****

ATM Forum Document Number: ATM Forum/97-1086R1

****

Title: Per-VC Rate Allocation Techniques for ABR Feedback in VS/VD Networks

****

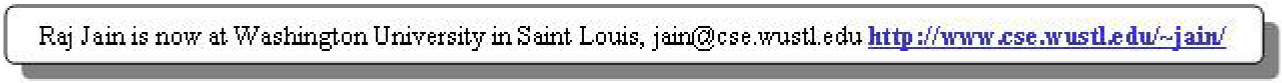
Abstract:

We describe issues in designing rate allocation schemes for virtual source / virtual destination (VS/VD) switches. Improper design of VS/VD such schemes can result in poor performance and large steady state queues. We propose a rate allocation scheme for VS/VD switches. This scheme is based on the ERICA+ algorithm, and uses per-VC queuing and per-VC control. We analyze the performance of this scheme, and conclude that VS/VD can help in limiting buffer requirements of switches, based on the length of their VS/VD control loops. *VS/VD is especially useful in isolating terrestrial networks from the effects of long delay satellite networks by limiting the buffer requirements of the terrestrial switches.* We present simulation results to substantiate our conclusions.

****

Source:

Rohit Goyal, Xiangrong Cai, Raj Jain, Sonia Fahmy, and Bobby Vandalore
The Ohio State University Department of Computer and Information Science
Columbus, OH 43210-1277


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

Kul Bhasin
Nasa Lewis Research Center
21000 Brookpark Road, Cleveland, OH 44135
Phone: 216-433-3676, Email: bhasin@lerc.nasa.gov

The presentation of this work at the ATM Forum is sponsored in part by NASA Lewis Research Center

****

Date: February 1997, Anaheim

****

Distribution: ATM Forum Technical Working Group Members (AF-TM)

****

Notice:

This contribution has been prepared to assist the ATM Forum. It is offered to the Forum as a basis for discussion and is not a binding proposal on the part of any of the contributing organizations. The statements are subject to change in form and content after further study. Specifically, the contributors reserve the right to add to, amend or modify the statements contained herein.

****

1 Introduction

The virtual source virtual destination (VS/VD) behavior specified in TM4.0 allows ATM switches to split an ABR control loop into multiple control loops. Each loop can be separately controlled by the nodes in the loop. The coupling between adjacent ABR control loops has been left unspecified by the forum, and is implementation specific.

VS/VD control can isolate different networks from each other. For example, two ABR networks can be isolated from a non-ATM network that separates them. Also, long latency satellite networks can be isolated from terrestrial networks so as to keep the effects of large latency to within the satellite loop.

VS/VD implementation in a switch, and the coupling of adjacent control loops present several design options to switch manufacturers. A VS/VD switch is required to enforce the ABR end-system rules for each VC. As a result, the switch must be able to control the rates of its VCs at its output ports. Per-VC queuing and scheduling can be used to easily enforce the rate allocated to each VC. With the ability to control per-VC rates, switches at the edge of the VS/VD loops can respond to congestion notification from the adjacent loop by controlling their output rates. Switches can also use downstream congestion information, as well as their internal congestion information, to provide feedback to the upstream loop. The ability to perform per-VC queuing adds an extra dimension of control for switch traffic management schemes. Rate allocation mechanisms can utilize the per-VC control at every virtual end system (VS/VD end point) for dimensioning of resources for each VS/VD loop.

In this contribution, we present several issues in VS/VD switch design. We first describe the basic architectural components of a VS/VD switch. We then discuss possible caveats in designing rate allocation schemes for a VS/VD switch. Here, we describe problems that may arise from naive implementations of feedback control schemes taken from non-VS/VD switches. We then present a rate allocation scheme for feedback control in a VS/VD switch. This scheme is based on the ERICA+ scheme, but uses the per-VC information and control available with VS/VD. We present simulation results with this scheme to show that VS/VD can help in switch buffer sizing, and isolation of users sharing a link.

2 A VS/VD Switch Architecture

Figure 1 illustrates the basic architecture of an output buffered VS/VD switch. The figure shows two output ports of the switch, and the data and RM cell flow of a VC going through the switch. Data and RM cells arrive at the input side of port 1. Data cells are switched to the appropriate destination port to be forwarded to the next hop. RM cells are turned around and sent back to the previous hop. For the VC shown in the figure, port 1 acts as the VD that accepts the data cells and turns around the RM cells, while port 2 acts as the VS for the next hop. Port 1 provides feedback to the upstream node in the VC's path by inserting congestion and rate information in the appropriate RM cell fields. Port 2 sends the data to the next hop, generates an RM cell every *Nrm* cells, and enforces all the source rules specified in the ABR end-system behavior. Port 1 also accepts and processes the turned around BRM cells returned by the downstream end system in the VC's path.

Each port has a class queue for the ABR class, as well as per-VC queues for each ABR VC.[1] Each per-VC queue

[1] The class queue is not essential if per-VC queuing and scheduling are used, but we include it to illustrate a general architecture. The

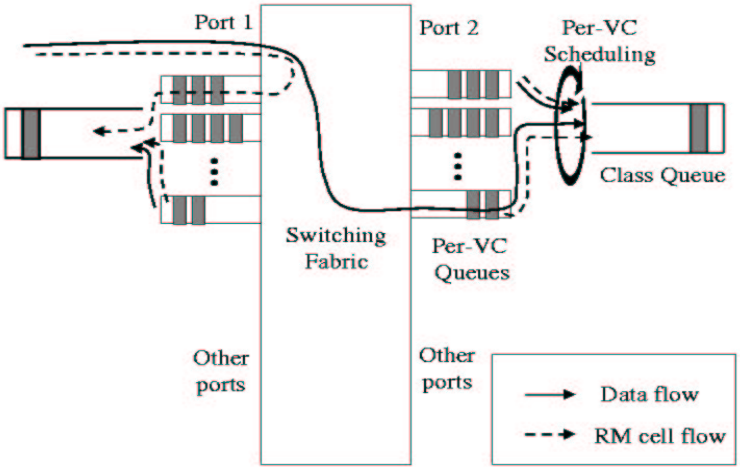


Figure 1: VS/VD switch architecture

contains the data cells and turned around RM cells for its VC. Each per-VC queue drains into the class queue at the ACR allocated to the corresponding VC. The class queue drains at the link rate of the outgoing link.

A scheduling mechanism ensures that each VC gets a fair share of the total link capacity. In principle, the scheduling policy must allow the VS to send at the rate that is allowed by a combination of the allocation policy and the end system behavior. However, when ACRs are overbooked, the scheduling policy must service the per-VC queues in some fair proportion of their ACR or MCR values. Details of scheduling policy design are a topic of future study.

In the next section, we present several issues that arise in designing a rate allocation scheme for VS/VD switches. We first present a queuing model for a non-VS/VD switch, and discuss the most common rate allocation techniques used by such switches. We then describe some problems that may arise when non-VS/VD rate allocation schemes are naively adapted to work in VS/VD switches. We discuss solutions and provide recommendations for designing schemes with VS/VD behavior.

3 Design Issues For Explicit Rate Allocation with VS/VD

Figure 2 shows a queuing model for a single port of an **output buffered non-VSVD switch** (node i). The port has one class queue for the ABR VCs. Cells from all the ABR VCs destined for the output port are enqueued in the class queue in a FIFO manner. Let the input rate of VC_j into node j be s_{ij}, and the input rate into the class queue be r_{ij}. In this case, since the node simply switches cells from the input to the output port, we have $s_{ij} = r_{ij}$. Let R_i be the output rate of the class queue at the given port of node i. Then R_i corresponds to the total bit rate of the link available to ABR. Let q_i be the queue length of the class queue. Let N be the number of ABR VCs sharing the link.

Many rate allocation algorithms use a parameter F_i, $(0 < F_i \leq 1)$ which is the target utilization of the link. The link

class queue can be removed without affecting the scheme presented in this contribution.

