A Per-Domain Behavior for Circuit Emulation in IP Networks

Kathleen Nichols
Pollere LLC
325M Sharon Park Drive #214
Menlo Park, CA 94025
nichols@pollere.com

Van Jacobson
Packet Design Inc.
3400 Hillview Ave.
Palo Alto, CA 94304
van@packetdesign.com

Kedarnath Poduri
Packet Design Inc.
3400 Hillview Ave.
Palo Alto, CA 94304
poduri@packetdesign.com

ABSTRACT

Circuit networks are expensive to build, difficult to operate, fragile, and not easily scalable. Many network operators would like to carry circuit traffic as an overlay on top of an IP network. With higher bandwidths, faster routers, and Differentiated Services [RFC2474, RFC2475] features in routers, this is now possible. In this paper we describe a simple set of mechanisms that are sufficient to allow an IP cloud to carry circuit replacement traffic. Then, using the framework of [RFC3086] for describing a Per-Domain Behavior (PDB), we explain where these mechanisms should be deployed and quantify how they should be configured in order to construct the appropriate edge-to-edge behavior. This "Virtual Wire" (VW) PDB makes it possible to replace dedicated circuits with IP transport. In the process we attempt to lay to rest two pieces of QoS mythology: first that a Diffserv approach requires substantial over-provisioning compared to int-serv, and second that Diffserv quality of service is inherently loose and not quantitative.

1 BACKGROUND AND APPLICABILITY

The Internet's datagram architecture is robust and capable of carrying a wide range of traffic so many network operators would like to carry legacy circuit traffic using an inexpensive IP infrastructure. Today's large optical bandwidths and high-speed routers with line-rate differentiated services capabilities supply the means to do so. Ths paper describes how to configure an IP cloud to deliver the required packet treatment using Differentiated Services, providing both the mathematical approach and the resultant recipe for circuit emulation on an IP network. By circuit emulation, we mean provision of a strictly bounded rate and delay variation transport.[1]

1.1 Network requirements

Network hardware has become sufficiently reliable that the overwhelming majority of network loss, latency and jitter (delay variation) are due to either short-term variation in network packet queues or routing changes. This paper focuses on configuration to ensure packets of the aggregate see no (or very small) queues over a time scale on the order of the edge-to-edge propagation time. The analysis makes three assumptions on routing: shortest path routing is used, routing is stable on time scales long compared to the edge-to-edge packet propagation times, and that all traffic between two points uses the same route (i.e., no equal cost multi-path, ECMP, splitting)[2] The authors have been involved in extensive measurements of large networks [RTG, FG, SUBMS] that have found them to be quite stable. For example [FG] continuously measured jitter across a transcontinental tier 1 network at 1 ms intervals for several months and found a delay variation of less than one millisecond 99.99% of the time for normal best effort traffic. All of the observed jitter was due to routing events and those events were largely due to fixable (and subsequently fixed) router bugs, not "acts of god." This routing stability on properly configured networks lets us focus on the forwarding path configuration of VW, confident that routing problems will be infrequent enough to allow Service Level Agreements to be met[3].

The importance of time scales is critical when trying to assess how poorly routing must be behaving before VW starts to misbehave. For example, for a transcontinental emulated cicuit (100ms end-to-end propagation) to be disrupted 0.01% of the time, there would need to be a routing flap on the path used by that circuit every 3 hours. This is several orders of magnitude worse than the worst we have measured.

[1] Although voice circuit traffic is typically viewed as "constant rate," VW is more general and will deliver a hard bound on jitter for variable rate circuit traffic as long as the rate stays below a pre-agreed bound. Thus VW handles silence-suppressed voice, variable-rate video, frame relay, etc.

[2] The "no ECMP" assumption is made only to simplify the exposition and is not intrinsic. We briefly discuss the effects of ECMP and how to account for them later in the paper.

[3] Only IP routing was measured. The stability and predictability of schemes such as MPLS is unknown.

1.2 Diffserv use and background

Differentiated Services provides a toolbox and a framework for delivering a range of treatment to distinct packet traffic aggregates [RFC2474, RFC2475, DSINT]. It is distinguished from approaches that are path-oriented and keep state in the center of the network. Most of the IETF work on Diffserv has focused on the definition of network node-level components that enable the differentiation of IP traffic, most notably the per-hop forwarding behavior (PHB). A PHB only describes behavior at a single hop, but for a meaningful behavior across a DS domain, traffic conditioning requirements must be combined with a PHB to deliver a behavior which concatenates and aggregates. A per-domain behavior (PDB) [RFC3086] is the technical specification of how to configure a DS domain and what quantifiable behavior can be expected, i.e. the manner in which PHBs are configured in the *collection* of nodes that make up a DS domain and the particular configuration of the domain's boundary traffic conditioners. This paper describes the Virtual Wire (VW) PDB, a scalable, low loss, low latency, low jitter, hard-limited peak bandwidth, edge-to-edge service that appears to the endpoints like an unshared, point-to-point connection or an emulated dedicated link. The development of this PDB is rooted in our own earlier differentiated services work [RFC2598, RFC2638, VWID][4], but contains significant new work and several key departures from past approaches.

A VW PDB is intended to send circuit replacement traffic across a Diffserv network. That is, VW 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 and egress routers adjacent to the originator/terminator. Inside the cloud, or DS domain, packets carrying the circuit data are only differentiated by the particular traffic aggregate to which they belong. This PDB 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). This definition is explicitly for carrying multiple rate circuits and explicitly network-oriented. The VW PDB major attributes are a guaranteed peak rate and a bounded jitter. It is possible to define a PDB with less rigorous requirements and only the first attribute, a "constant bit-rate" PDB, but this is not our objective. Three components are required. First, support in the IP forwarding path of commercial routers. Second, circuit-to-packet (and packet-to-circuit) conversion appliances at the edge of the cloud. Finally, specifications relating configuration and measured parameters of a network and its components to the rate and jitter bounds that can be provided. This paper gives requirements on the first and second and is mainly concerned with the third.

1.3 Related work

Attempts to define a suitable Internet service for "real-time" traffic predates DiffServ, notably in Golestani's papers [GOL1, GOL2] and IntServ's Guaranteed Quality of Service. The VW PDB has the same motivation as the Guaranteed Quality of Service [RFC2212] of the IntServ model, i.e., "datagrams will arrive within the guaranteed delivery time and will not be discarded due to queue overflows, provided the flow's traffic stays within its specified traffic parameters." However, the method of specification and delivery of this guarantee is quite different, being defined on a network domain, rather than a path, and not requiring signaling or state in the network interior. Further, the VW PDB is deliberately a low jitter service, designed for a much simpler and network-oriented implementation, requiring fewer features in the interior routers of a network while the IntServ Guaranteed Quality of Service puts a set of complex requirements on all network elements.

The VW PDB is more a descendant of Golestani's work than IntServ. [GOL2] identified some problems of aggregation and sending circuit traffic on packet networks. However, the approach to bounding jitter required re-timing at all the interior network nodes which is unnecessarily conservative. Though this approach leads to long delays and difficult scheduler implementations, it is a forerunner to our work in showing that timing windows do not have to be rigidly synchronized, but instead can be made to obey some other constraints. However, Golestani's conclusion was that control at the edge was not sufficient and that "more elaborate controls" were necessary and this paper shows that only a separate queue is required in addition to edge controls.

More recent work on the "hose" model [HOSE] also has the goal of providing performance assurances (specifically Service Level Agreements) at a traffic level that is more general than point-to-point. The VW PDB's performance bounds are more general than that of a hose, being defined on the entire network cloud, at the same time allowing for the use of more specific information to give tighter bounds, e.g., to a particular hose.

Mercankosk realized that an early document on VW [VWID] provided a context for his work on the theory of circuit emulation. His subsequent technical report and paper [VWANAL] is a good complement to this paper since it covers much of the same ground from a formal theoretical perspective rather than our more operational focus.

