An Opinionated View of the Current State of IP Differentiated Services

Kathleen Nichols

kmn@cisco.com

UC Berkeley MIG Seminar

September 29, 1999

Outline

Part I: Differentiated Services capsule tutorial

intent is to make sure there is sufficient background for Part II

Part II: Issues in evaluating differentiated services with an emphasis on forwarding path behavior

Not going to talk about:

- Allocation framework (e.g., Bandwidth Broker architecture)
- Applicability: service examples
- (tutorial slides from HotI99 might be of interest for these topics)

History (as near as I can figure)

1989 (?) QoS discussions in the IRTF end-to-end research group

March,1994 IETF WGs on Integrated Services and RSVP

March 1997 Future of Differentiated Services BOF

July, 1997 Differentiated Services discussed at Intserv WG meeting (see diffserv.lcs.mit.edu for minutes, slides, documents)

December, 1997 Differentiated Services at Intserv WG. Serious discussions of what a Diffserv WG would do.

January, 1998 Diffserv document team meets to review a strawman "spce", a strawman internet draft results

Feb 1998 Diffserv WG formed

March 1998 First Diffserv WG meeting - spec document split

December 1998 Publication of RFC 2474 and RFC 2475

Goal of Quality of Service?

Van Jacobson, from a talk on differentiated services and bandwidth brokers given at Bay Networks, November, 1997:

> “[The problem we are solving is to] Give ‘better’ service to some (at the expense of giving worse service to others - QoS fantasies to the contrary, it’s a zero sum game).”

In other words, QoS is managed <u>unfairness</u>.

Then “who gets ‘better’ service?” and “who decides?” are critically important questions for any viable QoS architecture

> It seems this is often forgotten, or at least neglected, as a topic

Differentiated Services in a Nutshell

- The differentiated services architectural model is an approach to delivering QoS in a scalable, incrementally deployable way that:
  - keeps control of QoS “local”
  - pushes work to the edges and boundaries
  - requires minimal standardization, encourages maximal innovation
- Diffserv’s model is based on an Internet made up of independently administered domains, each of which is connected to at least one other
- The IETF Differentiated Services WG is working on the “minimal standardization” part of this. See www.ietf.org/html.charters/diffserv-charter.html.

(The talk Dave Clark gave in December, 1997 lays out a lot of this philosophy. It’s the appendix of RFC2638.ps)

Scalability through Aggregation

- Fundamental to the diffserv approach is:
  - there are a relatively small number of ways to handle packets in the forwarding path
  - the number of traffic conversations requiring QoS may be quite large and subject to a wide range of rules which devolve from policy
- Packets are grouped by the forwarding behavior they are to receive within a cloud. Called a “behavior aggregate (BA)”
- Nodes in the center of a network only have to deal with the small number of traffic aggregates rather than keeping track of every separate traffic conversation that passes through

Aggregation and Conversations

- The per-conversation state is kept at the edges
- Flows or conversations are classified into aggregates and are "conditioned" to meet the rules of that aggregate
- Packets are not marked for the "services" individual conversations may be receiving. Many services may use the same marking. Any viable service must make sense under aggregation
- Don't distinguish between flows, so the treatment the behavior aggregate receives should not result in different performance for different traffic compositions of the behavior aggregate

Thus understanding how traffic aggregates is crucial to creating services.

Two Major Components to Diffserv QoS

<u>One</u> is the forwarding path behavior, the differential treatment an individual packet receives, such mechanisms as:

- queue service disciplines and/or queue management disciplines,
- the packet classification and traffic conditioning (e.g., shaping, policing) that can be used to control aggregates

For the most part, these primitives are well-understood and must be fairly simple as they are done per-packet at line rate.

Two Major Components to Diffserv QoS (cont'd)

The **second** is the control plane used to implement a cloud's policy goals and to configure the forwarding path accordingly.

This is much less well-understood. Yet, using simple policies and static configurations, it is possible to deploy useful differentiated services in networks.

There are a number of such ways to configure the forwarding path to create services (e.g., RFC 2475, RFC 2598, RFC 2638). Deployment of these will lead to experience that will guide more complex policies and allocations.

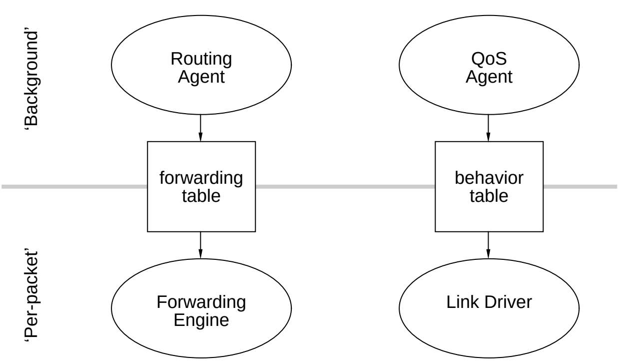
This approach has an analogy in the original evolution of the Internet in the development of forwarding and routing.

Mirrors the Development of the Internet

Packet forwarding is a simple task, performed on a per-packet basis as quickly as possible: use the packet header to find the routing table entry that determines the packet's output interface.

The work of routing is more complex and happens outside the forwarding path: routing sets and maintains the entries in that table and may need to reflect a range of transit and other policies as well as to keep track of route failures.

IP Router Architecture



RFC 2474 Fundamentals

- A bit-field in the packet header determines the packet's forwarding treatment. DS field covers the TOS octet in IPv4 and the Traffic Class octet in IPv6; within that uses bits 0-5 as a "codepoint" (DSCP)
- Codepoints should be looked at as an index into a table of packet forwarding treatments at each router.
- This table maps a DSCP to a particular forwarding treatment or "per-hop behavior" (PHB) that is applied to packets with that marking. PHBs are constructed by vendors from, for example, particular queue schedulers
- Behavior for only a few codepoints are to be globally assigned and diffserv-capable equipment must make codepoint to behavior mapping flexible and accessible
- Class Selector Compliant PHBs get DSCPs 000000-111000

Behavior Aggregates, DSCPs, Classifiers, and Traffic Conditioners

The DSCP indicates the packet's behavior aggregate within an administrative domain (or cloud)

A packet's DS field may be marked with a codepoint anywhere in the network (but marking is expected at edges and boundaries)

Marking can be based on microflow identification, the packet ingress link, the measured temporal characteristics of a microflow or aggregate, etc.

Traffic conditioners may alter the temporal characteristics of a behavior aggregate to conform to some requirements of a particular DS domain. Traffic conditioning includes metering, policing, shaping, and marking.

The Class Selector Compliant (CSC) PHB Group

This PHB group is defined in RFC2474 and seems to be widely underappreciated

- Motivation was to preserve some “backward compatibility” with the use of bits 0-2 of the TOS byte (the Precedence Field) while permitting evolution to a more useful PHB group
- RFC 2474 describes minimum requirements on a set of PHBs that are compatible with most of the deployed forwarding treatments selected by the IP Precedence field.
- The DSCPs 0-7 MUST map to PHBs meeting the requirements though the PHBs may impose *additional* requirements.
- Other codepoints MAY map to these same PHBs.

These can be utilized to deliver most types of QoS in use or under discussion today

CSC PHB Group Requirements

The CSC group must include at least two independently forwarded classes and the one codepoints 6&7 map to must “give packets a preferential forwarding treatment” compared to the PHB selected by DSCP 0 (the “default” behavior DSCP)

Packets with different Class Selector DSCPs may be reordered, though packets with the same DSCP which are not dropped should remain in order

A network node may enforce limits on the amount of the node's resources that can be utilized by each of these PHBs

Note that the CS Requirements can be met with two queues, one of which is mapped to by DSCPs 0-5, the other by DSCP 6&7.

On the other hand, the CS Requirements can be met with a sophisticated CBQ scheduler which may have more than eight queues.

Assured Forwarding (AF) PHB Group (RFC 2597)

Four independently forwarded AF classes and within each AF class,three levels of drop precedence (two okay)

Drop precedence of a packet determines the relative importance of the packet within the AF class. A congested AF node preferably discards packets with a higher drop precedence value

Packets with the lowest drop precedence value are assumed to be within a “subscribed profile”.

An AF-compliant node allocates resources sufficient to (at least) achieve the configured service bandwidth over “both large and small time scales.”

