Report Number: wucse-2009-70

2009

Enabling a Low-delay Internet Service via Built-in Performance Incentives

Maxim Podlesny and Sergey Gorinsky

The single best-effort service of the Internet struggles to accommodate divergent needs of different distributed applications. Numerous alternative network architectures have been proposed to offer diversified network services. These innovative solutions failed to gain wide deployment primarily due to economic and legacy issues rather than technical shortcomings. Our paper presents a new simple paradigm for network service differentiation that accounts explicitly for the multiplicity of Internet service providers and users as well as their economic interests in environments with partly deployed new services. Our key idea is to base the service differentiation on performance itself, rather than price. We... **Read complete abstract on page 2.**

Enabling a Low-delay Internet Service via Built-in Performance Incentives

Maxim Podlesny and Sergey Gorinsky

Complete Abstract:

The single best-effort service of the Internet struggles to accommodate divergent needs of different distributed applications. Numerous alternative network architectures have been proposed to offer diversified network services. These innovative solutions failed to gain wide deployment primarily due to economic and legacy issues rather than technical shortcomings. Our paper presents a new simple paradigm for network service differentiation that accounts explicitly for the multiplicity of Internet service providers and users as well as their economic interests in environments with partly deployed new services. Our key idea is to base the service differentiation on performance itself, rather than price. We design RD (Rate-Delay) network services that give a user an opportunity to choose between a higher transmission rate or low queuing delay at a congested network link. To support the two services, an RD router maintains two queues per output link and achieves the intended ratedelay differentiation through simple link scheduling and dynamic buffer sizing. Our extensive evaluation of the RD network services reports their performance, deployability, and security properties.

2009-70

Enabling a Low-delay Internet Service via Built-in Performance Incentives

Authors: Maxim Podlesny & Sergey Gorinsky

Abstract: The single best-effort service of the Internet struggles to accommodate divergent needs of different distributed applications. Numerous alternative network architectures have been proposed to offer diversified network services. These innovative solutions failed to gain wide deployment primarily due to economic and legacy issues rather than technical shortcomings. Our paper presents a new simple paradigm for network service differentiation that accounts explicitly for the multiplicity of Internet service providers and users as well as their economic interests in environments with partly deployed new services. Our key idea is to base the service differentiation on performance itself, rather than price. We design RD (Rate-Delay) network services that give a user an opportunity to choose between a higher transmission rate or low queuing delay at a congested network link. To support the two services, an RD router maintains two queues per output link and achieves the intended ratedelay differentiation through simple link scheduling and dynamic buffer sizing. Our extensive evaluation of the RD network services reports their performance, deployability, and security properties.

Type of Report: Other

Enabling a Low-delay Internet Service via Built-in Performance Incentives

Maxim Podlesny and Sergey Gorinsky

Technical report WUCSE-2009-70
Applied Research Laboratory
Department of Computer Science and Engineering, Washington University in St. Louis
One Brookings Drive, St. Louis, MO 63130-4899, USA
{*podlesny,gorinsky*}*@arl.wustl.edu*

August 2009

***Abstract*—The single best-effort service of the Internet struggles to accommodate divergent needs of different distributed applications. Numerous alternative network architectures have been proposed to offer diversified network services. These innovative solutions failed to gain wide deployment primarily due to economic and legacy issues rather than technical shortcomings. Our paper presents a new simple paradigm for network service differentiation that accounts explicitly for the multiplicity of Internet service providers and users as well as their economic interests in environments with partly deployed new services. Our key idea is to base the service differentiation on performance itself, rather than price. We design RD (Rate-Delay) network services that give a user an opportunity to choose between a higher transmission rate or low queuing delay at a congested network link. To support the two services, an RD router maintains two queues per output link and achieves the intended rate-delay differentiation through simple link scheduling and dynamic buffer sizing. Our extensive evaluation of the RD network services reports their performance, deployability, and security properties.**