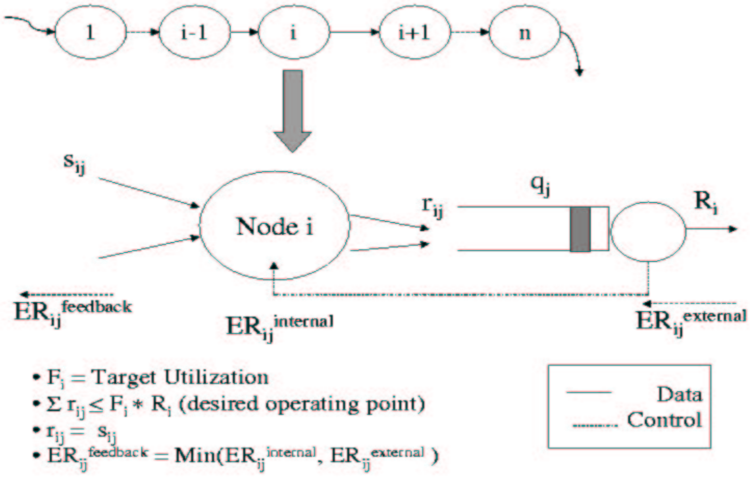

1
i-1
i
i+1
n
s_{ij}
Node i
r_{ij}
q_j
R_i
$ER_{ij}^{feedback}$
$ER_{ij}^{internal}$
$ER_{ij}^{external}$
• F_i = Target Utilization
• $\Sigma r_{ij} \leq F_i * R_i$ (desired operating point)
• $r_{ij} = s_{ij}$
• $ER_{ij}^{feedback} = Min(ER_{ij}^{internal}, ER_{ij}^{external})$
Data
Control

Figure 2: Queuing model for non-VS/VD switch

rate is allocated among the VCs so that

$$\sum_{j=1}^{j=N} r_{ij} \leq F_i R_i$$

i.e., the goal of the switch is to bring the total input rate into the class queue to the desired value of $F_i R_i$. Let $ER_{ij}^{internal}$ be the ER calculated by the node based on the internal congestion in the node. This is the rate at which the switch desires VC_j to operate. Node i also receives rate allocation information from the downstream node $(i+1)$. This is shown in the figure as $ER_{ij}^{external}$. Node i provides feedback to the upstream node $(i-1)$, as

$$ER_{ij}^{feedback} = Min(ER_{ij}^{internal}, ER_{ij}^{external})$$

At node $(i-1)$, $ER_{ij}^{feedback}$ is received as $ER_{(i-1)j}^{external}$, and node $(i-1)$ performs its rate calculations for VC_j in a similar fashion.

The internal explicit rate calculation is based on the local switch state only. A typical scheme like ERICA [3], uses several factors to calculate the explicit rate. In particular, the ERICA algorithm uses the total input rate to the class queue, the target utilization of the link, and the number of VCs sharing the link to calculate the desired operating point of each VC in the in the next feedback cycle, i.e.,

$$ER_{ij}^{internal} = \text{fn}(\sum_j r_{ij}, F_i R_i, N)$$

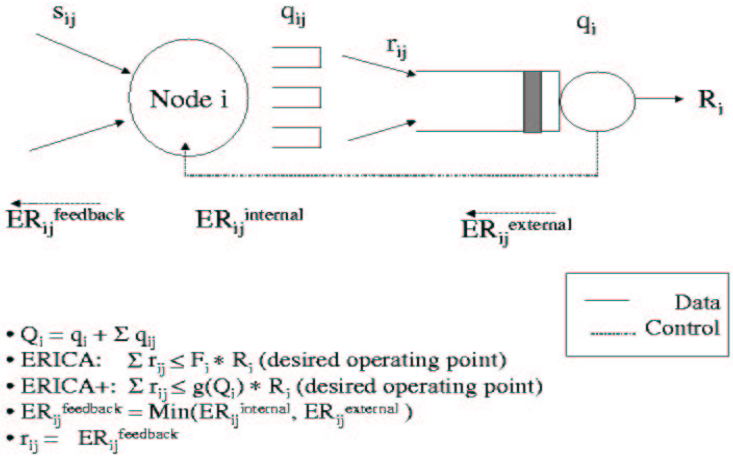
In steady state, the ERICA algorithm maintains $\sum_j r_{ij} = F_i R_i$, so that any queue accumulation due to transient overloads can be drained at the rate $(1 - F_i)R_i$. As a result, the ERICA algorithm only allocates a total of $F_i R_i$ to the VCs sharing the link, and results in $100F_i\%$ steady state link utilization of the outgoing link.

The ERICA+ algorithm can achieve 100% steady state link utilization by additionally considering the queue length of the class queue when it calculates the internal rate for VC_j, i.e., for ERICA+,

$$ER_{ij}^{internal} = \text{fn}(\sum_j r_{ij}, g(q_i)R_i, N)$$

where $g(q_i)$, $(0 < g_{min} \leq g(q_i) \leq g_{max})$ is a function known as the *queue control function*, that scales the total allocated capacity R_i based on the current queue length of the class queue. If q_i is large, then $g(q_i) < 1$ so that $\sum_j r_{ij} = g(q_i)R_i$ is the target operating point in the next feedback cycle, and $(1 - g(q_i))R_i$ can be used to drain the queue to a desired value (q_i^{target}). The queue control function is bounded below by $g_{min} > 0$ so that at least some minimal capacity is allocated to the VCs. A typical value for the ERICA+ algorithm of g_{min} is 0.5. When the queue is small, $(q_i < q_i^{target})$, $g(q_i)$ may increase to slightly more than 1 so that sources are encouraged to send at a high rate. As a result, switches try to maintain a pocket of queues of size q_i^{target} at all times.

In the remainder of this section, ERICA and ERICA+ are used as a basis for our discussion. However, the discussion is general, and applies to any rate allocation scheme that uses the target utilization and queue length parameters in its rate calculations. The discussion presents some fundamental concepts that should be used in the design of rate allocation algorithms for VS/VD switches.

