ns2 project

Rownok Ratul

February 2023

1 Introduction

To control bursty and unpredictable traffic in networking systems, congestion control schemes are used with the purpose of achieving a robust, stable, and efficient system. Active Queue Management (AQM) schemes are variants that utilize the concept of constant queue length at nodes. TCP congestion control schemes rely on packet loss due to buffer overflow that adapts the sending rate. Active Queue management systems are dynamic early congestion detectors that produce signals of congestion before the buffer is actually full. Along with early congestion notification, the low buffer occupancy has the advantage of:

- Small queueing delays
- Low queue length jitter
- Low packet loss and high link utilization

Here, a proportional integral based AQM (IAPI) is implemented in ns2, as a part of the learning process and also analyzed.

2 IAPI An Intelligent Adaptive PI AQM

IAPI is basically a feedback-based self-tuning queue management scheme theorized by Jinsheng Sun, Sammy Chan, and Moshe Zukerman [1]. the concept of proportional integral is based on the feedback integral,

$$p(t) = k_p e(t) + k_i \int_0^t e(\tau)\, d\tau \tag{1}$$

In discrete form,

$$p(k) = p(k-1) + k * p(e(k) - e(k-1)) + ki * e(k) \tag{2}$$

The k-values are PI parameters, self tuned by the following scheme.

1. The probability function $p(t)$ is calculated at a fixed sampling frequency (170 Hz) based on the k-parameters that reflect the current queue condition.

2. On each packet arrival, corresponding to a threshold e^*, the error rate $e(t)$ is calculated with a q_{ref} value. Overshooting the queue length is compensated with another parameter k_1 and k_{i0}.

3. Finally, on basis of the probability, it is decided whether to enqueue the packet or not.

The scheme can be understood by analysing the k_i-parameter equation,

$$k_i = ki_0 \cdot (1 + \frac{|e(k)| - e^*}{2e^*}) \tag{3}$$

As the instantaneous error rate e(k) deviates from the threshold error, the k_i increases (positive feedback), causing the probability the grow. This stablizes the long-run queue length error from q_ref value.

3 Implementation Environment

The scheme is implemented in ns2 simulator environment.

3.1 Parameter Setting

- Sampling frequency for probability function, $w = 170$
- Initial PI parameters, $k_p = 0.0001472$ and $k_{i0} = 0.0000004277$
- $Q_{\max} = 900$ with $q_{ref} = 200$, threshold error $e^* = 40$ for wired
- $Q_{\max} = 90$ with $q_{ref} = 20$ and threshold error $e^* = 4$ for wireless

3.2 Protocols

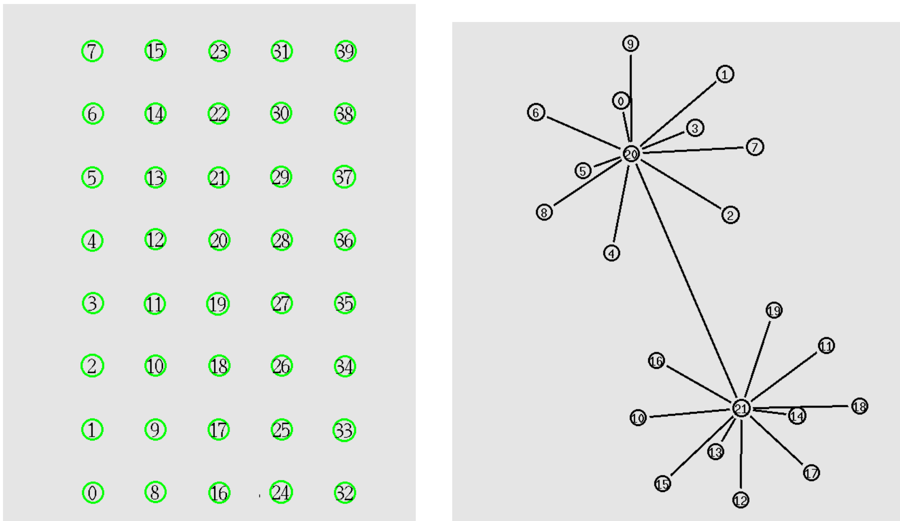
On Transport layer, TCPReno was used. For wireless transmission, two different MAC layer protocols were used: 802.11 and 802.15.4

3.3 Topology

For the wired system, a single bottleneck topology was created. For the wireless system, a grid-based mobile cluster was used.

4 Observation

Number of nodes, number of flows and packets sent per second were varied to observe the results. Subsequently, for wireless topology, the area, protocols and speed were varied accordingly too. Following subsections highlights the graphs found from different observations and characterizes the plot.



