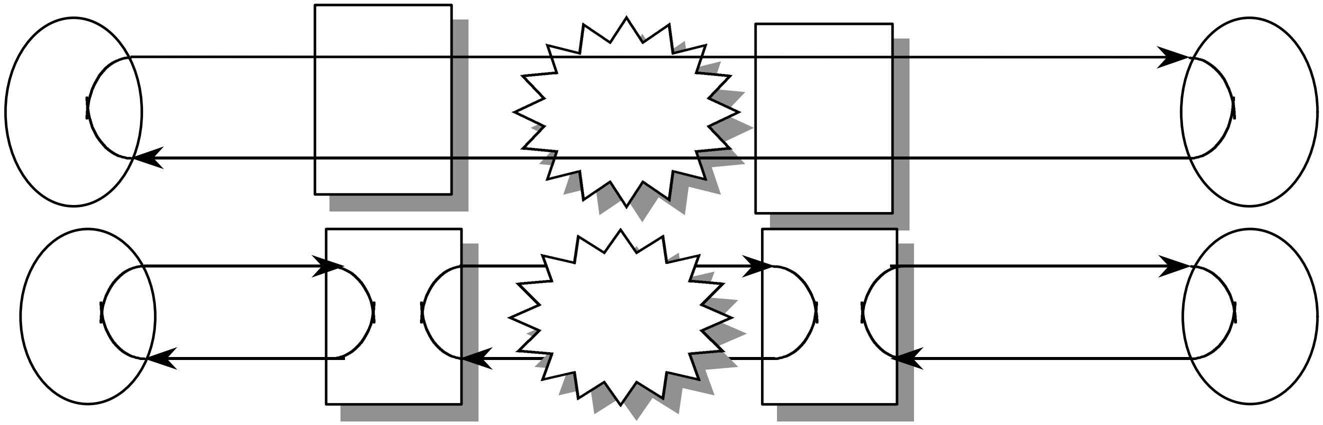
97-1086R1: Per-VC Rate Allocation Techniques For ABR Feedback in VS/VD Networks

Rohit Goyal, Xiangrong Cai, Raj Jain, Sonia Fahmy, Bobby Vandalore

Raj Jain is now at
Washington University in Saint Louis
Jain@cse.wustl.edu
http://www.cse.wustl.edu/~jain/

Virtual Source / Virtual Destination (VS / VD)

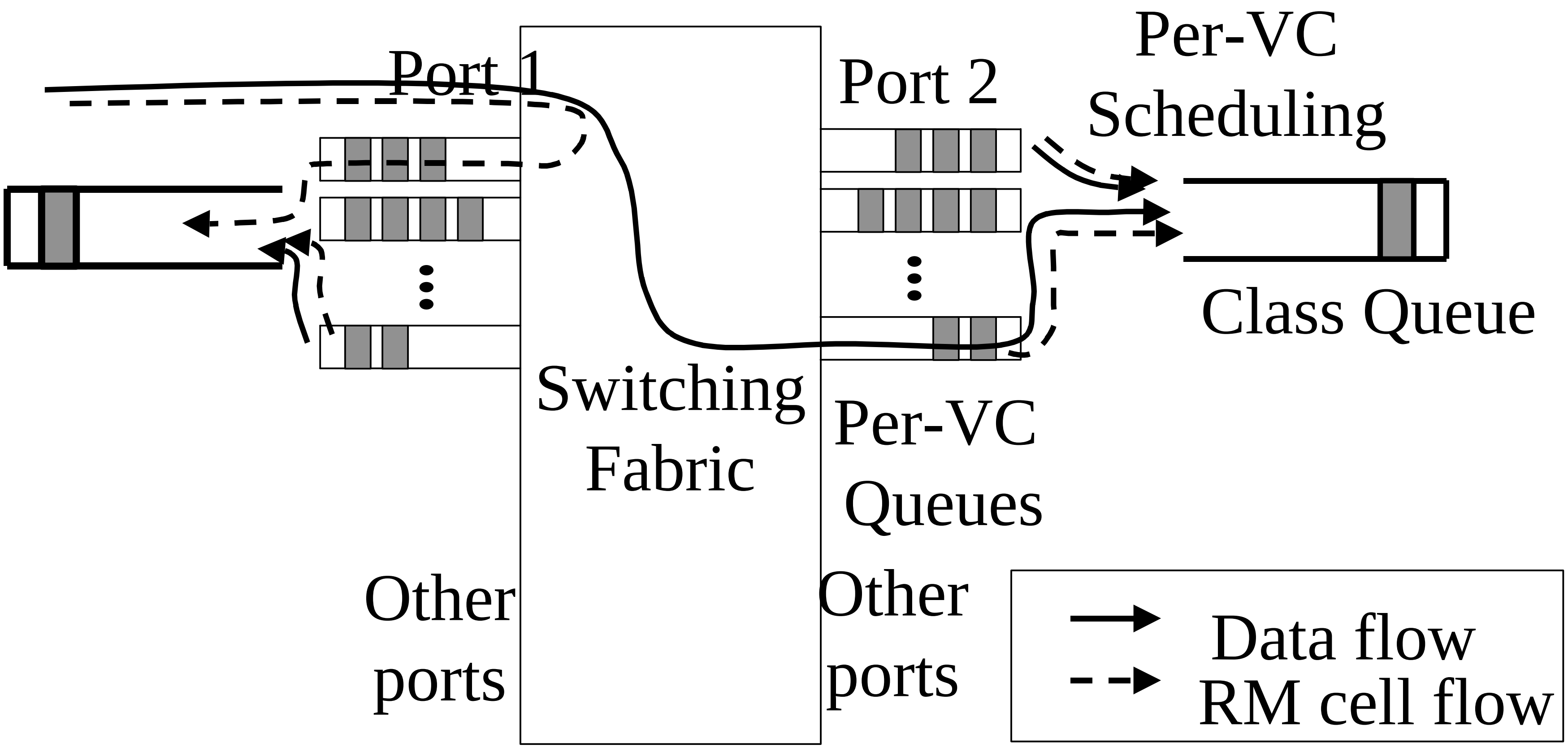


- Segments the end-to-end ABR control loop.
- Coupling between loops is implementation specific.
- VS/VD can help in buffer management across the network.
- ABR switches separated by non-ATM network could also implement VS/VD.

Goals

- Describe a VS/VD switch architecture.
- Discuss issues in designing rate allocation schemes for VS/VD switches.
- Present a per-VC rate allocation scheme for VS/VD.
- Discuss how VS/VD can help in buffer management across the network.

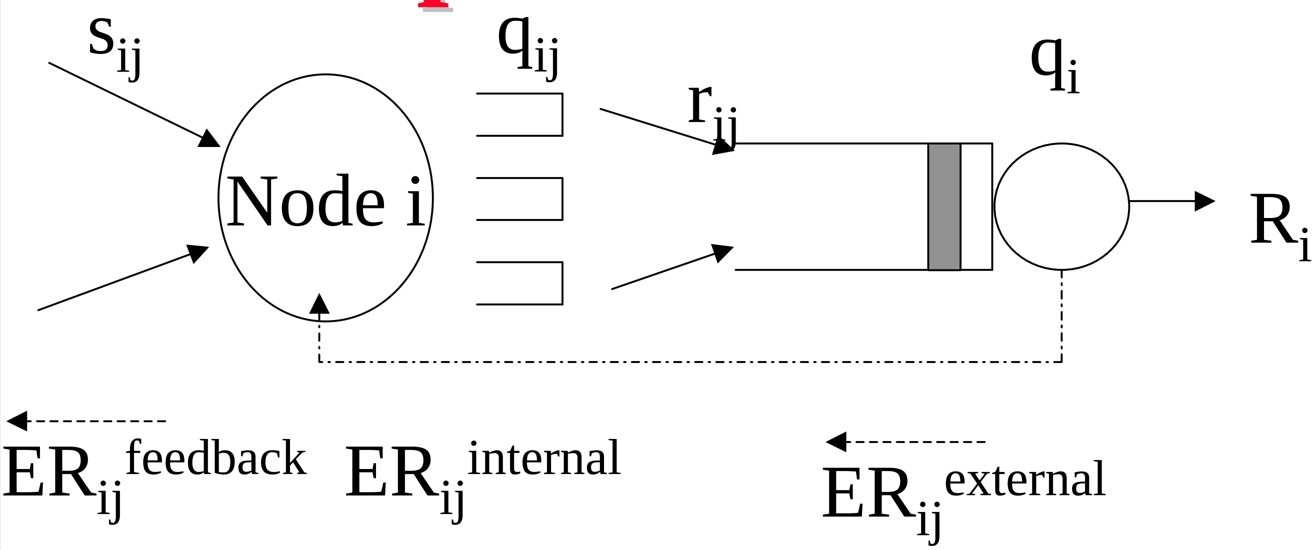
VS/VD Switch Architecture



VS/VD Switch Architecture

- Each switch port :
  - Class queue for each service category. (optional)
  - Per-VC queues drain into class queue or link
- When a cell is received :
  - Data cell : forwarded to destination port (VS).
  - FRM cell : turned around as BRM (VD).
  - BRM cell : ER is noted (VS).
- VS sends data + FRM cells at ACR to class queue.
- A scheduler services the per-VC queues.