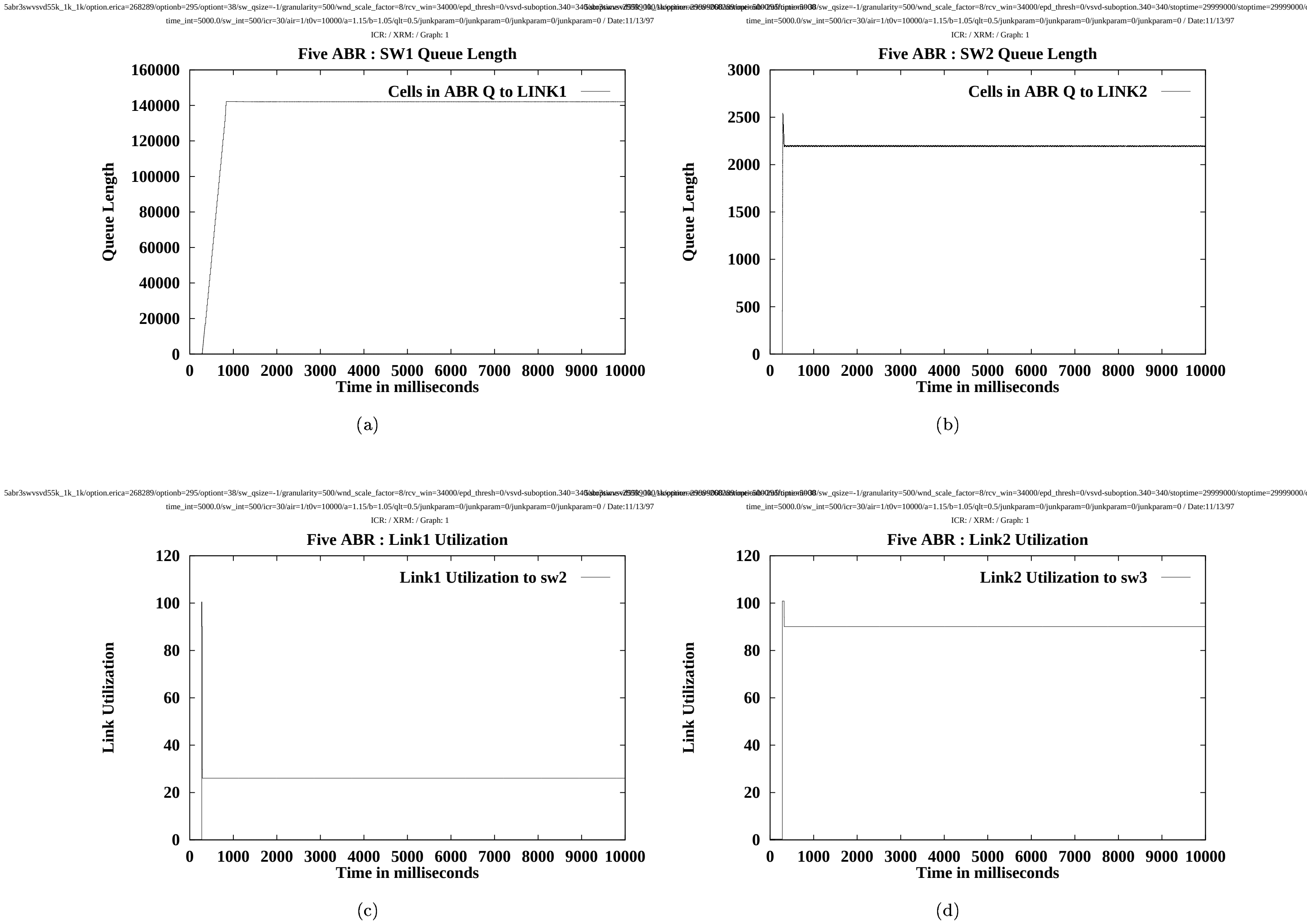


s_{ij}
q_{ij}
r_{ij}
q_i
Node i
R_i
$ER_{ij}^{feedback}$
$ER_{ij}^{internal}$
$ER_{ij}^{external}$
Data
Control
• $Q_i = q_i + \Sigma q_{ij}$
• ERICA: $\Sigma r_{ij} \leq F_j * R_j$ (desired operating point)
• ERICA+: $\Sigma r_{ij} \leq g(Q_i) * R_j$ (desired operating point)
• $ER_{ij}^{feedback} = \mathrm{Min}(ER_{ij}^{internal}, ER_{ij}^{external})$
• $r_{ij} = ER_{ij}^{feedback}$

Figure 3: Simple queuing model for a VS/VD switch

Figure 3 illustrates a simple adaptation of ERICA and ERICA+ to a VS/VD switch. The VS/VD switch can control the rates of the per-VC queues. r_{ij} is the rate at which VC_i's per-VC queue drains into the class queue. Like in ERICA, r_{ij} is set to the $ER_{ij}^{feedback}$ value calculated by the node. The explicit rate is calculated as before for both ERICA and ERICA+. ERICA+ uses the sum of the per-VC queues and the class queue for the queue control function. The key feature in this adaptation is that the output rate of the per-VC queue is set to the desired input rate at the class queue. This value is also fed back to the upstream hop of the previous loop. This simple approach can present problems in some cases.

Suppose that node i is the bottleneck node for VC_j, i.e., $ER_{ij}^{internal} < ER_{ij}^{external}$, and $ER_{ij}^{feedback} = ER_{ij}^{internal}$. As a result,$VC_j$ of node $(i-1)$ sends at a rate of $ER_{ij}^{internal}$, i.e., the input rate to VC_j's per-VC queue is $s_{ij} = ER_{ij}^{internal}$. Also, the VC_j's queue drains at the rate $r_{ij} = ER_{ij}^{internal}$. Thus, the per-VC queue of VC_i can not recover from transient overloads and results in an unstable condition. This is shown in the simulation results in figure 4. The figure shows the queue lengths and percentage link utilizations for the configuration shown in figure 6 and described in section 5. Both switch 1 and switch 2 queues build up during the open loop control phase of the simulation. When the closed loop VS/VD control sets in, the queues cannot drain because the input and output rates of each



(a)

(b)

(c)

(d)

Figure 4: Performance of incorrect implementation of VS/VD

switch are the same. When the queues build up, the link utilization of link 2 (the bottleneck link) should be 100%. However, the class queue in switch 2 is empty because the sum of the per-VC queues is only F_iRi with $F_i = 0.9$. As a result, the utilization of link 2 is 90% of the expected value.

The problem with the above scheme is that it ignores the existence of an ABR server at each VC-queue. The scheme uses the explicit rates calculated by the server at the class queue, and uses these as the output rates for the per-VC queues. As a result, the sum total of the output rates of the per-VC queues is limited to F_iR_i, hence limiting the drain rate of the class queue to the same value. The $(1 - F_i)R_i$ capacity is thus never usable since $\sum_j r_{ij} \leq F_iR_i$.

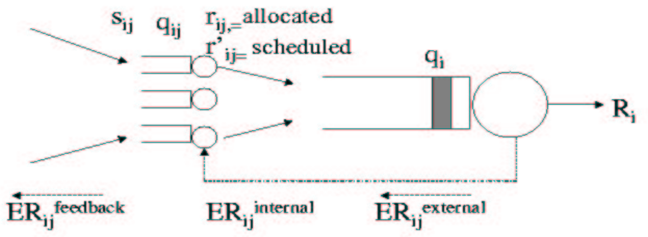
Figure 5 shows a better model for a VS/VD switch. The presence of servers at the per-VC queues is explicitly noted, and the input rates to the per-VC queues are not the same as their output rates. Separate servers are shown before each queue, because these servers process the cells before they enter the queue. The servers at the per-VC queues also control the output rates of their respective queues. In the case of ERICA, the sum total of the input rates to the class queue is limited by $F_i R_i$. This allows the class queue to drain in case of transient overloads from the per-VC queues. The input to the per-VC queues (s_{ij}) is limited by $F_i r_{ij}$, allowing the per-VC queues to also use a rate of $(1 - F_i)r_{ij}$ to recover from transient overloads. Moreover, for an ERICA+ like scheme that uses queue length information to calculate available capacity, additional per-VC queue information is now used to control s_{ij} in relation to r_{ij}. Thus, for ERICA+, the desired operating point is decided for the next feedback cycle such that

$$\sum_j r_{ij} \leq g(q_i)R_i$$

and

$$s_{ij} \leq g(q_{ij})r_{ij}$$

The feedback given to the previous loop is set to the desired per-VC operating point, which is the desired input rate to the per-VC queues. As a result, the per-VC feedback is further controlled by the VC's queue length. This can be used to further isolate the VCs on the same link from one another. Thus, if VC_j experiences a transient overload, only $ER_{ij}^{feedback}$ is reduced and the feedbacks to the remaining VCs are not affected by this temporary overload.



- ERICA: (desired operating point)
  - $\Sigma\, r_{ij} \leq F_i * R_i$
  - $s_{ij} \leq F_i * r_{ij}$
- $ER_{ij}^{internal}$ Min={Max(r'_{ij}/Overload , F_i*R_i/N), $ER_{ij}^{external}$ }
- r_{ij} = fn($ER_{ij}^{external}$, end system rules)
- $ER_{ij}^{feedback} = F_i * r_{ij}$

- ERICA+: (desired operating point)
  - $\Sigma\, r_{ij} \leq g(q_i) * R_i$
  - $s_{ij} \leq g(q_{ij}) * r_{ij}$
- $ER_{ij}^{internal}$ = Min{Max(r'_{ij}/Overload, $g(q_i)$*R_i/N), $ER_{ij}^{external}$ }
- r_{ij} = fn($ER_{ij}^{external}$, end system rules)
- $ER_{ij}^{feedback} = F_i * r_{ij}$

Figure 5: Queuing model for per-VC VS/VD switch

The following section presents a rate allocation scheme for VS/VD switches based on the above principles. This scheme is a variation of the ERICA+ algorithm, and converges to max-min fairness while achieving high link utilization. The scheme also limits the maximum buffer sizes in a switch to a function of the delay-bandwidth product of its upstream VS/VD control loop. Section 5 presents the simulation results of this scheme and illustrates how VS/VD can be used in switches to limit buffer sizes over non-VS/VD switches.