Our initial description of the Expedited Forwarding Per-Hop Behavior [RFC2598] appeared to confuse some readers and we agreed that it should be clarified. An excellent description, available as the Delay Bound (DB) PHB [RFC3248],

[4] RFC2598 is not a standards-track document but is referenced here as part of the authors' development of the ciruit emulation service.

was developed by a group under charter of the Diffserv working group. Simultaneously an independent group developed a specification for a new behavior, substantially different from the one described in [RFC2598], and requested that it be given the name and code point of EF. The working group voted to do this and the “new EF” is described in [RFC3246]. Subsequently there have been a number of publications on this EF PHB but these are not relevant here for two reasons. First, focusing on isolated per-node forwarding behavior tends to obscure the different and difficult effects of aggregation and transit that arise when one tries to compose node behaviors to construct a network service; the focus of this paper is constructing such a service. Second, [RFC3246] does not meet the primary objective of [RFC2598] which was to provide a building block that could be used to construct a VW PDB. [RFC3246] insufficiently constrains forwarding behavior in that it is possible to have a router behave in a way that exactly meets the formal requirements of [RFC3246]’s Section 2.2, eq_1 to eq_4, but violates properties essential to the implementation of VW. [5]

Rather than attempt to define additional restrictions on the EF PHB of [RFC3246], the VW PDB should use the Delay Bound PHB described in [RFC3248]. A specification of the forwarding behavior required is given in 3.2 for completeness. A CS PHB [RFC2474] should also be configurable to the specification.

2 CHARACTERIZING VIRTUAL WIRE DELAY AND ITS VARIATION

There is much prior work showing that such circuit traffic as voice and video is feasible on packet networks with sufficient average bandwidth to handle the data rate. Applications for the MBone, a large scale experiment in sending audio and video on the open Internet [MBONE, VIC] , removed the timing distortion from delay variation across the Internet by measuring the worst case jitter (difference between maximum and minimum delays) and using this to set a play-out delay at the egress. One surprise was how large this delay needed to be, often several seconds. Delays this large are unacceptable for many circuit applications, thus we need to understand why the jitter can grow so large and how to constrain it. This requires specifying the components of a VW PDB.

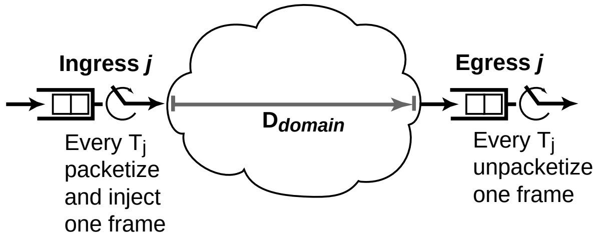
Ingress *j*
Every T$_j$ packetize and inject one frame
D$_{domain}$
Egress *j*
Every T$_j$ unpacketize one frame

Figure 1: Circuit transport across an IP domain

A VW PDB is characterized by specific properties of the domain on which it is configured: a (peak) rate R_{vw} (for a maximum packet size, S_{max}, where $S_{max} \leq MTU$) supportable without loss and a delay variation bound. The service offered to a particular circuit should be stated in terms of its rate and delay or, alternatively, the rate and a jitter bound. Constraints on the rates that may feasibly be allocated to individual circuits using the VW PDB (e.g., maximum, minimum, limits at particular ingress/egress links, maximum packet size) are also properties of a specific domain. (Here *jitter* refers to delay variation referenced against a fixed clock, not variation in delay between pairs of packets, or inter-arrival jitter.) An emulated circuit is characterized by a particular ingress, a particular egress, a peak rate, and a maximum packet size. The term *VW flow* will be used to describe the packets of one emulated circuit. Note that this definition of flow covers the range from a single microflow (e.g., a single phone call) to an aggregation of microflows entering at a single ingress (e.g., a trunk of calls). A VW flow is the stream of packets that enters the network edge compliant to one particular rate shaping and represents a single allocation “unit.”

Figure 1 illustrates the transport of an emulated circuit on an IP cloud. An ingress appliance packetizes a frame of the input circuit every T_j and sends $pckt_i$. It has duration $\frac{S}{B}$ where S is the packet size and B is the link bandwidth and this width must be $\leq T_j$. For each ingress, there is a maximum permissible packet size, S_j which cannot exceed the VW PDB maximum permissible packet size S_{max}.

Designating as zero the time the first packet enters the domain, $pckt_i$ enters at time $i \cdot T_j$[6]. Each $pckt_i$ must receive at least rate R_j, including the packetizing overhead. The ingress boundary of the network is at the entrance to the first network element, after the packetizing function. S_j, the maximum number of bytes of data emulated circuit or ingress j may send into the domain during T_j, is upper bounded by an MTU, though a smaller upper bound is possible (e.g., voice). At the egress, packets are played out onto the output wire one frame each

[5] For example, in the next section we show that jitter can grow quite large when packets of a VW flow queue behind other packets of the same flow. The DB PHB [RFC3248] precludes this by characterizing the ingress to egress behavior of a router in terms of a configured rate R such that if packets arrive at a rate strictly less than R no queue will form. The new EF [RFC3246] has a similar R but lacks the “no queue” guarantee. To see this, consider an output interface where packets of fixed length 1 arrive at constant rate R. Thus [RFC3246]’s eq_4 for target departure time will be $F_j = F_{j-1} + 1/R$ or $F_j = j/R$. Let the first packet depart at $D_1 = F_1$ and all subsequent packets be delayed by E_p then depart at constant rate R. Thus $D_j = F_j + E_p = j/R + E_p$ for $j > 1$ and since $D_0 = 0$, the average departure rate of the aggregate is just j/D_j or $\frac{j}{j/R+E_p} = R(1+RE_p/j)^{-1}$ which is strictly less than R for all $j > 1$ since $E_p > 0$ for any physically realizable router. Thus the average arrival rate exceeds the average departure rate at all times after $j = 1$ and a queue will form.

[6] Bounds on the actual timing required are made explicit in 3.1.

T_j at the circuit's rate. The egress boundary is at the exit of the last network element on the path; specifically, the play-out buffer, which absorbs variation in delay, is not considered part of the network cloud.

Inside the network, the packets of the VW PDB are not distinguished by their ingress or egress, but are said to belong to the VW Traffic Aggregate (TA), the collection of packets that have been admitted to the VW PDB.

The attributes of a VW PDB on any DS domain will be a function of both hardware limitations of that network and of configuration decisions. The former can be measured but not changed. The latter represent a trade-off between the range of feasibility determined by the hardware properties and decisions or requirements on the service levels to be offered and number of customers and requires looking inside the cloud. This paper aims to clarify and quantify these trade-offs.

2.1 Delay along the route

There are four main contributors to packet delay. T_j is the framing time or time between packetized frames of an emulated circuit. Next, D_{domain}, the IP cloud delay from first byte entering till first byte emerging and including physical media propagation delays, packet forwarding delays, and queuing delay. This can be written as $D_{domain} = D_{min} + J$ where $J \in [0, D_{max} - D_{min}]$, that is, D_{domain} takes on values between D_{min} and D_{max}, the maximum possible delay. The maximum jitter or delay variation bound is $J_{max} = D_{max} - D_{min}$ Third, d_{buffer}, play-out buffer delay, removes delay variation by holding the first packet $\geq J_{max}$ and includes the time for the complete packet to arrive at the buffer, $\frac{S_j}{B}$ (B is the link bandwidth). Fourth is the packetizing and depacketizing overheads, if any. This overhead is external to the network and can be subsumed into the play-out delay so is disregarded in the rest of this document. All parameters except for J depend solely on the current state of technology and are taken as given for VW PDB construction. Delay for each sample of emulated circuit j being carried across the IP cloud can be written as:

$$delay(sample_i) = T_j + D_i + d_{buffer} \tag{1}$$

where D_i is the value of D_{domain} seen by $pckt_i$. Here j pertains to a particular VW flow and i to a particular packetized sample of the flow.

