Internet Engineering Task Force
Differentiated Services Working Group
Internet Draft
Expires Aug, 2000
draft-ietf-diffserv-ba-vw-00.txt

Van Jacobson
Kathleen Nichols
Kedar Poduri
Cisco Systems, Inc.
March, 2000

The ‘Virtual Wire’ Behavior Aggregate <draft-ietf-diffserv-ba-vw-00.txt>

Status of this Memo

Abstract

This document describes an edge-to-edge behavior called ‘Virtual Wire’ (VW) that can be constructed in any domain supporting the diffserv EF PHB plus appropriate domain ingress policers. The VW behavior is essentially indistinquishable from a dedicated circuit and can be used anywhere it is desired to replace dedicated circuits with IP transport.

A pdf version of this document is available at ftp://ftp.ee.lbl.gov/papers/vw_ba.pdf

1.0 Introduction

[RFC2598] describes a diffserv PHB called expedited forwarding (EF) intended for use in building a scalable, low loss, low latency, low jitter, assured bandwidth, end-to-end service that appears to the endpoints like an unshared, point-to-point connection or 'virtual wire.'[1] For scalability, a diffserv domain supplying this service must be completely unaware of the individual endpoints using it and sees instead only the aggregate EF marked traffic entering and transiting the domain. This document provides the specifications necessary on that aggregated traffic (in diffserv terminology, a *behavior aggregate* or BA) in order to meet these requirements and thus defines a new BA, the *Virtual Wire behavior aggregate* or *VW BA*. Despite the lack of per-flow state, if the aggregate input rates are appropriately policed and the EF service rates on interior links are appropriately configured, the edge-to-edge service supplied by the domain will be indistinguishable from that supplied by dedicated wires between the endpoints. This note gives a quantitative definition of what is meant by 'appropriately policed and configured'.

Loss, latency and jitter are all due to the queues traffic experiences while transiting the network. Therefore providing low loss, latency and jitter for some traffic aggregate means ensuring that the packets of the aggregate see no (or very small) queues. Queues arise when short-term traffic arrival rate exceeds departure rate at some node(s). Thus ensuring no queues for some aggregate is equivalent to bounding rates such that, at every transit node, the aggregate's maximum arrival rate is less than that aggregate's minimum departure rate.

Creating the VW BA has two parts:

1. Configuring nodes so that the aggregate has a well-defined minimum departure rate. (`Well-defined' means independent of the dynamic state of the node. In particular, independent of the intensity of other traffic at the node.)
2. Conditioning the aggregate (via policing and shaping) so that it's arrival rate at any node is always less than that node's configured minimum departure rate.

[RFC2598] provides the first part. This document describes how one configures the EF PHBs in the *collection* of nodes that make up a DS domain and the domain's boundary traffic conditioners (described in [RFC2475]) to provide the second part. This description results in a diffserv behavior aggregate, as described in [BADEF].

The next sections describe the VW BA in detail and give examples of how it might be implemented. The keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", and "MAY" that appear in this document are to be interpreted as described in [RFC2119].

1. This service has also been called Premium service [RFC2638] and 'virtual leased line' (VLL). In the absence of the definitions supplied in this document, these terms have been (ab)used in ways that sometimes strayed far from the authors' intent. To minimize confusion with these various interpretations, we decided to choose a new name.

2.0 Description of the Virtual Wire BA

2.1 Applicability

A Virtual Wire (VW) BA is intended to send “circuit replacement” traffic across a diffserv network. That is, this BA is intended to mimic, *from the point of view of the originating and terminating nodes*, the behavior of a hard-wired circuit of some fixed capacity. It does this in a scalable (aggregatable) way that doesn’t require ‘per-circuit’ state to exist anywhere but the ingress router adjacent to the originator. This BA should be suitable for any packetizable traffic that currently uses fixed circuits (e.g., telephony, telephone trunking, broadcast video distribution, leased data lines) and packet traffic that has similar delivery requirements (e.g., IP telephony or video conferencing).

2.2 Rules

Each node in the domain MUST implement the EF PHB as described in section 2 of [RFC2598] but with the SHOULDs of that section taken as MUSTs. The bandwidth limit of each output interface SHOULD be configured as described in Section 2.4 of this document. In addition, each domain boundary input interface that can be the ingress for EF marked traffic MUST strictly police that traffic as described in Section 2.4. Each domain boundary output interface that can be the egress for EF marked traffic MUST strictly shape that traffic as described in Section 2.4.

2.3 Characteristics

“The same as a wire.” As long as packets are sourced at a rate <= the virtual wire’s configured rate, they will be delivered with a high degree of assurance and with almost no distortion of the interpacket timing imposed by the source. However, any packets sourced at a rate greater than the VW configured rate, measured over any time scale longer than a packet time at that rate, will be unconditionally discarded.

2.4 Parameters

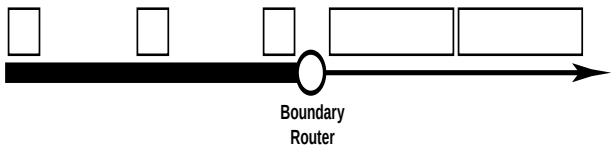

Figure 1: Time structure of packets of a CBR stream at a high to low bandwidth transition

Figure 1 shows a CBR stream of size *S* packets being sourced at rate *R*. At the domain egress border router, the packets arrive on a link of bandwidth *B* (= *nR*) and depart to their destination on a link of bandwidth *R*.

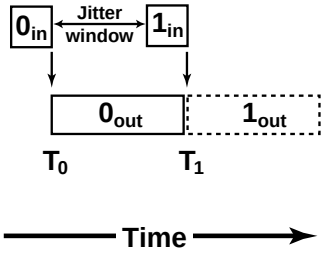

Figure 2: Details of arrival / departure relationships

Figure 2 shows the detailed timing of events at the router. At time T_0 the last bit of packet 0 arrives so output is started on the egress link. It will take until time $T_1 = T_0 + (S/R)$ for packet 0 to be completely output. As long as the last bit of packet 1 arrives at the border router before T_1, the destination node will find the traffic indistinguishable from a stream carried the entire way on a dedicated wire of bandwidth *R*. This means that packets can be *jittered* or displaced in time (due to queue waits) as they cross the domain and that there is a *jitter window* at the border router of duration

$$\Delta = \frac{S}{R} - \frac{S}{B} = \frac{S}{R} \times \left(1 - \frac{1}{n}\right) \qquad \text{(EQ 1)}$$

that bounds the sum of all the queue waits seen be a packet as it transits the domain. As long as this sum is less than Δ, the destination will see service identical to a dedicated wire.[1]

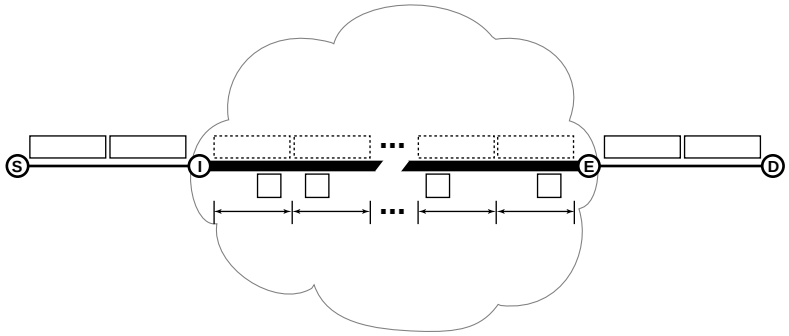

Figure 3: Packet timing structure edge-to-edge

Figure 3 shows the edge-to-edge path from the source to the destination. The links from *S* to *I* and *E* to *D* run at the virtual wire rate *R* (or the traffic is shaped to rate *R* if the links run at a higher rate). The solid rectangles on these links indicate the packet time *S/R*. The dotted lines carry the packet times across the domain since the time boundaries of these virtual packets form the jitter window boundaries of the actual packets (whose duration and spacing are shown by the solid rect-

1. Note that the jitter window is (implicitly) computed relative to the first packet of the flight of packets departing the boundary router and, thus, can only include *variable* delays. Any transit delay experienced by all the packets, be it propagation time, router forwarding latency, or even long-term average queue waits, gets removed by the relative measure so the sum described in this paragraph should only include terms that vary over the duration of the flight.

angles below the intra-domain link). Note that each packet's jitter is independent. E.g., even though the two packets about to arrive at *E* have been displaced in opposite directions so that the total time between them is almost 2Δ, neither has gone out of its jitter window so the output from *E* to *D* will be smooth and continuous.