4 A Per-VC Rate Allocation Algorithm for VS/VD

The scheme presented in this section is based on the ERICA+ scheme for ABR feedback [3]. The basic switch model is shown in figure 5. The switch maintains an averaging interval at the end of which it calculates the rate allocations

($ER_{ij}^{internal}$) for each VC to provide feedback to the previous hop. $ER_{ij}^{internal}$ is calculated for each VC based on the following factors:

- The actual (measured) scheduled rate of the VC queue into the class queue or the link ($\hat{r}_{ij}$).
- The allocated rate (ACR) of the VC queue into the class queue or the link (r_{ij}).
- The queue length of the class queue (q_i).
- The output rate of the class queue (R_i). This is also the total estimated ABR capacity of the link.
- The number of active ABR VC's sharing the class queue (N).
- The external rate allocation received by each VC from the downstream hop ($ER_{ij}^{external}$).
- The queue control function $g()$.

A portion of the link capacity $g(q_i)R_i$ is divided in a max-min fair manner among the per-VC queues[2]. The remaining portion is used to drain the class queue formed due to transient overloads. Then, the per-VC feedback is calculated for the upstream hop based on the per-VC queue length (q_{ij}) and the allocated rate (ACR) of the per-VC queue (r_{ij}). This calculation allocates a fraction (that depends on the queue length) of r_{ij} to the previous hop as $ER_{ij}^{feedback}$ so that s_{ij} in the next cycle is less than r_{ij} thus allowing the per-VC queue to drain out any transient overloads.

The basic design is based on the following principle. A desired input rate is calculated for each queue in the switch, and this desired rate is given as feedback to "the previous server" in the network. In the case of the class queue, the previous server controls the per-VC queues of the same node. The previous server for the per-VC queue is the class queue of the upstream hop in the VS/VD loop.

The basic algorithm consists of the following steps.

- When a BRM cell is received, the ER in the RM cell is copied to $ER_{ij}^{external}$.
- When an FRM cell is received, it is simply turned around, and its ER is stamped with the value $ER_{ij}^{feedback}$.
- Rate calculations are performed only once every averaging interval as follows

$$\begin{aligned}
\text{Overload} &\leftarrow \frac{\sum_j \hat{r}_{ij}}{g(q_i)R_i} \\
ER_{ij}^{internal} &\leftarrow \text{Min}\{\text{Max}\left(\frac{\hat{r}_{ij}}{\text{Overload}}, \frac{g(q_i)R_i}{N}\right), ER_{ij}^{external}\} \\
r_{ij} &\leftarrow \text{fn}(ER_{ij}^{internal}, \text{end-system rules}) \\
ER_{ij}^{feedback} &\leftarrow g(q_{ij})r_{ij}
\end{aligned}$$

This results in s_{ij} in the next feedback cycle to be $g(q_{ij})r_{ij}$. The remaining features and options of the algorithm are the same as the ERICA+ algorithm. For more details refer to [3].

[2] In the absence of a class queue, the function $g(q_i)R_i = F_iR_i$ where $F_i \leq 1$ is the target utilization of the link

5 Simulation Results

In this section we present simulation results to highlight the features of the VS/VD rate allocation scheme presented in this contribution, and its potential advantages over non-VS/VD switches. In particular, we are interested in comparing the buffer requirements of a VS/VD switch with those of a non-VS/VD switch.

Configuration

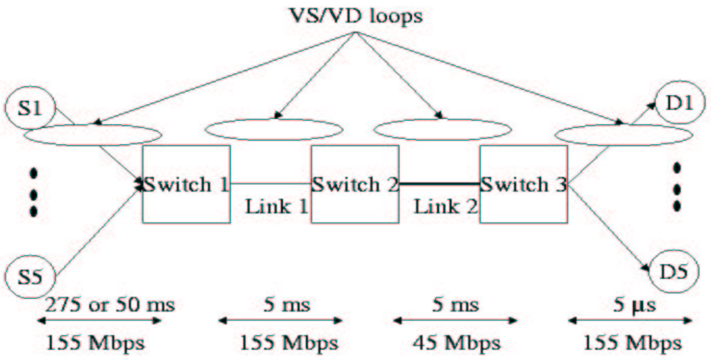


Figure 6: Five sources satellite configuration

Figure 6 shows the basic configuration used in the simulations. The configuration consists of three switches separated by 1000 km links. The one way delay between the switches is 5 ms. Five sources send data as shown in the figure. The first hop from the sources to switch 1 is a long delay satellite hop. We simulated two values of one way delay – 275 ms (GEO satellite delay), and 50 ms (LEO satellite delay). The link capacity of link 2 is 45 Mbps, while all other links ar 155 Mbps links. Most of our simulations use infinite ABR sources where the ABR source always has data to send at its allowed cell rate. We also use persistent TCP sources to show the performance of the scheme for bursty traffic. Initial cell rates are set to 30 Mbps in all experiments. For the TCP experiments, the TCP timer granularity is set to 500 ms.

Results

Figure 7 illustrates the difference in the maximum buffer requirements for a VS/VD switch and an non-VS/VD switch with the GEO satellite delay configuration. Switch 2 is the bottleneck switch since link 2 has a capacity of 45Mbps. Switch 1 is connected to the satellite hop and is expected to have large buffers. Switch 2 is a terrestrial switch, and its buffer requirements should be proportional to the delays experienced by terrestrial links. Without VS/VD, all queues are in the bottleneck switch (switch 2). The delay-bandwidth product from the bottleneck switch to the end system is about 150,000 cells (155 Mbps for 550 ms). This is the maximum number of cells that can be send to switch 2 before the effect of its feedback is seen by the switch. Figure 7(d) shows that without VS/VD, the maximum queue length in switch 2 is proportional to the feedback delay-bandwidth product of the control loop between the ABR source and the bottleneck switch. However, a terrestrial switch is not expected to have such large buffers, and should be isolated from the satellite network. In the VS/VD case, (figure 7 (a) and (b)), the queue is contained in switch 1 and not switch 2. The queue in switch 2 is limited to the feedback delay-bandwidth product of the control loop between switch 1 and switch 2. The observed queue is always below the maximum expected queue size of about 3000 cells (155 Mbps for 10 ms).

Figure 8 shows the corresponding result for the LEO satellite configuration. Again, with the VS/VD option, queue accumulation during the open loop period is moved from switch 2 to switch 1. The maximum queue buildup in switch 1 during the open loop phase is about 35000 (155 Mbps for 120 ms).

Figures 9 and 10 illustrate the corresponding link utilizations for link 1 and link 2 for the GEO and LEO configurations respectively. The figures show that the link utilizations are comparable for VS/VD and non-VSVD. Figures 11 and 12 show the ACRs allocated to each source for the GEO and LEO cases respectively. The ACR graphs show that the resulting scheme is fair in the steady state. The transient differences in the ACRs due to the small transient differences in the per-VC queue length.

Figure 13 shows the results of the above two configurations with persistent TCP sources. The TCP sources result in bursty traffic to the switch because of the window based TCP flow control. The figures show that even with bursty sources, the maximum buffer size requirements for VS/VD switch is still proportional to the feedback delay-bandwidth product of the upstream VS/VD loop.

This demonstrates that VS/VD can be helpful in limiting buffer requirements in various segments of a connection, and can isolate network segments from each other.