An egress appliance perfectly synchronized with the ingress could begin to transmit the contents of the play-out buffer D_{max} after the first packet was sent and play-out would proceed at the minimum delay to remain gap-free. Such synchronization is complex and expensive; instead, the arrival of the first packet at the egress is used to define the clocking. As long as the clock is started a sufficient delay after reception of the first packet, there will be no gaps at the output wire, and all bytes will have been delayed by the same amount. The delay of the first packet must be within the range D_{min} to D_{max}, so assume D_{min}, and hold it for (at least) J_{max} after its arrival to accommodate any subsequent arrival of a maximally delayed packet[7]. A play-out delay of J_{max} minimizes overall delay and buffer requirements. Actual delay is within J_{max} of the synchronized case, since $J_{max} \leq D_0 - D_{min}$. Then for a packet crossing the network:

$$delay(packet_i) = D_{min} + J_i + \frac{S_j}{B} \tag{2}$$

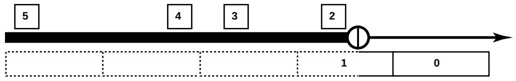

Figure 2: Transit time variation and egress re-timing

2.2 Delay variation

As routing changes occur on much larger timescales than inter-packet times, consider a network where routing is stable and all delay variation is due to interference with other traffic that causes variable waiting or queuing time along the path. Delay can be split into two terms, one due to all the other traffic in the network and one from earlier packets of the VW flow carrying the circuit. Let d_i be the delay seen by $pckt_i$ as it crosses an IP cloud from edge to edge, all_i the delay due to interference with all other traffic and *fwd*, the sum of the all the constant forwarding delays experienced along the path, and, α, a sequential delivery constraint that contributes some fraction of the previous packet's delay:

$$d_i = \alpha \cdot d_{i-1} + all_i + fwd \tag{3}$$

which has a solution [DIFFEQ] $d_i = C \cdot \alpha^i + \frac{all_i + fwd}{1-\alpha}$ where C is a constant. If $\alpha = 1$, delay grows without bound, but this only occurs if the arrival rate exceeds the network capacity. The solution is bounded for $\alpha < 1$, though the values of d_i can grow rapidly and quite large with increasing time samples, i, particularly for $\alpha > 0.5$, accounting for the large delays seen in the MBone. The smallest variation in delay occurs when $\alpha = 0$ and $d_i = all_i + fwd$, i.e., subsequent packets of a VW flow cannot queue behind earlier packets, so that packet *i* of a VW flow never queues behind packet *i-1* for all *i*.

2.3 Packet shadows and maximum flow rate

Packets of each ingress or emulated circuit can be associated with a *shadow packet* across the domain where each shadow's

[7] If packets vary in size, the first packet must be held for an additional $\frac{S_j - S_i}{B_{min}}$ to account for later arrivals of maximum sized packets, S_j. This can be ignored in later derivations.

right hand boundary corresponds to a packet launch time from at its ingress and the shadow's current location bounds the location of the physical packet at the current time. The shadow takes D_{min} to transit the cloud and the start of each subsequent packet, sent T_j after the previous one, must remain within J_{max} of the front of its shadow as it transits the cloud. Recall that S_j is the maximum number of bytes of data emulated circuit or ingress j may send into the domain during T_j(the maximum packet size), and that each packet (including overhead) must receive at least rate R_j. The packet shadow's duration T_j is defined and lower bounded by:

$$T_j = \frac{S_j}{R_j} \geq J_{max} + \frac{S_{max}}{B_{min}} \tag{4}$$

and the number of these spanning the domain is D_{min}/T_j. If the shadow's duration is long enough to accommodate all forwarding path jitter plus the duration of the physical packet at any link, shadows *will not experience any variation in their delay*. When a shadow's left hand boundary has reached a location, the physical packet must also have reached that location. A physical packet completely arrives at the egress appliance D_i after its launch time into the cloud plus $\frac{S_j}{B}$, the packet time on the cloud's internal link(s). The smallest possible T_j for a domain determines the largest allocatable rate on the domain, a *shadow size constraint*

$$R_{max} = \frac{S_{max}}{T_{min}},\ T_{min} = J_{max} + \frac{S_{max}}{B_{min}}. \tag{5}$$

where R_{max} is the largest possible value for any R_j on the domain. The leading edge of the shadow packet of flow j that enters the domain at $i \cdot T_j$ arrives at E at $i \cdot T_j + D_{min}$; all packets arrive at the egress within J_{max} of this leading edge. This is not a constraint on inter-arrival time; physical packets can arrive separated by more than T_j and still be completely received at E before needed to transmit (see for example packets 4 and 5 in figure 2).

The total rate allocation available for the VW PDB on the domain is defined as R_{vw}, where $\sum_j R_j \leq R_{vw}$ and $R_{max} \leq R_{vw}$. There must be a T_{vw} the minimum time window over which R_{vw} must not be exceeded. If we define S_{vw} as the maximum number of VW PDB bytes that can be sent into the domain during interval T_{vw}, then we have $\sum_j S_j \leq S_{vw}$ and $R_{vw} = \frac{S_{vw}}{T_{vw}}$.

2.4 VW rates for α = 0 and no non-VW traffic

Requiring that physical packets must (completely) depart each node before the trailing edge of their shadows ensures $\alpha = 0$ in eqn 3. Assuming that VW flows enter a domain with the timing requirements above, this section uses packet networks carrying only VW traffic and shows how to enforce $\alpha = 0$ to illustrate concepts and develop intuition.

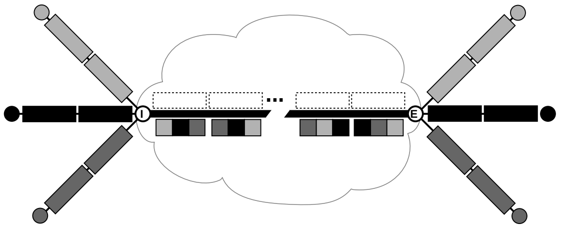

Figure 3: Service order independence for VW flows

2.4.1 When the wire is not virtual

A single packetized circuit on a wire. If the domain consists of a wire of capacity B, $B \geq R_j$, the physical packet containing the original circuit data remains at the front of each shadow. D_{domain} is the wire's propagation delay, D_{prop}, and, with a packet's origin time, locates the packet: $pckt_i$ exits at $i \cdot T_j + D_{prop}$. Transmission begins $\frac{S_j}{B}$ after each packet begins to arrive.

Multiple circuits of the same rate. For B sufficiently larger than R_j, a wire can carry multiple packetized circuits. Figure 3 shows three independent emulated circuits (light gray, black and dark gray) of the same rate, R_j where $n \cdot R_j \leq B$. $T_j = T_{min}$ and $S_j = S_{max}$ for all j. If the circuits have worst-case phasing, 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 the VW TA queue, no packet will get pushed outside its window, T_{min}. Node I ships a different perturbation of the three customer aggregate in every window yet this has no effect on the edge-to-edge VW properties. It doesn't matter what order the packetized circuits are served at the ingress node as long as the egress node holds the first packet of the circuit for at least the domain jitter window, $J_{max} = 2 \cdot \frac{S}{B} = T_{min} - \frac{S}{B}$. For this domain, T_{min} is $3 \cdot \frac{S}{B}$.

Multiple circuits with different rates. The wire carries three packetized circuits, each with a different rate R_j and a different T_j. All have the same value for S and $R_1 = 2 \cdot R_0$ and $R_2 = 3 \cdot R_0$. The time to send a physical packet is$\frac{S}{B}$ and $T_0 = 12 \cdot \frac{S}{B}$, $T_1 = 6 \cdot \frac{S}{B}$, and $T_2 = 4 \cdot \frac{S}{B}$. Delay variation comes only from differences in scheduling delays while one of the other packetized circuits is being launched onto the internal wire, thus $J_{max} = 2 \cdot \frac{S}{B}$. Maximal jitter occurs when packet arrivals are aligned and the service order randomly permutes, e.g., first serving $0-1-2$ and next $1-2-0$. Circuits are reconstructed by holding the first packet of each circuit for J_{max}. For circuit 2, the play-out buffer will sometimes hold more than one packet; this does not violate the requirement that a physical packet arrive at a time no later than the arrival of the shadow. To maintain $\alpha = 0$, no other VW flows can be added, since circuit 2's incoming packets could meet its previous packet at the ingress queue. The smallest possible T_j is

$3 \cdot \frac{S}{B}$, as in the previous example. Although each emulated circuit has a different rate and a different sized shadow packet, each shares the same J_{max} and each maintains the arrival of its packet flow without timing gaps independently from the other. Further, timing alignment between the flows is completely immaterial, giving *jitter independence* between the flows.

Though the two examples have the same T_{min}, efficiency is lost with mixed rates and the requirement that α=0. That is, without synchronization on the framing, the total bytes sent during any T_{min} must maintain $S_{vw} = \sum_j S_j \leq \frac{T_{min}}{B_{min}}$. This is equivalent to allocating the maximum rate, $R_j = \frac{S_{max}}{T_{min}}$ to all VW flows even though, as in this case, some of them may be operating at a lower rate. Methods of recapturing some lost efficiency (without time synchronization) will be discussed in later sections.