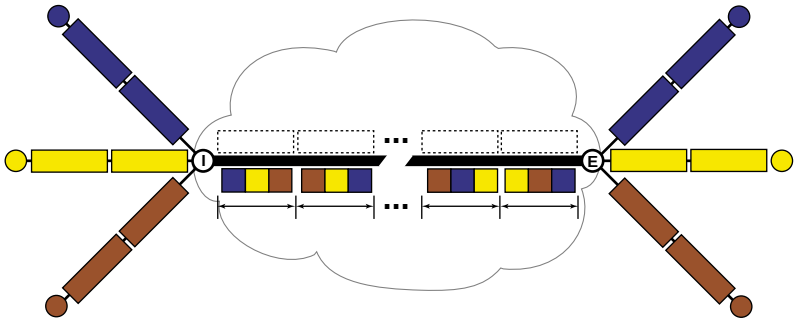

Figure 4: Three VW customers forming an aggregate

This jitter independence is what allows multiple 'virtual wires' to be transparently aggregated into a single VW BA. Figure 4 shows three independent VW customers, blue, yellow and red, entering the domain at *I.* Assume that their traffic has worst-case phasing, i.e., that one packet from each stream arrives simultaneously at *I.* Even if the output link scheduler makes a random choice of which packet to send from its EF queue, no packet will get pushed outside its jitter window. For example, in Figure 4 node *I* ships a different perturbation of the 3 customer aggregate in every window yet this has no effect on the edge-to-edge VW properties).

The jitter independence means that we only have to compare the jitter bound of Equation 1 to the worst case of the total queue wait that can be seen by a single VW packet as it crosses the domain. There are three potential sources of queue wait for a VW packet:

1. it can queue behind non-EF packets (if any)
2. it can queue behind another VW packet from the same customer
3. it can queue behind VW packet(s) from other customers

For case (1), the EF 'priority queuing' model says that the VW traffic will never wait while a non-EF queue is serviced so the only delay it can experience from non-EF traffic is if it has to wait for the finish of a packet that was being sent at the time it arrived. For an output link of bandwidth B, this can impose a worst-case delay of *S/B.* Note that this implies that if the (low bandwidth) links of a network are carrying both VW and other traffic, then *n* in Equation 1 must be at least 2 (i.e., the EF bound can be at most half the link bandwidth) in order to make the jitter window large enough to absorb this delay.[1]

Case (2) can only happen if the previous packet hasn't completely departed at the time the next packet arrives. Since the incoming VW stream is strictly shaped to a rate *R*, any two packets will be separated by at least time *S/R* so having leftovers is equivalent to saying the departure rate on

1. Several authors have confused this limit on the EF bandwidth with 'over provisioning'. The limit actually has nothing to do with provisioning but is a consequence of the fact that the link scheduler is non-pre-emptive at the packet level and *any* service scheme that wants to bound jitter on mixed traffic must include a similar limit.

some link is $<R$ over this time scale. But the EF property is precisely that the departure rate MUST be $>R$ over any time scale of S/R or longer so this can't happen for any legal VW/EF configuration. Or, to put it another way, if case (2) happens, either the VW policer is set too loosely or some link's EF bound is set too tight.

Case (3) is a simple generalization of (2). If there are a total of n customers, the worst possible queue occurs if all n arrive simultaneously at some output link. Since each customer is individually shaped to rate R, when this happens then *no* new packets from any stream can arrive for at least time S/R so having leftovers is equivalent to a departure rate $< nR$ over this time scale. But the EF property for any link capable of handling the aggregate traffic is that the departure rate be $> nR$ over any time scale longer than $S/(nR)$ so, again, this can't happen in any legal VW/EF configuration.

For case (1), a packet could be displaced by non-EF traffic once per hop so the edge-to-edge jitter is a function of the path length. But this isn't true for case (3): The strict ingress policing implies that a packet from any given VW stream can meet any other VW stream in a queue at most once. This means the worst case jitter caused by aggregating VW customers is a linear function of the number of customers in the aggregate but completely independent of topology.

2.5 Assumptions

The topology independence of VW service actually holds only while routing is relatively stable. Since packets can be duplicated while routing is converging, and since path lengths can be shorter after a routing change, it is possible to violate the VW traffic bounds and thus jitter stream(s) more than their jitter window for a small time during and just after a routing change.

2.6 Example uses

Say an ISP wants to carry RTP encapsulated telephony traffic in addition to data traffic. Assume that want to retain all the robustness of IP (re-)routing which is equivalent to saying that all traffic can show up on any link. This implies that the lowest bandwidth backbone link constrains the total number of calls that can be carried. If the smallest backbone link is OC-3 and each call generates at most a 200 byte packet every 20ms, then the total number of VW customers that can be admitted into the backbone must be less than

$$\frac{155 \times 10^6}{200 \times 8 / 0.02} = 1938$$

(For comparison, an OC-3 TDM telco trunk could admit 2421 customers so there is a 25% bandwidth penalty paid for the ability to efficiently mix voice and data.) Since this limit does not depend on the topology, these call slots can be assigned to customers, either statically or dynamically, in any way that doesn't violate the VW/EF bound on the customer tail circuits.

Note that each call looks like two 'customers', one each direction, so this overly simple bound is actually less than half the capacity of an equivalent telco system. If one adds the topological assumption that none of the simplex traffic streams between two endpoints will ever travel both directions over the same link, then the number VW customers becomes 1938 each direction so the domain has roughly the same telephony capacity as an equivalent telco system.

2.7 Environmental concerns

Routing instability will generally translate directly into VW service degradation.

The analysis in Section 2.4 would hold in a world where traffic policers and link schedulers are perfect and mathematically exact. When computing parameters for our world, 5-10% fudge factors should be used.

3.0 Security Considerations

There are no security considerations for the VW BA other than those associated with the EF PHB which are described in [RFC2598].

4.0 References

[RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S.Bradner, www.ietf.org/rfc/rfc2119

[RFC2474] “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, K.Nichols, S. Blake, F. Baker, D. Black, www.ietf.org/rfc/rfc2474.txt

[RFC2475] “An Architecture for Differentiated Services”, S. Blake, D. Black, M.Carlson,E.Davies,Z.Wang,W.Weiss, www.ietf.org/rfc/rfc2475.txt

[RFC2597] “Assured Forwarding PHB Group”, F. Baker, J. Heinanen, W. Weiss, J. Wroclawski, ftp://ftp.isi.edu/in-notes/rfc2597.txt

[RFC2598] “An Expedited Forwarding PHB”, V.Jacobson, K.Nichols, K.Poduri, ftp://ftp.isi.edu/in-notes/rfc2598.txt

[RFC2638] “A Two-bit Differentiated Services Architecture for the Internet”, K. Nichols, V. Jacobson, and L. Zhang, www.ietf.org/rfc/rfc2638.{txt,ps}

[BADEF] “Definition of Differentiated Services Behavior Aggregates and Rules for their Specification”, K.Nichols, B.Carpenter, draft-ietf-diffserv-ba-def-00.[txt, pdf]

[CAIDA] The nature of the beast: recent traffic measurements from an Internet backbone. K Claffy, Greg Miller and Kevin Thompson. http://www.caida.org/Papers/Inet98/index.html

[NS2] The simulator ns-2, available at: http://www-mash.cs.berkeley.edu/ns/.

[FBK] K. Nichols, “Improving Network Simulation with Feedback”, Proceedings of LCN’98, October, 1998.

[RFC2415] RFC 2415, K. Poduri and K. Nichols, “Simulation Studies of Increased Initial TCP Window Size”, September, 1998.