A Simple VS/VD Model

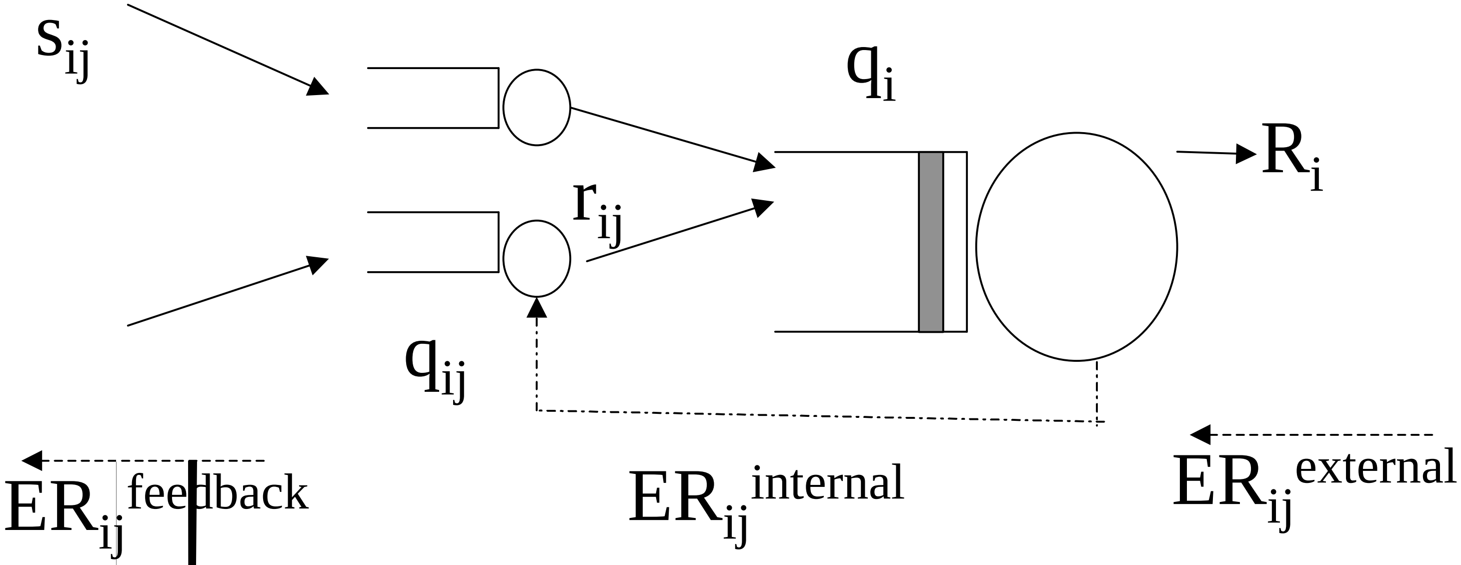


- Internal Service Rate = f(External/Downstream Feedback, Local congestion)
- Local Congestion = f(Q_i); $Q_i = q_i + \Sigma\, q_{ij}$
- Upstream feedback = Internal service rate
- Example: Downstream = 100 Mbps, Internal =90 Mbps = Upstream Feedback

Simple VS/VD Model

- Desired input rate to class queue is also fed back to the upstream switch.
- **Problem**:
  - Transient per-VC queues cannot drain.
    Input rate s_{ij} = Output rate r_{ij}
  - Queues that build up during open loop phase do not drain.

Correct VS/VD Model

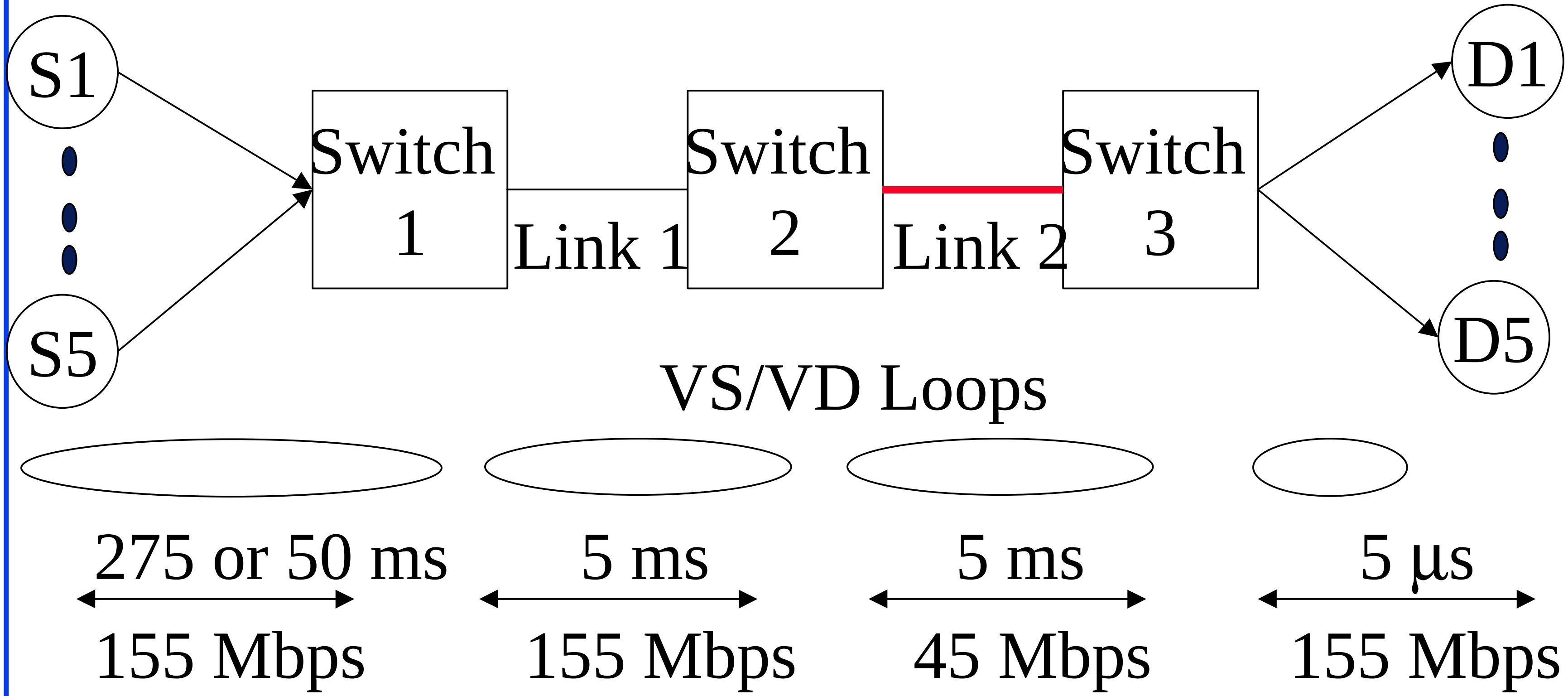


- Internal Service Rate = f(External/Downstream Feedback, Switch algorithm using q_i)
- ACR_{ij} = f(Internal service rate, end system rules)
- Upstream feedback =f(q_{ij})ACR_{ij}
- Example: Downstream = 100, Service =90, ACR=80, Upstream feedback=70 Mbps

Per-VC ERICA+

- BRM received :
  - $ER_{ij}^{external}$:= ER in RM cell
- FRM received :
  - ER in RM := $ER_{ij}^{feedback}$
- At the end of each averaging interval :
  - $ER_{ij}^{internal}$
    $:= \text{Min}\{ \text{Max} (r_{ij}/\text{Overload}, g(q_i)R_i/N), ER_{ij}^{external}\}$
  - Output rate
    $ACR_{ij} = r_{ij} := \text{fn}\{ER_{ij}^{internal}$, end system rules}
  - $ER_{ij}^{feedback} := g(q_{ij})r_{ij}$