2.4.2 Transiting a multi-hop packet network

Delay variation is increased due to encountering other traffic at each hop that can further move a physical packet around with respect to the boundaries of its shadow. A packet network may carry many VW flows using the VW PDB. Every link of the domain must have VW bandwidth available equal to or exceeding the maximum amount of VW TA that may transit it. To simplify terminology, we confine ourselves to the case where any link of the domain might need to carry the entire VW TA, $R_{vw} = \Sigma_{j=1}^{n} R_j$.

For heterogeneous link bandwidths, a physical packet duration varies as it crosses the network, never smaller than T_{min}. Thus the VW aggregate must receive at least R_{vw} on every internal link over the time T_{min}. The maximum rate that can be assigned to a VW flow is $R_{max} = \frac{S_{max}}{T_{min}}$. For multiple VW flows of different rates, R_j, and for B_{vw_l}, the amount of bandwidth reserved for the VW aggregate on link l:

$$R_{vw} = \Sigma_{j=1}^{n} R_j \leq B_{vw_l}. \qquad (6)$$

T_{min} defines the smallest time frame during which we can send no more than the amount of VW data which the domain can handle, S_{vw}. Let B_{min} be the smallest B_{vw_l}. Then with VW traffic alone, the *shadow size constraint* is $T_{min} = \frac{S_{vw}}{B_{min}}$.

2.4.3 A shadow frame, T_{vw}

T_{vw} was previously defined as the minimum time window over which R_{vw} must not be exceeded. During each T_{vw}, the sum of all the emulated circuits' individual packets must not exceed S_{vw}, a constraint which must be enforced on the domain's boundary. This leads us to further generalize T_{vw} as the *shadow frame,* similar in function to a TDM timing frame in which individual circuits are allocated particular time slots. Unlike a TDM system, there is no need to enforce a relative timing reference on the various emulated circuits in the domain. Instead, we require that $\sum_j R_j \leq R_{vw}$ and $\sum_j S_j \leq S_{vw}$. Although it does not have a synchronized timing reference across all entry points of the domain boundary, shadow frames are all the same duration and consist of the jitter due to the VW TA, the jitter from servicing packets of other traffic aggregates, and the transmission time of the packet.

Enforcement of $\alpha = 0$ in eqn 3 requires that $min_j T_j \geq T_{vw} \geq T_{min}$ (T_{min} as in eqn 5). As a result, any VW flow can have *at most* one packet in a shadow frame and all flows are allocated a "slot" in T_{vw}, whether they use it or not. Section 2.6 presents approaches to increasing the efficiency thus lost.

2.5 VW in general networks

2.5.1 Adding non-VW traffic

In eqn 3 all_i includes both the delay due to non-VW traffic aggregates, and the delay due to packets of the VW traffic aggregate. The jitter due to variation in these delay values is from the point of view of packets of the VW TA: *other jitter* caused by a VW packet queuing for packets of non-VW traffic aggregates and *self jitter* caused by queuing behind another VW packet, from the same VW flow or another VW flow. The worst case values for each of these makes up the J_{max} term in eqn 5, determining the minimum size shadow frame and thus the maximum rate that can be supported on the domain. Rewrite the jitter window constraint as:

$$T_{vw} \geq J_{max_{other}} + J_{max_{self}} + \frac{S_{max}}{B_{min}} \qquad (7)$$

2.5.2 Edge constraints: policing the ingress

The definition of T_{vw} requires that no more than S_{vw} bytes enter the domain during any shadow frame. A packet that would cause violation of this requirement will be discarded. From section 2.4.3, enforcement of this requirement can be distributed among the n VW flows by enforcing a rate limit R_j over duration T_j for each VW flow j, e.g. by strictly policing the jth ingress to packets no larger than S_j spaced at intervals T_j. A precise definition of this edge constraint is given in section 3.1 to use in implementing traffic conditioning.

2.5.3 Required per-hop behavior

The departure rate for the VW TA on all network links must be $\geq R_{vw}$ over T_{vw}. Conceptually, a scheduler where the queue holding the VW aggregate gets a token good for S_{vw} bytes every T_{vw} suffices; at the time granularity of T_{vw}, the VW TA can always get the S_{vw} bytes out. We call this generalized forwarding behavior *virtual priority*. A wide range of packet scheduler implementations can be configured this way, some more effective in providing small jitter values than others.

For any output link packet scheduler, there is a *scheduler cycle time*, τ_{cycle}, between initial service times of the VW TA's queue. τ_{cycle} is made up of the time the scheduler spends servicing the VW queue, τ_{vw}, and the time the scheduler is preempted (from the point of view of the VW TA) by non-VW traffic. For a particular output link l, $\tau_{cycle_l} = \tau_{other_l} + \tau_{vw_l}$. For a work conserving scheduler, τ_{cycle} can vary between 0 and an upper bound imposed by the scheduler's characteristics and configuration: we are interested in the upper bound. The cycle time imposes a constraint on T_{vw}. During each cycle, the scheduler sends τ_{vw}/B_l bytes of the VW TA. If k is the minimum number of worst case cycles (maximum time servicing *other* queues and the minimum time servicing the *VW* queue) it takes to send S_{vw} bytes, then $T_{vw} \geq k \cdot \tau_{cycle}$where $S_{vw}/R_{vw} \leq T_{vw}$. Note that τ_{vw} is always long enough to send at least one packet.

2.5.4 Determining self jitter

Packets of the same VW flow. A packet can only be jittered by packets of its own VW flow if a previous packet hasn't completely departed at the time a packet arrives. Since a VW flow enters the domain with strictly enforced rate R_j (over duration T_j) and must receive at least that rate on every link inside the domain, for two packets of the same VW flow to meet, the departure rate on some link must be less than R_j over T_j. Eqn 6 requires the departure rate for all links $\geq R_{vw}$ during T_{vw} so proper VW configuration will not allow this condition; no statement can be made for a misconfigured network. Thus each packet of a VW flow has a variation within the shadow frame that is independent of the other packets in the flow.

Packets of other VW flows. Using the results of section 2.4.1 and eqn 5 allows multiple VW flows to be transparently aggregated into a single VW PDB. In the case where the VW PDB is made up of *n* VW flows, the worst possible self-queue occurs if packets of all *n* arrive simultaneously at some output link. Enforcing the jitter window constraint on T_{vw} includes the maximum variation in time displacement any VW packet can experience, then *no* new packets from any VW flow can arrive for at least time T_{vw}. At the end of this time, there can only be packets left in the VW TA's queue if the queue's departure rate is less than R_{vw} over this time scale, T_{vw}. This violates eqn. 6.

Strict ingress policing combined with eqns 4 and 6 prevents a packet of a VW flow from being delayed by a packet of any other VW flow more than once. Thus the *worst case* self-jitter caused by aggregation is a linear function of the number of VW flows in the aggregate that is independent of topology. The self-jitter that a particular VW flow can encounter is bounded by:

$$J_{self(j)} \leq (k \cdot \tau_{cycle} - \frac{S_j}{B_{min}}) \qquad (8)$$

where $k \cdot \tau_{cycle}$is the time to clear the VW queue after simultaneous arrival of the *n* packets when there is a backlog of *other* traffic, $k \leq n$. Additional restrictions on the maximum number of VW flows that can share node output links would lead to a smaller upper bound on *k*.

Self jitter is bounded by the number of separate rate allocations or emulated circuits in the domain. Worst case self-jitter is a *domain* attribute that can be stated without reference to the detailed interior structure of the domain.

2.5.5 Determining jitter from other TAs

Varying levels of non-VW traffic can affect the jitter at each hop, but the *worst case* delay variation does *not* require knowledge of the intensity of the non-VW TAs. Each VW packet can be jittered by non-VW TAs as much as τ_{other_l} at each link where τ_{other_l} is the longest possible value. Then:

$$J_{other_{max}} \leq \sum_{l=1}^{h} \tau_{other_l} \qquad (9)$$

where the summation is over the contribution to jitter of each node along the route and τ_{other_l} must be determined for each link if the network link bandwidth is not homogeneous. A looser bound can be obtained by replacing the summation with $h \cdot \tau_{other_{max}}$.