The concept of drop precedence within a class is a descendant of the MIT work on RIO (“RED with In and Out”, see diffserv.lcs.mit.edu/Papers/exp-alloc-ddc-wf.pdf), a two-level DP. Cisco has generalized this to “weighted RED” (WRED)

Expedited Forwarding (EF) PHB (RFC 2598)

This is a forwarding behavior of general use, but its most likely applicability is for “virtual leased lines” (VLLs) and for VoIP

In simple terms, it is a rough equivalent of PQ, can be implemented by PQ with some safety mechanism

More precisely (from RFC 2598):

> “the departure rate of the aggregate’s packets from any diffserv node must equal or exceed a configurable rate.
>
> The EF traffic SHOULD receive this rate independent of the intensity of any other traffic attempting to transit the node.
>
> It SHOULD average at least the configured rate when measured over any time interval equal to or longer than the time it takes to send an output link MTU sized packet at the configured rate. “

(Where “configured rate” means the configured rate of the EF aggregate for that output link)

Implications of the EF Definition

At most 50% of a link can be composed of the EF aggregate. Otherwise, an MTU-sized packet of some other aggregate followed or proceeded by an MTU-sized EF packet will violate the "SHOULDs" quoted on the previous slide.

The worst case displacement (which leads to jitter) of an EF packet is composed of two parts:

- First, the packet might encounter a packet from each of the other microflows which compose the EF aggregate. (If the EF configuration conforms to RFC 2598 then a packet from any microflow can encounter at most one packet from each of the other EF microflows.)
- Second, the packet might have to wait for an MTU-sized packet of some other aggregate at every hop along its path.

To see this as jitter, one packet of a stream must get worst case displacement, the next packet must wait for *no* packets

Services in the Diffserv Framework

- That covers the forwarding path, but what about building services?
- Services are built by adding rules to govern behavior aggregates:
  - initial packet marking
  - how particular aggregates are treated at boundaries
  - temporal behavior of aggregates at boundaries
- Classifiers and traffic conditioners at DS boundaries are configured to enforce rules in accordance with the service specification
- Different user-visible services can share the same aggregate
- Services must be sensible and quantifiable under aggregation

About Allocation of QoS Services

- A technical specification of a particular service first must make sense and must perform as expected
- Then we can begin to explore how it can be allocated within a domain to meet technical constraints and policy considerations.
- Traffic conditioners at boundaries as well as the mechanisms that implement PHBs must be properly configured to the allocation
- For services which cross domain boundaries, must communicate and specify the relevant data about the service between clouds.
- (Initially, expect the information that crosses boundaries is simple and static or quasi-static)

Part II: Issues in Evaluating Differentiated Services

There are many Internet Drafts, conference papers, and journal papers that purport to be evaluations of differentiated services

Beware

Many of these make sweeping conclusions based on very simple traffic models and topologies.

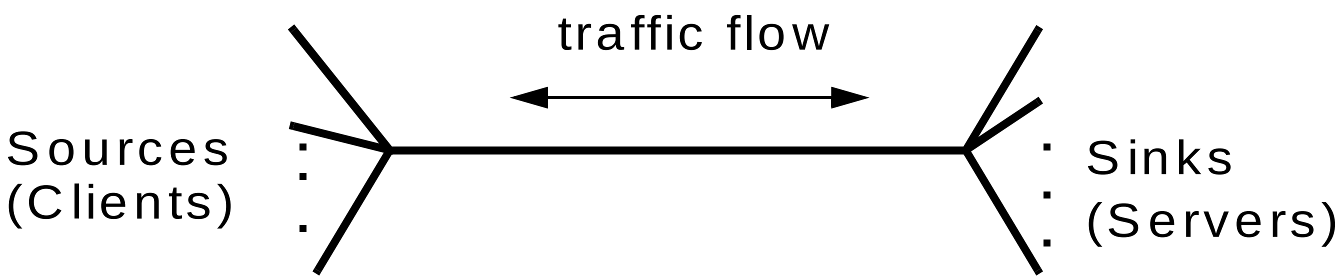
Most are based on “dumbbell” topologies and use a traffic model consisting *only* of long-lived TCPs with identical RTTs as their traffic model. Aggregation issues are invisible.

Simulations should be run long enough and for enough variation in random starting conditions.

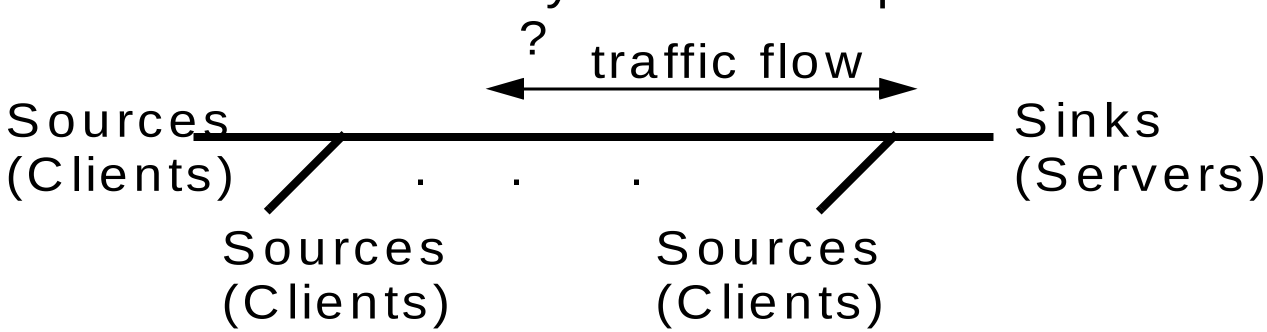
Ask yourself: What **topology** captures the salient features of a realistic network? Do the **traffic models** capture the salient features of real loads? Were **traffic loads** sufficient to exercise a range of operating points?

Topology: What kind of network?