(a) Wireless Topology (mobile nodes)   (b) Wired single bottleneck

4.1 Experiment on Wired Topology

4.2 Experiment on Wireless Topology

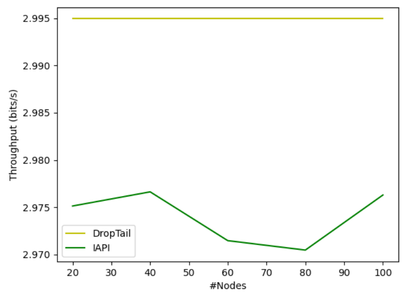
Here, accumulation of all tests on wireless grid topology is presented. Only of the parameters from coverage area, #nodes, #flows, #speed, #packets/s is changed at a time.

4.2.1 802.11 Protocol

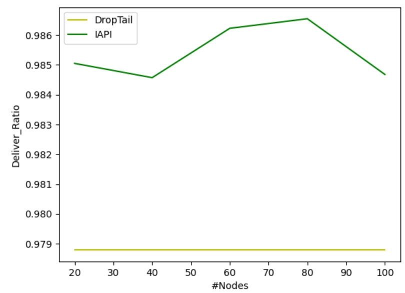
4.2.2 802.15.4 Protocol

5 Conclusion

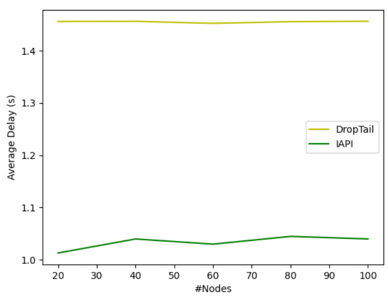
The above graph signifies with quantitative measure of the AQM-IAPI scheme. The most important of the findings is depicted in Figure 4.1. As we can see, the queue length of the bottleneck link of topology demonstrates the variance of queue size becomes stable with IAPI. The value stablizes when it is close to provided $q_ref = 200$. This poses an indication towards lower jitter through lower queue length usage and high link utilization. Moreover, the scheme is tested under variant of environment and network parameters such as nodes, flows and speed(for wireless) and so on. Though the result is not always promising, which may indicate the lack of test cases. The scheme is tested only on 5 different values on each parameter, making the graphs bit zig-zagged. Sometimes, IAPI performs less likely to opt under controlled parameters. But, this cannot be concluded solely on a simulator. But overall, the opt to stablize queue length at a preferred value is quite accomplished under varying network connections. The



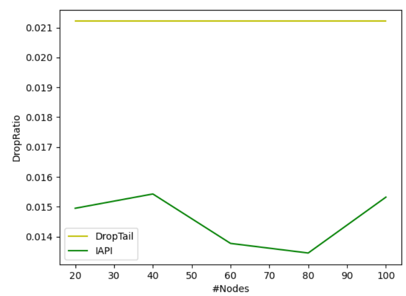
(c) Throughput



(d) Delivery Ratio



(e) Average Delay



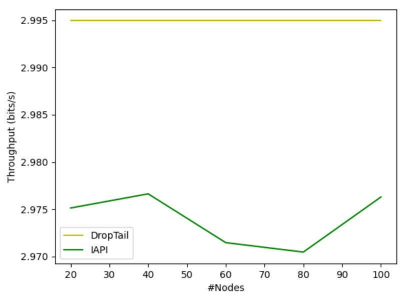
(f) Drop Ratio

Figure 1: Network Metrics vs Number of Nodes

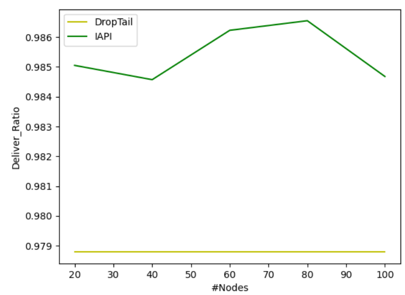
tests were done both for wired and wireless 802.11 and 802.15.4 MAC protocols, indicating robustness and independence to the underlying layer type. Also, PI integral probability functions are light weight, and k-value parameters are easy to compute. In essence, the experiment on the above term project evaluates the practical applicability of IAPI in the testing environment and conclusion can be drawn by impying the applicability only when low-jittering and buffer length is expected highly. On the other occasions, RED based systems may outperform IAPI, but being computation heavy, causing a higher power dissipation.

References

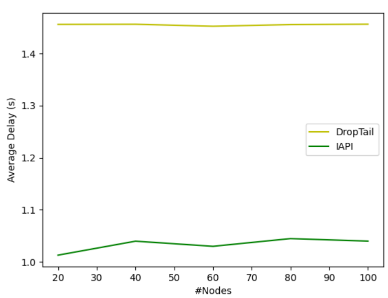
[1] Jinsheng Sun, Sammy Chan, and Moshe Zukerman. Iapi: An intelligent adaptive pi active queue management scheme. *IEEE Communications Letters*, 11(8):675–677, 2007.