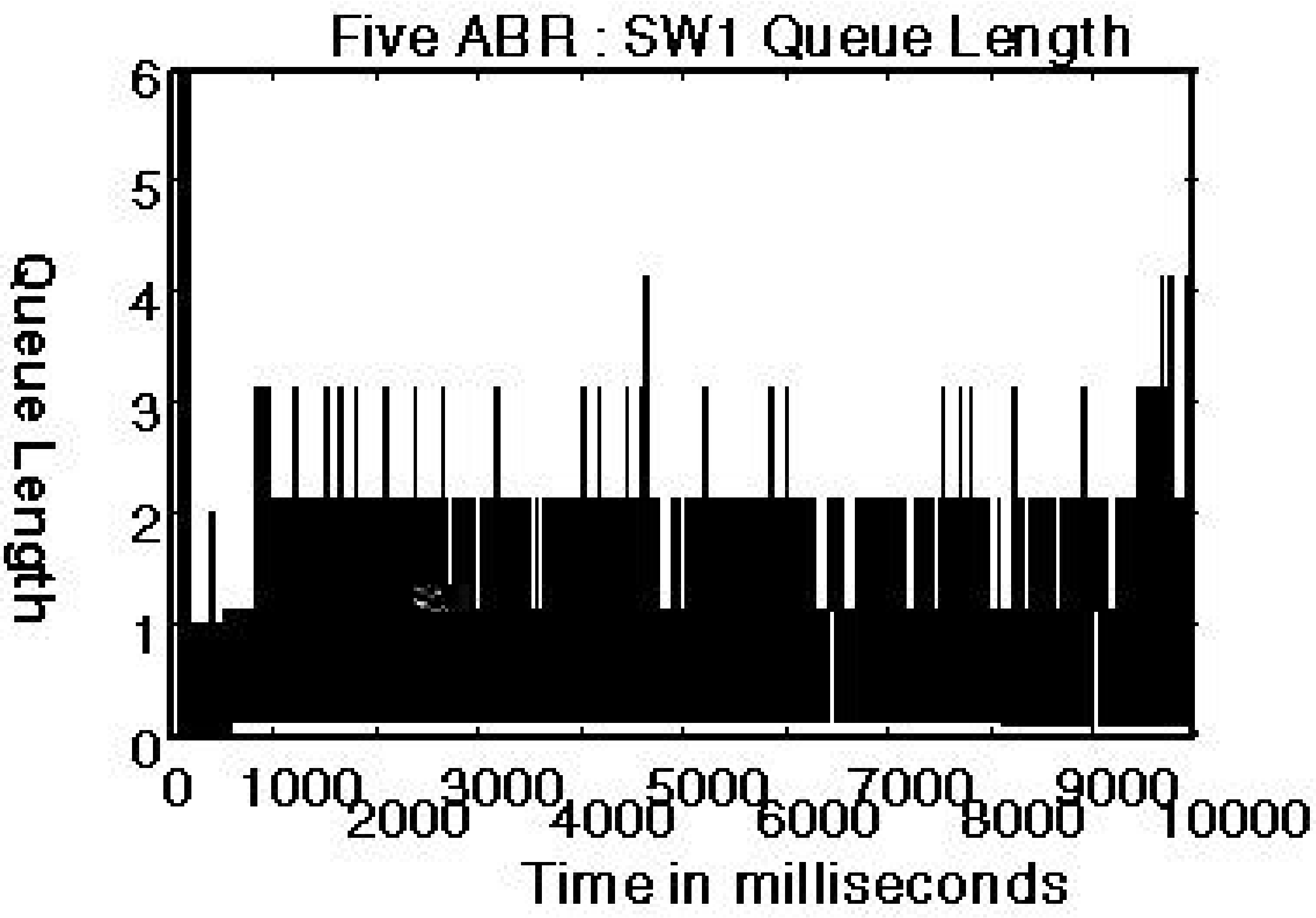
Simulation Model



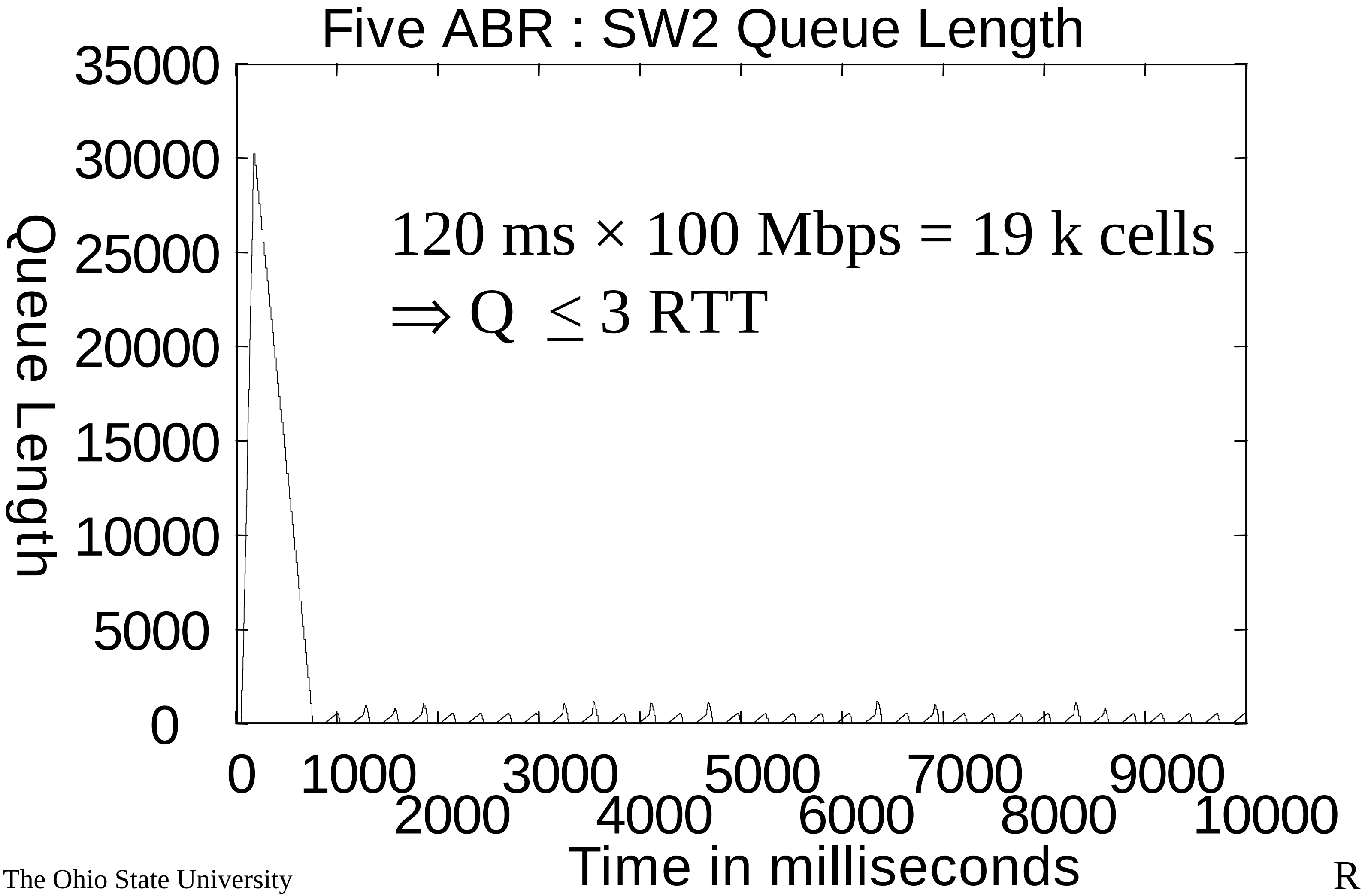
Parameters

- VS/VD and Non-VS/VD configurations.
- First hop = Satellite hop with 1 way delay:
    - LEO = 50 ms
    - GEO = 275 ms
- Link 2 = 45 Mbps (Bottleneck Link).
- All other links = 155.52 Mbps (149.76 with SONET)
- Persistent ABR sources: ICR = 30 Mbps
- Persistent TCP sources: Timer granularity = 500 ms. At 45 Mbps, 100 ms causes timeouts in GEO. Known problem with TCP Std deviation measurement.