I. Introduction

Numerous architectures with diversified network services have been proposed to remedy the inability of the Internet architecture to serve different applications in accordance with their diverse communication needs. IntServ (Integrated Services) [9], a prominent representative of the architectural innovations, offers users a rich choice of services that include guarantees on end-to-end throughput and delay within a packet flow. The IntServ design incorporates complicated admission control and link scheduling mechanisms such as WFQ (Weighted Fair Queuing) [12] and WF^{2}Q (Worst-case Fair Weighted Fair Queueing) [7]. While IntServ failed to gain ubiquitous adoption, early IntServ retrospectives attributed the failure to the complexity of supporting the per-flow performance guarantees, especially in busy backbone routers. DiffServ (Differentiated Services) [8], a subsequently proposed architecture, addresses the scalability concerns by restricting complex operations to the Internet edges and offering just few services at the granularity of traffic classes, rather than individual flows. DiffServ did not deploy widely either in spite of its simpler technical design.

The deployment failures of the diversified-service architectures suggest that technical merits of an innovative solution is not the main factor in determining its success. Economic and legacy issues become a crucial consideration because the current Internet is a loose confederation of infrastructures owned by numerous commercial entities, governments, and private individuals [11]. The multiplicity of the independent stakeholders and their economic interests implies that partial deployment of a new service is an unavoidable and potentially long-term condition. Despite the partial deployment, the new service should be attractive for adoption by legacy users and ISPs (Internet Service Providers).

Our paper explores a simple novel paradigm for network service differentiation where deployability is viewed as the primary design concern. We explicitly postulate that partial deployment is unavoidable and that the new design should be attractive for early adopters even if other ISPs or users refuse to espouse the innovation. Moreover, we require that the benefits of network service diversification should not come at the expense of legacy traffic. The imposed constraints are potent. In particular, they imply that the new architecture cannot assume that traffic shaping, metering, pricing, billing, or any other added functionality will be supported by most ISPs, even by most ISPs on Internet edges.

To resolve the deployability challenge, we utilize built-in performance incentives as a basis for network service differentiation. While prior studies have established a fundamental trade-off between queuing delay and link utilization [34], [23], the Internet practice favors full utilization of bottleneck links at the price of high queuing delay. Unfortunately, delay-sensitive applications suffer dearly from the long queues created by throughput-greedy applications at shared bottleneck links. Our proposal of RD (Rate-Delay) services resolves this tension by offering two classes of service: an R (Rate) service puts an emphasis on a high transmission rate, and a D (Delay) service supports low queuing delay. Each of the services is neither better nor worse per se but is simply different, and its relative utility for a user is determined by whether the user's application favors a high rate or low delay. Hence, the RD

architecture provides the user with an incentive and complete freedom to select the service class that is most appropriate for the application. Packet marking in the sender realizes the selection of the R or D service.

The interest of users in the low-delay D service is viewed as an indirect but powerful incentive for ISPs to adopt the RD services. By switching to the RD architecture, an ISP attracts additional customers and thereby increases revenue. We also envision an RD certification program championed by early adopters. Being RD-certified is expected to give an ISP a differentiation advantage over legacy ISPs when competing with them for users and provider peering agreements. The RD certification will thereby act as a catalyst for virulent deployment of the RD architecture.

The RD design achieves its objectives primarily through packet forwarding in routers. The RD router serves each output link from two FIFO (First-In First-Out) queues and supports the intended rate-delay differentiation through dynamic buffer sizing and simple transmission scheduling. The RD router treats legacy traffic as belonging to the R class. The simplicity of the RD forwarding makes the design amenable to easy implementation even at high-capacity links.

The overall architecture remains in the best-effort paradigm. Neither R nor D service offers any rate or loss guarantees. Besides, the architecture modifies forwarding but not routing. Although the RD services provide users and ISPs with incentives to adopt the services, the architecture does not eliminate most security problems of the Internet. In particular, a malicious ISP can disrupt the rate and delay characteristics of transient RD traffic. While security is not the main focus of this study, we believe that the RD services do not introduce any fundamentally new vulnerabilities. For example, a user can mark some packets as R-class and other packets as D-class to increase throughput. However, such behavior is essentially the same as the well-known Internet technique of running multiple flows in parallel. Moreover, the two-queue RD design alleviates some existing threats. For example, if a D flow transmits excessively to create heavy losses for other flows at the shared bottleneck link, the RD router limits the damage from the denial-of-service attack to the D class and, thus, preserves the high transmission rates of concurrent legacy and R flows.