5.0 Authors' Addresses

Van Jacobson
Cisco Systems, Inc.
170 W. Tasman Drive
San Jose, CA 95134-1706
van@cisco.com

Kathleen Nichols
Cisco Systems, Inc.
170 W. Tasman Drive
San Jose, CA 95134-1706
kmn@cisco.com

Kedar Poduri
Cisco Systems, Inc.
170 W. Tasman Drive
San Jose, CA 95134-1706
poduri@cisco.com

6.0 Appendix: On Jitter for the VW BA

The VW BA's bounded jitter translates into the generally useful properties of network bandwidth limits and buffer resource limits. These properties make VW useful for a variety of statically and dynamically provisioned services, many of which have no intrinsic need for jitter bounds. IP telephony is an important application for the VW BA where expected and worst-case jitter for rate-controlled streams of packets is of interes; thus this appendix is primarily focused on voice jitter. The appendix focuses on jitter for individual flows aggregated in a VW BA, derives worst-case bounds on the jitter, and gives simulation results for jitter.

6.1 Jitter and Delay

The VW BA is sufficiently restrictive in its rules to preserve the required EF per-hop behavior under aggregation. These properties also make it useful as a basis for Internet telephony, to get low jitter and delay. Since a VW BA will have link arrival rates that do not exceed departure rates over fairly small time scales, end-to-end delay is based on the transmission time of a packet on a wire and the handling time of individual network elements and thus is a function of the number of hops in a path, the bandwidth of the links, and the properties of the particular piece of equipment used. Unless the end-to-end delay is excessive due to very slow links or very slow equipment, it is usually the jitter, or variation of delay, of a voice stream that is more critical than the delay.

We derive the worst case jitter for a a VW BA in a DS domain using it to carry a number of rate-controlled flows. Jitter is defined as the absolute value of the difference between the arrival time difference of two adjacent packets and their departure time difference, that is:

$$\text{jitter} = |(a_k - a_j) - (d_k - d_j)| \quad \text{(EQ 2)}$$

The maximum jitter will occur if one packet waits for no other packets at any hop of its path and the adjacent packet waits for the maximum amount of packets possible. There are two sources of jitter, one from waiting for other EF packets that may have accumulated in a queue due to simultaneous arrivals of EF packets on several input lines feeding the same queue and another from waiting for non-EF packets to complete. The first type is strictly bounded by the properties of the VW BA and the EF PHB. The second type is minimized by using a Priority Queuing mechanism to schedule the output link and giving priority to EF packets and this value can be approached by using a non-bursty weighted round-robin packet scheduler and giving the EF queue a large weight. The total jitter is the sum of these two.

Maximum jitter will be given across the domain in terms of T, the virtual packet time or cyle time. It is important to recall the analysis of section 3.0 showing that this *jitter across the DS domain* is completely invisible to the end-to-end flow using the VW BA if it is within the jitter window at the egress router.

6.1.1 Jitter from other VW packets

The jitter from meeting other packets of the VW aggregate comes from (near) simultaneous arrival of packets at different input ports all destined for the same output queue that can be completely rearranged at the next packet arrival to that queue. This jitter has a strict bound which we will show here.

It will be helpful to remember that, from RFC 2598, a BA using the EF PHB will get its configured share of each link at all timescales above the time to send an MTU at the rate corresponding to that share of that link.

Focus on the DS domain of figure 5. Unless otherwise stated, in this section assume M Boundary Routers, each having N inputs and outputs. We assume that each of the BR's ingress ports receives a flow of EF-marked packets that are being policed to a peak rate R. If each flow sends a fixed size packet, then it's possible to calculate the fixed time, T, between packets of each of these MxN flows that enters the DS domain, a quite reasonable assumption for packets carrying voice. For example, assume a domain traversed by MxN flows of 68 byte voice packets sent at 20 ms time intervals. Note we assume all ingress links have some packets that will be marked for the VW aggregate. Thus the total number of ingress EF-marked streams to the VW aggregate is I = MxN.

To construct a network where the maximum jitter can occur, a single flow traversing the network must be able to meet packets from all the other flows marked for the EF PHB and it should be possible to meet them in the worst possible way.

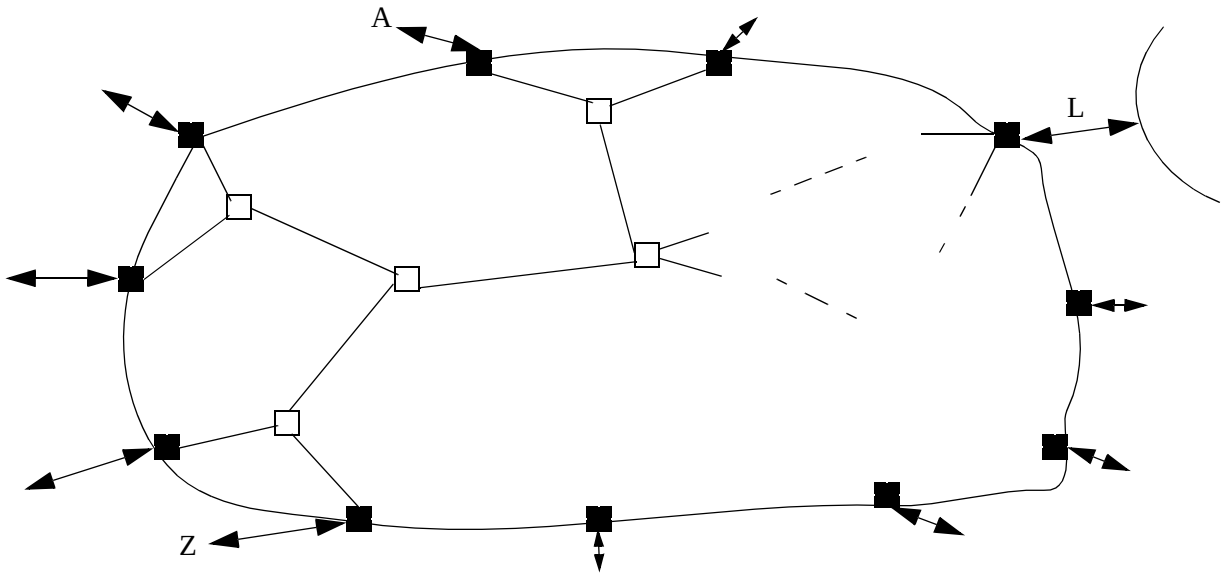

Figure 5: A DS domain

Unless otherwise stated, assume that all the routers of the domain have N inputs and outputs and that all links have the same bandwidth B. Although there are a number of ways that the individual streams from different egress ports might combine in the interior of the network, in a properly configured network, the arrival rate of the VW BA must not exceed the departure rate at any network node. Consider a particular flow from A to Z and how to ensure that packets entering the VW BA at A meet every other flow entering the domain from all egress points as they traverse the domain to Z. Consider three cases: the first is a single bottleneck, the second makes no assumptions about routing in the network and the third assumes that the paths of individual flows can be completely specified.

Assume there are H hops from A to Z and that **delay** is the minimum time it takes for a packet to go from A to Z in the absence of queuing. Both packets experience **delay** and thus it subtracts in the jitter calculation. Recall that the packets of the flow are separated in time by T, then (normalizing to a start time of 0):

$$d_j = 0 \quad \text{(EQ 3)}$$

$$d_k = T \quad \text{(EQ 4)}$$

$$a_j = \text{delay} \quad \text{(EQ 5)}$$

$$a_k = \text{time spent waiting behind all other packets} + \text{delay} + T \quad \text{(EQ 6)}$$

Then we can use:

$$\text{jitter} = \text{time spent waiting behind all other packets} \quad \text{(EQ 7)}$$

as we explore calculating worst case jitter for different topologies.

The next step is to establish some useful relationships between parameters. First, assume that some fraction, f, of a link's capacity is allocated to EF-marked packets. Since we are assuming that all the flows that are admitted into this DS domain's VW aggregate generate packets at a spacing of T, this can be expressed in time as fxT. Then the amount of time to send an EF packet on each link can be written as fxT/(total number of EF-marked flow crossing the link). Note that f should be less than 0.5 in order that an MTU-sized non-EF packet will not cause the EF condition to be violated.