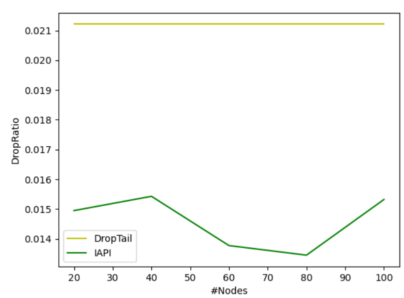
(a) Throughput



(b) Delivery Ratio

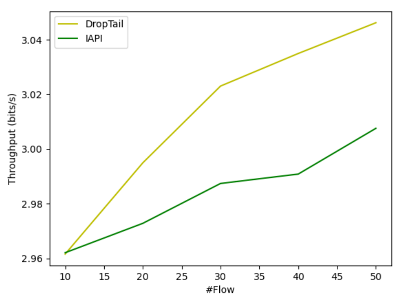


(c) Average Delay

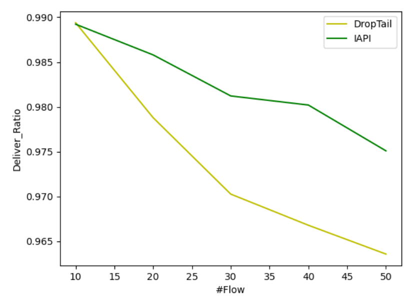


(d) Drop Ratio

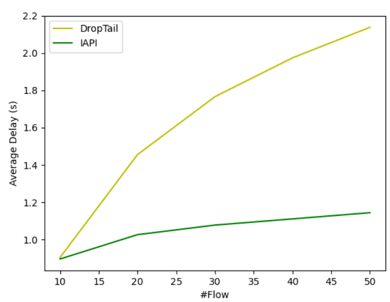
Figure 2: Network Metrics vs Number of Nodes



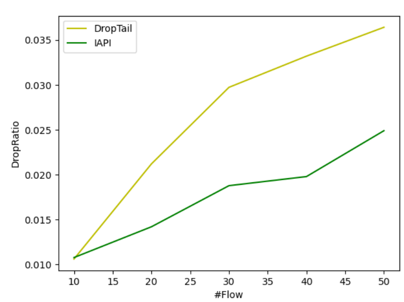
(a) Throughput



(b) Delivery Ratio

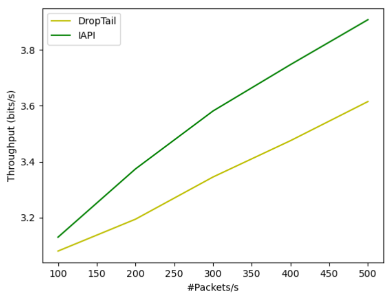


(c) Average Delay

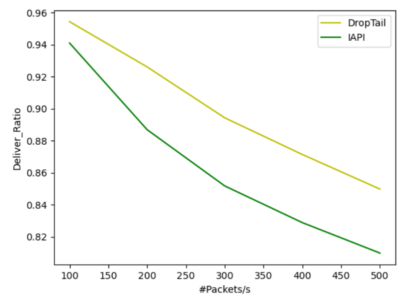


(d) Drop Ratio

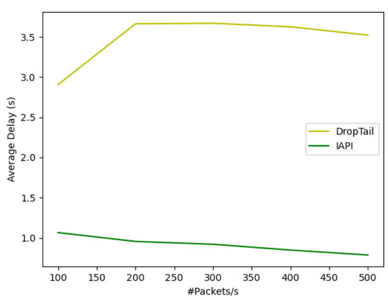
Figure 3: Network Metrics vs Number of Flows



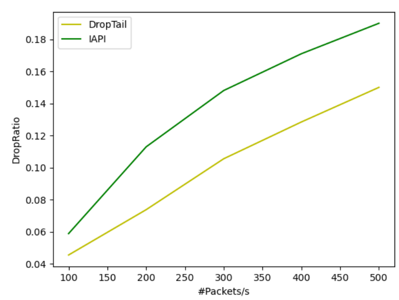
(a) Throughput



(b) Delivery Ratio



(c) Average Delay



(d) Drop Ratio

Figure 4: Network Metrics vs Max Sequence Number (Packets/s)