We present an experimental study that sheds some light on the security properties of the RD architecture. The experiments are certainly not exhaustive. New behavioral patterns induced by the RD architecture and their security aspects require thorough separate investigation. More generally, it will be interesting to examine whether design for incremental deployment is intrinsically less robust and whether the focus in securing such architectures needs to be shifted from purely technical to legal mechanisms.

In the work, in which the RD services were originally proposed [32], to ensure strict delay guarantees the router kept per-packet arrival times. In this paper, we present a new version of the RD router implementation that does not require to support any per-packet state. The key difference of the new version is a use of new control rule for a buffer size of the D queue.

The rest of the paper has the following structure. Section II presents our design principles. In Section III we describe the conceptual framework of the RD services. Section IV clarifies analytical foundations for RD router operation. In Section V we deliver details of our design. Section VI provides the theoretical analysis of the design. Section VII reports the extensive performance evaluation of the RD services. Sections VIII and IX report our assessment of application-perceived performance for VoIP (Voice over the Internet Protocol) and web browsing respectively. Section X discusses related work. In Section XI we suggest directions for future work. Finally, Section XII concludes the paper with a summary of its contributions.

II. Model and Principles

In our model, the Internet is an interconnection of network domains owned and operated by various ISPs. ISPs generate revenue by selling network services to their direct customers. Users are the customers whose applications run at end hosts and send flows of packets over the Internet. In general, a network path that connects the end hosts of a distributed application traverses the infrastructure belonging to multiple independent ISPs.

Different applications have different communication needs. The single best-effort service of the current Internet matches the diverse interests of the users imperfectly. In response to this tension, numerous architectures with diversified network services have been proposed. Although technically brilliant, even the best of the proposals failed to gain wide deployment. We attribute the failures to ignoring the serious economic challenges of deploying a new service in a confederated infrastructure governed by numerous independent stakeholders. Instead of treating the deployment as an afterthought, we base our design on principles that explicitly acknowledge the multiplicity of Internet parties and their economic rationale in deciding whether to adopt new services.

First, we explicitly recognize that partial deployment is an unavoidable and potentially long-term condition for any newly adopted service. Hence, the new design should be attractive for early adopters even if other ISPs or users refuse to embrace the innovation:

Principle 1: A new service should incorporate incentives for both ISPs and end users to adopt the service despite the continued presence of legacy traffic or other ISPs that do not espouse the new service.

The above principle has a more specific but nevertheless important implication that the new design should not worsen the service provided to legacy Internet users. Doing otherwise is against the economic interests of ISPs due to the danger of losing a large number of current customers who keep communicating via legacy technologies. This consideration leads us to the following principle:

Principle 2: Adoption of a new service should not penalize legacy traffic.

III. Conceptual Design

Below, we apply the principles from Section II to derive a conceptual design for Rate-Delay (RD) services, our solution to the problem of network service differentiation. As the name reflects, the RD services enable a user to choose between a higher transmission rate or short queuing delay at a congested network link.

Our Principle 1 prescribes providing both end users and ISPs with incentives for early adoption of the RD services. The constraint of the partial deployment excludes the common approach of pricing and billing, e.g., because a user should be able to opt for the RD services despite accessing the Internet through a legacy ISP that provides no billing or any other support for service differentiation. With financial incentives not being an option, our key idea is to make the performance itself a cornerstone of the service differentiation. While the performance is subject to a fundamental trade-off between queuing delay and link utilization [34], [23], different applications desire different resolutions to the tension between the two components of the performance. Hence, the RD services consist of two classes:

- R (Rate) service puts an emphasis on a high transmission rate, and
- D (Delay) service supports low queuing delay.

Each of the two services is neither better nor worse per se but is simply different, and its relative utility for a user is determined by whether the user's application favors a high rate or low delay. Since the network services are aligned with the application needs, each user receives an incentive to select the service of the most appropriate type, and the RD service architecture empowers the user to do such selection by marking the headers of transmitted packets.

