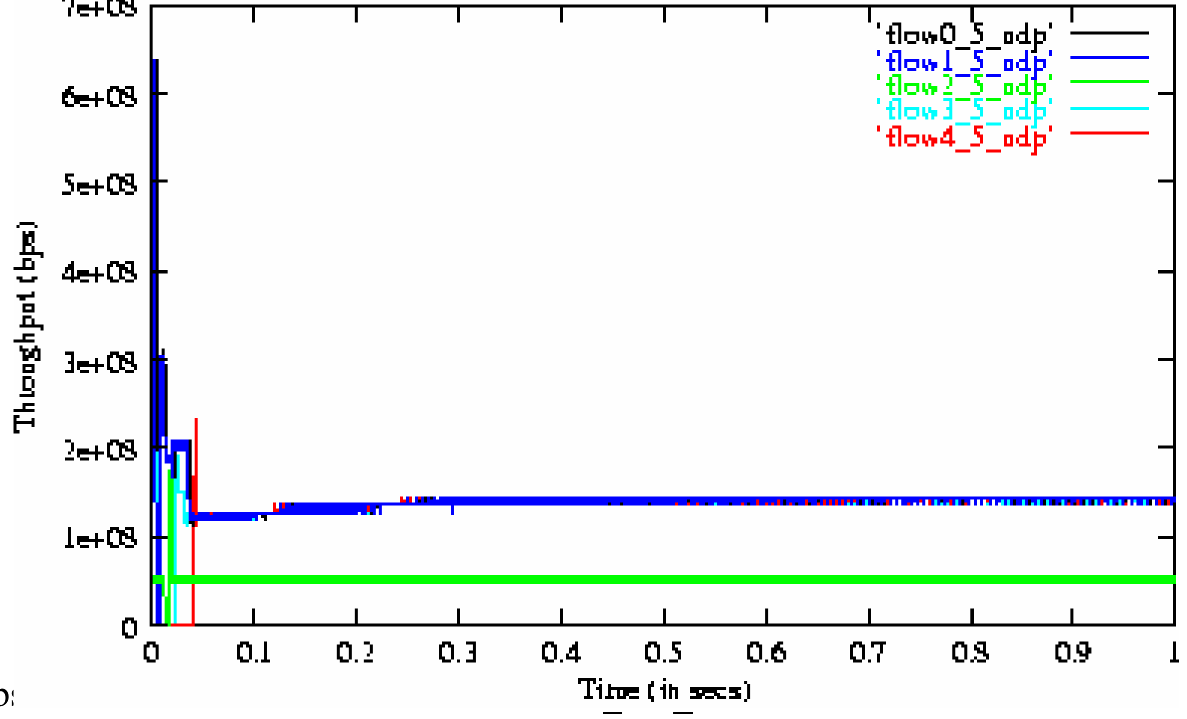
- Condition for dual queue MAC simplified as:

  (STQDepth < STQLowThreshold) && (RTTWorthofIntervalsHavePassed)

- Action computes LocalFairRate as:

  LocalFairRate = min(unreservedRate, localFairRate+(unreservedRate-lpAddRate-lpFwRate)/RAMPCOEF);