2.5.6 Example PHBs and resultant PDB parameters

Jitter for priority schedulers. Delay variation in a mixed traffic network is minimized when VW packets have *strict priority* in forwarding nodes, i.e., a VW packet will only queue for a non-VW packet if it is already in service when the VW packet arrives.[8] The largest jitter to a VW flow occurs if all flows arrive at the minimum bandwidth hop simultaneously such that the flow's $pckt_i$ is at the front of the VW queue, $pckt_{i+1}$at the end. Then:

$$J_{self(f)} \leq J_{self(f)_{max}} = \Sigma^{n}_{j=1_{j \neq f}} \frac{S_j}{B_{min}} \leq (n-1) \cdot \frac{S_{max}}{B_{min}} \qquad (10)$$

where the second term bound is tighter if the S_j are close in size. If additional restrictions on the maximum number of VW flows that share links are present and known, *n* in eqn 10 can

[8] An IP link scheduler is non-preemptive at the packet level. Fragmentation is only used very occasionally and on low bandwidth links. This would make the bound lower, but the analysis is identical, only using a smaller worst case wait, so it is not discussed further in this paper.

be replaced with the upper bound on the number of VW flows to share a single output link

Since the VW traffic aggregate is uncorrelated with the other traffic aggregates, a VW packet may arrive at any time during the service of a non-VW packet at a particular link. In the worst case, a VW packet at link l will have to wait for $\tau_{other_l} = \frac{MTU}{B_l}$.

Thus for a network domain of h hops, $J_{other_{max}} = \Sigma_{l=1}^{h} \frac{MTU}{B_l}$, and using eqn 10 in eqn 7:

$$T_{vw} \geq \sum_{l=1}^{h} \frac{MTU}{B_l} + \Sigma_{j=1_{j\neq f}}^{n} \frac{S_j}{B_{min}} + \frac{S_f}{B_{min}} = \sum_{l=1}^{h} \frac{MTU}{B_l} + \frac{S_{vw}}{B_{min}}.$$

Jitter for a round-robin scheduler. Though a fairly common vendor implementation choice, Weighted-Round-Robin and Deficit-Round-Robin schedulers can lead to large jitter. Nevertheless, such schedulers can be suitable for the VW PDB.Let weight W_q be the maximum number of bytes sent at each visit to queue q. The worst case cycle time of a WRR or a DRR scheduler can be written as $\tau_{cycle} = \Sigma_q \frac{W_q}{B}$. Then $\tau_{other} = \frac{1}{B} \cdot (\Sigma W_q - W_{vw})$. Many implementations compute weights in number of packets, as in the example here. Packet sizes for the non-VW TA are bounded by MTU and packet sizes for the VW TA are bounded by $S_{max} \leq MTU$. Assume the upper limits and write:

$$\tau_{other} = \frac{MTU}{B} \cdot (\Sigma W_q - W_{vw})$$

where the W_q are in packet counts and must be integer.

In figure 4, the VW traffic aggregate is using queue 5 configured at 50% of the output link bandwidth (weight 5), queue 4 has 40% (weight 4)and queue 1 has 10% (weight 1). The worst case variation in *q5*'s inter-departure time occurs if *q1* and *q4* are both backlogged. In this case 4 packets from *q4* and one packet from *q1* are sent immediately after every 5 packets from *q5* so 17% of *q5*'s packets are separated by 6 MTU times and the remainder are back-to-back.

For this scheduler note that, since the weights must be integers, the maximum jitter is dependent on the *smallest* share allocated. For example, if *q1* has a share of 1% of the output link, *q4* 49% and *q5* still has 50%, the total number of shares in τ_{cycle} is 100 and the scheduler will delay 50 packet times before revisiting *q5*. Even if the amount of traffic admitted to the TA using *q5* is limited to far less than 50% of the link, the cycle time must still include the 50 packets of non *q5* traffic. Thus, increasing the share of *q5* beyond its allocation cannot decrease J_{max}.

Schedulers with different ways of sharing the link could decrease the maximum jitter contribution by decreasing τ_{cycle}. The above example is one of the poorer choices of link scheduler for realizing a VW PDB but it can still be utilized to deliver an appropriate virtual priority per-hop behavior.

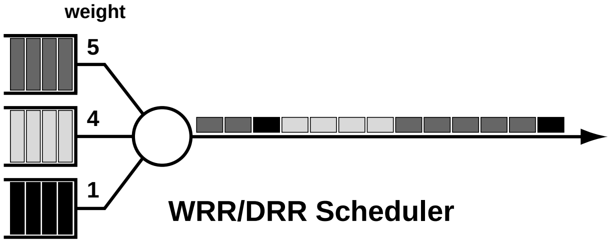

Figure 4: Example of delay variation due to scheduler

2.5.7 On relating VW rate allocations and jitter

The VW PDB rate allocation trades off with jitter bounds. As $R_{vw} \leq S_{vw}/T_{vw}$, we now have

$$T_{vw} \geq max_l(k \cdot \tau_{cycle_l}) + \Sigma_{l=1}^{h} \tau_{other_l} \qquad (11)$$

Since $R_{vw} \geq \Sigma_{j=1}^{n} R_j$ and $T_{vw} \leq T_j$, rate allocations among the different ingresses or flows can only be varied by increasing individual T_j's and by variations in S_j's. Then largest rate that can be allocated is $R_j = \frac{MTU}{T_{vw}}$. A small T_{vw} minimizes jitter, but may limit S_{vw} so that either the total rate allocation is small or the number of individual VW flow allocations is quite limited. A large T_{vw} increases domain jitter. Further, as more separate allocations are made, n increases, and the self-jitter increases.

2.6 Extending the definition to gain efficiency

2.6.1 Mixed rates and inefficiency

T_{vw} defines a time frame during which we can send no more than the amount of VW data which the domain can handle, S_{vw} and thus far we have further constrained T_{vw} to keep $\alpha = 0$ in eqn 3 which requires treating all allocations as if they were R_{max}. Suppose there are 3 VW flows using the same packet size, S, two of rate R and one of rate $2 \cdot R$ to be sent through a single wire domain, like the one shown earlier in figure 3. Then each flow must be allocated a rate $2 \cdot R$ and $6 \cdot R \leq B$, an inefficient use of the bandwidth.

Suppose that $B = 4 \cdot R$, exactly the sum of the actual VW flow rates and we attempt to apply the VW PDB. Figure 5 shows the lower rate flows, black and light gray, allocated 1/4 share each while the higher rate flow (dark gray) gets a 1/2 share. Dark gray's shadow is $T_{dg} = 2 \cdot S$ (the dotted dark gray line), the others are $4 \cdot S$. The emulated circuits can have any relative time phase, so packets from all three may arrive simultaneously at I. I has no per-flow state, thus the serving order for the three is random and the two low rate circuits may be served before the high rate. Thus there are no dark gray packets in its third shadow frame and two in its fourth. This results in a non-zero α in eqn 3 though only one other packet could be encountered

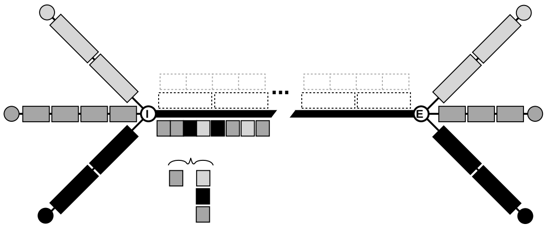

Figure 5: Effect of different circuit rates

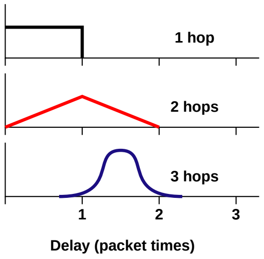

Figure 6: Probability density of non-VW jitter after 1 to 3 hops

in this case, thus $\alpha = 0$ for all even values of i. If the dark gray circuit uses $d_{playout} = max_j T_j \leq J_{max}$, it can still be reconstructed without gaps.