An ISP finds the RD services attractive due to the potential to boost revenue by adding customers who are interested in the D service. We envisage an RD certification program championed by a nucleus of early adopters. The RD certification will serve as a catalyst for virulent deployment of the RD architecture because being RD-certified will give an ISP a differentiation advantage over legacy ISPs when competing with them for users and provider peering agreements.

To support the RD services on an output link, a router maintains two queues for packets destined to the link. We refer to the queues as an R queue and D queue. Depending on whether an incoming packet is marked for the R or D service, the router appends the packet to the R or D queue respectively. The packets within each queue are served in the FIFO (First-In First-Out) order. Whenever there is data queued for transmission, the router keeps the link busy, i.e., the RD services are work-conserving.

By deciding whether the next packet is transmitted from the R or D queue, the router realizes the intended rate differentiation between the R and D services. In particular, the link capacity is allocated to maintain a rate ratio of

$$k = \frac{r_R}{r_D} > 1 \tag{1}$$

where r_R and r_D refer to per-flow forwarding rates for packet flows from class R and D respectively.

The router supports the desired delay differentiation between the R and D services through buffer sizing for the R and D queues. As common in current Internet routers, the size of the R buffer is chosen large enough so that the oscillating transmission of TCP (Transmission Control Protocol) [26] and other legacy end-to-end congestion control protocols utilizes the available link rate fully. The D buffer is configured to a much smaller dynamic size to ensure that queuing delay for each forwarded packet of the D class is small and at most d. The assurance of low maximum queuing delay is attractive for delay-sensitive applications and easily verifiable by outside parties. An interesting direction for future studies is an alternative design for the D service where queuing delay stays low on average but is allowed to spike occasionally in order to support a smaller loss rate.

In agreement with our overall design philosophy, parameters k and d are independently determined by the ISP that owns the router. The ISP uses the parameters as explicit levers over the provided RD services. Our subsequent experimental study reveals suggested values for parameters k and d.

As per our Principle 2, adoption of the RD services by an ISP should not penalize traffic from legacy end hosts. While the R service and legacy Internet service are similar in putting the emphasis on a high transmission rate rather than low queuing delay, the legacy traffic and any other packets that do not explicitly identify themselves as belonging to the D class are treated by an RD router as belonging to the R class, i.e., the router diverts such traffic into the R queue. Since those flows that opt for the D service acquire the low queuing delay by releasing some fraction of the link capacity, the adopters of the D service also benefit the legacy flows by enabling them to communicate at higher rates.

Due to the potentially partial deployment of the RD services, R and D flows might be bottlenecked at a link belonging to a legacy ISP. Furthermore, the R and D flows might share the bottleneck link with legacy traffic. This has an important design implication that end-to-end transmission control protocols for the R and D services have to be compatible with TCP.

IV. Analytical Foundation

While Section III outlined the conceptual design of the RD services, we now present an analytical foundation for our specific implementation of RD routers.

A. Notation and assumptions

Consider an output link of an RD router. Let C denote the link capacity and n be the number of flows traversing the link. We use n_R and n_D to represent the number of flows from the R and D class respectively. Since the router treats legacy traffic as belonging to the R class, we have

$$n_R + n_D = n. \tag{2}$$

Notation	Semantics
x	class of the service, R or D
n_x	number of flows from class x
L_x	amount of data transmitted from queue x since the last update of L_x
B_x	buffer allocation for queue x
q_x	size of queue x
p	packet
t_p	arrival time of p

Fig. 1. Internal variables of the RD router algorithms in Figures 3, 4, and 5.

For analytical purposes, we assume that both R and D queues are continuously backlogged and hence

$$R_R + R_D = C \tag{3}$$

where R_R and R_D refer to the service rates for the R and D queues respectively. Also, we assume that every flow within each class transmits at its respective fair rate, r_R or r_D:

$$R_R = n_R r_R \tag{4}$$

and

$$R_D = n_D r_D. \tag{5}$$

Our experiments with dynamic realistic traffic including a lot of short-lived flows confirm that the above assumptions do not undermine the intended effectiveness of the RD services in practice.