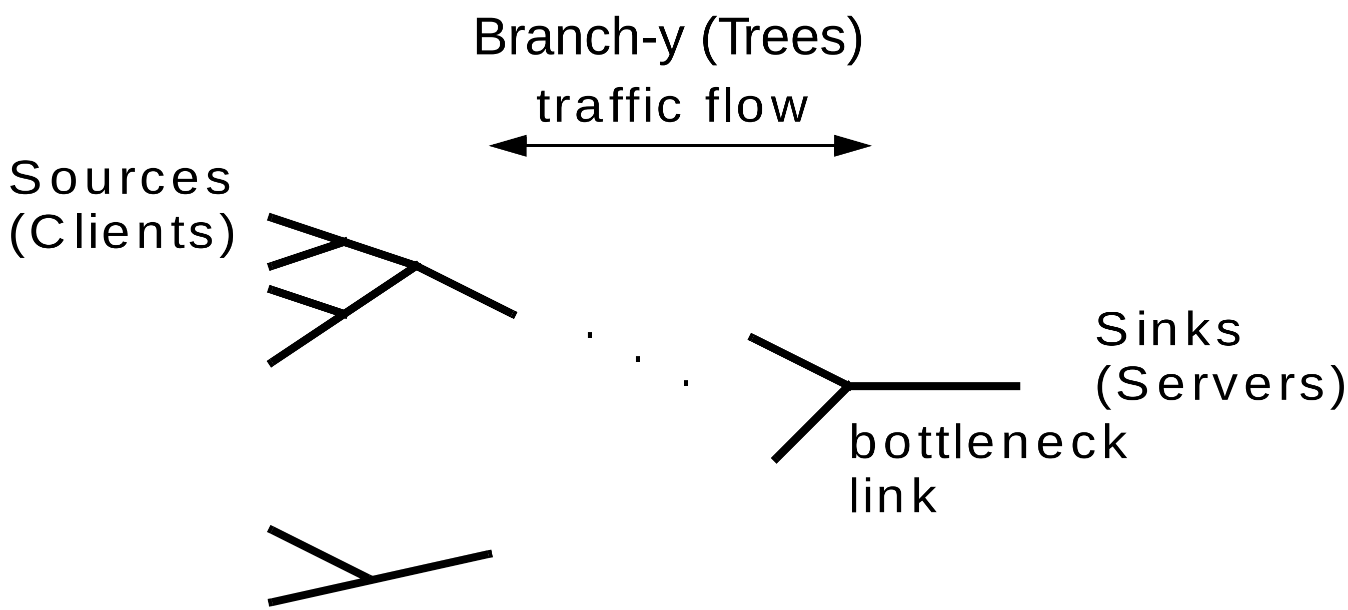
Dumbbells and Dancehalls



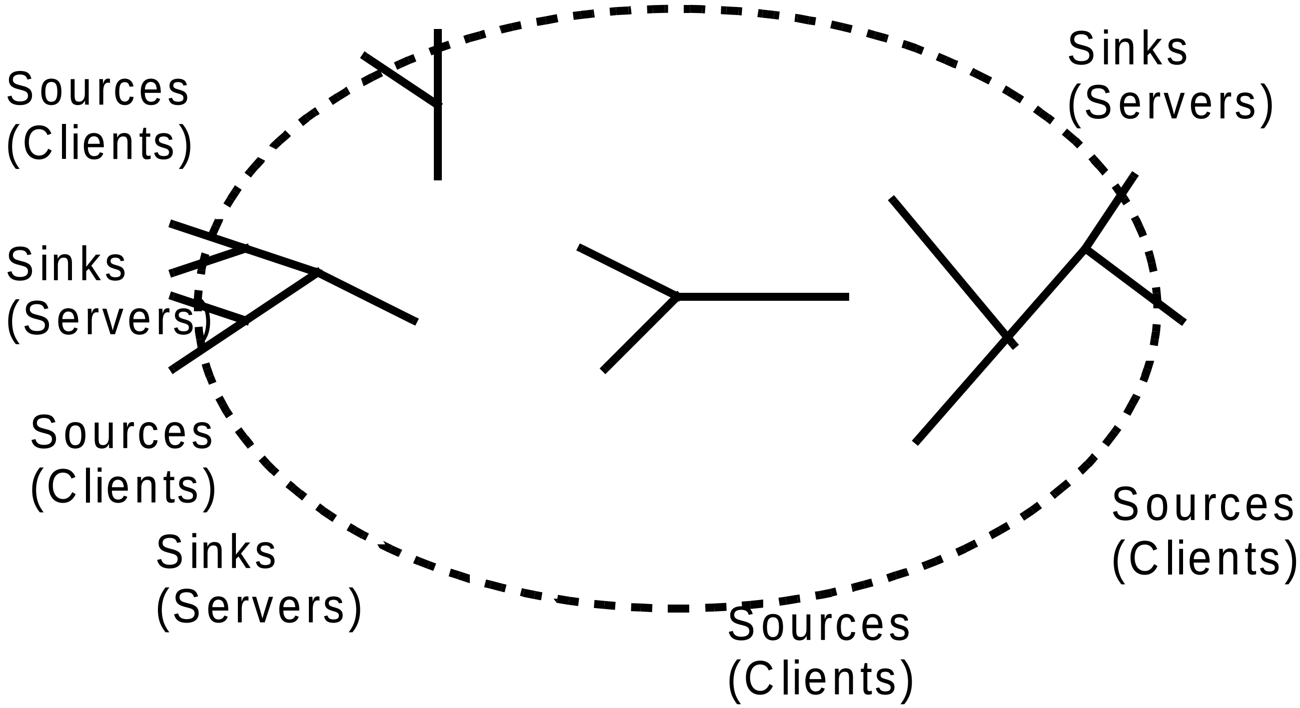
Freeway with On-ramps



More Topologies



Branch-y and Mesh-y



This one is both hard to draw and hard to simulate.

This makes it less appealing for graduate students who want to produce papers in a short time period.

Traffic Models: Many Ways to Go Wrong

Simulator models often oversimplify any or all of: the application model, the TCP protocol model, packet size selection.

Most published differentiated services simulations have used the ns-2 basic TCP, developed for the study of long-lived TCPs, which uses one size for all packets, does not include the SYNs and FINs of TCP connection set-up and tear-down, and does not permit data to flow in both directions. The effect of dropping a SYN packet is different from that of dropping a DATA packet.

Worse yet, some simulations rely on “open-loop” source models that look nothing like real traffic under congestion.

TCP’s feedback leads to network behavior which is time-varying and buffer dynamics that will vary with transfer sizes.

(Related discusssions in K.Nichols, “Improving Network Simulation with Feedback”, Proceedings of 98 IEEE LCN Conference)

Traffic Loads

Most published studies use long-lived TCPs (steady state). Some add CBR as “bad” flows to be restricted relative to the LL TCPs.

Most credible measurements put the percentage of HTTP traffic in networks at 60-90% of flows.

Measurement studies have found the distribution of file sizes fetched by HTTP transactions are Pareto distributed with most objects fitting into a single packet.

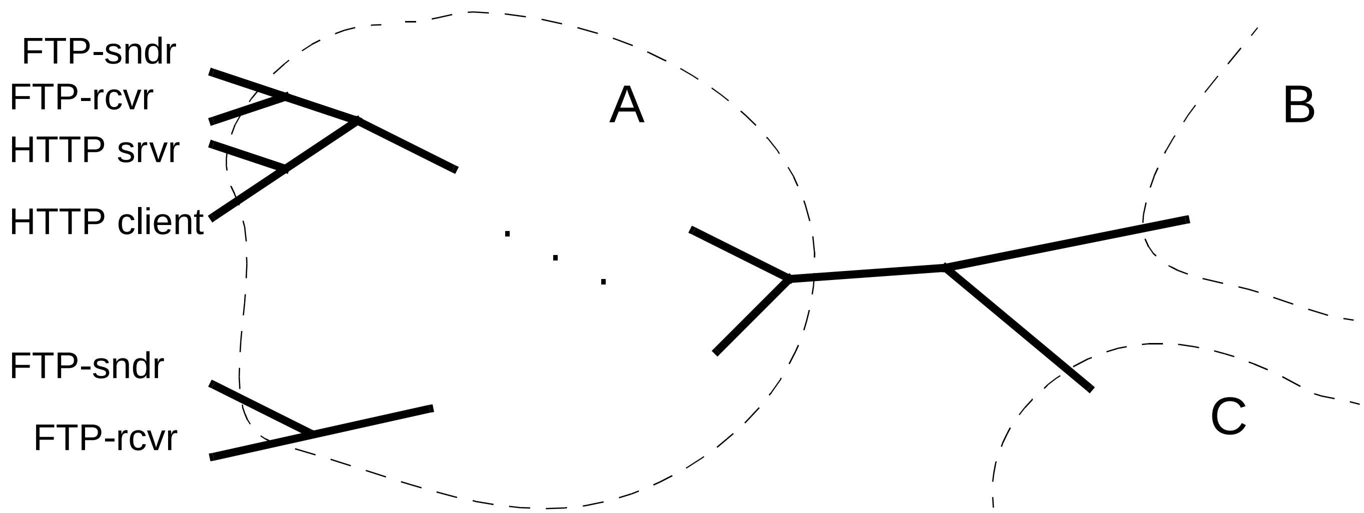
When a user downloads a web page, typically many small transfers take place. Such flows have different characteristics from a long-lived TCP (like FTPing a large file)

What we know about Internet traffic is that it will change over time, so we must design for and evaluate over a range of operating points.

Per-flow Fallacies

The use of dumbbells and long-lived TCPs obscures network dynamics and aggregation effects. Perpetuates myth of “single flow”, i.e., a long-lived TCP as a representative of “the customer”.

End up solving the wrong problems. Unnecessary complexity.

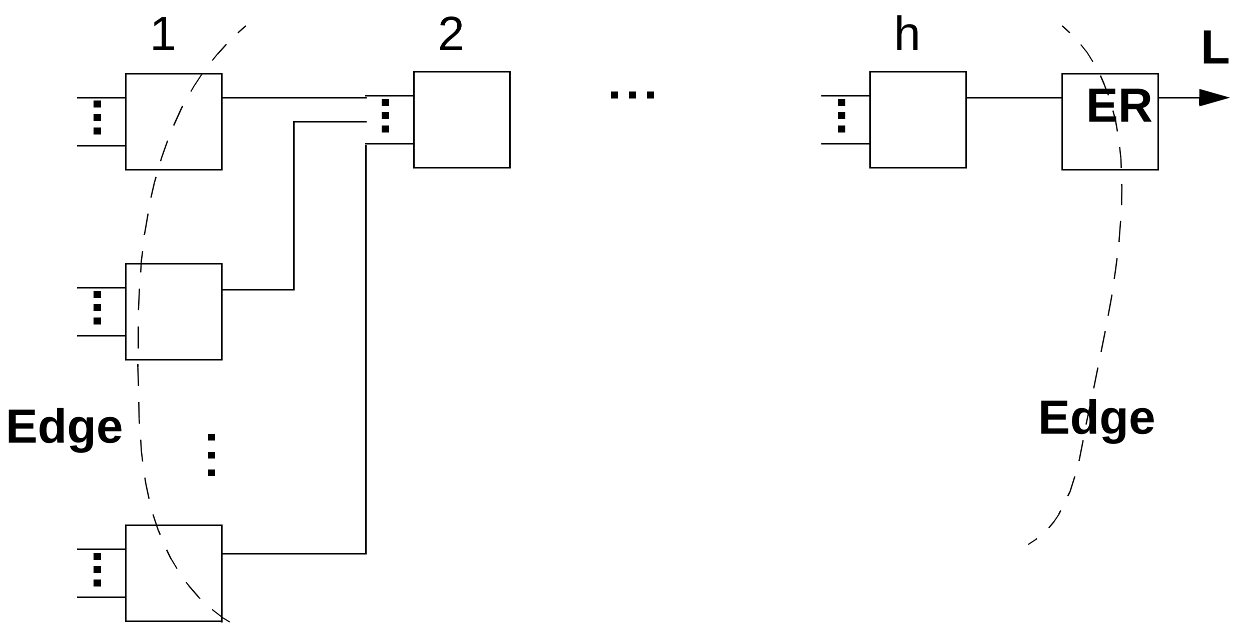


At each link above: who’s the “customer”? What’s “fair”?