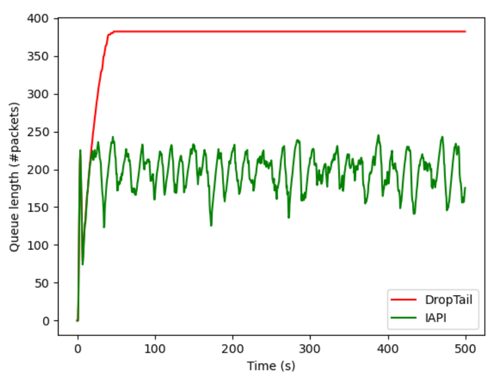
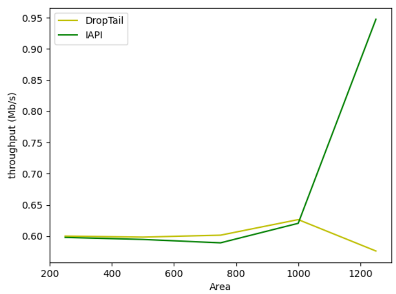
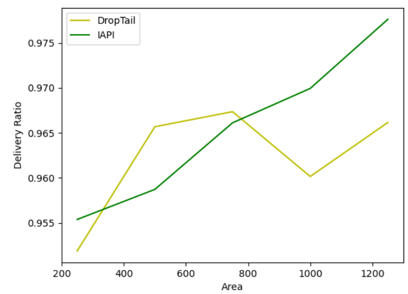


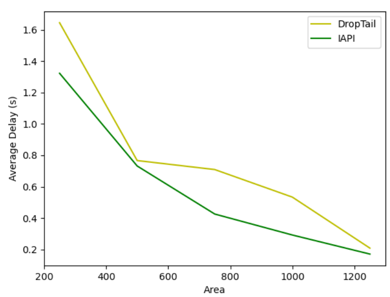
Figure 5: Queue Length (Common Bottleneck Link) vs Time



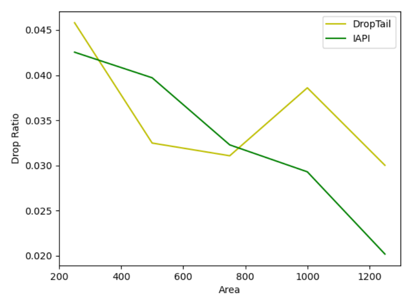
(a) Throughput



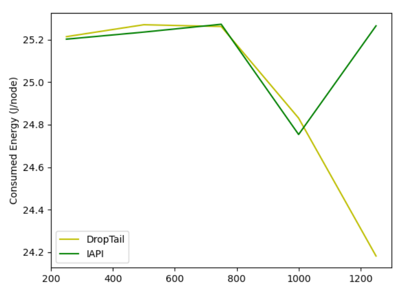
(b) Delivery Ratio



(c) Average Delay

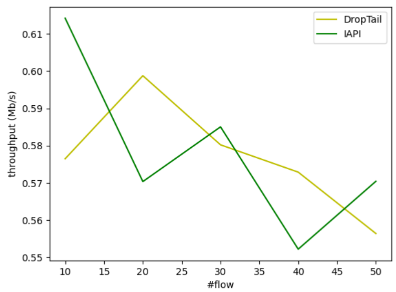


(d) Drop Ratio

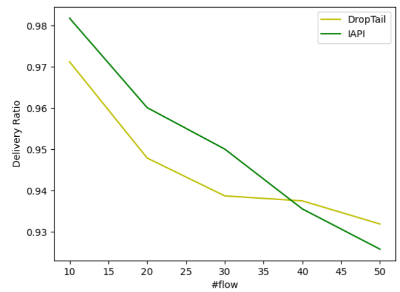


(e) Drop Ratio

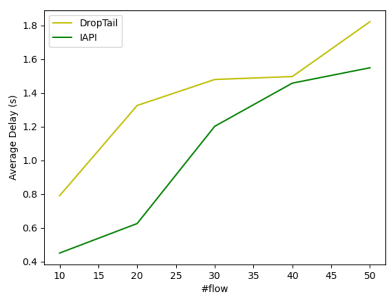
Figure 6: Network Metrics vs Coverage Area