ERICA+ Non-VS/VD LEO



ERICA+ Non-VS/VD LEO

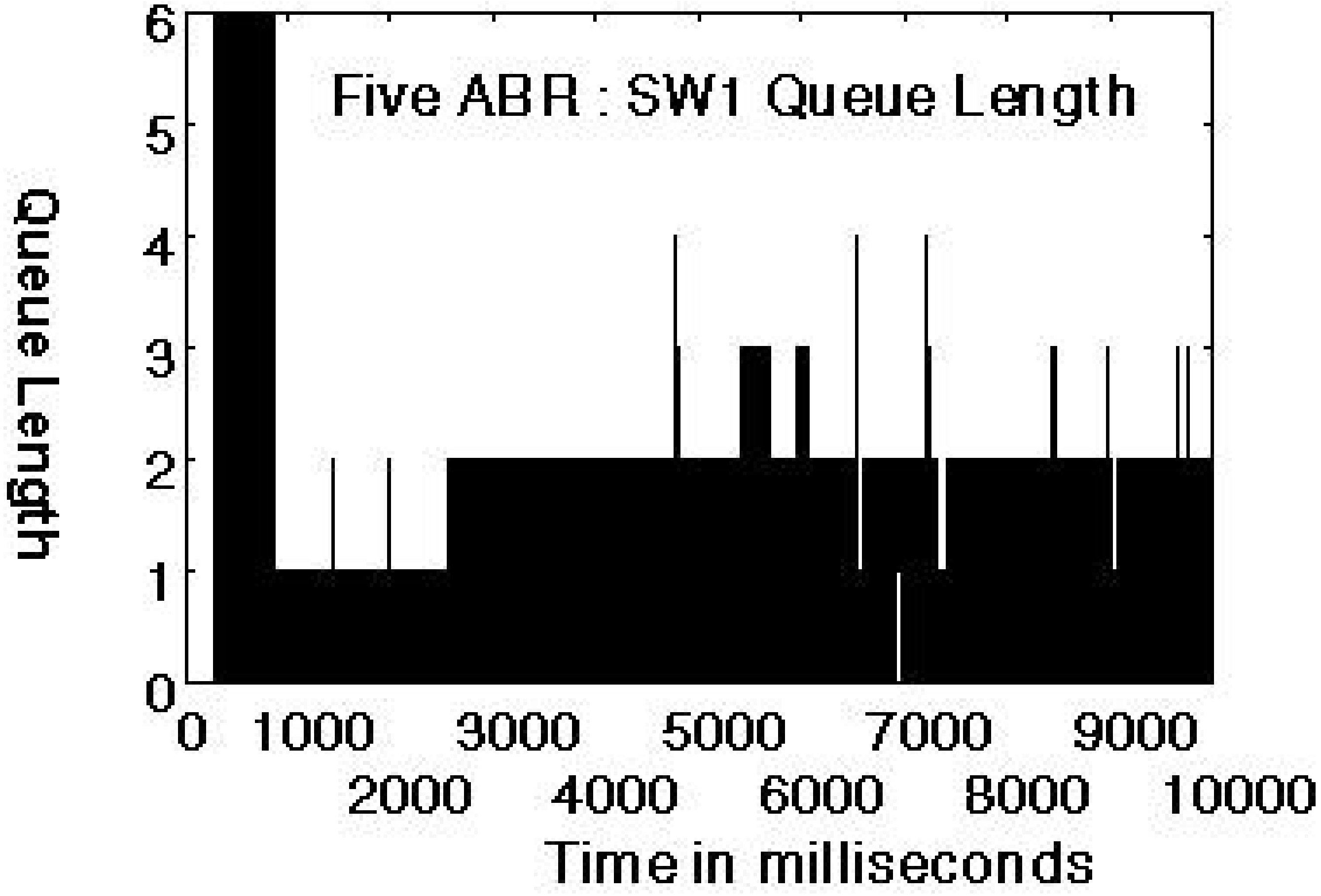


Five ABR : SW2 Queue Length

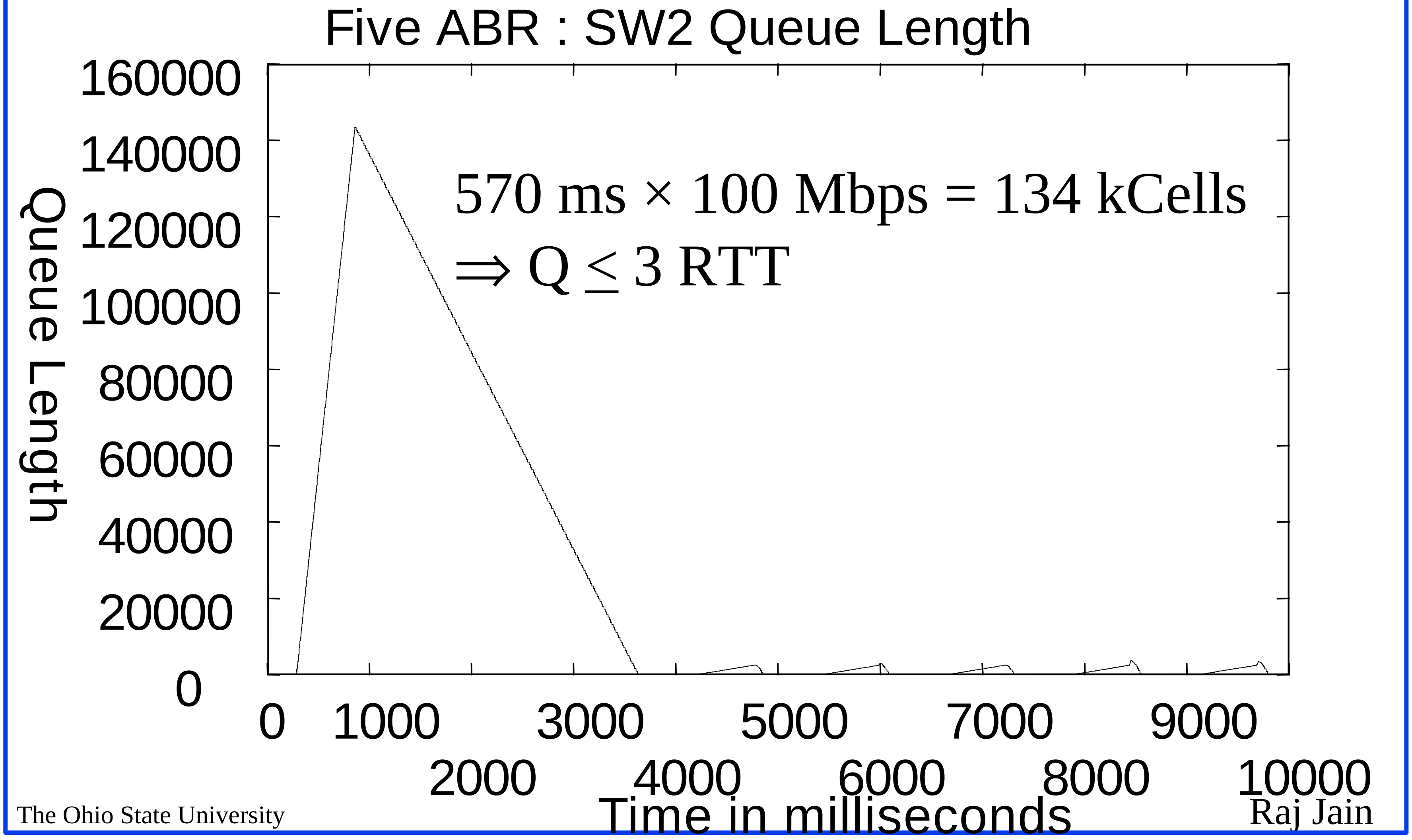
120 ms × 100 Mbps = 19 k cells

⇒ Q ≤ 3 RTT

ERICA+ Non-VS/VD GEO



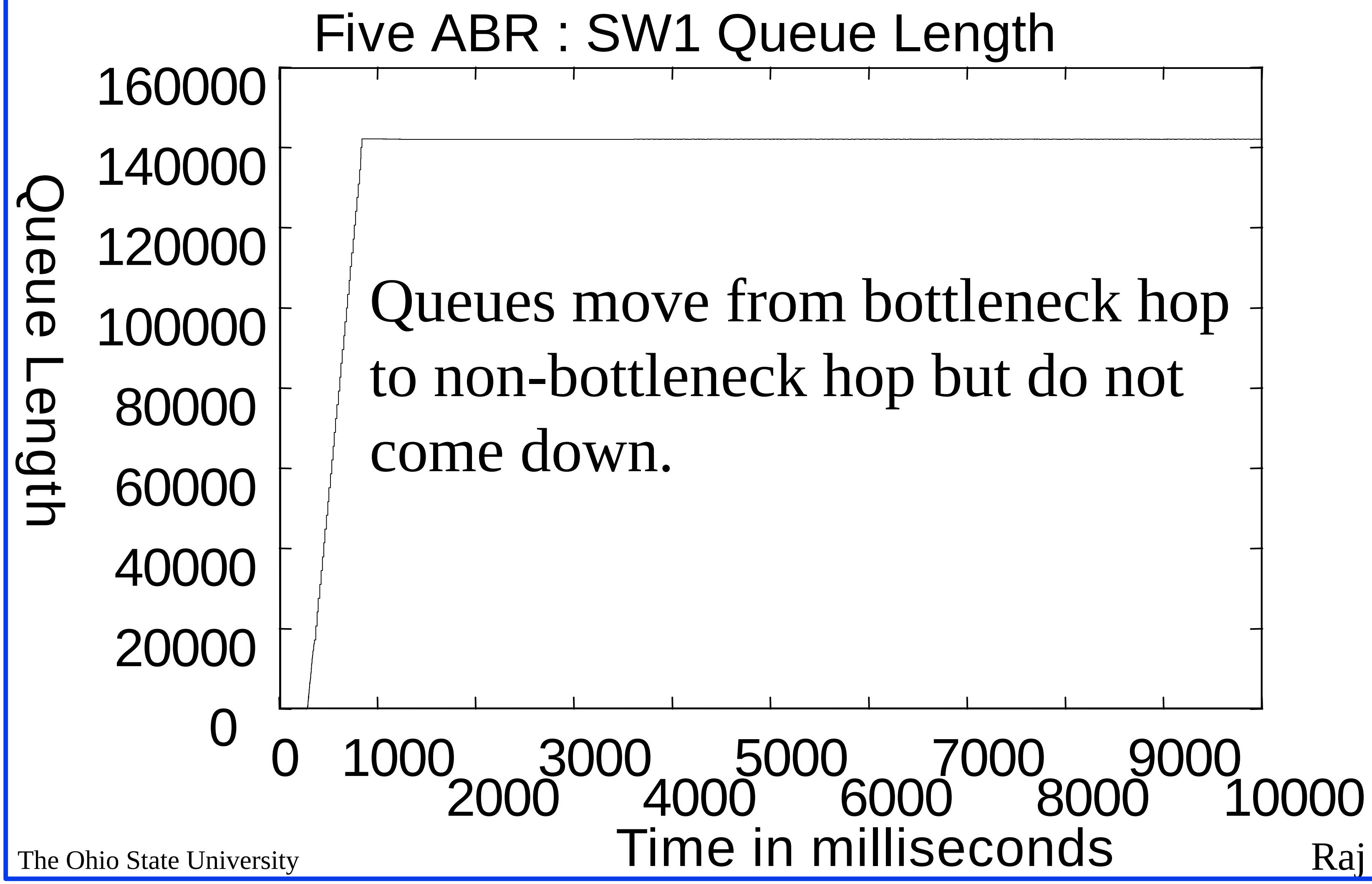
