Dynamic Time Windows:
New Developments in Quality of Service Provision
and
Experimental Results

Ted Faber
Lawrence Landweber

University of Wisconsin at Madison

faber@cs.wisc.edu and landweber@cs.wisc.edu

--

--

Outline

- Dynamic Time Windows (DTW) Goals and Philosophy
- DTW Overview
- Quality of Service and DTW
- Experimental Results

Dynamic Time Windows

- Dynamic Time Windows (DTW) is a congestion control system for high–speed wide area ATM cell computer networks

- The network
  * high speed/wide area : high bandwidth-delay product, *e.g.* XUNET
  * ATM cells : small packets and virtual circuits

Philosophy

- The integrated system
  * queueing disciplines
  * switch resource management
  * feedback-based traffic control
- Burstiness control
  * enforce negotiated rates at sources
  * *time window* controls granularity of average rate enforcement
  * allows sources to maintain average throughput into the network

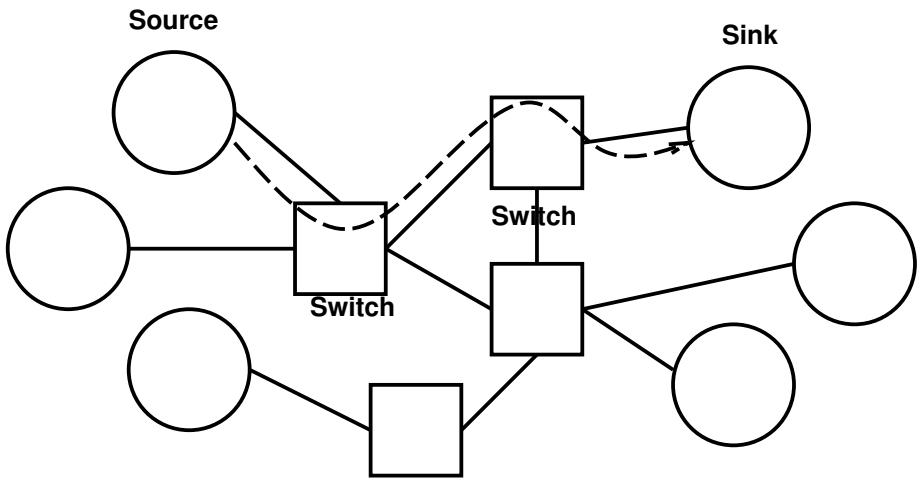
--

--

Network Elements

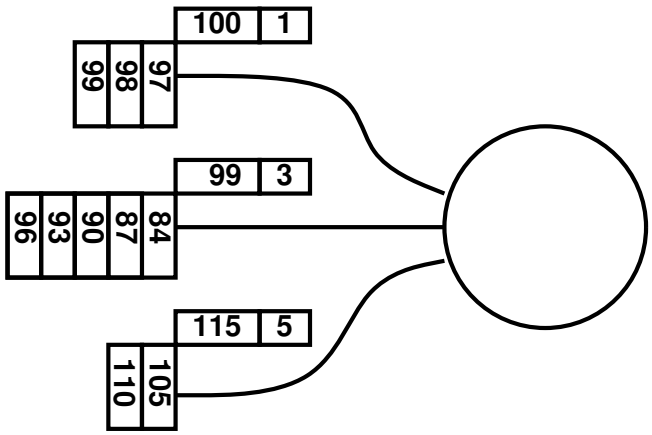
- Sources
  * virtual circuit endpoints
  * traffic sources and shapers

- Switches
  * cell switch
  * traffic monitor
  * cell injector



Mapping the System to the Model Switches

- Switches use Weighted Fair Queueing



  * cells are queued per–virtual circuit
  * each arriving cell is stamped with the virtual clock for the circuit it arrives on
  * the cell with the lowest stamp is served

Mapping the System to the Model
Switches

- Switches maintain a Maximum Time Window (MTW) for sources using it
  * when congested, a switch reduces its MTW multiplicatively
  * when uncongested, a switch periodically increases its MTW almost linearly
  * changes to the MTW are propagated to the sources via feedback cells

Mapping the System to the Model

Sources

- Sources enforce the Time Window Criterion for their traffic
  * peak rate ($\hat{\lambda}$)
  * average rate ($\bar{\lambda}$)
  * time window (I)