Worst Case Clump of “Special” (SP) Packets

Assume: each node has in-degree, i, each internal link has bandwidth **B**, the egress link has bandwidth L, each input may burst b SP packets. How large a clump of SP packets can appear at link L?

If each input has an allocation, then clump(b,i,h) = $b * i^h$



Implications and Considerations

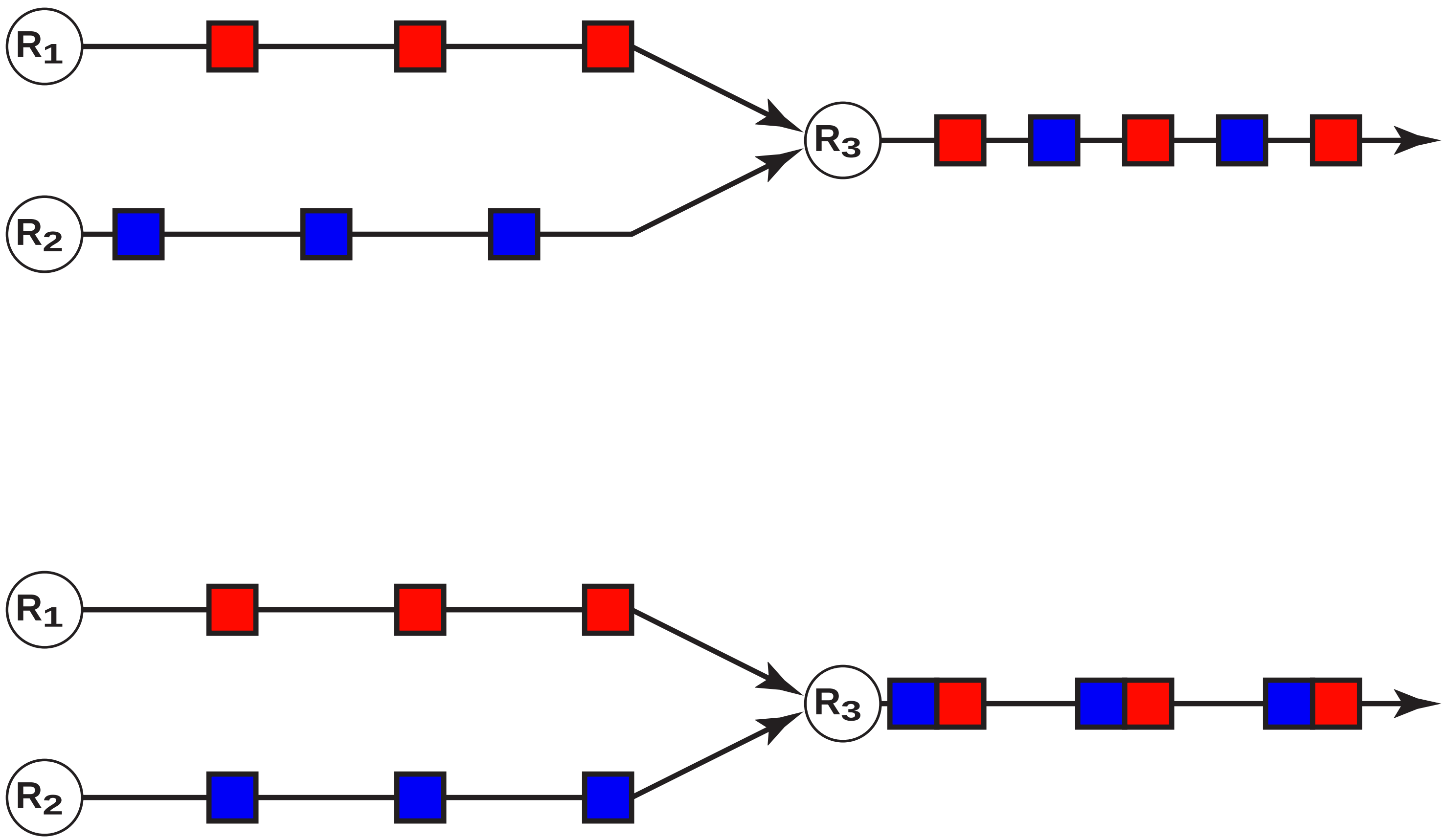
The entire egress link can be used by the SP traffic for the period it takes to transmit clump(b,i,h) packets at rate L even if edge meters are set up to only mark as SP rate $a*L/i^h$ $(a<1.0)$ at burst b

If the total allocation is restricted to a number, $n < a*L/r$ where r is the “subscribed rate” for an input, then the maximum clump is limited to $b*n$ packets.

If $L < B$, then must buffer a clump*(B-L)/B packets to avoid loss. Reasonable values of B/L are in the range from 1 to 100. At B/L of 5, buffering 80% of the clump; at 10, 90%; at 100, 99%.

If $b=1$ and the allocation is retricted to n, the maximum clump of SP packets is n. No loss if can buffer n packets.

Even CBR Streams can form Clumps



Drop Preference

Dave Clark of MIT proposed a service based on RIO droppers and edge marking by "time sliding window" markers

RIO droppers use two '93 RED algorithms running on the same queue. For "outs", dropping starts at a lower average queue size and has a higher probability. The queue average used for the "outs" includes all packets, for the "ins", only the "in" packets are counted

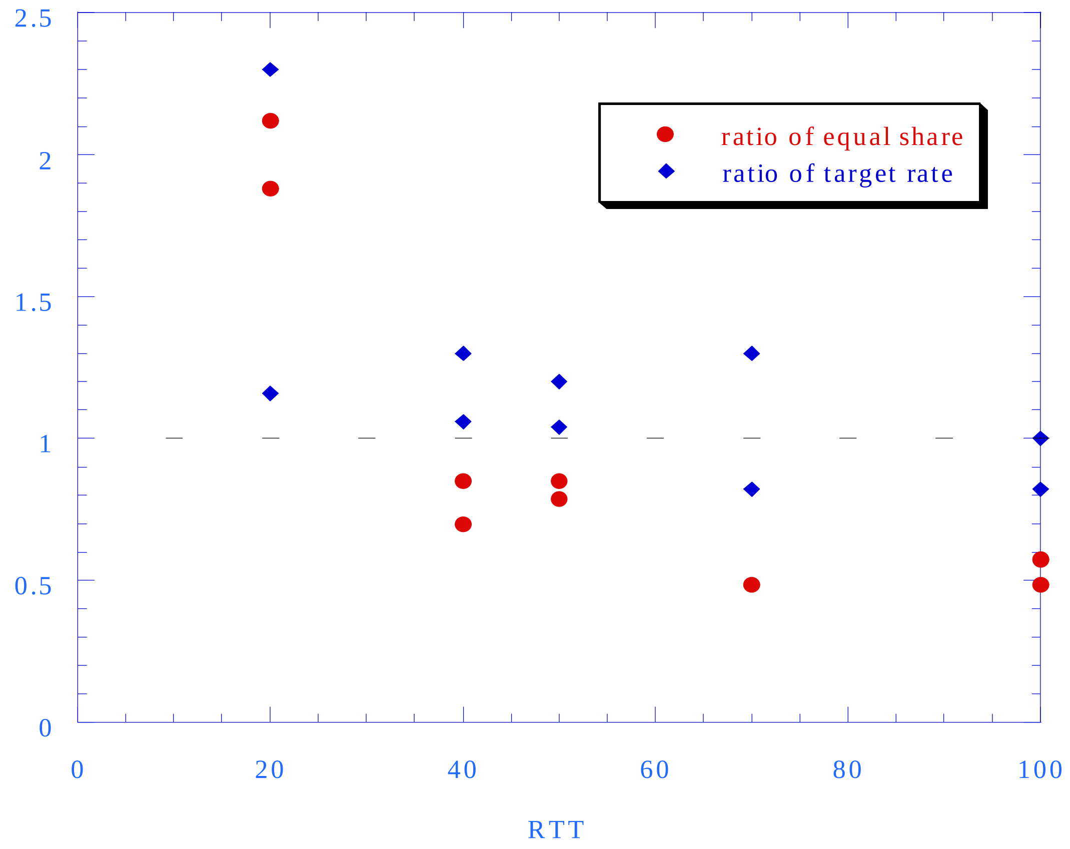
Result were reported in "Explicit Allocation of Best Effort Packet Delivery Services" by Feng and Clark, published in Transactions on Networking (URL given previously)

Table 1 of that document shows results for a topology of a single shared 33 Mbps link with 10 sources on one end, 10 sinks on other, 91% of the link was allocated, each flow had a target rate of either 1 Mbps or 5 Mbps and one of 5 RTTs (20-100ms). I plotted the table to get a better picture of the claimed RTT effect.