ERICA+ Non-VS/VD GEO



570 ms × 100 Mbps = 134 kCells

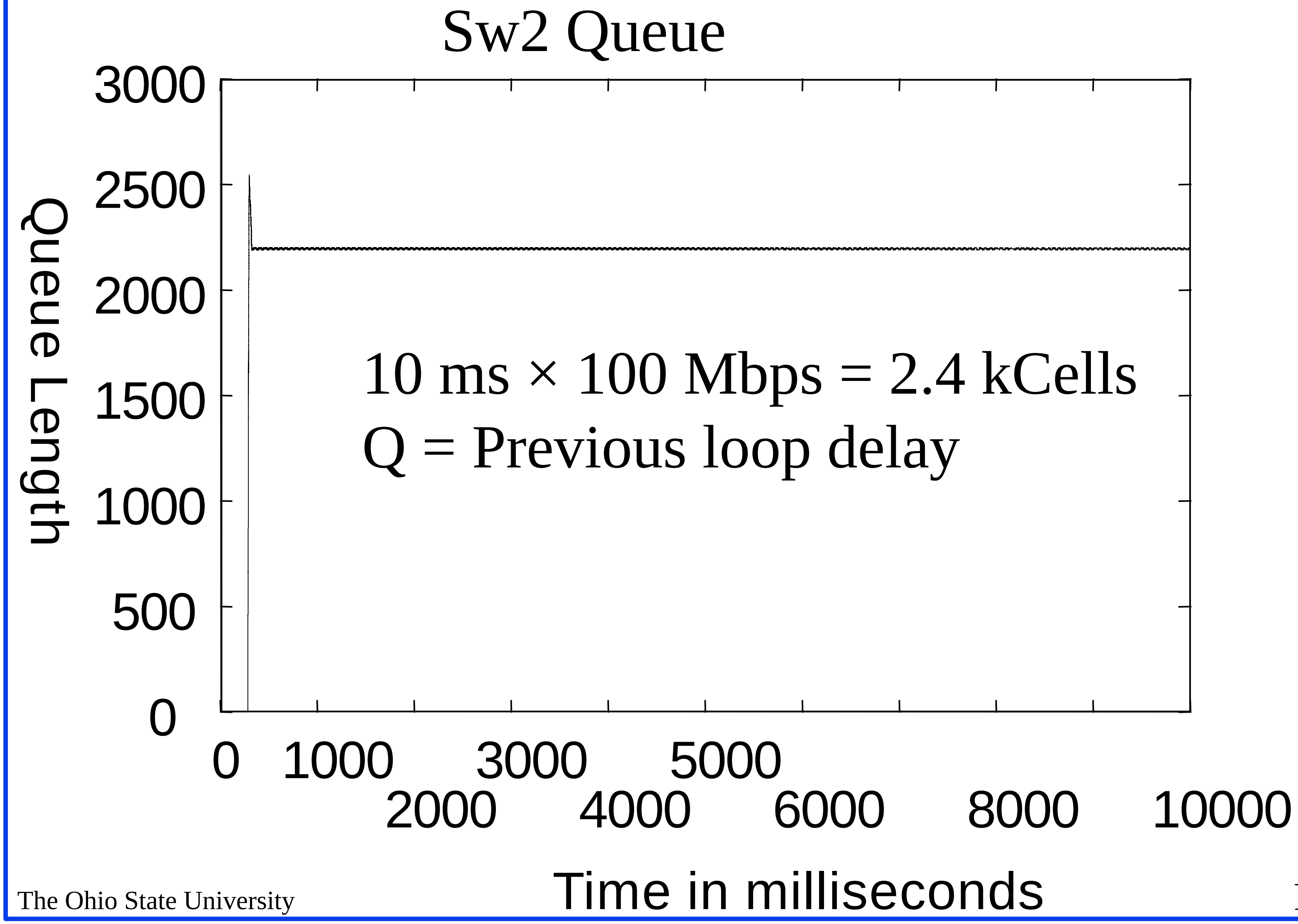
⇒ Q ≤ 3 RTT

ERICA w/ Simple VS/VD



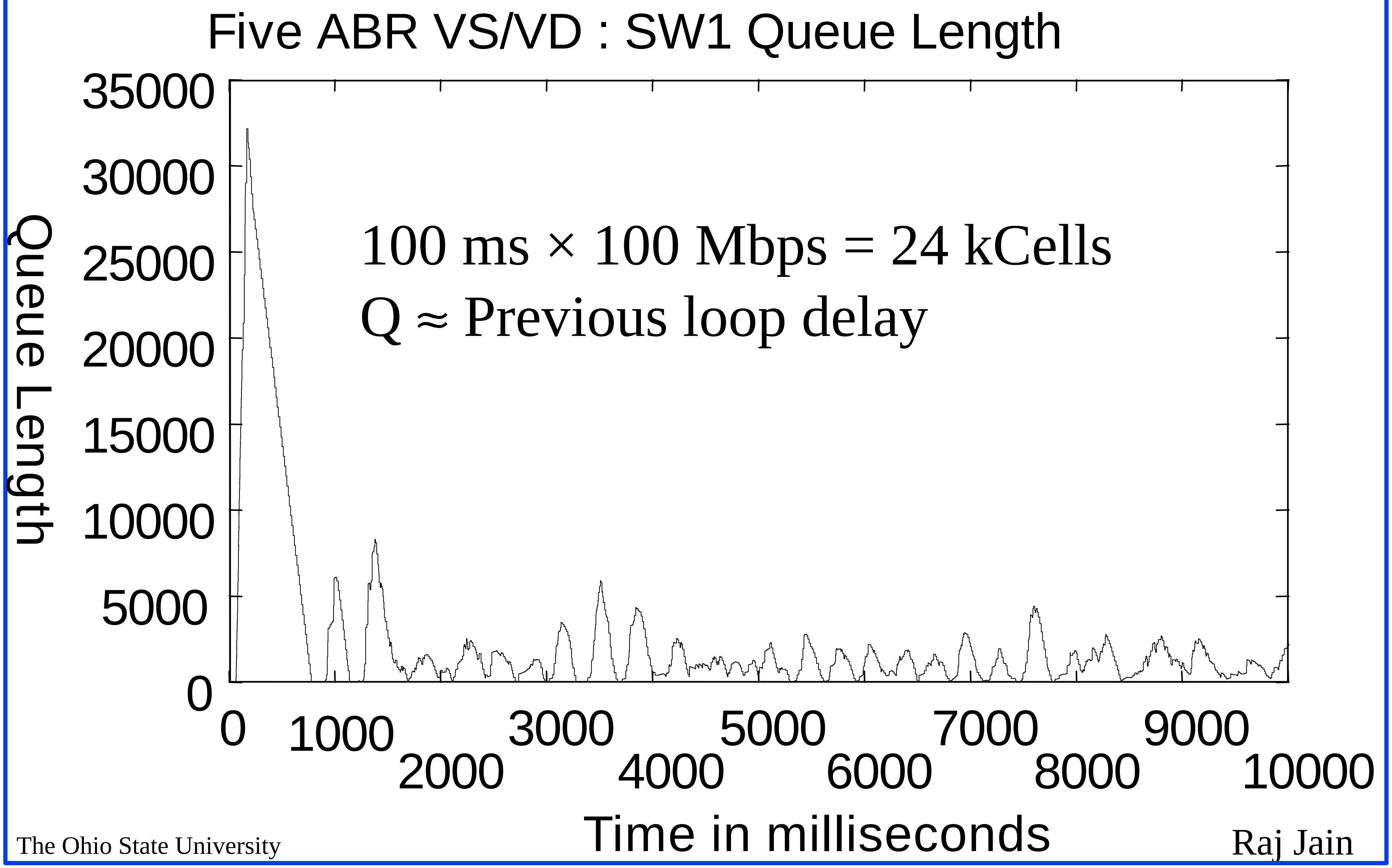
Queues move from bottleneck hop to non-bottleneck hop but do not come down.