  Over any time of length I at most $\bar{\lambda}I$ bits may be sent

- A source adjusts its time windows based on feedback from switches its circuit uses

  * sources record each switch's current MTW

  * the minimum of these MTWs is used to compute its time window

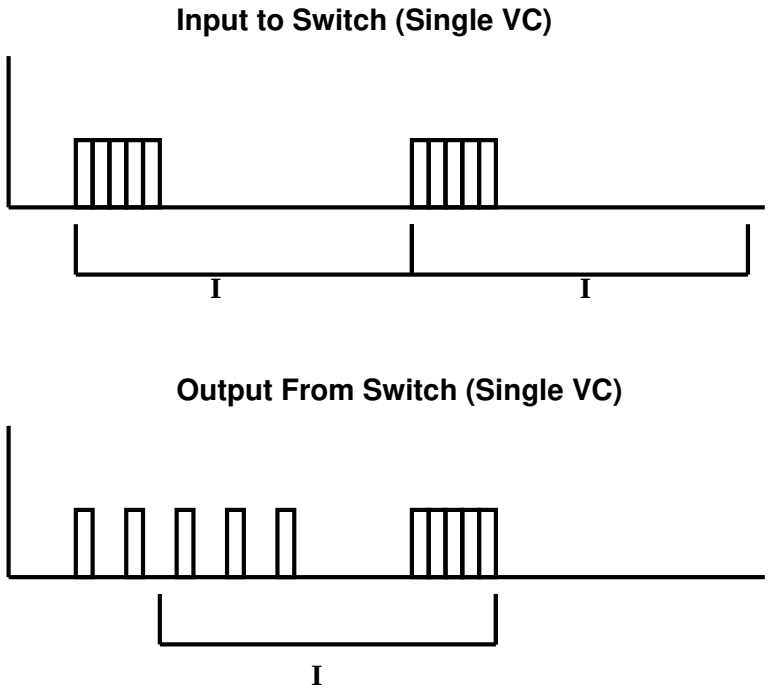
--

--

Stability (one switch)

- Under DTW a switch will be have no cells in any queue one MTW after a cell arrives to an empty switch
  - * requires the $\sum\bar{\lambda}$ to be less than the service rate of the switch (μ)
  - * requires the switch to be work-conserving
  - * requires that the sources' time windows to be integral divisors of the MTW

Stability (many switches)



- after passing through a switch, a source's average rate is $k\bar{\lambda}$ for $k \geq 1$.
- further, $1 \leq k \leq 2$ after passing through any number of switches
- requires $\sum k\bar{\lambda} \leq \mu$ at each switch
- requires all switches MTW to share a least common multiple

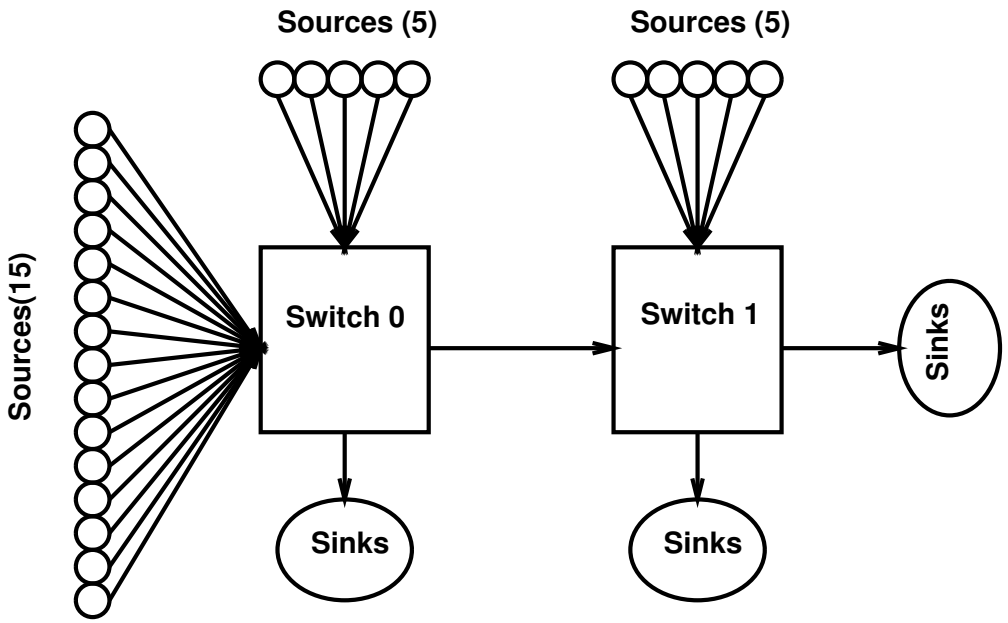
Providing Diverse Services
Allocating Switch Resources