We denote the sizes of the R and D queues as q_R and q_D respectively and the buffer allocations for the queues as B_R and B_D respectively. If the corresponding buffer does not have enough free space for an arriving packet, the router discards the packet.

B. Sizing and serving the R and D queues

Combining equations (1), (3), (4), and (5), we determine that the service rates for the R and D queues should be respectively equal to

$$R_R = \frac{kn_R C}{n_D + kn_R}, \tag{6}$$

and

$$R_D = \frac{n_D C}{n_D + kn_R}. \tag{7}$$

To ensure that queuing delay for any packet forwarded from the D queue does not exceed d, the buffer allocation for the queue should be bounded from above as follows:

$$B_D = \lfloor R_D(d-w) \rfloor^+ \tag{8}$$

where:

$$w = \frac{2}{C}\Big(S_D^{max}\frac{kn_R}{n_D} + S_R^{max}\Big) \tag{9}$$

In Section VI we prove that equation (8) indeed ensures bounded queuing delay. Taking equation (7) into account, we establish the following buffer allocation for the D queue:

$$B_D = \Big\lfloor \frac{n_D C(d-w)}{n_D + kn_R} \Big\rfloor^+. \tag{10}$$

Parameter	Semantics
d	upper bound on queuing delay experienced by a packet of class D
k	ratio of per-flow rates for classes R and D
T	update period
E	flow expiration period
b	timestamp vector size

Fig. 2. Parameters of the RD router algorithms.

In practice, we expect B_D to be much smaller than overall buffer B that the router has for the link. Manufacturers equip current Internet routers with substantial memory so that router operators could configure the link buffer to a high value B_{max}, chosen to support throughput-greedy TCP traffic effectively [38]. Thus, we recommend to allocate the buffer for the R queue to the smallest of $B - B_D$ and B_{max} (and expect B_{max} to be the common setting in practice):

$$B_R = \min\left\{B_{max};\ B - \Big\lfloor \frac{n_D C(d-w)}{n_D + kn_R} \Big\rfloor^+\right\}. \tag{11}$$

V. Design Details

A. End hosts

As per our discussion at the end of Section III, the RD services do not demand any changes in end-to-end transport protocols. The only support required from end hosts is the ability to mark a transmitted packet as belonging to the D class. We implement this requirement by employing bits 3-6 in the TOS (Type of Service) field of the IP (Internet Protocol) datagram header [33]. To choose the D service, the bits are set to 1001. The default value of 0000 corresponds to the R service. Thus, the RD services preserve the IP datagram format.

B. Routers

The main challenge for transforming the analytical insights of Section VI into specific algorithms for RD router operation lies in the dynamic nature of Internet traffic. In particular, while expressions (6), (7), (10), and (11) depend on n_R and n_D, the numbers of R and D flows change over time. Hence, the RD router periodically updates its values of n_R and n_D. Sections V-B1, V-B2, and V-B3 describe our algorithms for processing a packet arrival, serving the queues, and updating the algorithmic variables at the RD router respectively. Figure 1 summarizes the internal variables of the algorithms. In addition to the internal variables, a number of parameters characterize the RD router operation. Figure 2 sums up these parameters.

1) Processing a packet arrival: Figure 3 presents our simple algorithm for dealing with packet arrivals. When the router receives a packet destined to the link, the router examines bits 3-6 in the TOS in the packet header to determine whether the packet belongs to class R or D. If the corresponding buffer is already full, the router discards the packet. Otherwise, the

```
p ← the received packet;
x ← the class of p;
S ← size of p;
if q_x + S ≤ B_x
        append p to the tail of queue x;
        q_x ← q_x + S;
        if x = D
            t_p ← current time;
 else
        discard p
```