6.1.1.1 Worst case jitter in a network with a dumbbell bottleneck

Consider a DS domain topology shown in figure 6. In order for a packet of the (A,Z) flow to arrive behind packets of all the other flows, a packet from each ingress must arrive at each of the M border routers at the same time and must be transmitted to the interior router's queue for the bottleneck link B at the same time. Further the links between the border routers and the bottleneck router must be enough larger than B that the packets are still sitting in B's queue when our (A,Z) packet arrives at the end of a burst of N packets, that is L > NxB. Then the

$$\text{jitter}_{\text{worst case}} = \text{MxNx(time to send an EF packet on B)} \qquad \text{(EQ 8)}$$

Since we expressed the EF aggregate's allocation on B as fxB, the time to send an EF packet on B is fxT/(MxN), so

$$\text{jitter}_{\text{worst case}} = \text{fxT} \qquad \text{(EQ 9)}$$

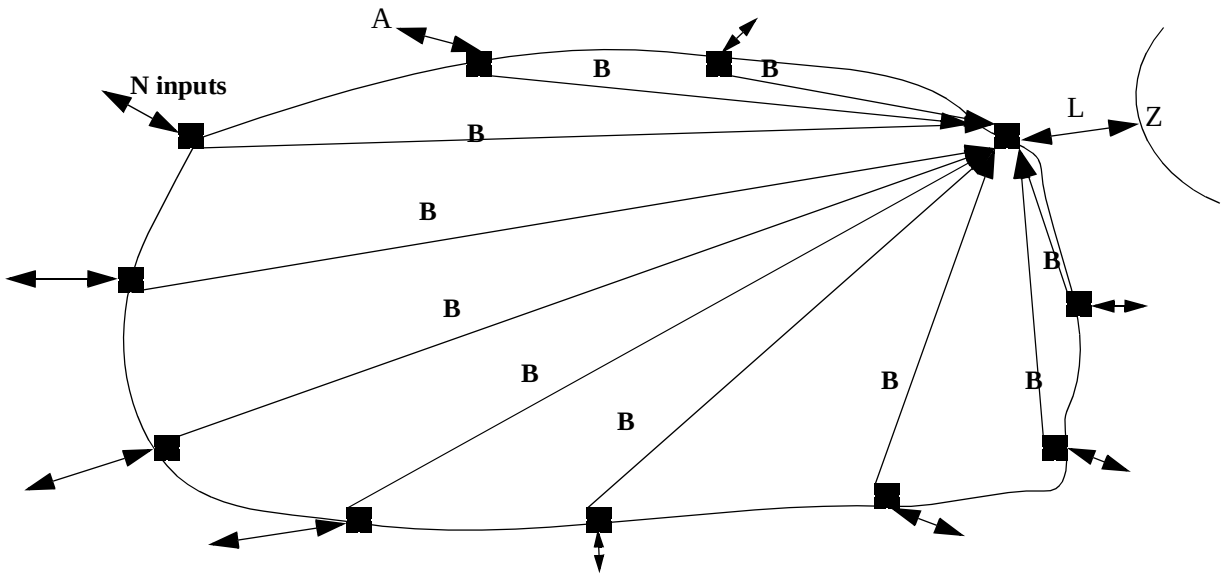

Figure 6: A dumbbell bottleneck

This result shows that the worst case jitter will be less than half a packet time for any VW-compliant allocation on this topology. For the worst case to occur, all N packets must arrive simultaneously at all M border routers. By assuming independence, an interested person should be able to get some insight on the likelihood of this happening. Simulation results in a later section will show this.

6.1.1.2 Worst case jitter in an arbitrary network

Consider the network of figure 5 and in this case, one packet of the (A,Z) flow must arrive at the same time, but be queued behind a packet from each of the other flows as it passes through the network. This can happen at any link of the network and even at the same link. In this case, assume all links have bandwidth B but we don't know the path the individual EF packets or flows of the aggregate will follow. Then the worst case jitter is

$$\text{jitter}_{\text{worst case}} = Ix(fxT/I) = fxT \qquad \text{(EQ 10)}$$

the same as the bottleneck case. Note that, in allocation, if it is possible to know that not all flows of the aggregate will take the same path, then one could allocate each link to a smaller number of flows, but this would also imply that the number of flows that it's possible to meet and be jittered by is smaller. Allocation can be kept to under 0.5 times the bandwidth of a core link, while the existence of multiple paths offers both fault tolerance and an expectation that the actual load on any link will be less than 0.5.

How likely is this case to happen? One packet of the (A,Z) flow must encounter and queue behind every other individual shaped flow that makes up the domain's VW aggregate as it crosses the domain.

6.1.1.3 Maximal jitter in a network with "pinned" paths per flow

Then at each hop the (A,Z) packet has to arrive at the same time as an EF packet from the (N-1) other inputs and the (A,Z) packet has to be able to end up anywhere within that burst of N packets. In particular, for two adjacent packets of the (A,Z) flow, one must arrive at the front of every hop's burst and the other at the end of every hop's burst. This clearly requires an unrealistic form of path pinning or route selection by every individual EF-marked flow entering the DS domain. This uni-directional path is shown in figure 7 where all routers have N inputs and at each of the H routers on the path from A to Z, N-1 flows are sent to other output queues, while N-1 of the shaped input flows that have not yet crossed the A to Z path enter the router at the other input ports.

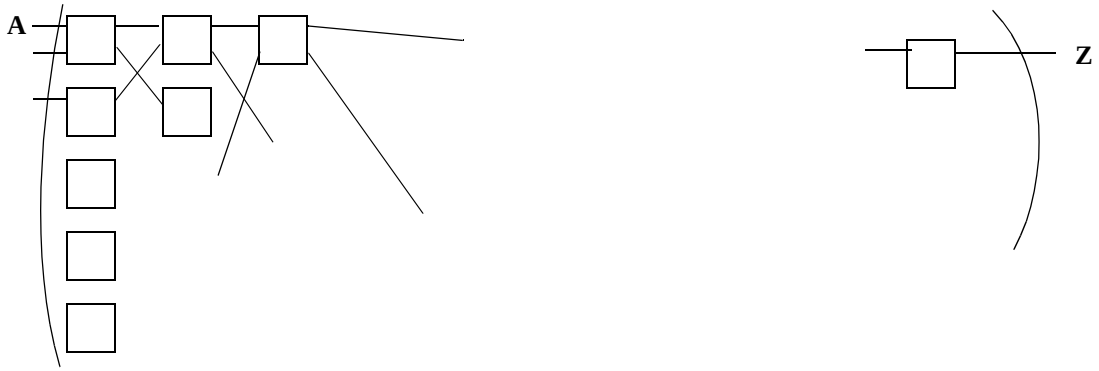

Figure 7: Example path for maximal jitter across DS domain from A to Z

It should be noted that if the number of hops from A to Z is not large enough, it won't be possible for one of its packets to meet all the other shaped flows and if the number of hops is larger than what's required there won't be any other shaped flows to meet there. For the flow from A to B to meet every other ingress stream as it traverses a path of H hops:

$$Hx(N-1) = MxN - 1 \quad \text{(EQ 11)}$$

then compute the maximum jitter as:

$$\text{jitter} = Hx(N-1)x(\text{time to send an EF packet on each link}) \quad \text{(EQ 12)}$$

If the total number of ingress streams exceeds Hx(N-1) + 1, then it's not possible to meet all the other streams and the maximum jitter is

$$\text{jitter}_{\text{worst case}} = H \times (N-1) \times fT/(\text{number of ingress-shaped EF flows on each link}) \quad \text{(EQ 13)}$$

Otherwise the max jitter is

$$\text{jitter}_{\text{worst case}} = (MxN - 1) \times fT/(\text{number of ingress-shaped EF flows on each link}) \quad \text{(EQ 14)}$$

Then the maximum jitter depends on the number of hops or the number of border routers. In this construction, the number of ingress-shaped EF flows on each link is N, thus:

$$\text{jitter}_{\text{worst case}} < \text{smaller of } (HxfT, MxfxT) \quad \text{(EQ 15)}$$

Dividing out T gives jitter in terms of the number of ingress flow cycle times (or virtual packet times). Then, for the jitter to exceed the cycle time (or 20 ms for our VoIP example),

$$fxH > 1 \text{ and } fxM > 1 \quad \text{(EQ 16)}$$

If f were at its maximum of 0.5, then it appears to be easy to exceed a cycle time of jitter across a domain. However, it's useful to examine what f might typically be. Note that for this construction:

$$f= NxR/B \quad \text{(EQ 17)}$$

For our example voice flows, a reasonable R is 28-32 Kbps. Then, for a link of 128 Kbps, f = 0.25xN; for 1.5 Mbps, f = 0.02xN; for 10 Mbps, f = 0.003xN; for 45 Mbps, f = 0.0007xN; and for 100 Mbps, f = 0.0003xN. Then such a network of DS3 links can handle almost 1500 individual shaped flows at this rate. Another way to look at this is that the hop count times the number of ingress ports of each router must exceed the link bandwidth divided by the VoIP rate in order to have a *maximum* jitter of a packet time at the VoIP rate.

$$HxN > B/R \quad \text{(EQ 18)}$$

For a network of all T1 links, this becomes HxN > 50 and for larger bandwidth links, it increases.