2.6.2 Delay variation and use of the domain shadow frame when $\alpha > 0$

Use of a non-zero α leads to more efficiency in the VW PDB, but it must be applied so that delay variation is bounded and an intuitive use of the domain shadow frame results. Revisiting section 2.2 and eqn 3, delay is bounded if, for some m, $pckt_i$ never queues behind $pckt_{i-m}$, i.e., a maximum of m packets of the same VW flow may be present in the same queue inside the domain. This implies that in eqn 3, $\alpha = 0$ for all $i = m, 2m, 3m, \ldots$ and that jitter from these m packets cannot accumulate past the shadow frame time T_{vw}. With this extension, larger rate allocations can be made by permitting individual flows to send more packets during T_{vw}. That is, individual rate allocations R_j are not limited to a maximum of one MTU during T_{vw} and the individual circuit frames, T_j, can be *smaller* than T_{vw}. The rate bound remains the same, $R_{vw} \geq \Sigma_{j=1}^{n} R_j$, but now we can allocate individual rates up to $R_j = \frac{m \cdot S_j}{T_{vw}}$ where m is an integer. Then ingress j can send one packet of size $\leq S_j$ every $T_j = \frac{T_{vw}}{m}$. Clearly, no value of T_j is less than $\frac{S_{min}}{R_{vw}}$ where $S_j = S_{min}$, the smallest packet size possible for the domain.

More specifically, for each emulated circuit or ingress, there is an m_j which gives both $T_j = \frac{T_{vw}}{m_j}$ and the maximum number of packets of VW flow j that a packet of flow j can queue behind, $m_j - 1$. Define

$$m_j \cdot T_j = k \cdot T_{vw}$$

where both m_j and k are arbitrary integers larger than 1, but subject to the constraints above and

$$T_{vw} \geq \frac{\Sigma_j m_j \cdot S_j}{R_{vw}}.$$

with worst-case self-jitter for a particular flow

$$J_{self_{max}} = \frac{\Sigma_{j \neq i} m_j \cdot S_j}{B_{min}}$$

The most efficient allocation then is when $k = 1$. That is, pick T_{vw} for a domain as in eqn 5 such that the *smallest* desired rate allocation is expressed as a basic rate, $\frac{S_{max}}{T_{vw}}$. This means that every shadow frame will hold *at least* (rather than *at most*) one packet of an actively sending ingress or emulated circuit j. This treats the larger rate flows like an aggregate of minimum rate flows, a particularly appropriate choice if the flow is indeed an aggregate, e.g., a trunk containing many voice calls. In the allocation and provisioning process, each emulated circuit or ingress should be given a rate allocation that is a multiple of the basic rate for the domain. Then the worst case self-jitter occurs by queuing behind one packet of each of these basic rate allocations[9], or

$$J_{self_{max}} = \frac{R_{vw}}{\frac{S_{max}}{T_{vw}}} \cdot \frac{S_{max}}{B_{min}} = \frac{R_{vw} \cdot T_{vw}}{B_{min}}$$

2.7 Putting the jitter effects together: practical limits

Relaxing the dependence on hop count. The worst case jitter from non-VW traffic is linearly dependent on the length of the path (in hops) that a packet follows. This *worst case* bound is *highly* unlikely and where it is sufficient to statistically bound jitter to a very high probability, the dependence on hop count can be greatly relaxed. In general, the approach of statistically bounding service quality is quite practical and frequently used, e.g. "five nines" availability.

If the *other* traffic fills its share of the network (a congested network) and all packets have length MTU, then the probability of waiting at each hop is uniformly distributed from 0 to τ_{other}. Assuming that all hops have the same τ_{other}, the spread of possible values after h hops is from *0* to $h \cdot \tau_{other}$. The joint probability function for all the hops is the convolution of that of the individual hops, illustrated graphically in figure 6 where the single hop worst case is normalized to unity. ([SLFCNV] contains a useful and readable analysis of the evolution of the

[9] The arithmetic is altered somewhat if $S_j < S_{max}$, but the results are straightforward.

parameters of the convolved density function with h.) The convolved pdf approaches Gaussian distribution, $\frac{e^{\frac{-x^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}$ with mean value $\mu_h = \frac{h}{2} \cdot \tau_{other}$. This is not a tight estimate until h is large, but the spread of the Gaussian is *larger* than that of the convolution, overestimating the spread, thus suiting our needs quite well. A bound of 99th percentile for the Gaussian distribution ($\pm 3\sigma$ of the mean) used to estimate the actual distribution is a higher probability bound ($\geq$ 99%) for the actual distribution.

When delays are distributed uniformly over τ_{other} at each hop,

$$\sigma_h = \sqrt{\frac{h \cdot \tau_{other}}{12}} \, and \, 3\sigma_h \cong 0.9\sqrt{h \cdot \tau_{other}}$$

With stationary link utilization levels, $6\sigma_h$ would be useful as a high probability estimate of the $J_{other_{max}}$ about the mean. Experimentally convolving these functions shows that, at each hop, $2\sigma_h$, or $\cong 0.6\sqrt{h \cdot \tau_{other}}$ is sufficient to yield a 97.5th percentile bound. As stationary utilization levels can not be assumed in general, a high probability bound for domain jitter that holds without regard to fluctuations in traffic load must cover the range from one packet seeing the minimum delay to an adjacent packet seeing the maximum delay a packet is likely to see under high load conditions or $J^{est}_{other_{max}} = \mu_h + 2\sigma_h = \frac{h \cdot \tau_{other}}{2} + 0.6\sqrt{h \cdot \tau_{other}}$. Most ISP domains have maximum diameters of about six hops and most enterprise domains about 10 to 12 hops. If, as a practical matter, a value of k = $\sqrt{h}$ of 3 to 5 is used, the resulting bound covers jitter to a very high probability.

Effects of realistic utilizations and packet sizes. Two of the above assumptions lead to larger values of τ_{other} than in realistic networks, 1) MTU-sized packets and 2) high link utilization. Network studies show more than half of all packets are smaller than 100 bytes and the effect of the presence of smaller-than-MTU sized *other* packets will decrease τ_{other} . Further, most network links are utilized at levels much less than 100% making the probability of incurring *no* delay at a hop more likely. If the per-hop probability is weighted by the link's utilization, it changes the per-hop distribution to an impulse weight of $1 - util$ at 0 and, roughly, the rest of the probability, $utilization$, uniformly distributed from *0* to τ_{other}. The latter has the effect of decreasing the mean value to $utilization \times \mu_h$ without the weight at zero delay. A practical, but still conservative, jitter bound is:

$$J^{est}_{other_{max}} = util \cdot (\mu_h + 2\sigma_h) = util \cdot (\frac{h \cdot \tau_{other}}{2} + 0.6\sqrt{h \cdot \tau_{other}}) \quad (12)$$

when all hops have the same τ_{other}. If not, replace $h \cdot \tau_{other}$ with $\sum_{l=1}^{h} \tau_{other_l}$. If the values of τ_{other_l} are greatly different, it may be best to put the hop with the large value of τ_{other} into another DS domain, re-timing at the edge. That is, where specific highly utilized links exist next to several less utilized links (e.g. at an access network), those links should be made to belong to a separate DS domain.

Implications for real networks. Although the hop count contribution to $J_{other_{max}}$ at first appears daunting, it becomes less so when 1) a high probability bound is acceptable in place of the upper bound and 2) typical hop counts for domains are considered. Further, in many cases, the rate allocations of the VW PDB and, indeed the total R_{vw}, are modest compared to the capacity of the network links, enabling simplifications in provisioning and use. These computations are useful in understanding delay variation and its bounds and trade-offs in provisioning, but may also be used to give confidence to work from measurement samples of the real probability distribution of jitter across the target network domain, using this understanding of the actual distribution to inform setting the value of J_{max}. An example of a closely related empirical approach is outlined in [EMP] for a tier 1 ISP.

3 TECHNICAL SPECIFICATION OF THE VW PDB

The results of the previous sections can be applied to more formally specify a VW PDB and its resultant attributes as in [RFC3086]. This specification covers traffic conditioning at the domain edge, PHB configuration, parameterized attributes, and rules for concatenating this PDB across domains to build a service description.

3.1 The VW PDB at the edge: traffic conditioning

Figure 7 illustrates a DS domain on which a VW PDB has been defined with parameters $(R_{vw}, T_{vw}, J_{max}, S_{max})$.The edge of **D**, shown as a "cloud" outline, must enforce the rules on entry to the VW PDB's traffic aggregate. As long as packets enter the VW traffic aggregate (by crossing the domain edge) at less than or equal to the VW PDB's configured rate R_{vw}, the packets will be delivered across the domain with a very high probability and with almost no distortion of the inter-packet timing imposed by the source. However, any packets sourced at a rate greater than the VW configured rate will be unconditionally discarded. In general, a VW PDB is said to have a configured rate $R_{vw} = \frac{S_{vw}}{T_{vw}}$.