- Switches allocate service rate and buffer space to circuits
- Allocations are opportunistic
- Service rate (μ) allocations are WFQ weights ($1/s$)
- Buffer (B) allocations are similar weights (b)
- Source i is guaranteed $\sum_j \frac{b_i}{b_j} B$ buffers and $\sum_j \frac{s_i}{s_j} \mu$ service rate
- $\sum_i b_i \leq 1$ and $\sum_i s_i \leq 1$

Providing Diverse Services
Feedback Modifications

- Sources are assigned to groups
- Feedback is generated to each group separately
- A group's feedback may affect other groups directly
- Groups may have different feedback triggers
  * exceeding group buffer allocation
  * exceeding global buffer usage limit

Simulations



- Source Parameters
  * Average Rate: 50 Mb/sec
  * Peak Rate: 100 Mb/sec
  * Range of Time Window ratios
- Switch Service rate: 1 Gb/sec
- Five sources are real time,
  others best effort

--

--

Results (Average Delay)

Service Share (s)	Delay (msec)
0.05	35.1
0.05	34.9

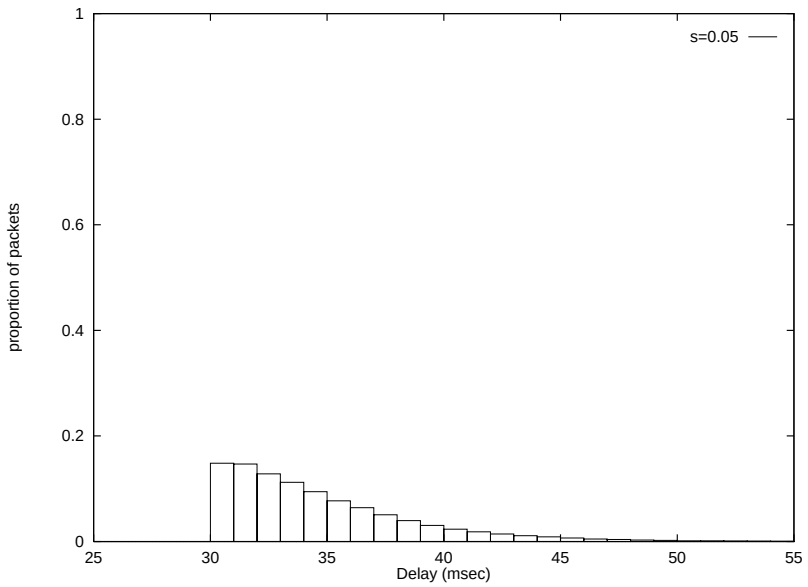
Service Share (s)	Delay (msec)
0.04	36.0
0.075	33.7

Service Share (s)	Delay (msec)
0.03	37.8
0.10	32.3

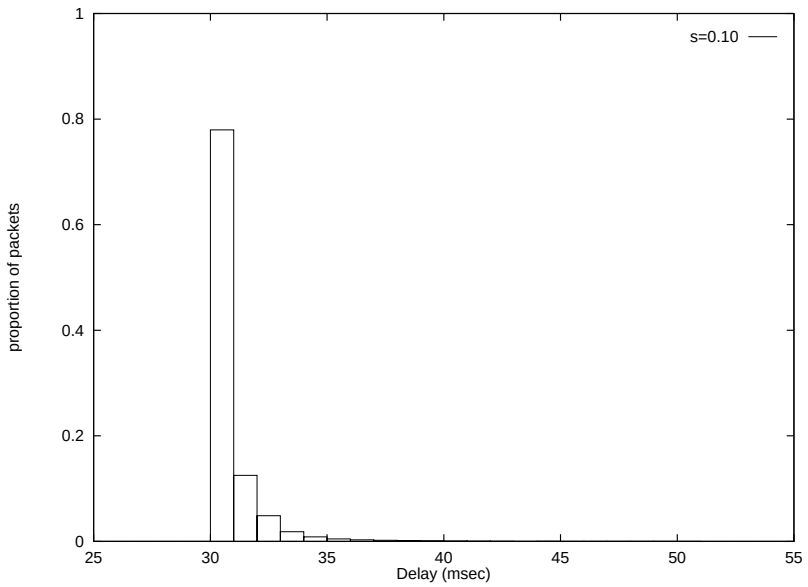
--

--

Results (Packet Distribution)