Fig. 3. RD router operation upon receiving a packet destined to the link.

```
\* select the queue to transmit from *\
if q_R > 0 and q_D > 0
      if kn_R L_D > n_D L_R
          x ← R;
      else
          x ← D;
else \* exactly one of the R and D buffers is empty *\
      x ← class of the non-empty buffer;
p ← first packet in the x queue;
S ← size of p;
if p != null
      \* update the L variables *\
      if q_R > 0 and q_D > 0
          L_x ← L_x + S;
          δL ← (L_R n_D)/(k n_R) − L_D;
          if δL < 0 δL ← 0;
      else \* only D buffer is empty *\
      if q_R > 0 and q_D = 0
          L_R ← 0; L_D ← 0;
      else \* only R buffer is empty *\;
            if δL > 0 δL ← δL − S;
            if δL > 0 L_D ← −δL;
            else
                L_D ← 0;
            L_R ← 0;
      transmit p into the link;
      q_x ← q_x − S
```

Fig. 4. Router operation when the RD link is idle, and the link buffer is non-empty.

router appends the packet to the tail of the corresponding queue.

2) Serving the R and D queues: The original version of the algorithm serving the queues [32] used the arrival times of enqueued D packets to ensure that queuing delay of forwarded D packets does not exceed upper bound d. More specifically, if the packet at the head of the D queue has been queued for longer, the router discards the packet. The situation might arise due to the dynamic nature of Internet traffic: since the population of flows changes, the service rate for the D queue might decrease after the packet arrives. The version presented in this paper supports delay constraint without keeping per packet arrivals times. It is realized through using a new control rule for the size of the buffer for the D queue, B_D.

```
B_D^old ← B_D;
update n_R and n_D as in [28];
update B_R and B_D according to equations (11),(10);
if δL > 0 L_D ← −δL;
   else L_D ← 0;
L_R ← 0;
if q_D > B_D or B_D^old < B_D
   discard all packets from the D queue;
   q_D ← 0;
else while q_R > B_R
          p ← last packet in the R queue;
          S ← size of p;
          discard p;
          q_R ← q_R − S
```

Fig. 5. Update of the RD algorithmic variables upon timeout.

Figure 4 reports further details of the algorithm for serving the R and D queues. While the RD services are work-conserving, the router transmits into the link whenever the link buffer is non-empty. Since the router can transmit at most one packet at a time, the intended split of link capacity C into service rates R_R and R_D can be only approximated. The router does so by:

- monitoring L_R and L_D, the amounts of data transmitted from the R and D queues respectively since the last reset of these variables;
- transmitting from such queue that $\frac{L_R}{L_D}$ approximates $\frac{R_R}{R_D} = \frac{kn_R}{n_D}$ most closely.

More specifically, when $kn_R L_D > n_D L_R$, the router transmits from the R queue; otherwise, the router selects the D queue.

We derived the above algorithm from the assumption that all flows within a class transmit at the same fair rate, r_R or r_D. While the assumption is clearly unrealistic, one specific problematic scenario occurs when the total transmission rate of the D flows is much less than $n_D r_D$, the maximum service rate for the D queue. Then, a throughput-greedy flow has an incentive to mark its packets as D packets and thereby achieve a much higher forwarding rate than the one offered by the intended R service. In our simulations we consider this scenario, and the results reveal that the unintended selection of the D service by the throughput-greedy flow does not disrupt the D service.

To avoid overflow of the values L_D and L_R, they are periodically assigned to zero values. In particular, the assigning happens in two cases. The first one occurs if only one queue is backlogged. In this case, both the values of L_D and L_R are zeros until both the queues get backlogged again. The second one happens upon a timeout for recalculating control parameters. The problem is that exceeding the delay constraint

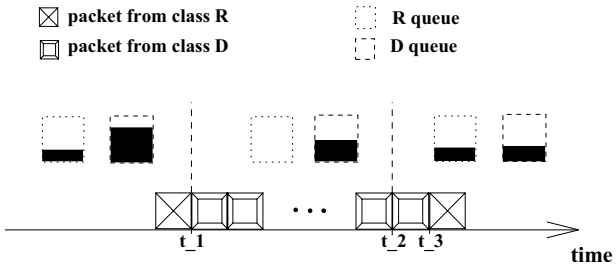


Fig. 6. The exceeding of the delay constraint: combination of both and one queue backlogged.

may occur right after the assignment of zero values to L_D and L_R, as we discuss below.