Suppose that the ingress flows are not the same rate. If the allocation, f, is at its maximum, then this means the number of ingress flows must decrease. For example, if the A to Z flow is 10xR, then it will meet 9 fewer packets as it traverses the network. Even though the assumptions behind this case are not realistic, we can see that the jitter can be kept to a reasonable amount. The rules of the EF PHB and the VW BA should make it easy to compute the worst case jitter for any topology and allocation.

6.1.1.4 Achievablility of the maximum

Now that we've examined how to compute the worst case jitter, we look at how likely it is that this worst case happens and how it relates to the jitter window.

In addition to the topological and allocation assumptions that were made in order to allow a flow to have the opportunity of meeting every other flow, events must align so that the meeting actually happens at each hop. If we could assume independence of the timing of each flow's arrival within an interval of T, then that probability is on the order of $(fT/N)^N$ For this to happen at every hop we need the joint probability of this happening at all H nodes. Further we need the joint probability of that event in combination with an adjacent packet not meeting any other packets. For each additional hop, the number of ways the packets can combine increases exponentially, thus the probability of that particular worst case combination decreases.

6.1.1.5 Jitter from non-VW packets

The worst case occurs when one packet of a flow waits for no other packets to complete and the adjacent packet arrives at every hop just as an MTU-sized non-EF packet has begun transmission. That worst case jitter is the sum of the times to send MTU-sized packets at the link bandwidth of all the hops in the path or, for equal bandwidth paths,

$$\text{jitter} = HxMTU/B \qquad \text{(EQ 19)}$$

Note that if one link has a bandwidth much smaller than the others, that term will dominate the jitter.

If we assume that the MTU is on the order of 10-20 times the voice packet size in our example, then the time to send an MTU on a link is 10 or 20 times fxT/N so that our jitter bound becomes 20 x H x fxT/N.

What has to happen in order to achieve the worst case? For jitter against the default traffic, one packet waits for no default traffic and the adjacent packet arrives just as an MTU of the default type begins transmission on the link.

The worst case is linear in the number of hops, but since the joint probability of an EF packet arriving at each queue precisely at the start of a non-EF packet on the link decreases in hop count, measured or simulated jitter will be seen to grow as a negative exponential of the number of hops in a path, even at very high percentiles of probability. The reason for this is that the number of ways that the packets can arrive at the EF queue grows as p^H so the probability is on the order of p^{-H}. When the link bandwidth is small, it may be necessary to fragment non-EF packet to control jitter.

How should we relate jitter in terms of source cycle times or virtual packet times to the jitter window defined in section 3.0? Note that we can write

$$\text{jitter window} = Sx(1/R - 1/((nxR)/f)) \qquad \text{(EQ 20)}$$

and noting that T = S/R, we get:

$$\text{jitter window} = Tx(n-f/n) \qquad \text{(EQ 21)}$$

So that, in many cases, the jitter window can be approximated by T.

6.2 Quantifying Jitter through Simulation

Section 1 derived and discussed the worst-case jitter for individual flows of a diffserv behavior aggregate (BA) based on the EF PHB. We showed that the worst case jitter can be bounded and calculated these theoretical bounds. The worst case bounds represent possible, but not likely, values. Thus, to get a better feel for the likely worst jitter values, we used simulation.

We use the ns-2 network simulator; our use of this simulator has been described in a number of documents [NS2,FBK,RFC2415]. The following subsections describe the simulation set-up for these particular experiments.

6.2.1 Topology

Figure 8 shows the topology we used in the simulations. A and Z are edge routers through which traffic from various customers enters and exits the Diffserv cloud. We vary the topology within the Diffserv cloud to explore the worst-case jitter for EF traffic in various scenarios. Jitter is measured on a flow or set of flows that transit the network from A to Z. To avoid per hop synchronizations, half the DE traffic at each hop is new to the path while half of the DE traffic exits the path. For the mixed EF and DE simulations, half the EF flows go from A to Z while, at each hop, the other half of the 10% rate only crosses the path at that hop. As discussed in section 1, this is an unlikely construction but we undertake it to give a more pessimistic jitter For the EF-only simulations, we emulate the case analyzed in section 1.1.3 by measuring jitter on one end to end flow and having (N-1) new EF flows meet that flow at every hop. Note that N is determined by the maximum number of 28 kbps flows that can fit in the EF share of each link, so N=share x bandwidth/28 kbps.

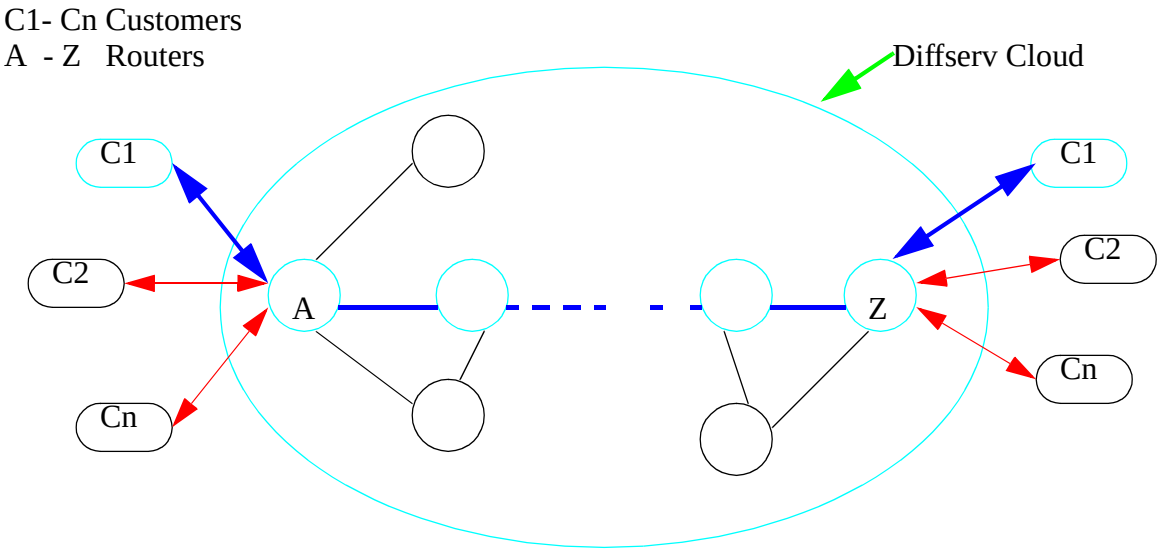

Figure 8: Simulated topology

6.2.2 Traffic

Traffic is generated to emulate G.729 voice flows with packet size (B) of 68 Bytes and a 20 ms packetization rate. The resultant flows have a rate of 27 Kbps. As previously discussed, jitter experienced by the voice flows has two main components; jitter caused by meeting others flows in the EF queue, and jitter due to traffic in other low prioirity classes. To analyze the first component, we vary the multiplexing level of voice flows that are admitted into the DS domain and for the second, we generate data traffic for the default or DE PHB. Since we are interested in exploring the worst case jitter, data traffic is generated as long-lived TCP connections with 1500 Byte MTU segments. Current measurements show real Internet traffic consists of a mixture of packet sizes, over 50% of which are minimum-sized packets of 40 bytes and over 80% of which are much smaller than 1500 Bytes [CAIDA]. Thus a realistic traffic mix would only improve the jitter that we see in the simulations.

6.2.3 Schedulers and Queues

All the nodes(routers) in the network have the same configuration: a simple Priority Queue (PQ) scheduler with two queues. Voice traffic is queued in the high priority queue while the data traffic is queued in the queue with the lower priority. The scheduler empties all the packets in the high priority queue before servicing the data packets in a lower priority queue. However, if the scheduler is busy servicing a data packet at the time of arrival of a voice packet, the voice packet is served only after the data packet is serviced completely, i.e., the scheduler is non-preemptive. For priority queuing where the low priority queue is kept congested, simulating two queues is adequate.