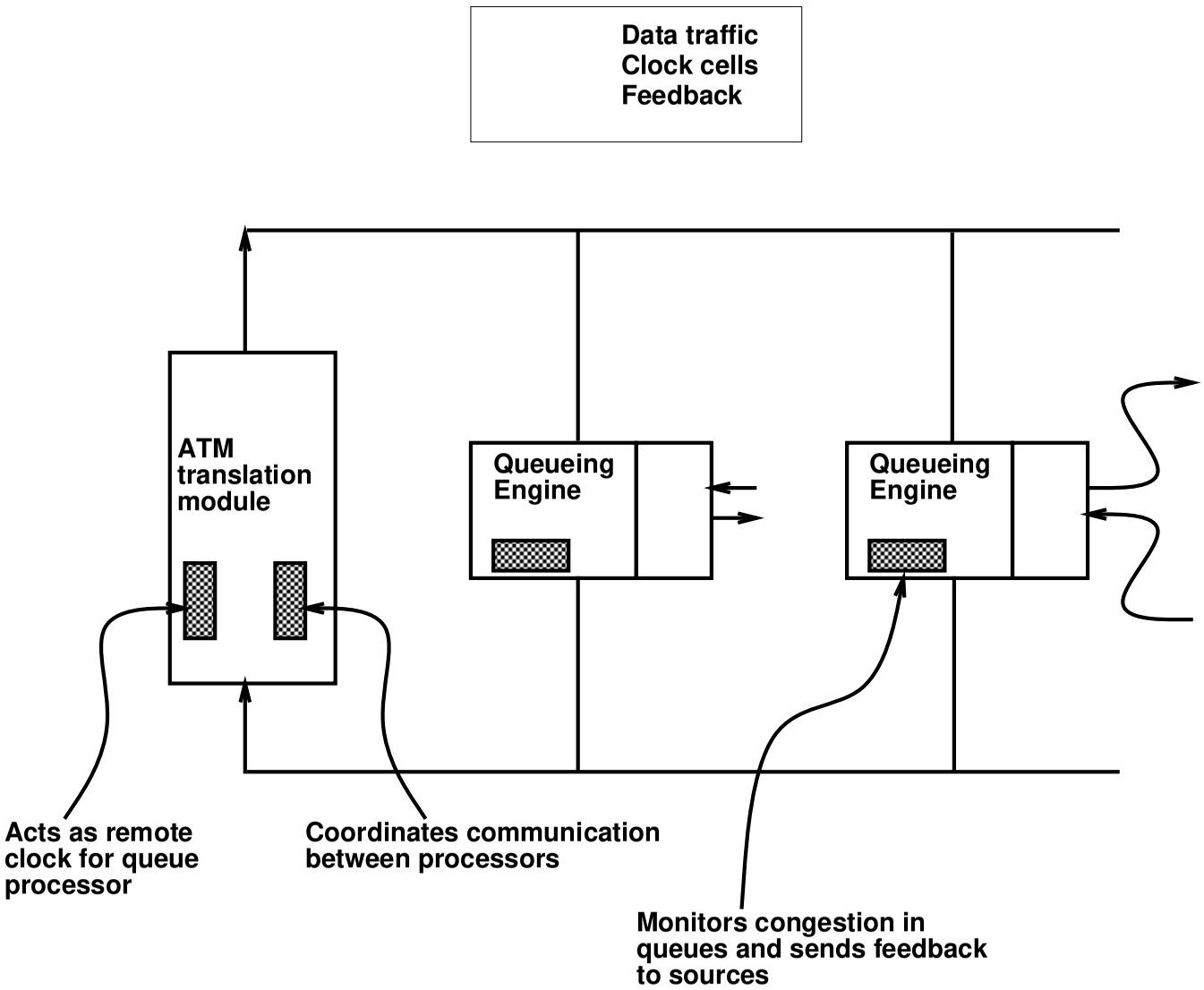


$s = 0.05$

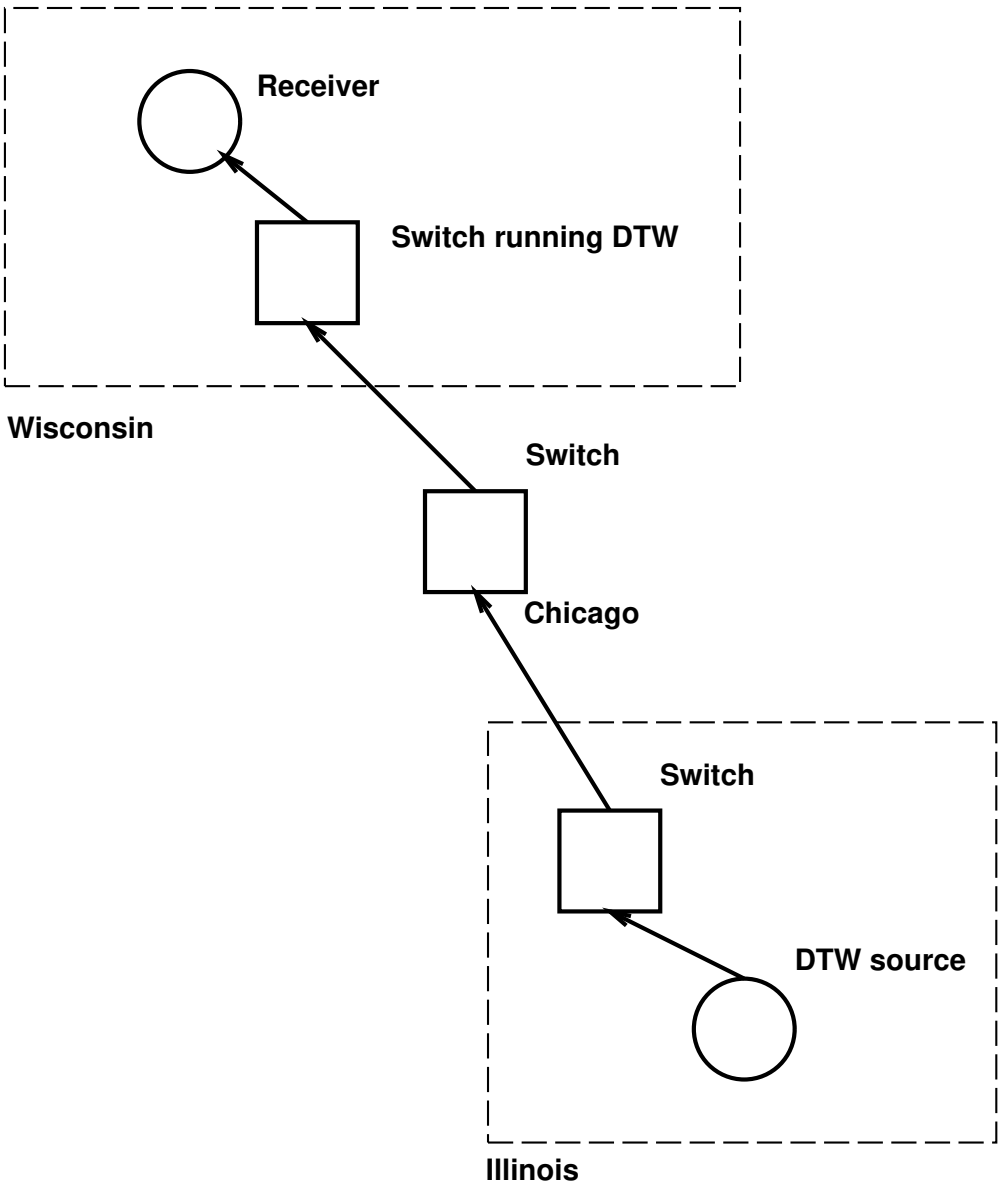


$s = 0.10$

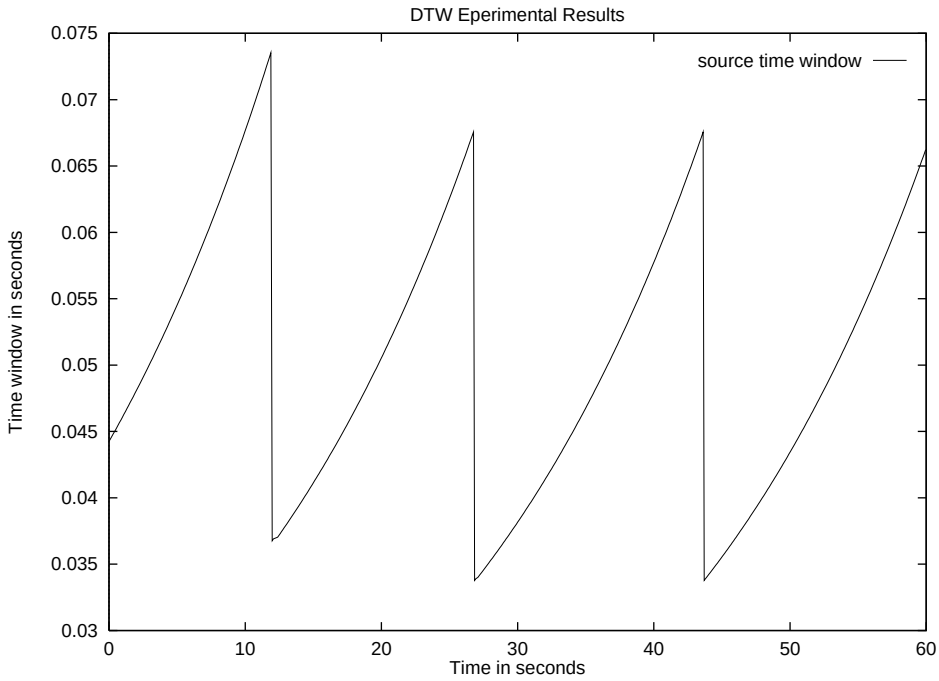