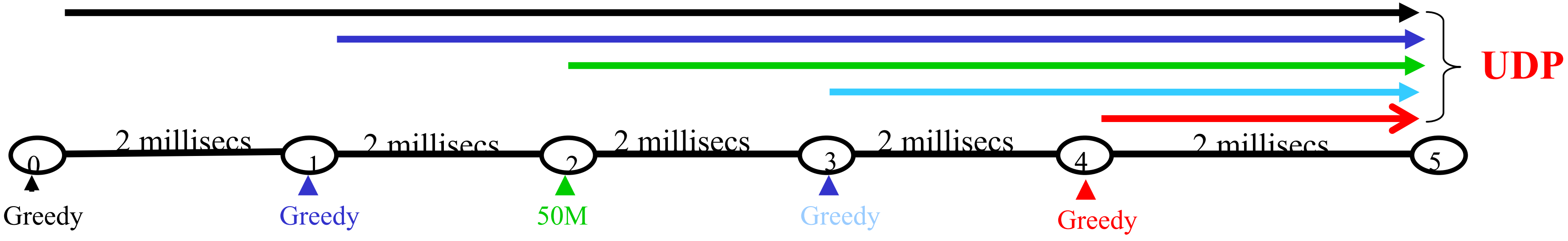




# Without Row 6 modification for LocalFairRate calculation

- Below: action computes LocalFairRate as in Draft 2.1:

LocalFairRate = min(unreservedRate, localFairRate+(unreservedRate- localFairRate)/RAMPCOEF);