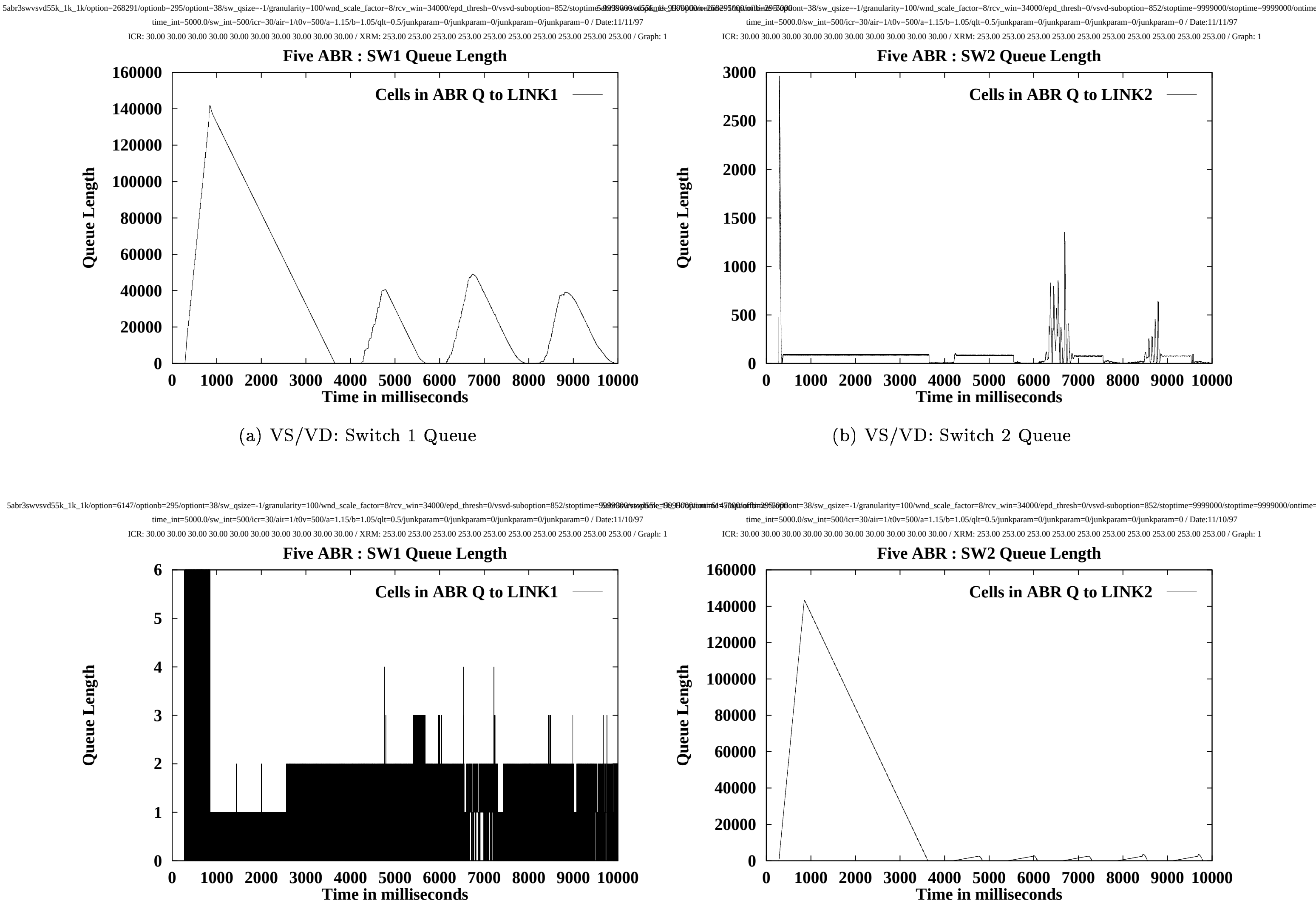
6 Summary and Future Work

In this contribution, we have examined a few basic issues in designing VS/VD feedback control mechanisms. VS/VD can effectively isolate nodes in different VS/VD loops. As a result, the buffer requirements of a node are bounded by the feedback delay-bandwidth product of the upstream VS/VD loop. However, improper design of VS/VD rate allocation schemes can result in an unstable condition where the switch queues do not drain.

A VS/VD switch port has a per-VC queue for each VC that goes through the port. These per-VC queues drain into a single class queue for the ABR class. A server at the head of each queue 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 upstream server. When providing feedback, each server should only allocate upto the rate at which it is allowed to drain. During queue build up due to congestion, the server should only allocate a part of its output rate to the previous hop so that its queue can drain quickly.

We have presented a per-VC rate allocation mechanism for VS/VD switches based on ERICA+. This scheme retains the basic properties of ERICA+ (max-min fairness, high link utilization, and controlled queues), and isolates VS/VD control loops thus limiting the buffer requirements in each loop. The scheme has been tested for infinite ABR and persistent TCP sources.

We have shown that VS/VD, when implemented correctly, helps in reducing the buffer requirements of terrestrial