ERICA w/ Simple VS/VD



10 ms × 100 Mbps = 2.4 kCells
Q = Previous loop delay

ERICA+ VS/VD LEO

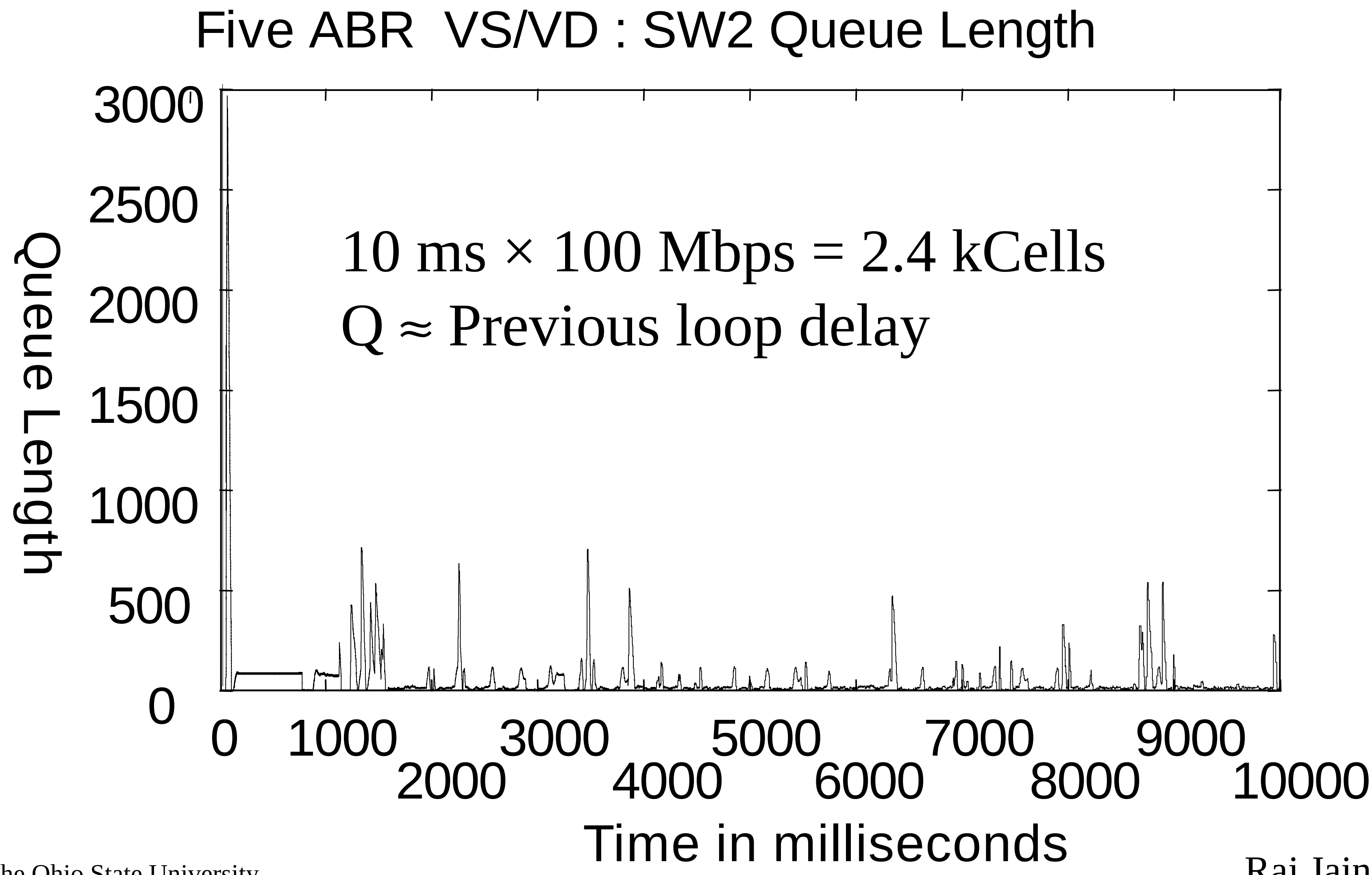


Five ABR VS/VD : SW1 Queue Length

100 ms × 100 Mbps = 24 kCells

Q ≈ Previous loop delay

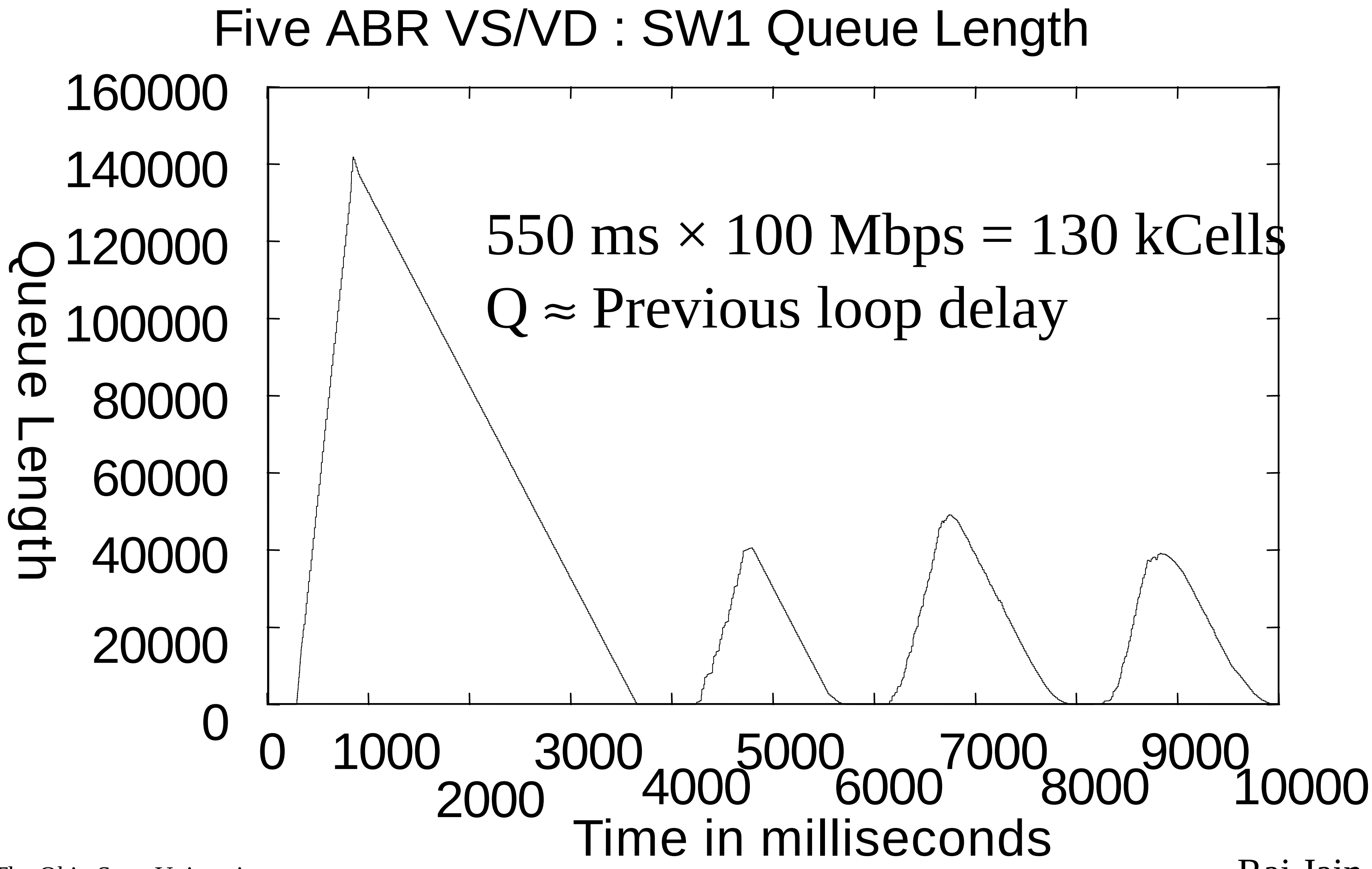
ERICA+ VS/VD LEO



10 ms × 100 Mbps = 2.4 kCells

Q ≈ Previous loop delay

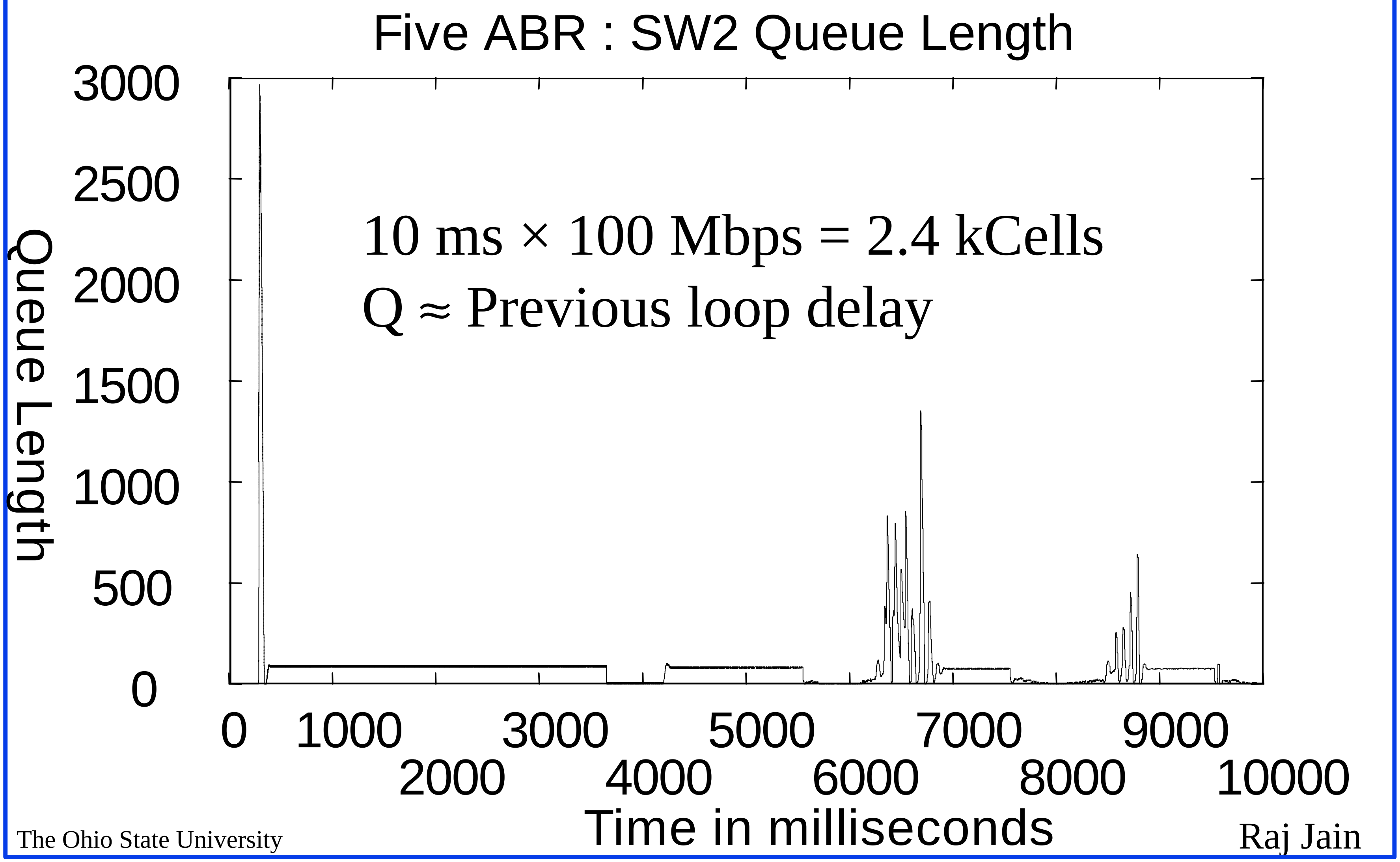
VS/VD GEO Sw1 Queue



550 ms × 100 Mbps = 130 kCells

Q ≈ Previous loop delay

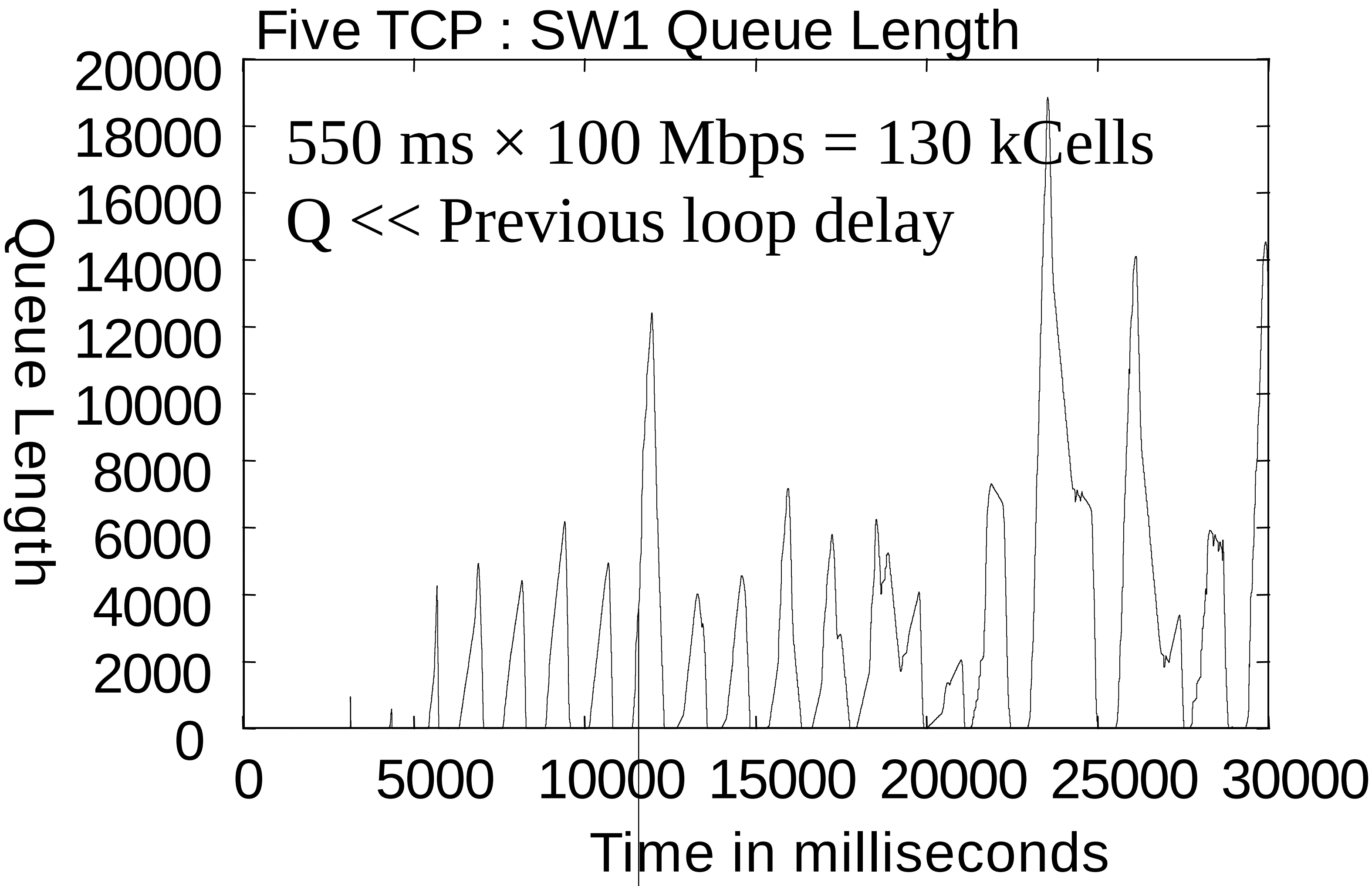
ERICA+ VS/VD GEO



10 ms × 100 Mbps = 2.4 kCells

Q ≈ Previous loop delay

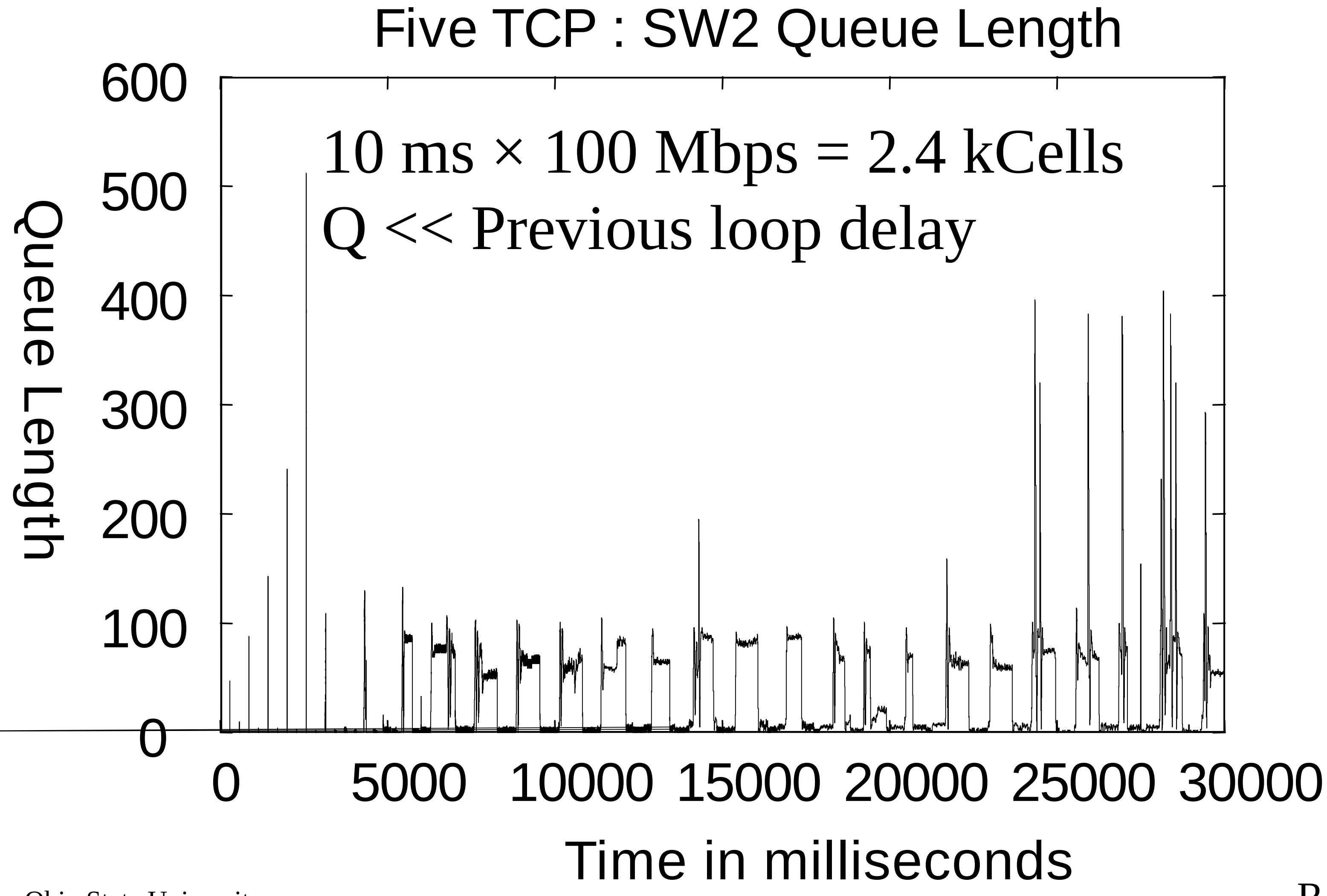
ERICA+ VS/VD GEO TCP



550 ms × 100 Mbps = 130 kCells

Q << Previous loop delay

ERICA+ VS/VD GEO TCP



10 ms × 100 Mbps = 2.4 kCells

Q << Previous loop delay

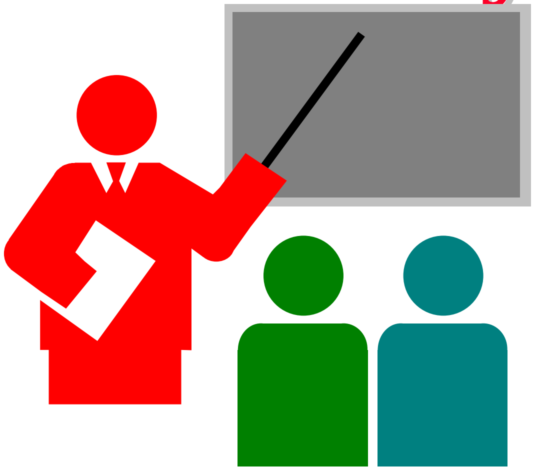
Simulation Results

VS/VD	Sw.	Feedback delay	B/w diff.	Max Exp. Q (cells)	Max Obs. Q (cells)
OFF	Sw1	120	0	0	0
OFF	Sw2	120	100	3*28 K	30 K
ON	Sw1	100	100	3*25 K	30 K
ON	Sw2	10	100	3*2.4 K	3 K
OFF	Sw1	570	0	0	0
OFF	Sw2	570	100	3*135 K	140 K
ON	Sw1	550	100	3*130 K	140 K
ON	Sw2	10	100	3*2.4 K	3 K

Observations

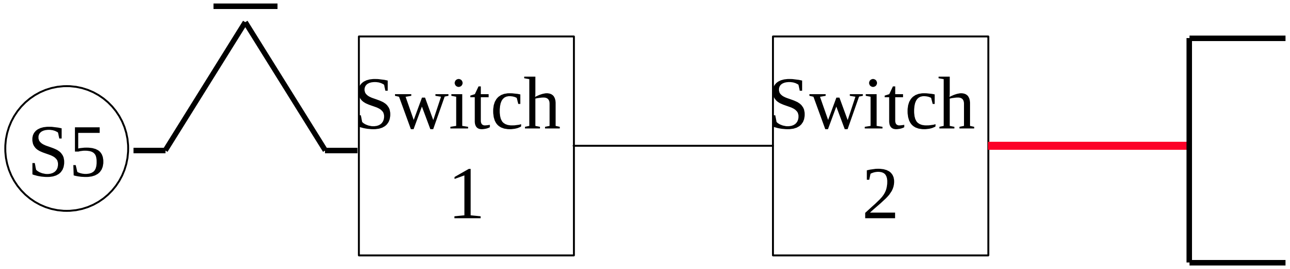
- Without VS/VD:
  - Single control loop for the entire connection.
  - All queues are in the bottleneck switch.
  - Buffer requirements for terrestrial switch are proportional to satellite propagation delay.