Achieved-to-Target vs. RTT

Round/red are the all BE flows results, divided by 3.3 Mbps

Diamond/blue are the RIO results divided by target rate



What do the Results Mean?

Does use of “ins” really overcome RTT effects? Not conclusive, clear, or quantifiable.

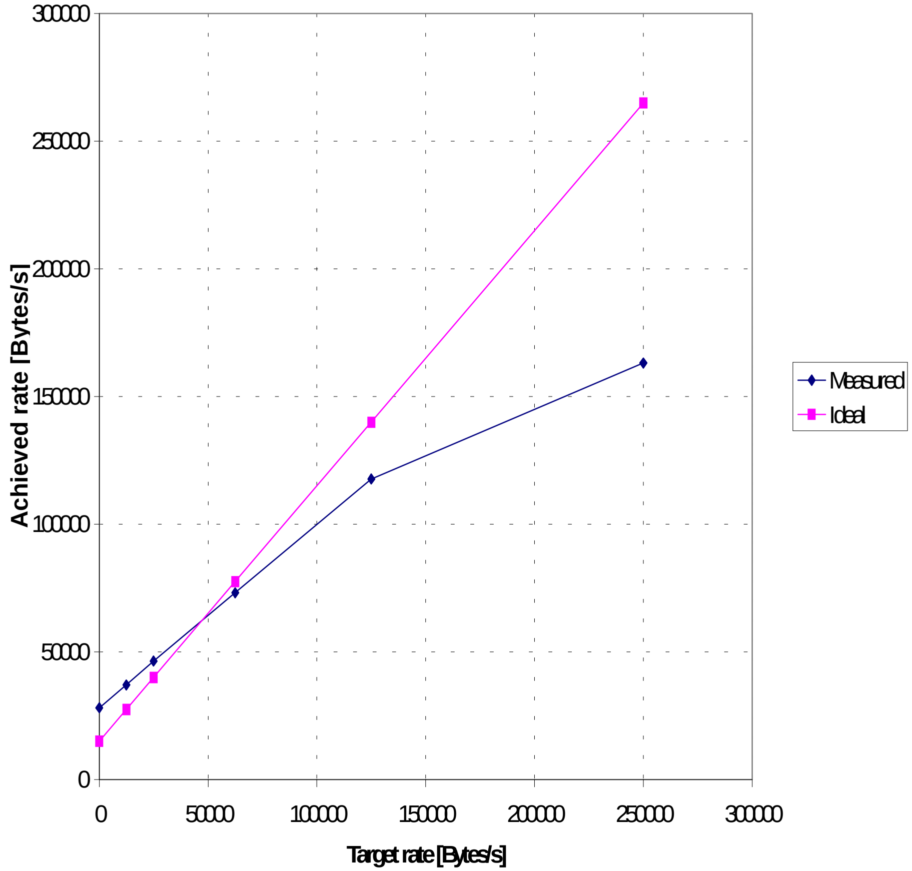
What if more experiments were run? (Is the point at 20 ms anomalous?)

What would happen if there was more BE traffic mixed in (of various types) and more interesting topologies?

Juan-Antonio Ibanez simulated a similar topology, 50 sources and sinks through a single link, using only long-lived TCPs. He found sensitivities to RIO parameters and metering (how the decision to mark “in” is made). An edited form of his report was published as an Internet Draft in July of 1998 and is available at http://ec.eurecom.fr/~ibanez/internet_draft.html.

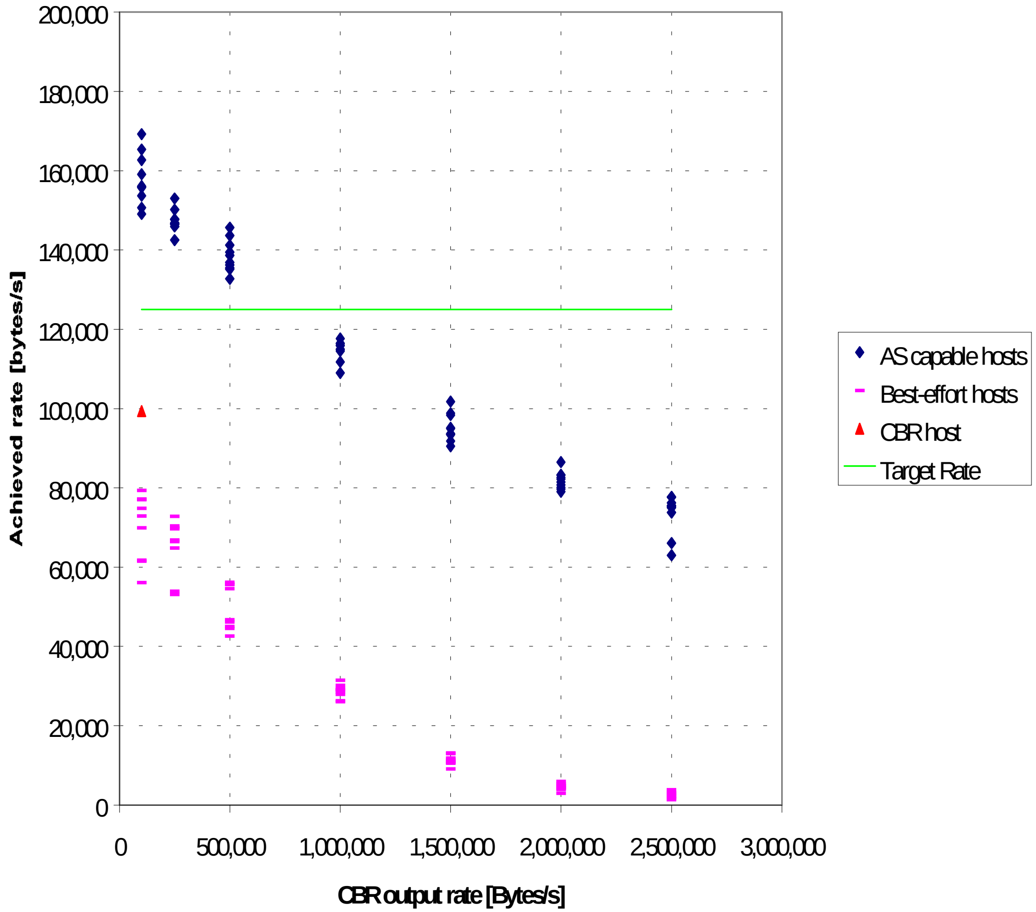
Some of those results follow:

Performance against Target Rates



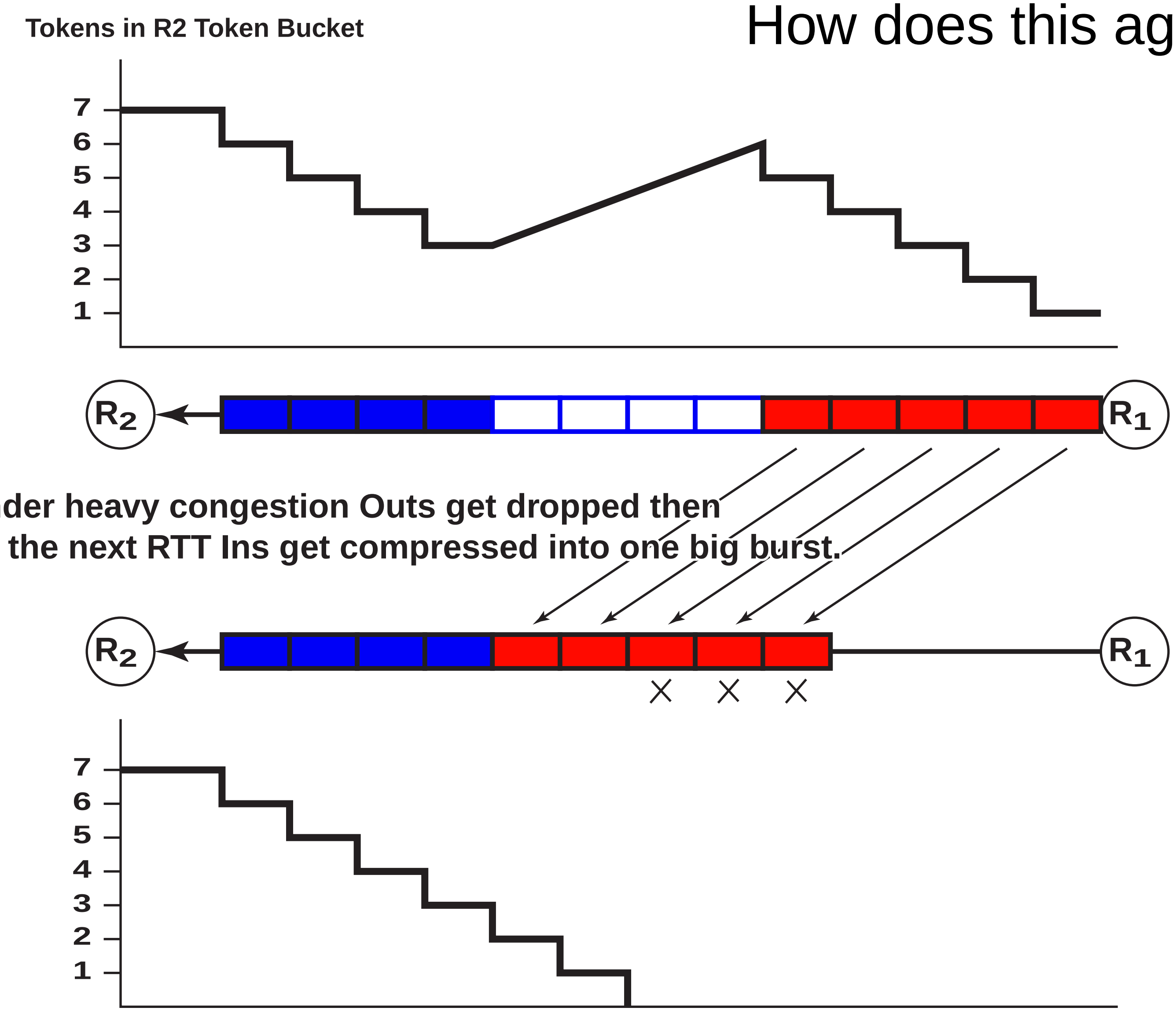
Mix of 20 RIO, 20 BE FTPs where 76% of shared link bandwidth was allocated to “ins”. Five target rates were allocated to sets of four hosts each. The rates were: 0.1, 0.2, 0.5, 1.0, and 5.0 Mbps.

Effect of a Non-responsive Source



Added a non-responsive BE source to 10 RIO, 10 BE FTPs (same RTT, same target rate). The “out” packets degrade the performance.

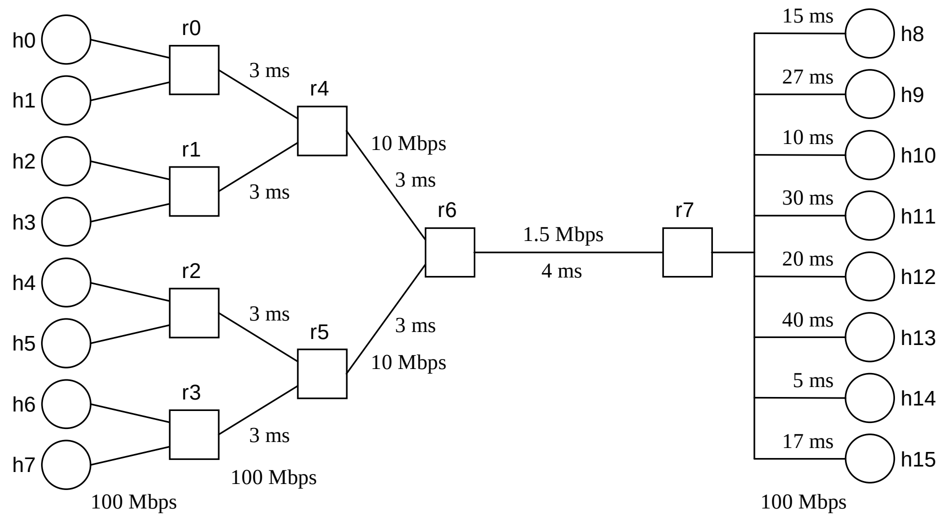
How does this aggregate?



Under heavy congestion Outs get dropped then on the next RTT Ins get compressed into one big burst.

Preliminary Work with a Branchy Topology

Eight FTPs, each with a target rate of 100 Kbps. Police at merges. Five of the FTPs saw one to three drops of “ins”. 7.5% of “ins” remarked “outs”



Host id	0	1	2	3	4	5	6	7
RTT (ms)	50	74	40	80	60	100	30	54
Measured ratio	1.9	1.7	1.7	1.7	2.0	1.5	2.2	1.6

Other Studies: AF-based Services

Claim is that marking non-responsive sources as “way out”, can solve problems. This raises questions many new questions.

Most published evaluations use a simple dumbbell. Allocated traffic is LL FTPs; non-responsive traffic is all considered “undesirable”

One of the better of these is: “Effect of Number of Drop Precedences in Assured Forwarding”, Mukul Goyal, Arian Durresi, Raj Jain, ftp.isi.edu/internet-drafts/draft-goyal-dpstdy-diffserv-01.txt. In best case, appears that “in-profile” traffic needs to be limited to 50% of the link and this for a “dumbbell dancehall” topology where there are two levels of mixing, one for 5 senders, the next merges these 10 merged streams. Traffic model of LL TCPs with unresponsive CBR.

Note that this is all very much untried with the primary traffic of the Internet: web surfing

Other Open Questions about DP

- Is it possible to build a service which aggregates well from this PHB? Resources are allocated for service rate, but bursts are permitted. The effect on the applications is not clear if bursts are all remarked to a higher precedence.
- How do you allocate in order to accomodate bursts of lower drop precedence? The effect on the network is not clear if multi-packet bursts of low drop precedence are allowed.
- How much re-marking happens in a branchier network? Do the remarked packets later get dropped? Is the re-marking “fair”? In what sense can a flow’s “expected capacity” be predicted?
- Can one non-reponsive flow take over the high DP (non-allocated) bandwidth?
- What about when “non-responsive” does not equate to “bad”?

Evaluating EF PHB's Virtual Leased Line

RFC 2598 proposes a "virtual leased line" (VLL) service built from the EF PHB combined with border policers that drop packets exceeding a peak rate. Bursts are not permitted.

Shaping is expected to be employed to conform to the policers, where an "upstream" border will shape its traffic to avoid drop.

With these rules, it is easy to aggregate such a service since the peak rates just add.

The notion of a VLL is that it "looks like a wire" to the application in terms of its performance.

Since it has the properties of low jitter and guaranteed rate, VLL is often cited as a means to provide QoS to VoIP. However, it is intended to provide the equivalent of a dedicated link, useful for *any* mixture of network traffic, potentially useful for VPNs.

Performance Against a Dedicated Link

(Results due to K. Poduri. Were previously presented in 8/98, see: http://www.nren.nasa.gov/workshops3.html)

The percentage of the bottleneck link allocated to “marked” packets varies: 20, 40, and 60 percent.

Eight FTPs were rate-limited and marked for EF PHB.

Results are the average transfer rate of each flow (all were the same).

percentage of bottleneck	rate (Kbps)	measured line rate	measured EF rate
20	100	90	90
40	150	143	143
60	225	213	215

VLL's Forwarding Path Performance

- Forwarding path performance of VLLs for general Internet traffic is frankly not that interesting: as long as the traffic is shaped properly (sufficiently sized holding buffers and good queue management), the VLL acts like a dedicated link at the subscribed rate.
- For VoIP type traffic in a VLL, care about jitter. Values are easy to bound in worst case, but we'd like to understand "typical" behavior and how different EF PHB implementations and allocation percentages affect jitter performance.
- K. Poduri simulated EF performance on the six-hop branchy topology with a mixture of flow types and using a mix of BE traffic (HTTP, FTP, CBR). Included in RFC 2598.

Simple Analysis to Bound EF Jitter

Earlier, noted the two components of jitter: assume a worst case for both the displacement from other EF microflows and the per-hop wait for a full MTU-sized packet of another aggregate. The worst case jitter is the worst case displacement if the next packet of the stream doesn’t queue. (Very unlikely!)