UDP

2 millisecs

2 millisecs

2 millisecs

2 millisecs

2 millisecs

0

1

2

3

4

5

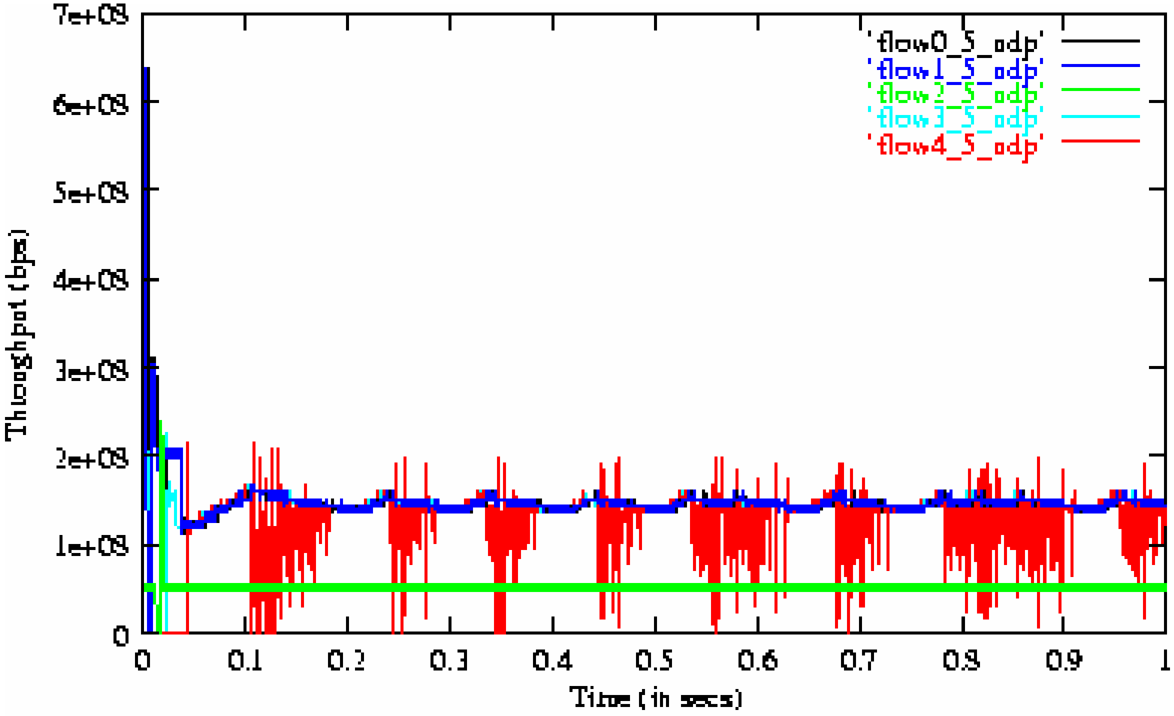
Greedy

Greedy

50M

Greedy

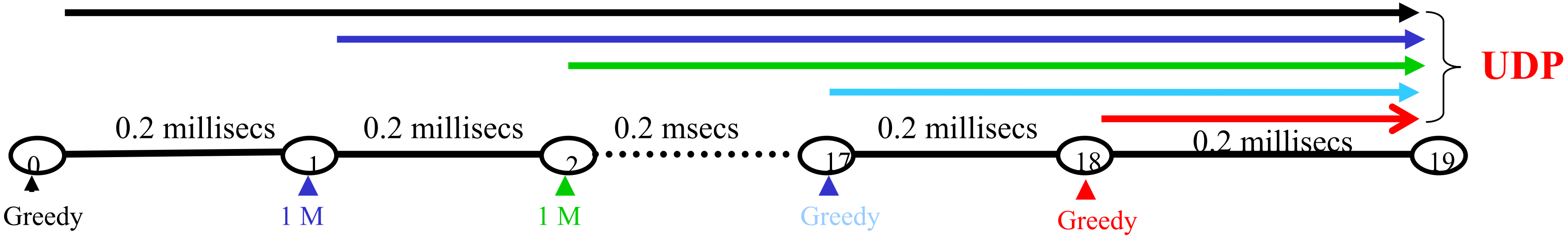
Greedy

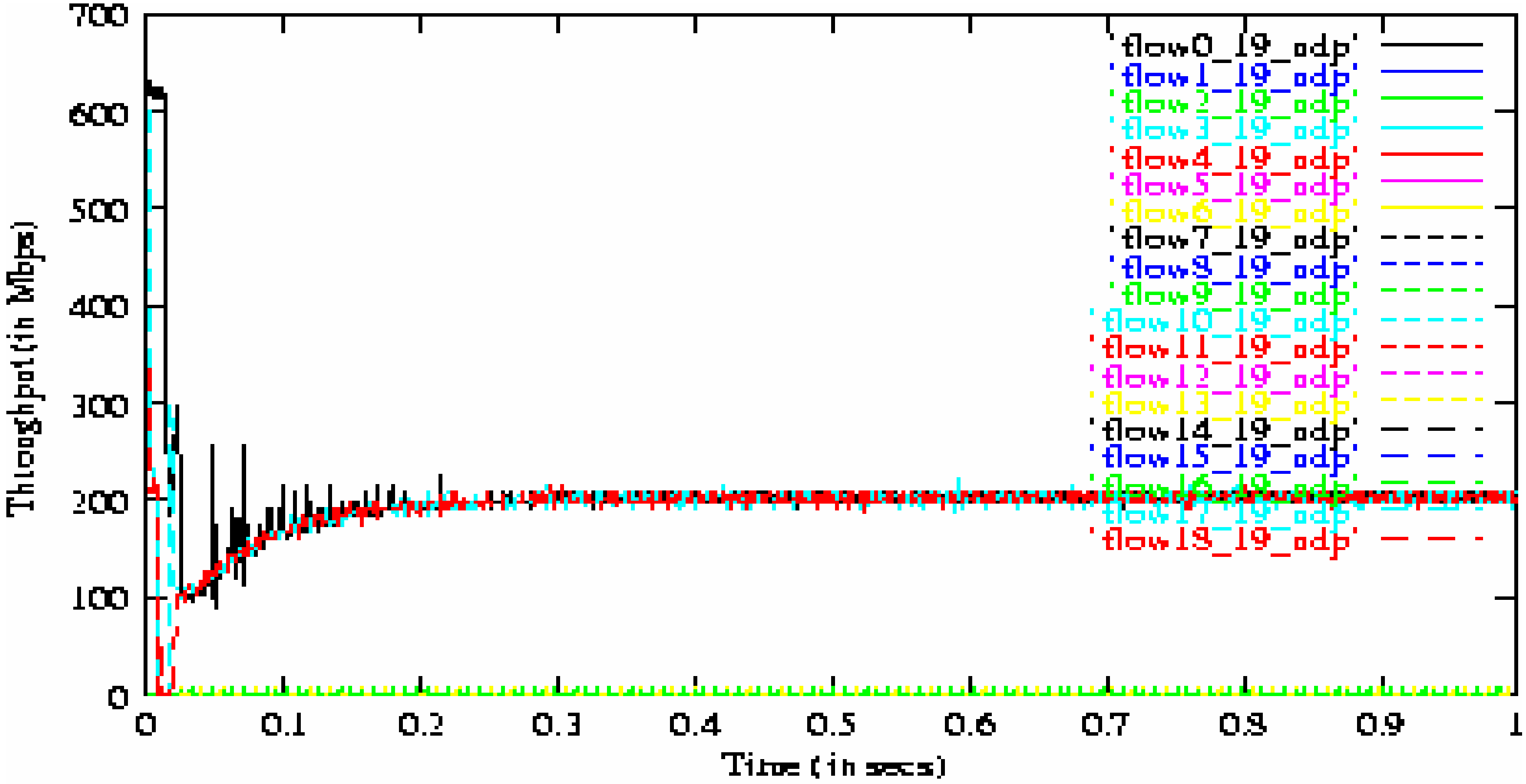


# Measurement of Active Stations/Weights

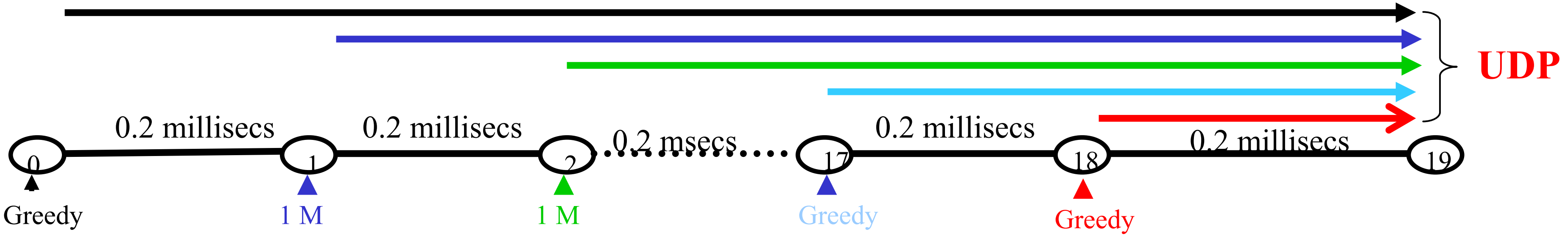
- We determined that it is desirable to measure active weights (possibly once every 10 milliseconds).
  - If a station sends a packet during the measurement interval, it is counted as an active station.
- What is the consequence of it as we scale up the system, with very small flows?
- We modeled a ring with 20 stations in the congestion domain
  - Station 0 is a greedy flow
    - To ensure the congestion domain spans all 20 stations
  - Station 1-16 are small flows, with a demand of only 1 Mbps
    - They impact the measurement of active stations, but use very little of their fair share
  - Station 17 and 18 are greedy flows
  - All stations send traffic to station 19.
- What is the overall performance of such a ring with this workload
  - Measuring every aging interval vs. every 10 milliseconds?