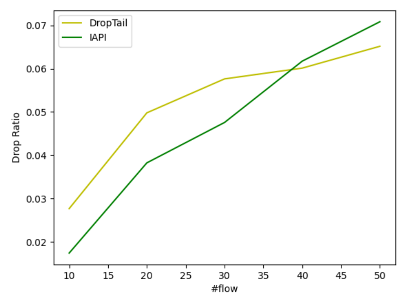
(a) Throughput



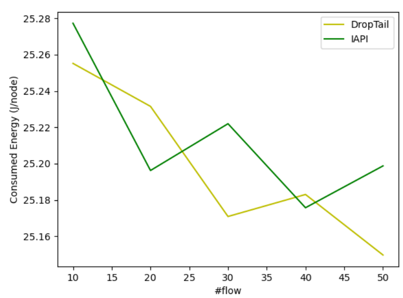
(b) Delivery Ratio



(c) Average Delay

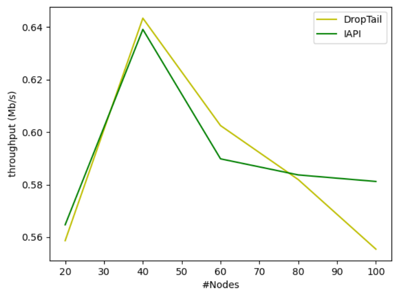


(d) Drop Ratio

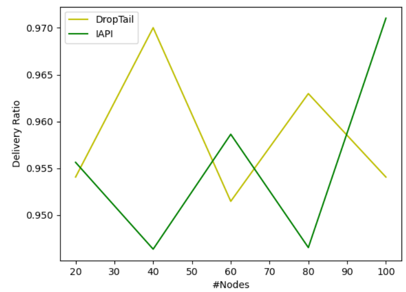


(e) Drop Ratio

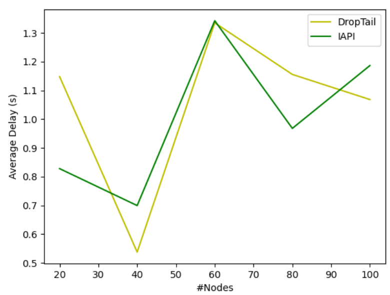
Figure 7: Network Metrics vs Number of Flows



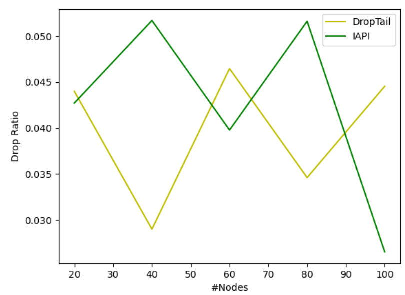
(a) Throughput



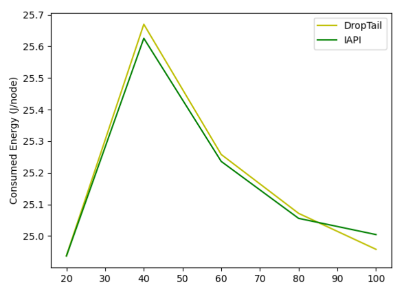
(b) Delivery Ratio



(c) Average Delay

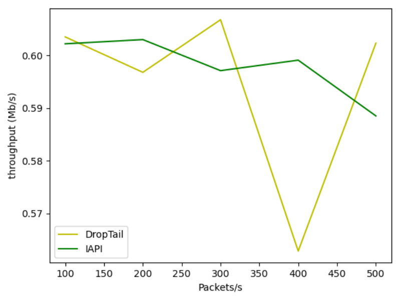


(d) Drop Ratio

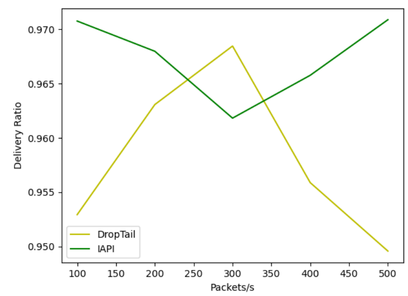


(e) Drop Ratio

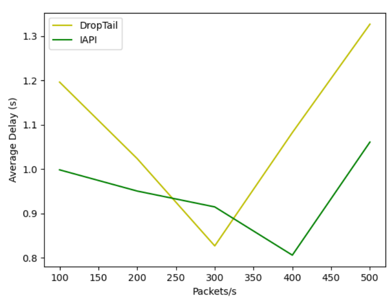
Figure 8: Network Metrics vs Number of Nodes