Assume the bottleneck bandwidth, **B**, is the bandwidth of every link in the domain. Worst case is an allocation, **a**, of 50%. Compare to a 30% allocation to see how that affects jitter.

We can refine this to be a sort of “average” worst case by first noting that, at each hop, we wait on average for a half of an MTU-sized packet and that the chances of meeting a packet of another aggregate depend on how highly utilized the link is. The former halves the jitter component due to the number of hops, **h**, and the latter multiplies it times the utilization percentage.

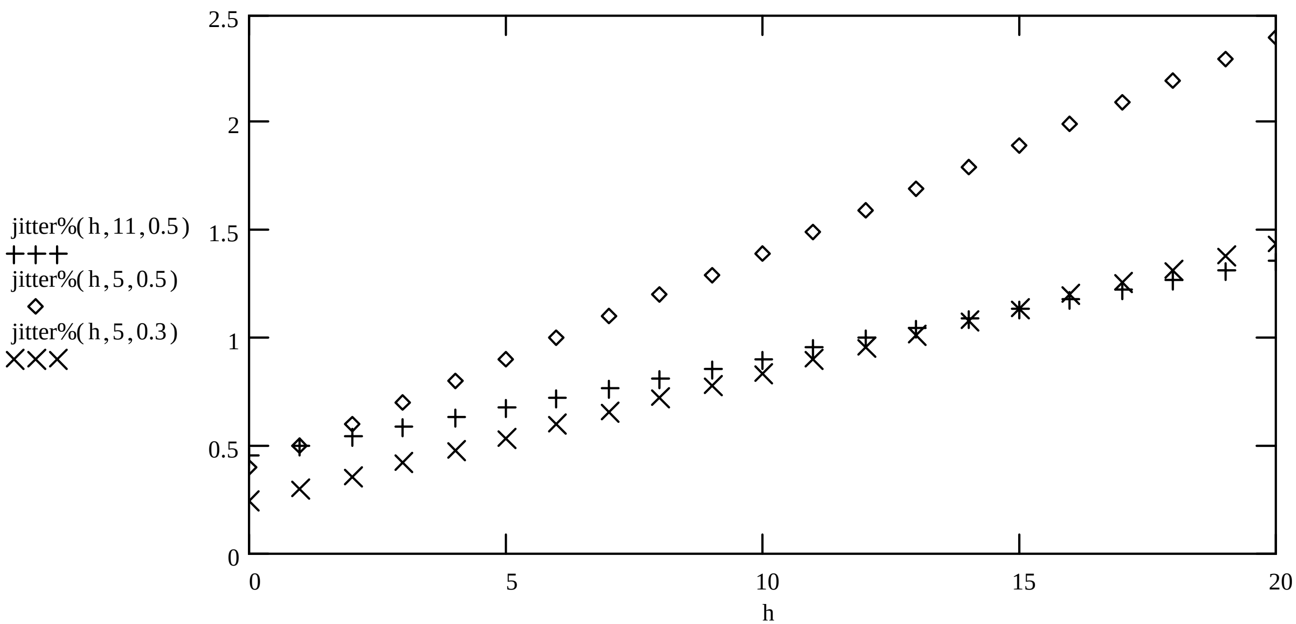
Worst-case Bound on EF Displacement

$$\text{jitter\%}(h,n,a) := \frac{\left[(n-1)\cdot\left(\frac{MTU}{B}\right)\right] + h\cdot\left(\frac{MTU}{B}\right)}{\left(\frac{MTU}{a\cdot\frac{B}{n}}\right)}$$

$$\text{jitter\%}(h,n,a) := \frac{(n-1)+h}{\frac{n}{a}}$$

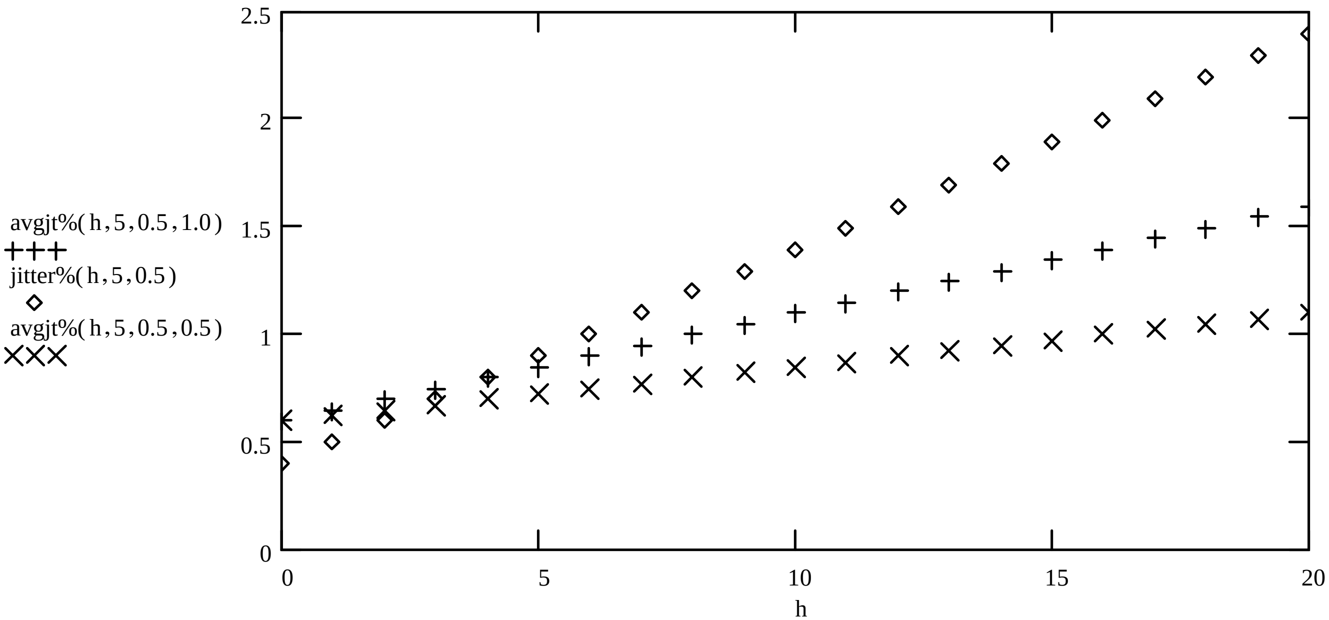
Jitter could be larger as a fraction of a smaller packet size, but the number of bytes and thus the time would remain constant

The maximum allocation of a=0.5 gives the largest jitter



Relaxing the Bound with Utilization and Average

$$\text{avgjt\%}(h,n,a,u) := \frac{(n+1)+\frac{u}{2}\cdot h}{\frac{n}{a}}$$



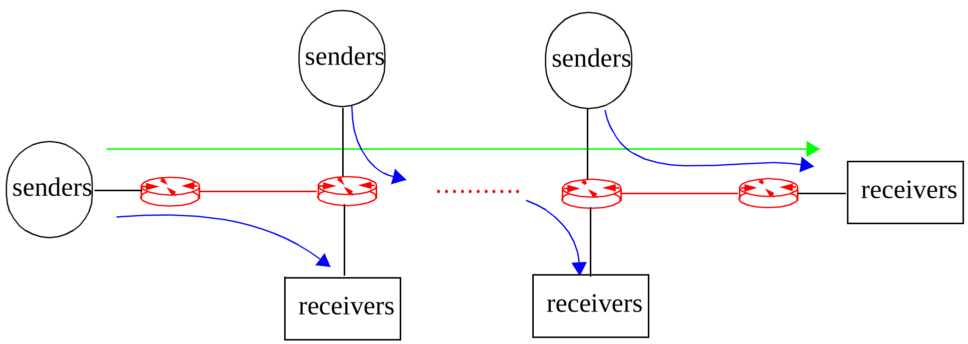
Using an average wait of half an MTU makes a larger difference than varying the link utilization, but both give some feel for what conditions have to occur to have jitter exceed one MTU at rate