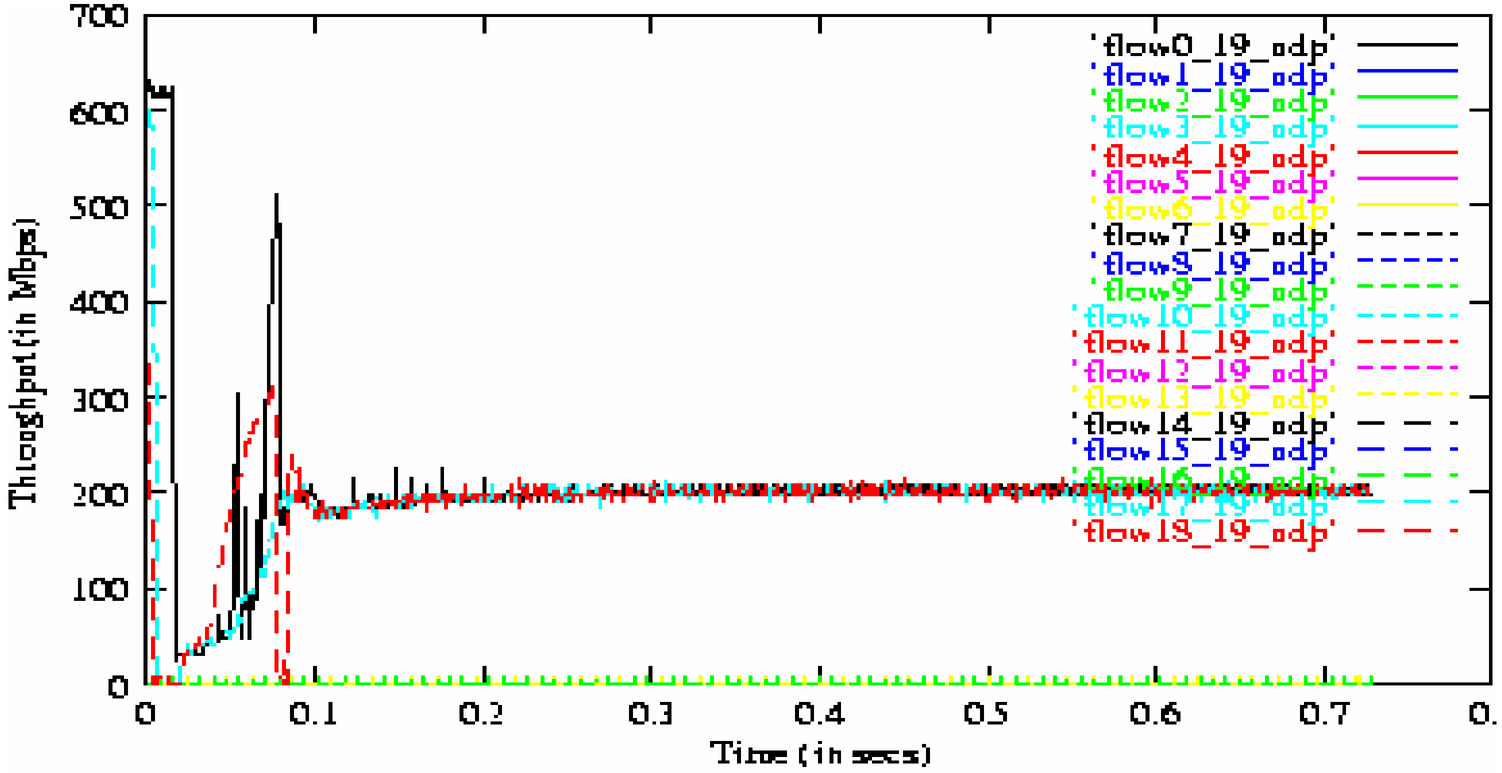
# Measurement of Active Stations/Weights every aging interval