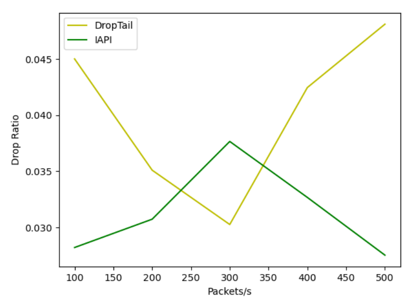
(a) Throughput



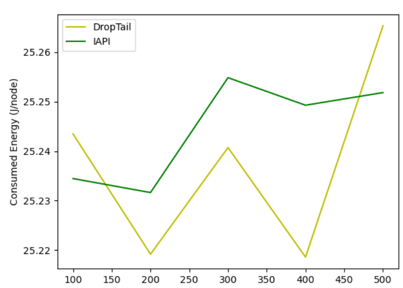
(b) Delivery Ratio



(c) Average Delay

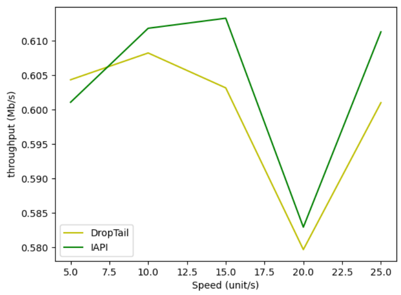


(d) Drop Ratio

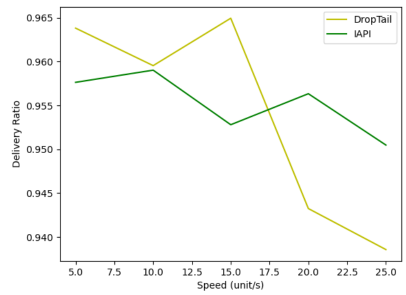


(e) Drop Ratio

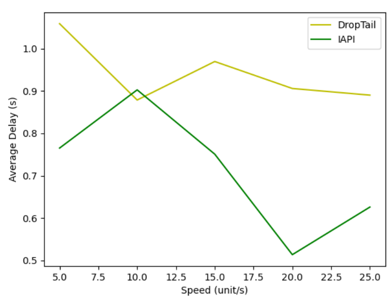
Figure 9: Network Metrics vs Packets/s (Max Sequence Number)