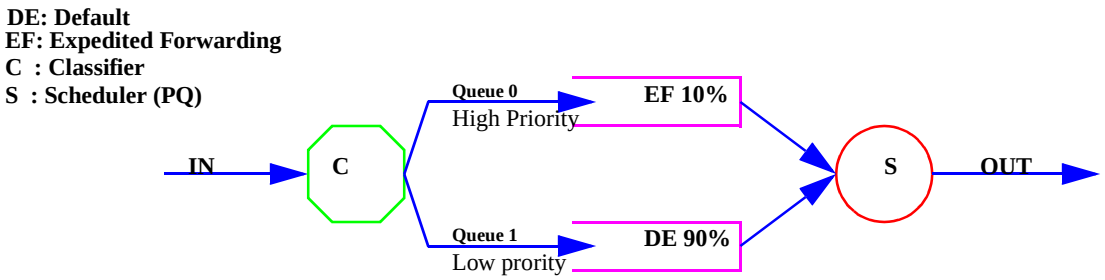

Figure 9: Link scheduling in the simulations

6.2.4 Results

In the following simulations, three bandwidth values were used for the DS domain links: 1.5 Mbps, 10 Mbps, and 45 Mbps. Unless otherwise stated, the aggregate of EF traffic was allocated 10% of the link bandwidth. The hops per path was varied from 1 to 24. Then, the 1.5 Mbps links can carry about 5 voice flows, the 10 Mbps about 36 voice flows, and 45 Mbps about 160.

6.2.4.1 Jitter due to other voice traffic only

To see the jitter that comes only from meeting other EF-marked packets, we simulated voice traffic only. Figure 10 shows results from 10 Mbps links with 10% of the link share assigned to the voice flows. For a single bottleneck link in a dumbbell, the worst case jitter possible for this scenario is 2 ms. Note that the values in Figure 10 are far less than that. This source of jitter is quite

small, particularly compared to the jitter from traffic in other queue(s) as we will see in the next section. (Note: Figure 10's results are quite preliminary. Further simulations will be performed that jitter the individual sources slightly.)

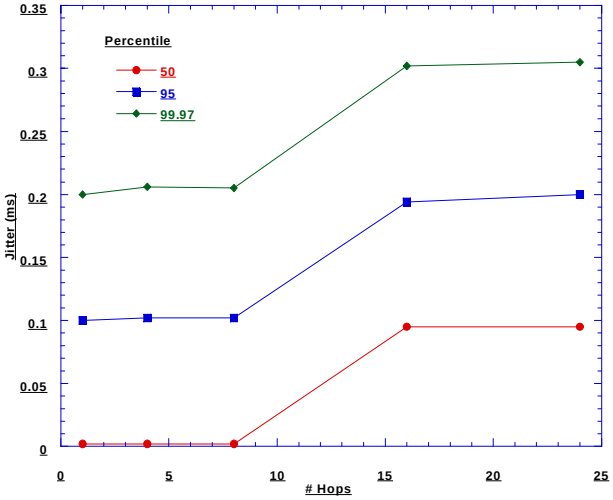

Figure 10: Jitter from meeting other EF-marked traffic

6.2.4.2 Jitter in a voice flow where there is a congested default class

Our traffic model for the DE queue keeps it full with mostly 1500 byte packets. From section 1, the worst case jitter is equal to the number of hops times the time to transmit a packet at the link rate. The likelihood of this worst case occuring goes down exponentially in hop count, and the simulations confirm this. Figure 11 shows several percentiles of the jitter for 10 Mbps links where the time to transmit an MTU at link speed is 1.2 ms.

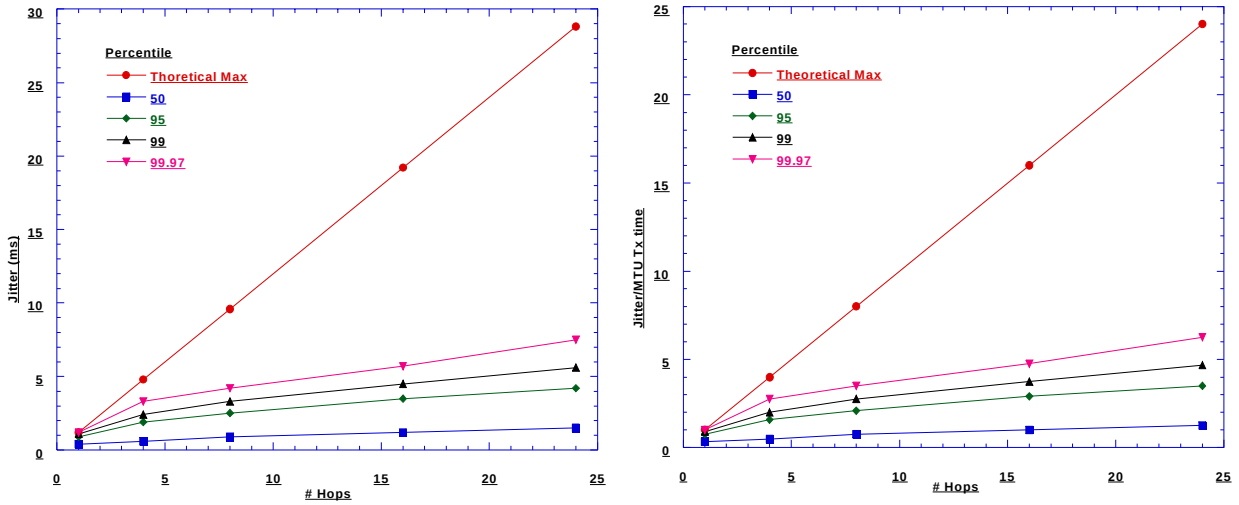

Figure 11: Various percentile jitter values for 10 Mbps links and 10% allocation

Recall that the period of the voice streams is 20 ms and note that, the jitter does not even reach half a period. The median jitter gets quite flat with number of hops. Although the higher percentile values increase at a somewhat higher rate with number of hops, it still does not approach the calculated worst case. The data is also shown normalized by the MTU transmission time at 10

Mbps. Now the vertical axis value is the number of MTU sized packets of jitter that the flow experiences. This normalization is presented to make it easier to relate the results to the analysis, though it obscures the impact (or lack thereof) of the jitter on the 20 ms flows.

Figure 12 shows the same results for 1.5 Mbps links and Figure 13 for 45 Mbps links.

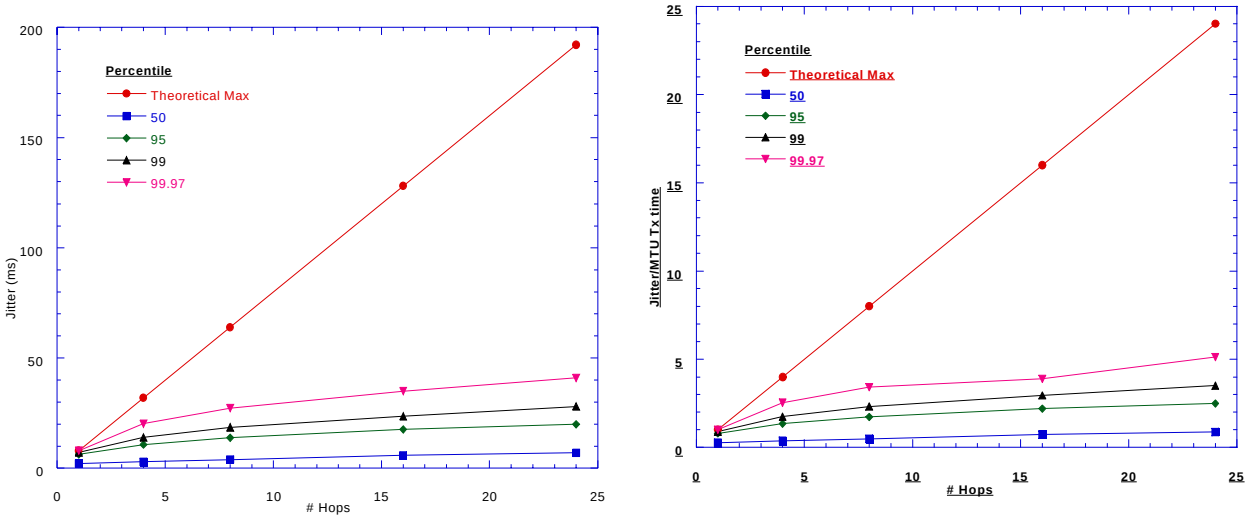

Figure 12: Various percentiles of jitter for 1.5 Mbps links and 10% share

Notice that the worst case jitter for the 1.5 Mbps link is on the order of two cycle times while, for 45 Mbps, it is less than 10% of the cycle time. However, the behavior in terms of number of MTUs is similar.