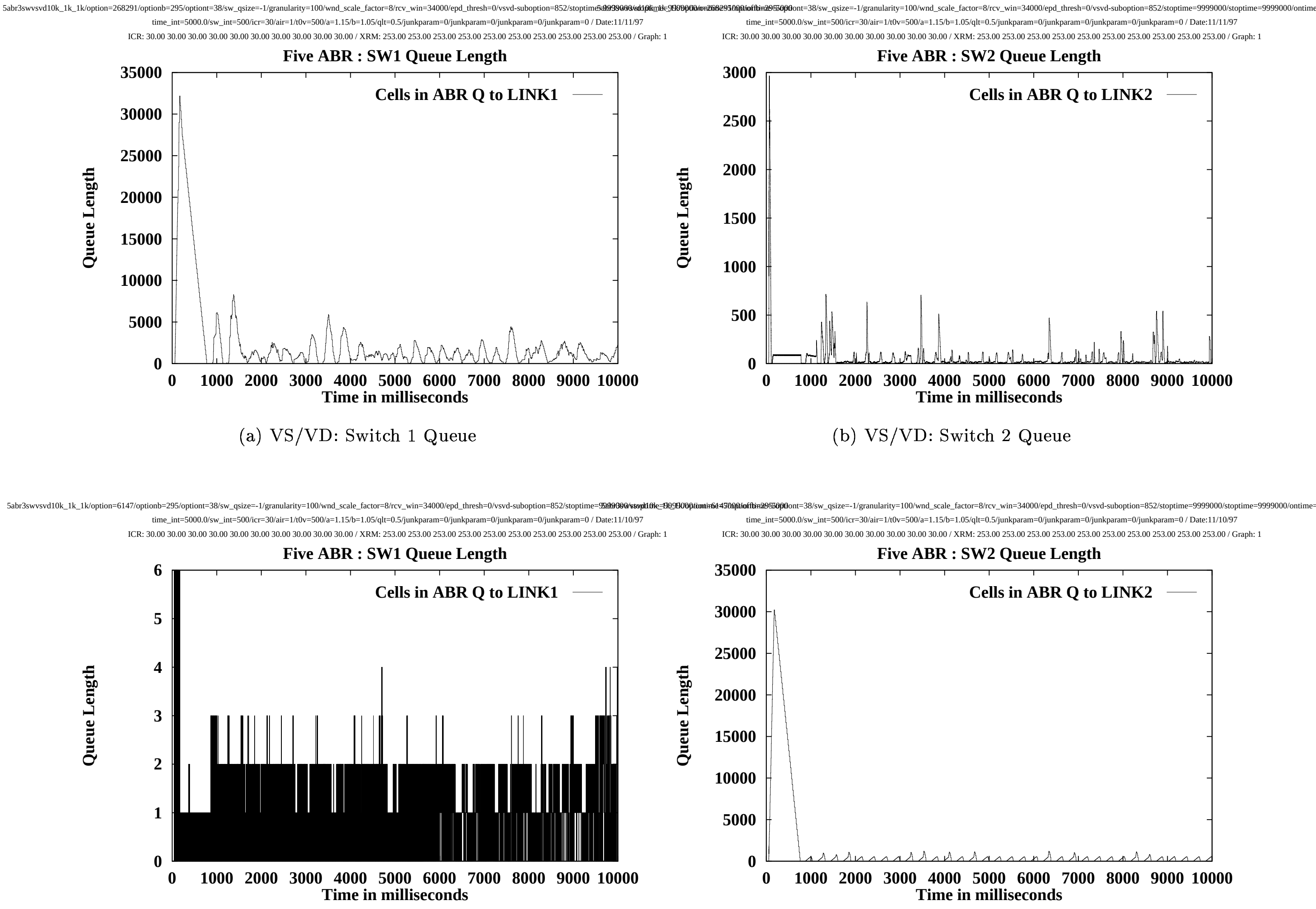
(a) VS/VD: Switch 1 Queue

(b) VS/VD: Switch 2 Queue

(c) Non-VS/VD: Switch 1 Queue

(d) Non-VS/VD: Switch 2 Queue

Figure 7: Switch Queue Length for VS/VD and non-VS/VD:GEO



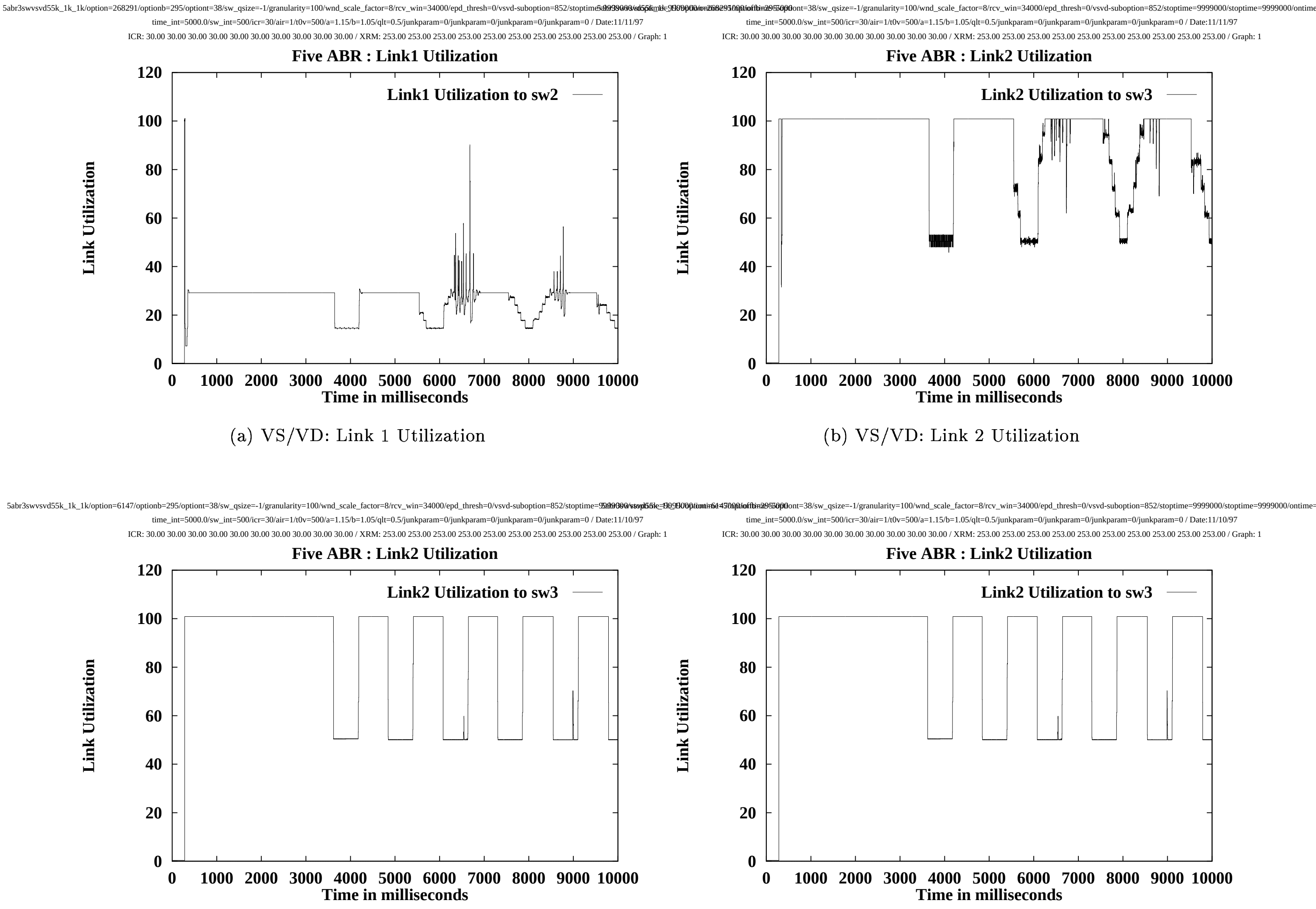
(a) VS/VD: Switch 1 Queue

(b) VS/VD: Switch 2 Queue

(c) Non-VS/VD: Switch 1 Queue

(d) Non-VS/VD: Switch 2 Queue

Figure 8: Switch Queue Length for VS/VD and non-VS/VD Case: LEO



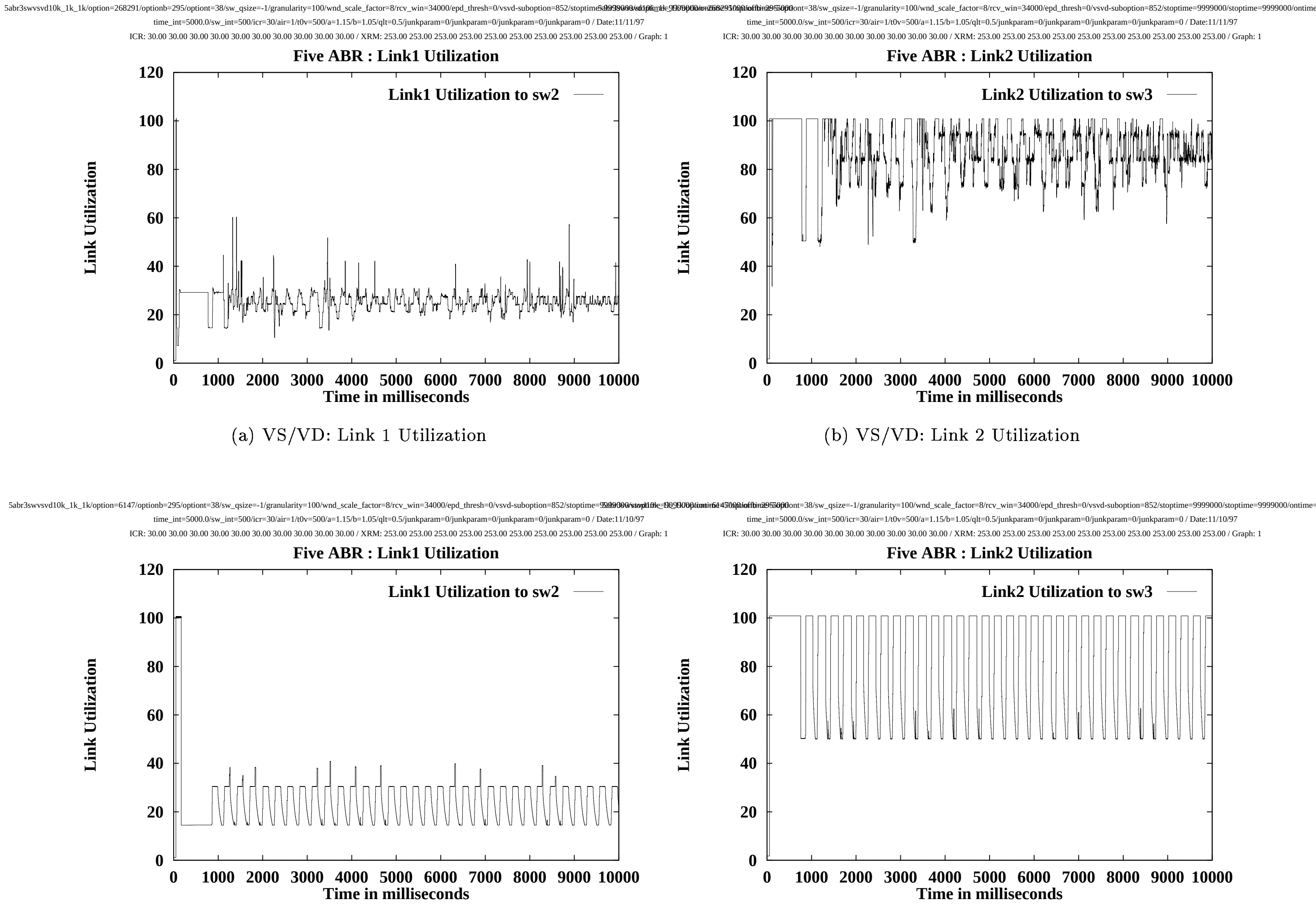
(a) VS/VD: Link 1 Utilization

(b) VS/VD: Link 2 Utilization

(c) Non-VS/VD: Link 1 Utilization

(d) Non-VS/VD: Link 2 Utilization

Figure 9: Link Utilizations for VS/VD and non-VS/VD:GEO



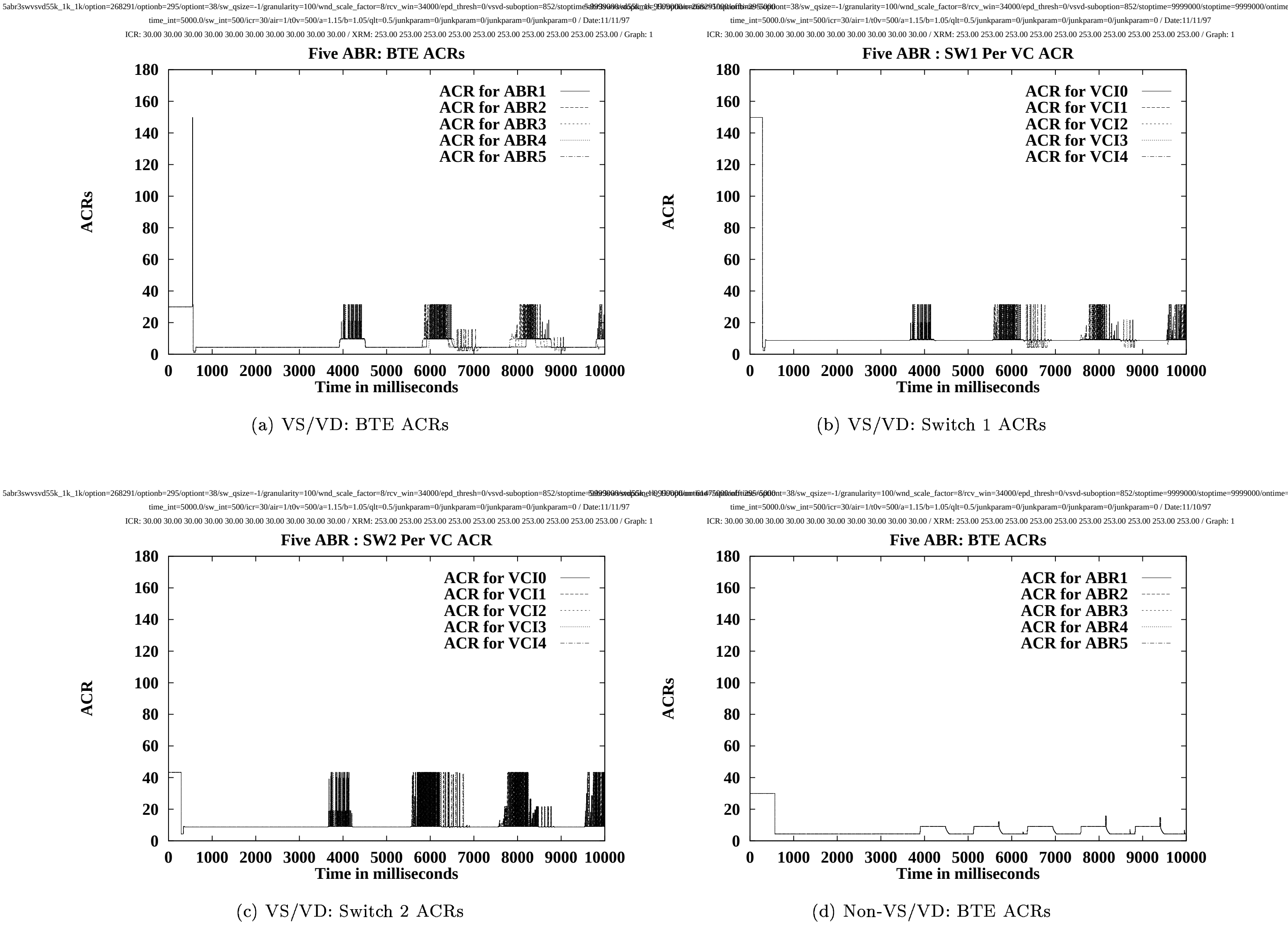
(a) VS/VD: Link 1 Utilization

(b) VS/VD: Link 2 Utilization

(c) Non-VS/VD: Link 1 Utilization

(d) Non-VS/VD: Link 2 Utilization

Figure 10: Link Utilizations for VS/VD and non-VS/VD: LEO



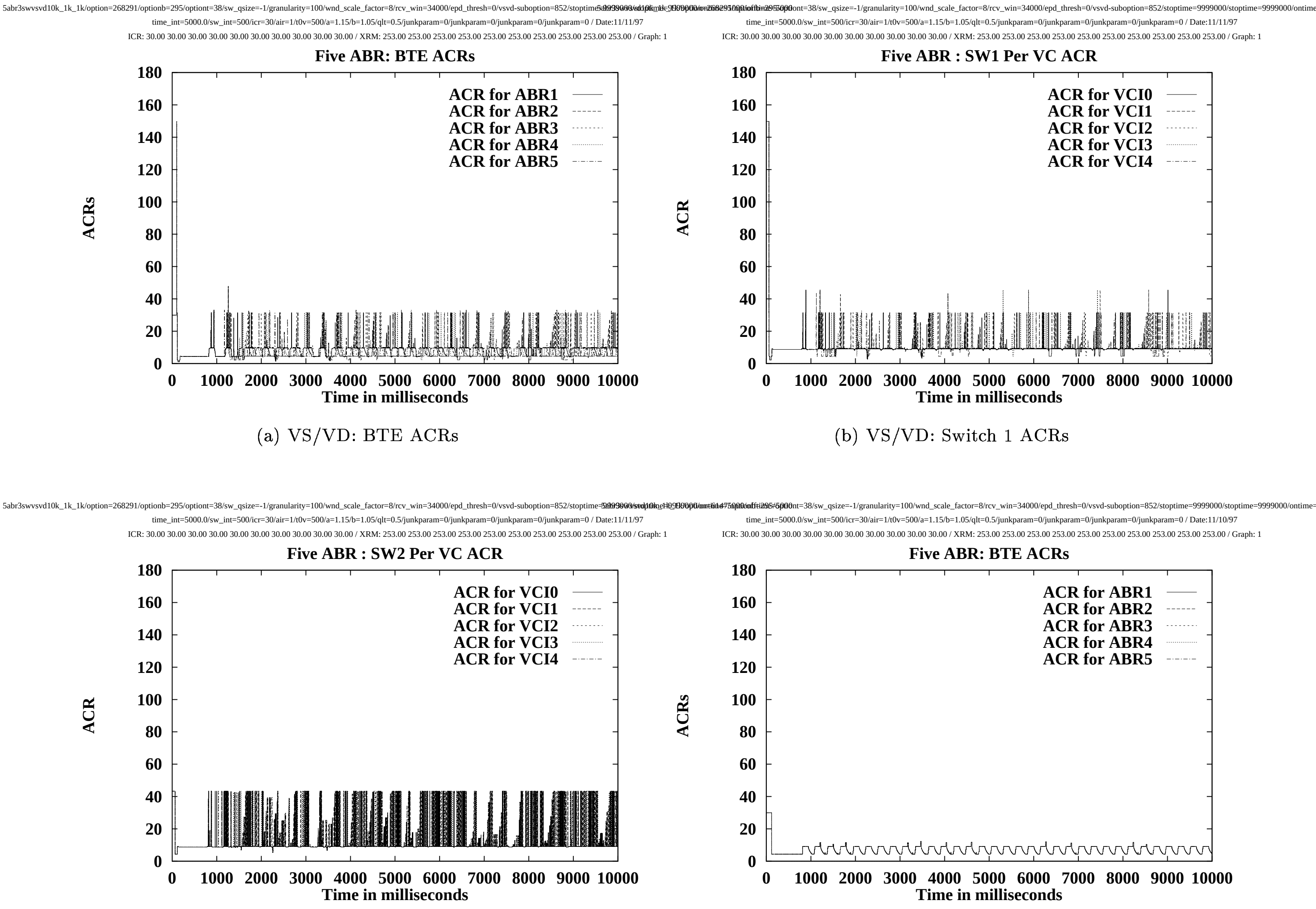
(a) VS/VD: BTE ACRs

(b) VS/VD: Switch 1 ACRs

(c) VS/VD: Switch 2 ACRs

(d) Non-VS/VD: BTE ACRs

Figure 11: ACRs for VS/VD and non-VS/VD:GEO



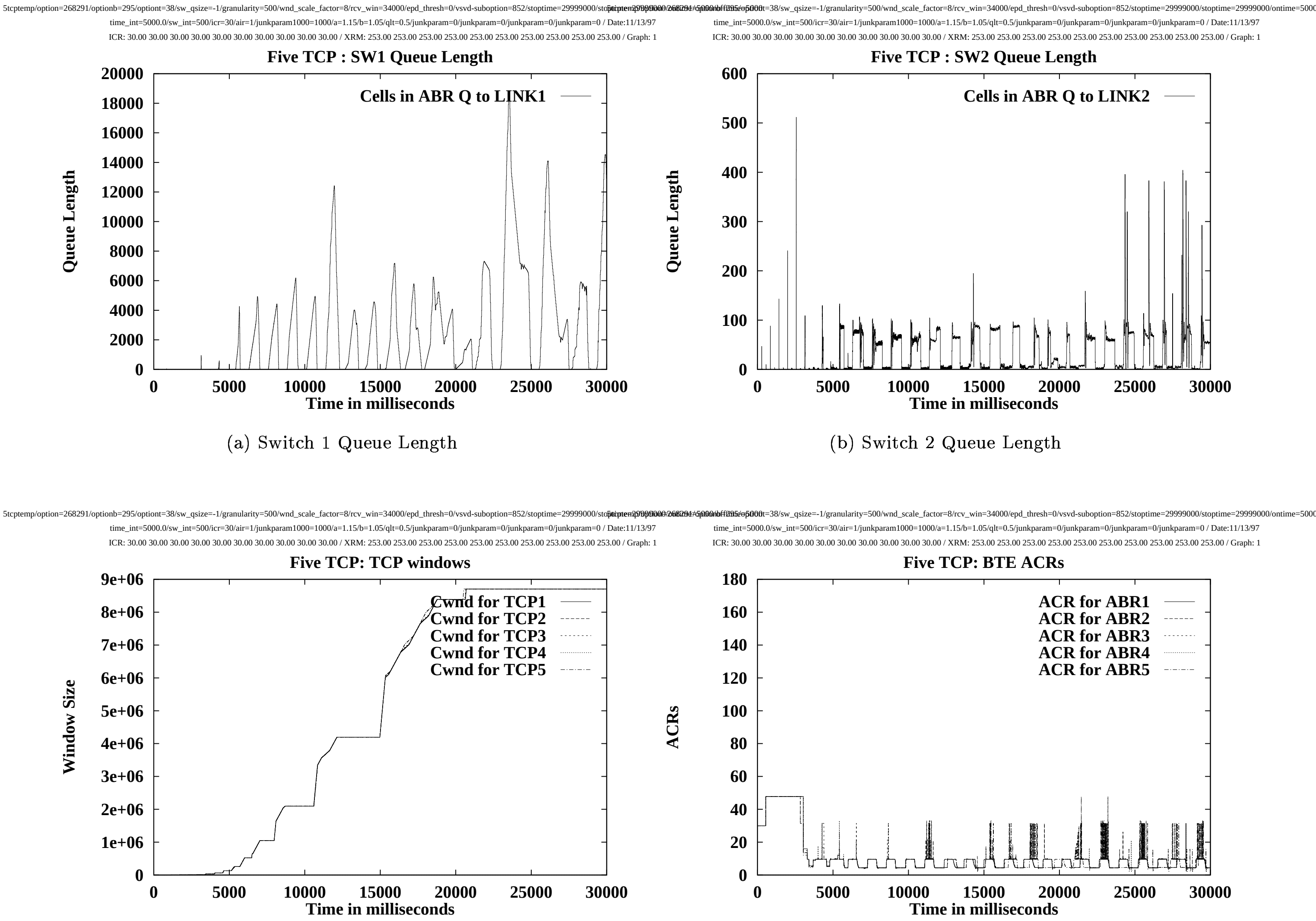
(a) VS/VD: BTE ACRs

(b) VS/VD: Switch 1 ACRs

(c) VS/VD: Switch 2 ACRs

(d) Non-VS/VD: BTE ACRs