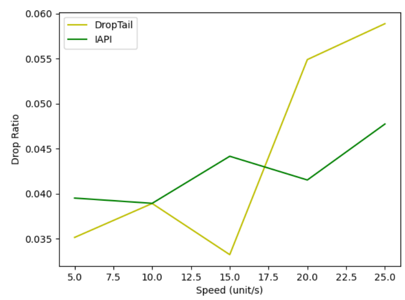
(a) Throughput



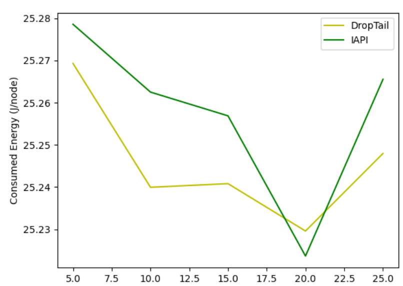
(b) Delivery Ratio



(c) Average Delay

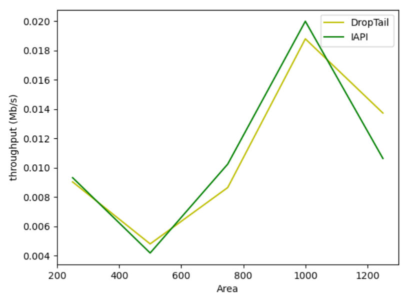


(d) Drop Ratio

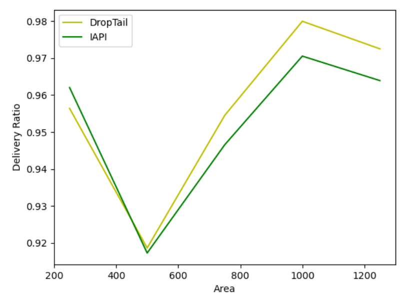


(e) Drop Ratio

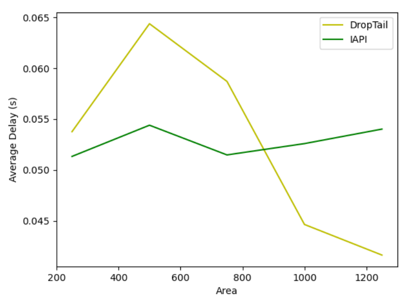
Figure 10: Network Metrics vs Speed



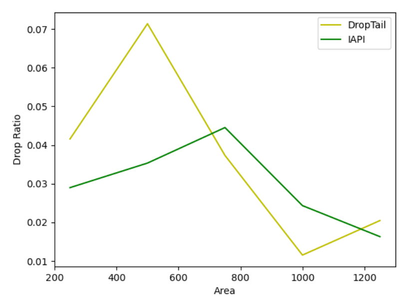
(a) Throughput



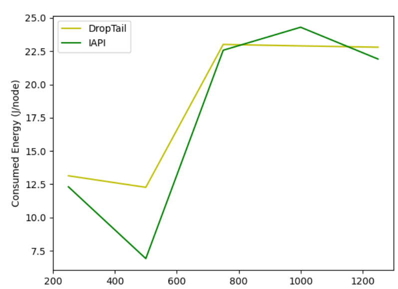
(b) Delivery Ratio



(c) Average Delay

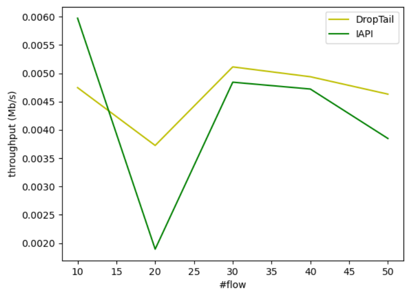


(d) Drop Ratio

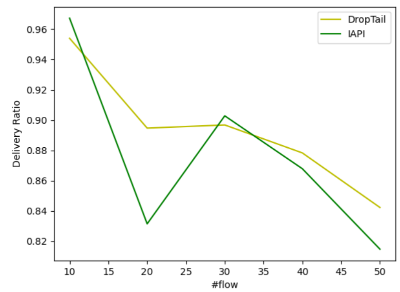


(e) Drop Ratio

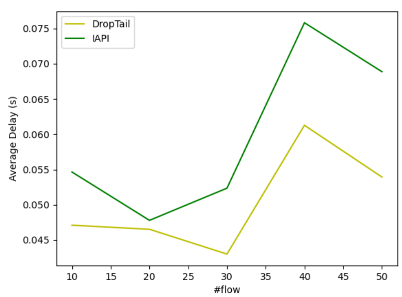
Figure 11: Network Metrics vs Area



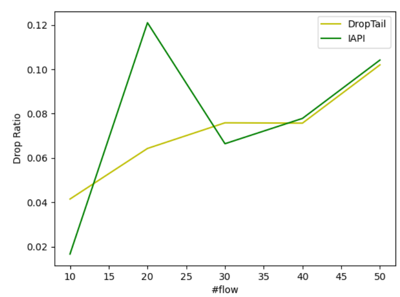
(a) Throughput



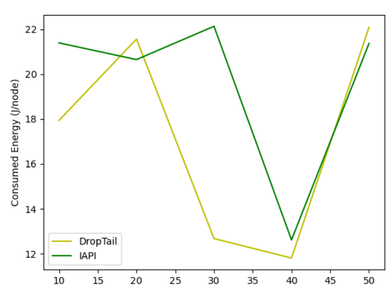
(b) Delivery Ratio



(c) Average Delay

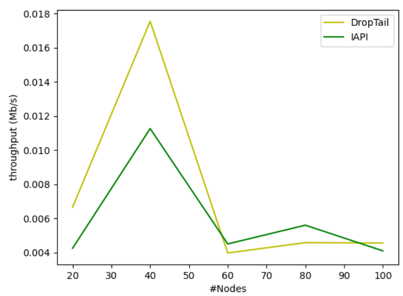


(d) Drop Ratio

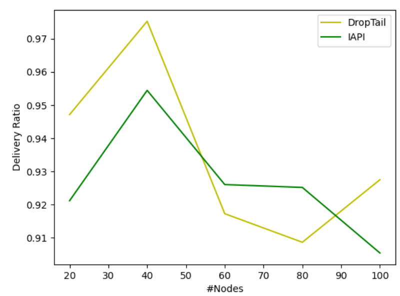


(e) Drop Ratio

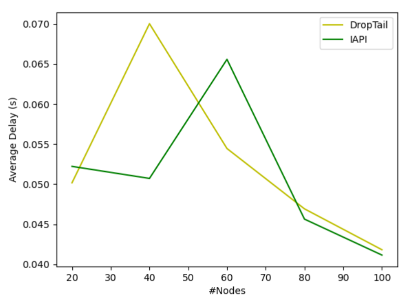
Figure 12: Network Metrics vs Number of Flows