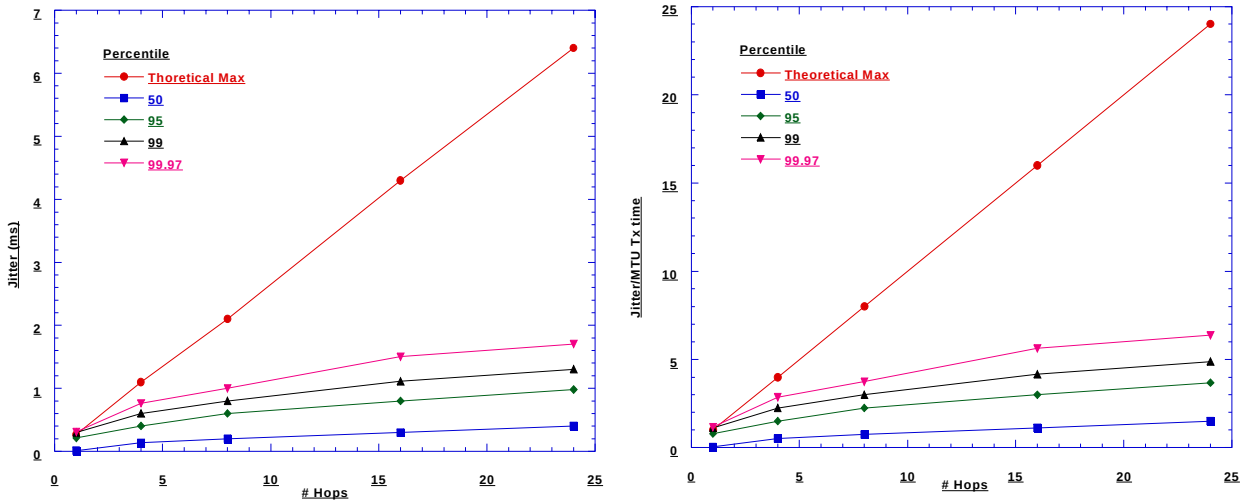

Figure 13: Various percentiles of jitter for 45 Mbps links and 10% share

The jitter in time and thus as a fraction of the virtual packet time of the flow being measured clearly increases with decreasing bandwidth. Even the smallest bandwidth, 1.5 Mbps can handle nearly all jitter with a jitter buffer of 2 packets. The two higher bandwidths don't even jiter by one virutal packet time, thus staying within the jitter window. Figures 14, 15, and 16 compare the median, 99th percentile and 99.97th percentile (essentially the worst case). It's also interesting to normalize the results of each experiment by the MTU transmission time at that link bandwidth.

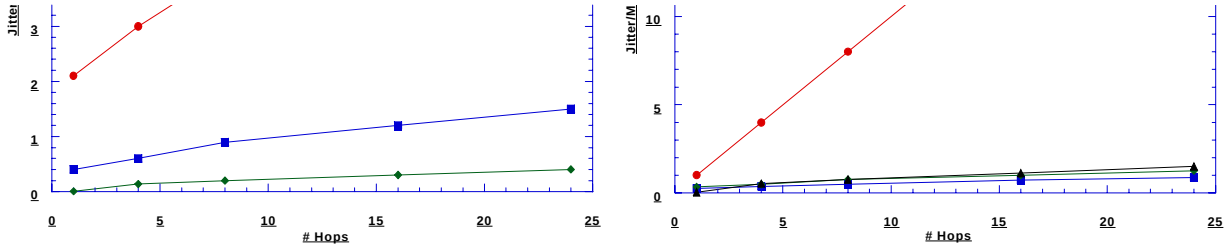

Figure 14: Median jitter for all three bandwidths by time and normalized

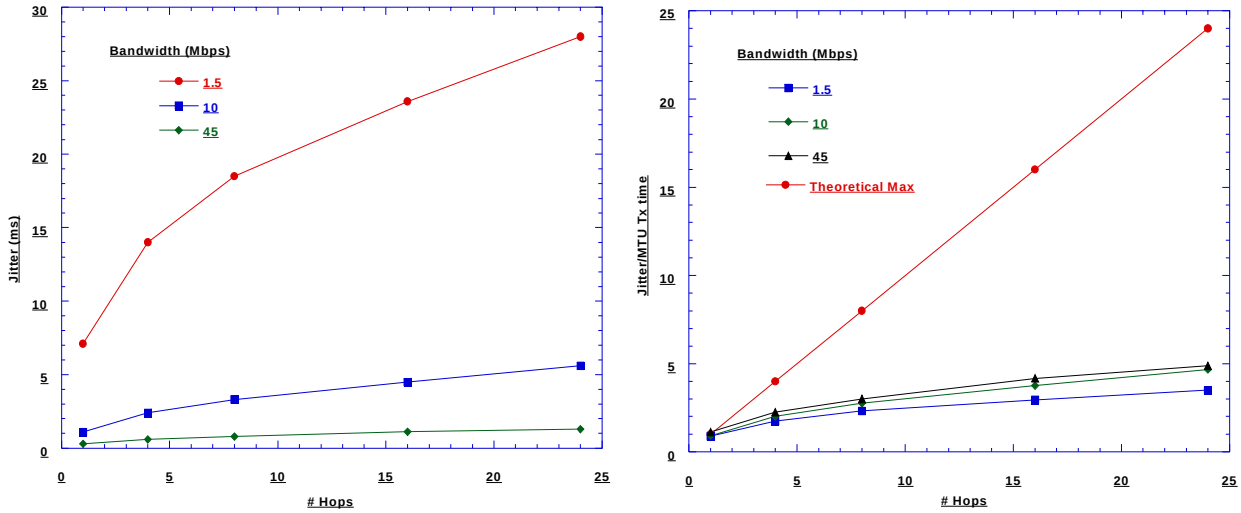

Figure 15: 99th percentile of jitter for the three bandwidths; absolute time and normalized

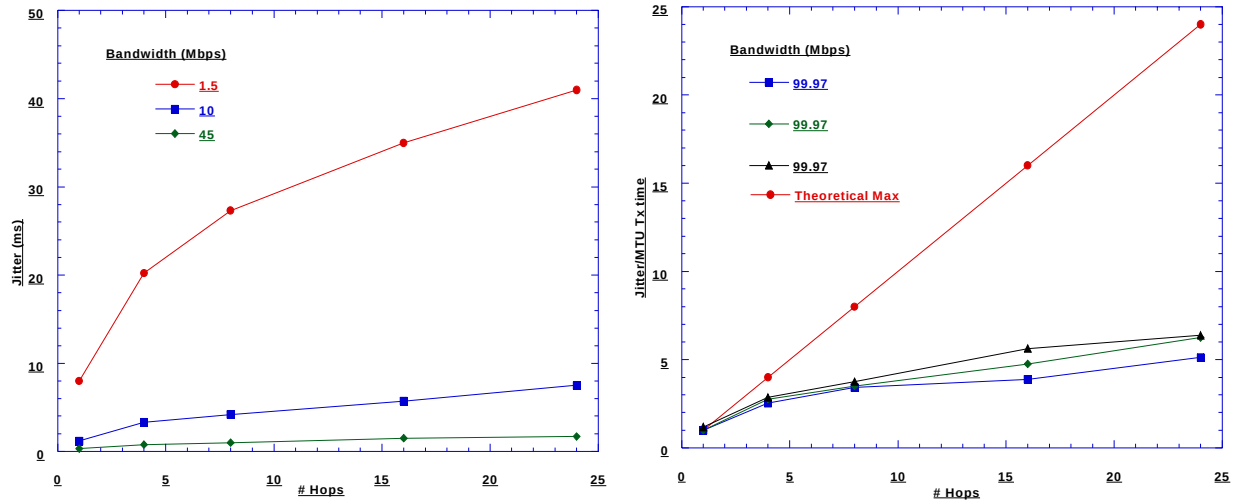

Figure 16: 99.97th percentile of jitter; absolute time and normalized by MTU transmission times

The simulation experiments are not yet complete, but they clearly show the probability of acheiveing the worst case jitter decreasing with hop count and show that jitter can be controlled. The normalization shows that the jitter *behavior* is the same regardless of bandwidth. The absolute times differ by scale factors that depend on the bandwith.