DTW Implementation on the XUNET Switch

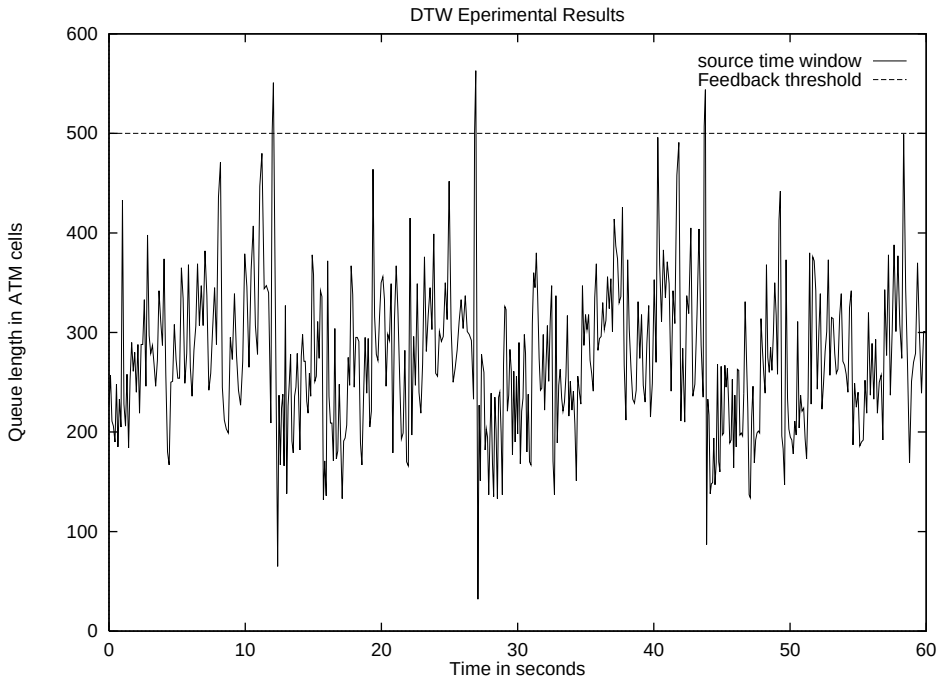


Experimental Configuration



Implementation





--

--

Conclusions

- Generality and flexibility of DTW proven by quality of service modifications

- DTW is implementable on the XUNET