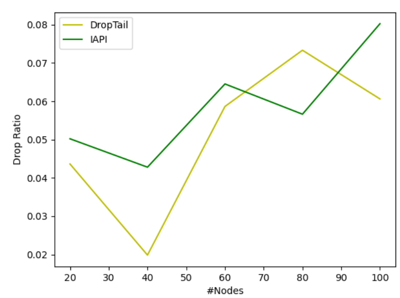
(a) Throughput



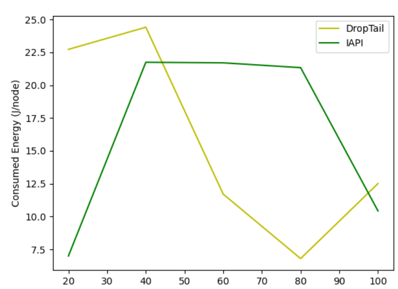
(b) Delivery Ratio



(c) Average Delay

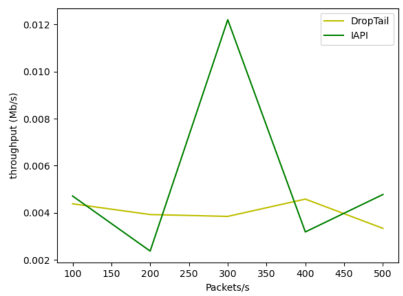


(d) Drop Ratio

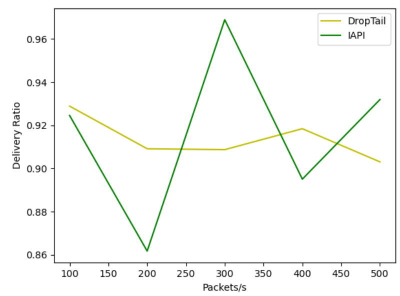


(e) Drop Ratio

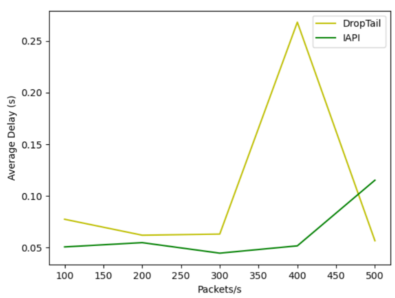
Figure 13: Network Metrics vs Number of Nodes



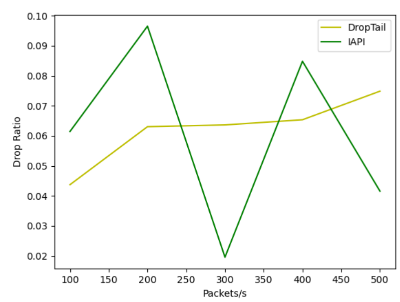
(a) Throughput



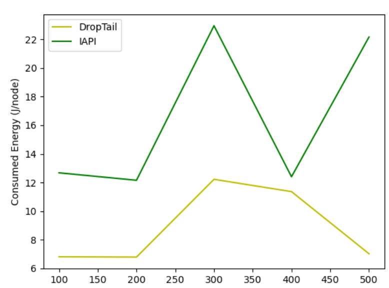
(b) Delivery Ratio



(c) Average Delay

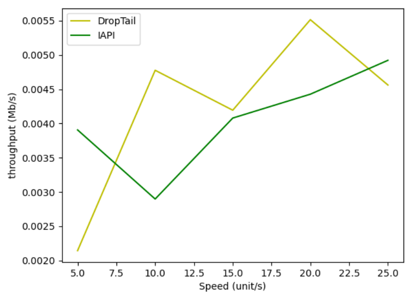


(d) Drop Ratio

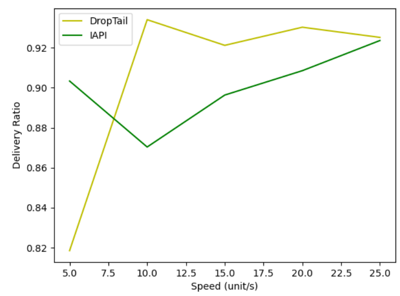


(e) Drop Ratio

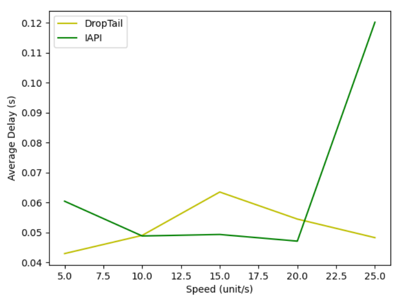
Figure 14: Network Metrics vs Packets/s (Max Sequence Number)



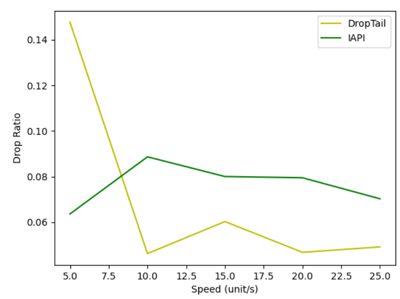
(a) Throughput



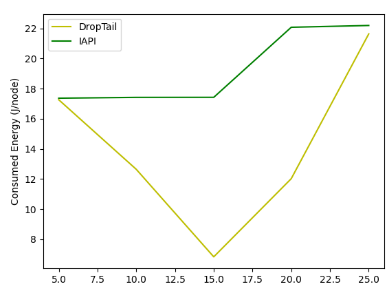
(b) Delivery Ratio



(c) Average Delay



(d) Drop Ratio



(e) Drop Ratio

Figure 15: Network Metrics vs Packets/s (Max Sequence Number)