- With VS/VD:
  - Control loop broken at each switch.
  - Queues remain at the switch between the satellite and the terrestrial loop (satellite switch).
  - Terrestrial switch only requires small buffers.

Summary



- VS/VD switch architecture:
  - Per-VC queues drain at an ACR based only on the external congestion and class Q
  - Feedback to upstream queue must include external congestion, class Q, and per-VC Q.
  - Each queue must monitor its input and output rate.

Summary (Cont)



- With correct implementation of VS/VD:
  Maximum queue at each switch
  ≤ Bandwidth delay product of the previous loop
  ⇒ Can help isolate long-delay hops from short-delay hops.
- Workgroup switches on satellite paths will not need buffering proportional to round-trip even if they are the bottleneck.
- Motion: Add sample VS/VD scheme to baseline text

Future Work

- More complex configurations.
- Presence of VBR background.
- Analysis of complexity of VS/VD switch.
- Scheduling policies for per-VC and class queues.

Motion

- Add the following two paragraphs to I.5.4 of the baseline text.

I.5.4 A Sample Explicit Rate VS/VD Switch Algorithm

One simple method to implement VS/VD is to have a separate queue (per-VC queue) for each VC. A server at the head of each of these queues monitors the input rate of the queue, provides feedback to the upstream queue, and controls the output rate of the queue based on the feedback from the corresponding downstream server. When providing feedback, each server only allocates up to the rate at which it is allowed to output (ACR). However, if queues are large, the server may allocate only a part of its ACR to the previous hop so that its queues can drain quickly. The main features and options of the algorithm are similar to the ERICA+ algorithm. ERICA+ is an extension of the ERICA algorithm, and uses queue length to dynamically set the target ABR capacity.

Motion (contd.)

The basic rate allocation algorithm consists of the following steps at the end of every averaging interval. The port overload is calculated as the ratio of the total measured service rate of the per-VC queues and the target ABR capacity. The fair share term for VCs is calculated as the ratio of the target ABR capacity to the number of active ABR VC. VCshare is calculated for each VC as the ratio of its measured service rate to the overload. The ER for each VC is calculated as ER = Min(Max(Fair Share, VC share), ER from downstream node). The ACR at which the VC's queue drains is determined from this ER as well as the source-end-system rules for the VS. The feedback to the previous hop for the VC is calculated as a fraction (based on the VC's queue length) of the calculated ACR.