6.2.4.3 Jitter with an increased allocation

In the following, the experiments of the last section are repeated, but using a 20% link share, rather than a 10% link share Figure 17 shows the jitter percentiles for 10 Mbps links and a 20%

share. The values are also plotted with the 10% share results (on the right hand side) to show how similar they are..

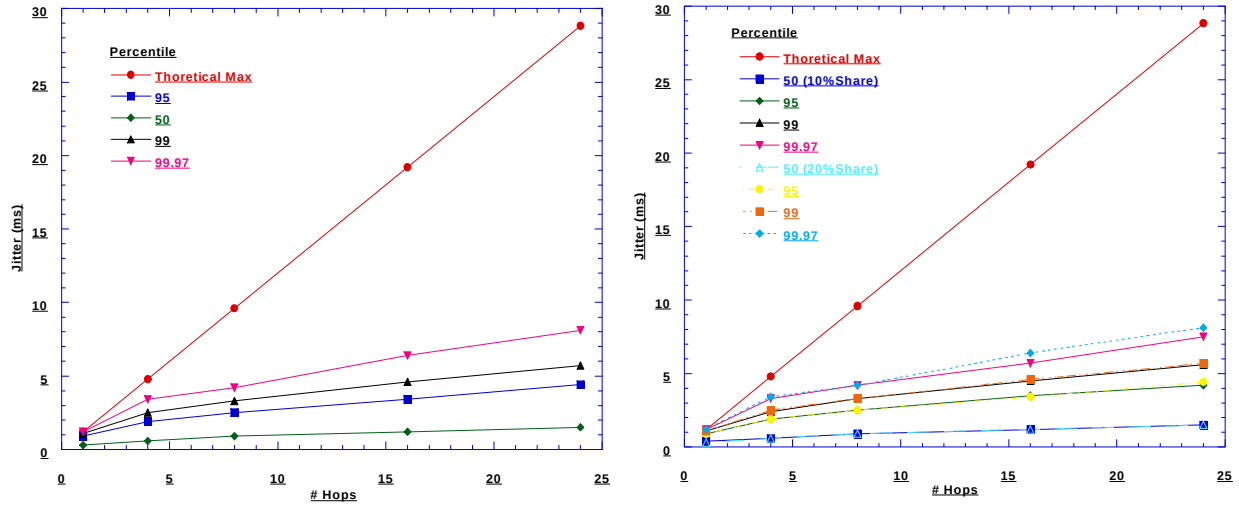

Figure 17: Jitter percentiles for 10 Mbps links and 20% EF share

Using the previous section, we would believe that the results for other bandwidths would have the same shape, but be scaled by the bandwidth difference. Figure 18 shows this to indeed be the case. Thus it is sufficient to simulated only a single bandwidth.

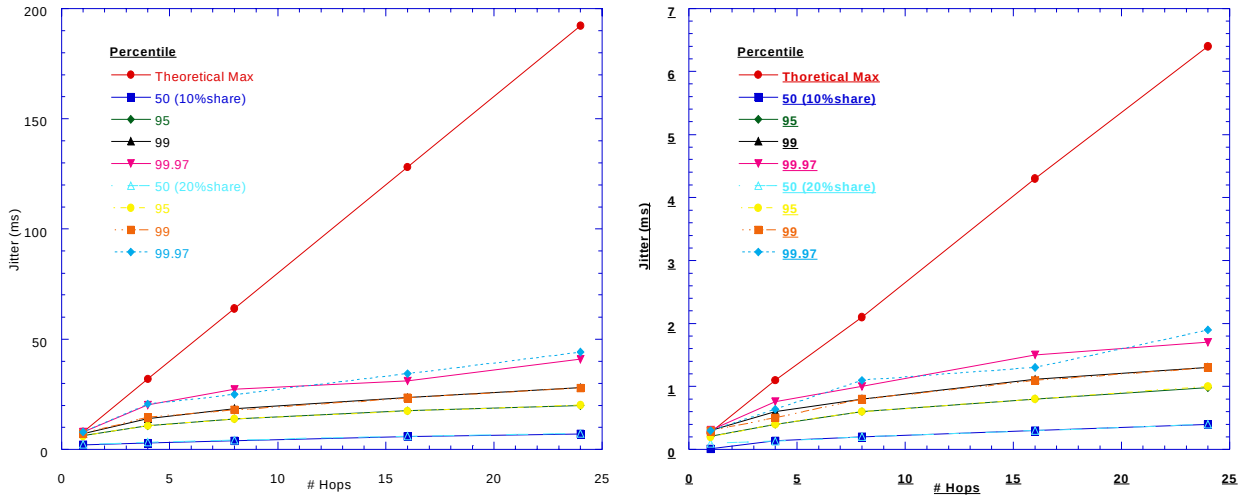

Figure 18: Jitter for 1.5 Mbps links (on left) and 45 Mbps links (on right)

In all the experiments, it can be clearly seen that the shape of the jitter vs. hops curve flattens because the probability of the worst case occuring at each hop decreases exponentially in hops. To

see if there is an allocation level at which the jitter behavior diverges, we simulated and show results for allocations of 10, 20, 30, 40, and 50 percent, all for 10 Mbps links in figure 19.

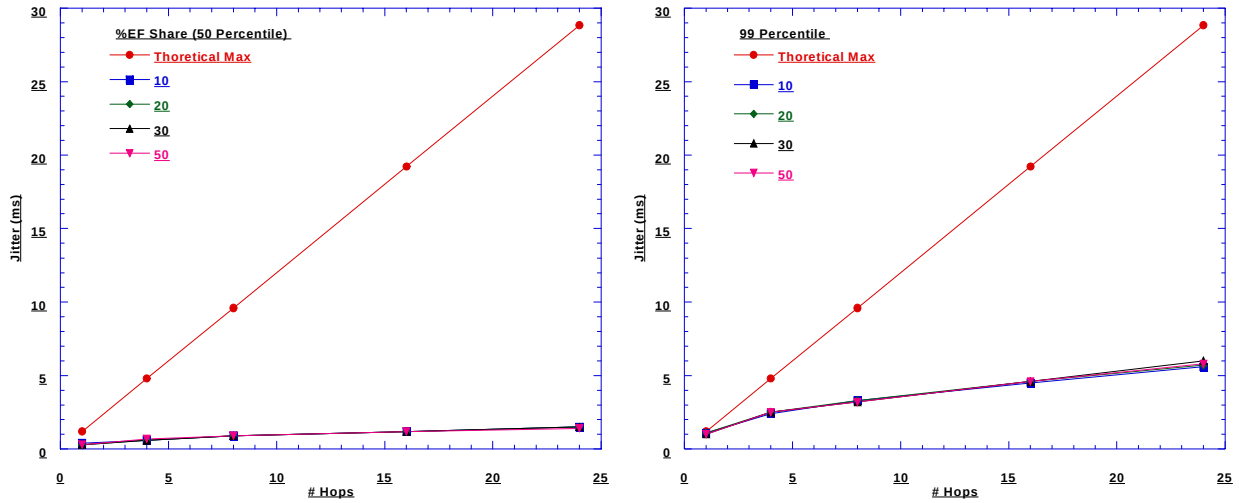

Figure 19: Median and 99th percentile jitter for various allocations and 10 Mbps links

What may not be obvious from figure 19 is that the similarity between the five allocation levels shows that jitter from other EF traffic is negligible compared to the jitter from waiting for DE packets to complete. Clearly, the probability of jitter from other EF traffic goes up with increasing allocation level, but it is so small compared to the DE-induced jitter that it isn't visible except for the highest percentiles and the largest hop count.