Evaluating Jitter for VoIP using VLLs

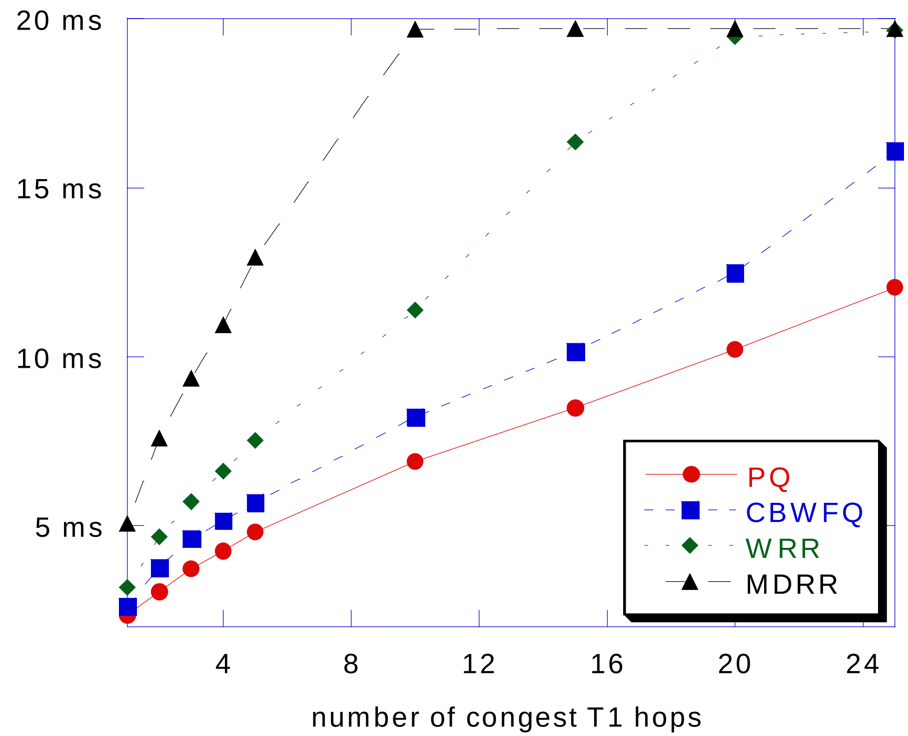
These simulation results focused on a VLL over multiple hops (from Yonghwan Kim of Cisco).

Jitter is the difference between the absolute value of the arrival time differences minus the absolute value of the departure time differences of two adjacent packets, $(|(a_j-a_i)-(d_j-d_i)|)$.

“Freeway” topology of varying numbers of hops. Each hop is 1.5 Mbps.



Median Jitter of VoIP using VLL

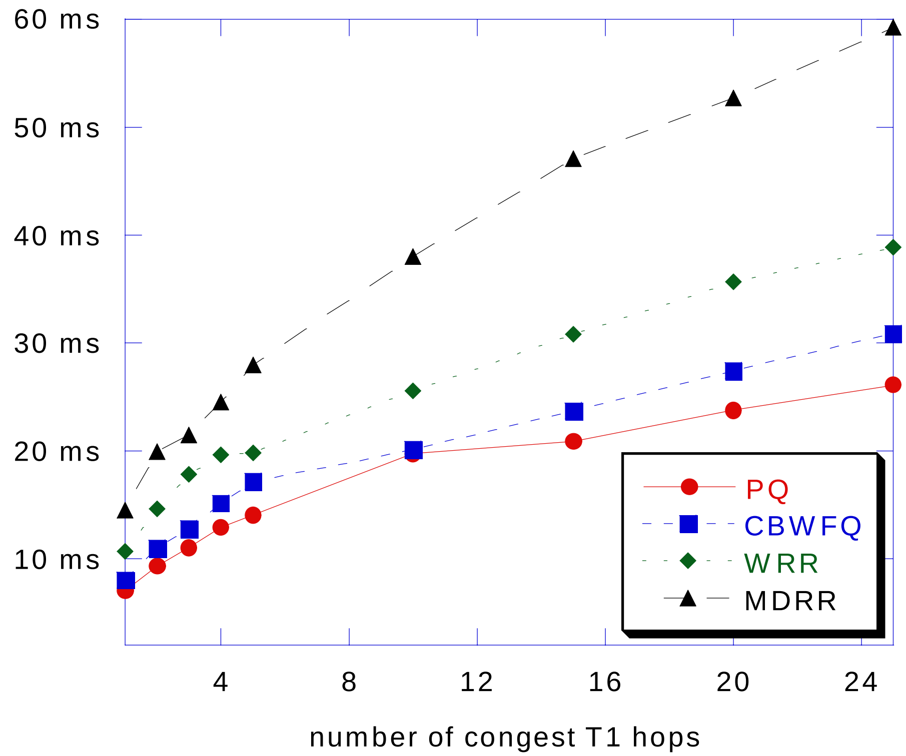


The “VoIP” flows were 24Kbps, 20 ms between 60 byte packets

10% of traffic is EF-marked, 60% gets other “special” treatment

Of the EF packets, half are long (1500 bytes), half short (100 bytes)

95th Percentile Jitter of VoIP using VLL



Note the shape of these curves is much better than the line of the simple analysis for a worst-case bound.

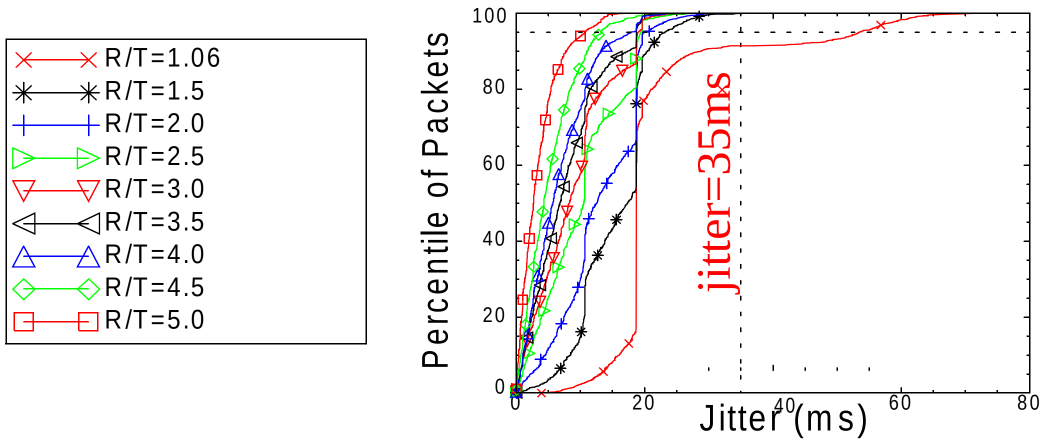
Jitter for a WRR EF Implementation

Uses a single bottleneck topology, six hops, middle is bottleneck

Here there are a total of 5 queues, one of which is EF.

EF gets 20% of each link.

A WRR queue scheduler is used and the ratio of the Rounding weight to the Target rate is varied, R/T.



Considerations to Wrap up With

Many questions are still open about building “better effort” services.

Marketing folks have driven a demand for a service that guarantees a minimum but allows excees when there is capacity in the network. This is a lot harder to do than folks simulating dumbbells think.

Need analysis of the effects of permitting bursts in “guaranteed” traffic.

Need to really understand the behavior of traffic under aggregation before we start to develop a lot of PHBs and services.

Flow-based thinking hampers deployability by introducing complexity and limiting the ability of solutions to scale. What problem does it solve?

QoS and its allocation are needed in order to express desired organizational policies. We should be focused on building the tools to make that possible.

"If you expect to see the final results of your work you simply have not asked a big enough question."   I.F. Stone

Pointers/Topics Not Covered in Talk

Bandwidth Broker architecture as proposed by Van Jacobson

slides available at www-nrg.ee.lbl.gov/talks

RFC 2638 was originally published as an Internet Draft in November '97. Also at ftp://ftp.ee.lbl.gov/papers/dsarch.pdf

Talk by Jacobson given at the Internet2 QoS workshop is available at: www.internet2.edu/media/qos8.ram and the proceedings from that workshop are at www.internet2.edu/qos/may98Workshop/9805-Proceedings.pdf

A Two-Tier Resource Management Model for Differentiated Services Networks. F. Reichmeyer et. al., ftp://ftp.isi.edu/internet-drafts/draft-rotzy-2-tier-management-00.txt

Performance of QoS Agents for Provisioning Network Resources: Olav Schelen et. al., www.cdt.luth.se/~olov/publications. Presents topology-aware QoS agents which it presents as a type of bandwidth broker.

More Pointers (cont'd)

DECIDES BOF, July 1999, www.ietf.org/proceedings/99jul/index.html

From Sigcomm99: www.acm.org/sigcomm/sigcomm99/:

A Flexible Model for Resource Management in Virtual Private Networks: N. Duffield et. al., This paper examines allocating VPNs based on the total ingress and egress amount of traffic of any one customer site.

Proportional Differentiated Services: Delay Differentiation and Packet Scheduling: C. Dovrolis, D. Stiliadis, P.Ramanathan. This paper uses the CSC PHBs in an interesting way.