"Sourced at a rate greater than the VW configured rate" is defined as sending more than S_{vw} bytes in T_{vw} with respect to a measurement interval $interval_i$. Measurement intervals are defined relative to the arrival time of the first VW packet, t_0, as

$$interval_i = [t_0 + i \cdot T_{vw} - 2\Delta, t_0 + i \cdot T_{vw} + 2\Delta) \quad (13)$$

where $i \geq 1$ and $T_{vw} \geq 2 \cdot \Delta \geq 0$. The intent of the parameter Δ is to 1) allow enforcement of the spacing of the arriving

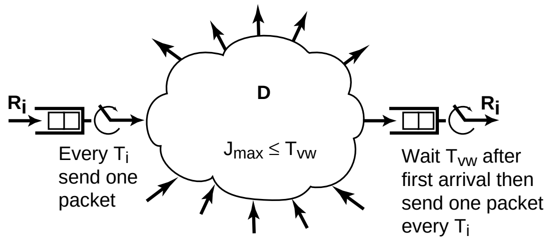
R_i
D
Every T_i send one packet
$J_{max} \le T_{vw}$
Wait T_{vw} after first arrival then send one packet every T_i
R_i

Figure 7: A VW PDB on DS Domain D

packets and 2) avoid placing the measurement interval where small perturbations of arrival time would result in a violation for an otherwise compliant traffic aggregate. Thus Δ should be the maximum permitted jitter in the arrival flow and, as such, must be included into its reconstruction delay at egress. If a range of packet sizes is permitted in the ingress flow, Δ will need to be large enough to include the variation, $\frac{S_{max}-S_{min}}{R}$. To reduce buffering and the resultant delay in the reconstructed signal, Δ should be kept small.[10]

A policer that *enforces the rules on entry* to the VW PDB's aggregate only allows the allocated number of packets to be admitted during each T_{vw}. It can be described as a token bucket of depth $\sum_j m_j$, where each token is good for one packet and the additional restriction can be enforced that all packets must be $\le S_{max}$. The token bucket fill rate is T_{vw}, with tokens delivered at the beginning of time $interval_i$ and expired (if not used) at the end of $interval_i$.[11] An appropriate shaper for a conformant packet flow for this PDB would send one packet (of size $\le S_{max}$) every $\frac{T_{vw}}{\sum_j m_j}$.

Synchronized enforcement over the entire boundary is not necessary. In practice, an allocation process distributes the VW PDB among some n ingress ports of the domain and enforcement is also distributed among the ingress ports. Each ingress then becomes responsible for policing its allocation of the VW PDB for the domain. The notion of a "VW flow", a convenient fiction inside the cloud, has meaning at the domain edge as one of the n allocation chunks. At the edge, classifiers and policers are used to construct a flow that conforms to both the requirements of the VW PDB and its allocation policies. Each ingress VW flow is strictly policed to its rate R_j where packet sizes do not exceed S_{max}, though an allocation policy can be applied that further restricts a particular domain ingress to packets not to exceed size $S_j \le S_{max}$, and packets enter spaced in time at $T_j = \frac{R_j}{S_j}$.

For distributed enforcement, the restriction can be written for each ingress by replacing T_{vw} and S_{max} in equation 13 with the ingress's T_j and S_j and a Δ_j specific to that ingress. For example, in figure 7, the VW PDB's rate R_{vw} can be split among the six ingress ports so that $\Sigma_{j=1}^{6} R_j = R_{vw}$, $S_j \le S_{max}$, and $m_j \cdot T_j = T_{vw}$. Any partitioning is possible that obeys those constraints. The worst case "self jitter" for the VW aggregate increases by m_j packets with each additional ingress policing point.

Packets that pass this policing must be marked in their DSCP field with the codepoint that selects the PHB configured as described in Section 3.2.

3.2 Individual node PHB requirements: virtual priority queue (VPQ)

This section specifies the required configuration for the PHB used by VW and the relationship of the per hop parameters to bounds on the VW parameters. In application, some iteration of parameters may be needed subject to realizability constraints.

Relating R_{vw}, the total VW rate on the domain, to PHBs. VW PDB parameters $(R_{vw}, T_{vw}, J_{max}, S_{max})$ derive from the limitation of the per-hop behavior that can be configured on each particular domain. The maximum rate for the VW TA that can be supported on a hop is $\frac{\tau_{vw}}{\tau_{cycle}} \cdot B_l$ (τ_{vw} and τ_{cycle} as in section 3.1). The minimum value of this bound on all the domain's hops upper bounds R_{vw}. Then T_{vw} is lower bounded by considering the maximum jitter of the domain and S_{vw} is the amount of VW traffic that can be transmitted by every hop over T_{vw}.

Relationship of S_{vw}. S_{vw} is the maximum number of bytes that may be sent in that time window, $S_{vw} = \frac{R_{vw}}{T_{vw}}$. However, it also can be used to compute the maximum self-jitter a VW packet of size S can see, $J_{self_{max}} = \frac{S_{vw}-S}{B_{min}}$. If there is a target number of emulated circuits for a domain or, alternatively, a target number of "basic rate allocations", where n is that target, then $S_{vw} = \sum_{j=1}^{n} S_j$.

Relating T_{vw} to PHBs. A VW PDB can be constructed on a domain if the individual nodes can be configured so as to bound the forwarding jitter of each packet. The *packet forwarding jitter* is defined as the difference between the node service time for any $pckt_i$ (of size S) that may experience competing traffic at the node and the node service time for the same size (and DSCP) packet (sent on a link of bandwidth B) when there is no competing traffic. Thus the packet forwarding jitter $\le (\tau_{cycle} + fwd) - (\frac{S}{B} + fwd) = \tau_{cycle} - \frac{S}{B}$ where τ_{cycle} is the scheduler cycle for the node as defined in section 2.5.5. This bound must hold so long as the arrival rate of the aggregate to its queue does not exceed its configured bound. The arrival rate of the aggregate is said to be *within its configured bound* if, in a network which is conditioned at the edge as described in section 3.1, there is some τ_{cycle} for which the number of

[10] It should be possible to more closely bound the measurement interval if some feedback mechanism is applied to the input stream.

[11] This is necessary as late arrivals will be permitted if tokens are not expired.

VW bytes arriving at the node does not exceed S_{vw}. The domain shadow frame is lower bounded by the largest τ_{cycle} on all possible VW hops, $max_h(\tau_{cycle_h}) \leq T_{vw}$.

For a priority queue (PQ) scheduler, $R_{vw} = \frac{S_{max}}{S_{max}+MTU} \cdot B_{min}$. If the desired allocations for the domain results in S_{vw} bytes per shadow frame, then $T_{vw} = S_{vw} \cdot \frac{S_{max}+MTU}{S_{max} \cdot B_{min}}$. If $S_{max} = MTU$ and all $S_j = S_{max}$, then $T_{vw} = 2 \cdot n \cdot \frac{MTU}{B_{min}}$. Then J_{max} can be reduced by decreasing *n*, and thus S_{vw}, but R_{vw} cannot be changed.

We define a node to exhibit virtual priority queue per-hop behavior over some time τ_{vpq} if the forwarding behavior “looks like” a priority queue to packets of the VW traffic aggregate over τ_{vpq} that is, there is a finite $\tau_{cycle} = \tau_{vpq} = \frac{S_{vw}}{B} + \tau_{other}$. We refer to the time period τ_{vpq} as the *VPQ bound* of a particular output link.

3.3 Concatenating VW PDBs

Each domain boundary output interface that can be the egress for VW traffic must strictly shape that traffic as described in section 3.1. The shaping parameters must not be more aggressive than the policing parameters the downstream domain uses at its ingress. As part of the VW PDB definition, each domain boundary input interface that can be the ingress for VW traffic must strictly police that traffic as described in section 3.1. The two DS domains must agree upon how the traffic from the upstream will be policed at the downstream. There are many methods of doing so, most current ones reflecting business practices rather than technical practices. Each DS domain reconstructs an unjittered stream, but at the cost of delay. This delay has a known bound that can also be exported to any downstream DS domains if necessary.