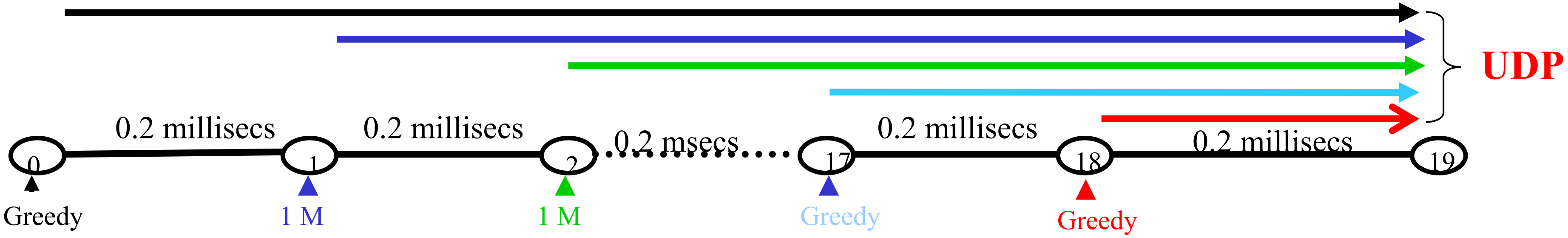


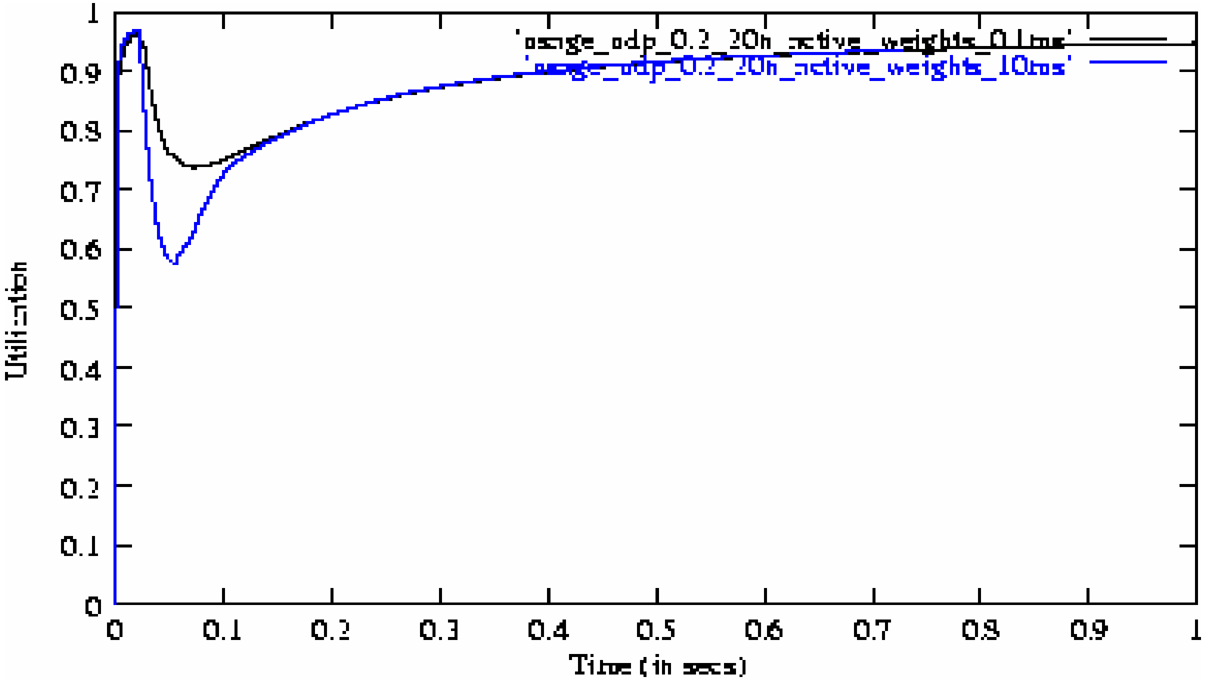
# Measurement of Active Stations/Weights every 10 milliseconds





# Measurement of Active Stations/Weights Link Utilizations





- ❑ Measurement of active stations every 10 msecs. impacts convergence and link utilization somewhat, but ultimately converges to desired value: Row 5 and 6!

# Summary

- ❑ Significant improvement in understanding of the conservative mode
  - ➢ Improvements have maintained the same basic framework for the conservative mode
- ❑ Achieved improvements in utilization, reduced possibility of starvation of head node and considerably improved oscillations
- ❑ Conservative mode performance is acceptable
  - ➢ Achieves high utilization
  - ➢ Achieves single choke fairness
  - ➢ Responsive to dynamics in workload
- ❑ Setting up Fairness Ad-Hoc was very useful
  - ➢ FAH has reached rough consensus on modifications