Figure 12: ACRs for VS/VD and non-VS/VD Case:LEO



(a) Switch 1 Queue Length

(b) Switch 2 Queue Length

(c) TCP Cwnd

(d) BTE ACRs

Figure 13: TCP Configuration with VS/VD

switches that are connected to satellite gateways. Without VS/VD, terrestrial switches that are a bottleneck, must buffer cells of upto the feedback delay-bandwidth product of the entire control loop (including the satellite hop). *With a VS/VD loop between the satellite and the terrestrial switch, the queue accumulation due to the satellite feedback delay is confined to the satellite switch. The terrestrial switch only buffers cells that are accumulated due to the feedback delay of the terrestrial link to the satellite switch.*

Some additional work needs to be done to fully test the performance of the scheme presented in this contribution. In particular, the performance of the scheme in the presence of VBR background traffic needs to be analyzed. Various other configurations like the parking-lot, upstream and the GFC-2 configuration need to be tested. Also, any VS/VD implementation is typically more complex than a non-VS/VD ABR switch. The additional complexity arises due to the per-VC queuing and implementation of the source rules. The complexity of the scheme needs to be analyzed in detail.

7 Motion

Appendix A in this contribution should be added to the baseline text as
I.5.4 A Sample Explicit Rate VS/VD Switch Algorithm

8 Appendix A

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 correspinding downstream server. When providing feedback, each server only allocates upto 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.

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.

References

[1] ATM Forum, “ATM Traffic Management Specification Version 4.0,” April 1996, available as ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps

[2] Raj Jain, Shiv Kalyanaraman, Rohit Goyal, Sonia Fahmy, “Source Behavior for ATM ABR Traffic Management: An Explanation,” *IEEE Communications Magazine*, November 1996[3].

[3] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanathan, “The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks, Part I: Description,” IEEE Transactions on Networking, submitted.

[4] Shivkumar Kalyanaraman, Raj Jain, Jianping Jiang, Rohit Goyal, and Sonia Fahmy, Seong-Cheol Kim, ”Virtual Source/Virtual Destination (VS/VD): Design Considerations,” ATM Forum/96-1759, December 1996.

[3] All our papers and ATM Forum contributions are available through http://www.cis.ohio-state.edu/˜jain