3.4 Real world considerations

Section 1.1 listed our assumptions on routing and their justification. Although routing instability will generally translate directly into VW service degradation, properly configured networks do not experience frequent routing changes as they can lead to packet loss and excessive delays. Networks that experience routing problems on time scales short enough to have a significant impact on the service level that can be specified for a VW service are not good candidates for VW. It is possible for a particular VW service to see no disruption after a routing event since effects depend on the severity of the problem and the VW packets should clear the network most quickly. A reasonable expectation on frequency of routing outages should be accommodated statistically in an SLA for a VW service.

Multipath routing of VW will, in general, increase the jitter and degrade the service unless either the paths are exactly the same length, technology, and configuration (so there is no effect on jitter) and/or the routing decision is such that it always sends any particular customer down the same path. The interested reader should be able to follow the analysis to see that each multipathing opportunity in the network could lead, at worst, to a VW packet being displaced one more time by a packet that it has previously met. In conservative allocations, this is likely to be subsumed by the allocation for “other” jitter, but it is certainly possible to account for it.

The analysis assumed traffic policers and link schedulers are perfect and mathematically exact. For real world applicability, incorporating 5-10% overhead factors should accommodate deviations from perfection.

4 IMPLEMENTATION CONCERNS AND EXAMPLES

Definition of a VW PDB is one step in IP circuit emulation. For large scale practical applicability, provisioning algorithms and software capable of handling a non-trivial number of circuits should be investigated and evolved, a process that can be highly dependent on topological effects. Such work is sufficient for another paper, but simple techniques are effective in some cases; the approach in [EMP] for example, or this small motivational example on enterprise voice-over-IP.

An enterprise uses VW to provision a large scale, internal VoIP telephony system. Internal links are all Fast Ethernet (100Mb/s) or above, arranged in a three-level hierarchy (switching/aggregation/routing) so the network diameter is 5 hops. Typical telephone audio codecs deliver a packet every 20ms. At this codec rate, RTP encapsulated G.711 voice is 200 byte packets and G.729 voice is 60 byte packets.[12]

Using G.711, $J_{other_{max}} = 5 \cdot \frac{1500\,bytes}{100Mbps} = 0.6\,ms$ and $J_{self_{max}} = n \cdot \frac{200\,bytes}{100Mbps}$. R is the basic rate allocation for this VW PDB of 80 kbps and the shadow frame T_{vw} is most sensibly defined at 20 ms, its minimum. Then the largest number of calls for this network is $n = \frac{19.4ms}{0.016ms} \cong 1200$ which has a $J_{max} = 19.84ms$. J_{max} can be reduced by using a smaller value of *n*, the smallest possible being a single call with a $J_{max} = 0.6ms$. Since J_{max} defines the additional delay added to packets of the call, its target value should be used to pick *n*. For example, if the target is to have no more than 10ms of additional delay, then only 600 calls should be admitted. An adaptive play-out mechanism can reduce the expected value of this delay with J_{max} then upper bounding it.

The preceding holds only *if the ingress can simultaneously police both packet and bit rate*. If the ingress can police only one of these then only 75 calls can be admitted because each packet might be as long as an MTU.

[12] We deliberately do a worst-case analysis, ignoring the effect of RTP header compression.

If the codecs use the smaller G.729 frames, about 4000 calls can be admitted if the maximum jitter is allowed or 2000 for a 10 ms J_{max}. This paper does not discuss methods of limiting the number of calls admitted, but, for some networks, policing the flows might be considered sufficient.

5 REFERENCES

[DIFFEQ] Difference Equations, Ronald E. Mickens, Van Nostrand Reinhold Company, Inc., 1987.

[GOL1] S. Golestani, "Congestion-Free Transmission of Real-Time Traffic in Packet Networks**,"** IEEE Infocomm Conference Proceedings, 1990.

[GOL2] S. Golestani, "A Stop-and-Go Queueing Framework for Congestion Management," ACM Sigcomm Conference Proceedings, 1990.

[DSINT] B. Carpenter and K. Nichols, "Differentiated Services in the Internet," Proceedings of the IEEE, vol 90 no 9, September, 2002, pp. 1479-1494.

[EMP] T. Telkamp, "Traffic Characteristics and Network Planning," NANOG-26, Eugene, OR, October 27-29, 2002, `http://nanog.org/mtg-0210/telkamp.html`

[FG] S. Casner, C. Alaettinoglu, and C-C Kuan, "A Fine-Grained View of High-Performance Networking," NANOG 22, May, 2001, `http://nanog.org/mtg-0105/casner.html`

[HOSE] N.G. Duffield et al, "A Flexible Model for Resource Management in Virtual Private Networks," Sigcomm, 1999.

[MBONE] S. Casner and S. Deering, "First IETF Internet Audiocast," ACM Computer Communications Review, 22(3), July 1992.

[RFC2212] "Specification of a Guaranteed Quality of Service," S. Shenker, C. Partridge, R. Guerin, `http://www.ietf.org/rfc/rfc2212.txt`, September, 1997.

[RFC2474] "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", K.Nichols, S. Blake, F. Baker, D. Black, `http://www.ietf.org/rfc/rfc2474.txt`, December 1998.

[RFC2475] "An Architecture for Differentiated Services", S. Blake, D. Black, M.Carlson, E.Davies, Z.Wang, W.Weiss, `http://www.ietf.org/rfc/rfc2475.txt`, December, 1998.

[RFC2598] "An Expedited Forwarding PHB", V. Jacobson, K. Nichols, K. Poduri, `http://www.ietf.org/rfc/rfc2598.txt`, June, 1999.

[RFC2638] "A Two-bit Differentiated Services Architecture for the Internet", K. Nichols, V. Jacobson, and L. Zhang, `http://www.ietf.org/rfc/rfc2638.ps`, July, 1999 (original I-D December, 1997).

[RFC3086] "Definition of Differentiated Services Per-domain Behaviors and Rules for their Specification", K.Nichols and B.Carpenter, RFC 3086, `http://www.ietf.org/rfc/rfc3086.txt`, April, 2001.

[RFC3246] "An Expedited Forwarding PHB", B. Davie, A. Charny, J.C.R. Bennett, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, D. Stiliadis, `http://www.ietf.org/rfc/rfc3246.txt`, March, 2002.

[RFC3248] "A Delay Bound alternative revision of RFC 2598", G. Armitage, B. Carpenter, A. Casati, J. Crowcroft, J. Halpern, B. Kumar, J. Schnizlein, `http://www.ietf.org/rfc/rfc3248.txt`, March, 2002.

[RTG] Cengiz Alaettinoglu and Steve Casner, "ISIS Routing on the Qwest Backbone: a Recipe for Subsecond ISIS Convergence", NANOG-24, Miami, FL, Februrary, 2002, `http://nanog.org/mtg-0202/cengiz.html`

[SLFCNV] S.Koscielniak, "Automated multiple self-convolution of functions using the Tchebychev-Hermite asymptotic expansion and its relation to the Central Limit Theorem," TRIUMF Design note TRI-DN-00-01, January, 2000, `http://www.triumf.ca/people/koscielniak/tridn-2000-01.pdf`

[SUBMS] C. Alaettinoglu, V. Jacobson, H. Yu, "Toward Millisecond IGP Convergence," draft-alaettinoglu-ISIS-convergence-00, November 2000, `http://www.packetdesign.com/news/industry-publications/drafts/convergence.pdf`, also talk at NANOG-20, October, 2000, `http://nanog.org/mtg-0010/igp.html`.

[VIC] S. McCanne and V. Jacobson. Vic: A flexible framework for packet video. In ACM Multimedia 95, pages 511–522, November 1995.

[VWANAL] G. Mercankosk and J. Siliquini, "The 'Virtual Wire' Per Domain Behaviour - Analysis and Extensions," IFIP Conference Proceedings, pp 265-276, October, 2002.

[VWID] V. Jacobson, K. Nichols, and K. Poduri, "The 'Virtual Wire' Per-Domain Behavior", (originally draft-ietf-diffserv-pdb-vw-00.txt), at `http://www.packetdesign.com/news/industry-publications/drafts/vw_pdb_0.pdf`.