Figure 6 illustrates the first case corresponding to only one backlogged queue. We will refer to the ratio between $L_D(t)$ and $L_R(t)$ as $\beta(t)$, and to the ratio between n_D and kn_R as α. During time interval $[t_1 - \delta t; t_1)$, $\delta t > 0$, both the queues R and D are backlogged. Then only one queue D gets backlogged during time interval $[t_1; t_2)$. Finally, both the queues are backlogged again starting from time t_2. Let us denote the amount of traffic sent during time interval $[t_1; t_3]$ as $\delta L'_D$. Besides, let us assume that traffic is such that $\beta(t_1) < \alpha$. Then if the following inequality:

$$\frac{L_D(t_1) + \delta L'_D}{L_R(t_1)} < \alpha \tag{12}$$

takes place, it is possible to exceed the delay constraint for D packets that arrive after time t_3, because R_D is less than the required rate during time interval $[t_1; t_3)$, and that is "forgotten" by the algorithm as L_D and L_R are assigned to zero values at time t_1. Applying similar speculations, one can demonstrate exceeding the delay constraint in the second case related to the timeout expiration. To handle the described problems, we use a special parameter δL that tracks the amount of traffic that needs to be departed from the D queue to avoid exceeding the delay constraint.

3) Updating the algorithmic variables: Whereas n_R and n_D play important roles in the presented RD router algorithms, we compare two approaches to computing the numbers of flows: explicit notification from end hosts and independent inference by the router. Since our design principles allow a possibility that many hosts do not embrace the RD services, it is likely that the router serves many legacy flows and needs to do at least some implicit inference. Furthermore, since we favor solutions with minimal modification of the current infrastructure, the router in our RD implementation estimates n_R and n_D without any help from end hosts.

To estimate the numbers of flows, we independently apply the timestamp-vector algorithm [28] to each of classes R and D. Our experiments confirm the excellent performance of the algorithm. Using a hash function, the algorithm maps each received packet into an element of the array called a timestamp vector. The timestamp vector accommodates b elements. The algorithm inspects the timestamp vector with period T and considers a flow inactive if the timestamp vector did not register any packets of the flow during the last period E. Following the guidelines in [16] and assuming $E = 1$ s, 10^5 active flows, and standard deviation $\epsilon = 0.05$, we recommend $b = 18,000$ as the default setting for the timestamp vector size.

The RD router updates n_R and n_D with period T. At the same time, the router updates the buffer allocations for the R and D queues. Even if n_R or n_D is zero, the router allocates a non-zero buffer for each of the queues. Our experimental results suggest that the specific allocation split is not too important; in the reported experiments, we initialize the buffer allocations to $B_D = \frac{4Cd}{4+k}$ and $B_R = \min\{B_{max};\ B - B_D\}$, which correspond to the 1:4 ratio between the numbers of flows from classes R and D. If both n_R and n_D are positive, the router updates the buffer allocations according to equations (10) and (11).

The update of B_R and B_D can make one of them smaller than the corresponding queue size. Figure 5 describes how the router deals with this issue. If the updated B_R is less than q_R, the router discards packets from the tail of queue R until q_R becomes at most B_R. The discards ensure that the D service receives the intended buffer allocation. If the update decreases B_D, i.e., $B_D^{old} > B_D$, where B_D^{old} is the previous value of the size of the D buffer, the router flushes all packets from queue D to ensure that neither of them will be queued for longer than d. The longer queueing might occur otherwise because the decrease of B_D also proportionally reduces the service rate for queue D.

Although the D buffer is typically small, discarding the burst of packets might affect the loss rate negatively and be even unnecessary because it might be still possible to forward at least some of the discarded D packets in time despite the reduced service rate. We explore the influence of discarding the packets in Section VII.

To select update period T, we observe that reducing T increases the computational overhead. Also, the operation might become unstable unless T is much larger than d. However, with larger T, the design responds slower to changes in the network conditions. Our experiments show that $T = 400$ ms offers a reasonable trade-off between these factors.

VI. Analysis

In this section we show that configuring the buffer size of queue D of the RD router design through equations (8) and (9) guarantees the strict support of the delay constraint without tracking packet arrival times or discarding packets at the head of the D queue.

We can distinguish two different cases of backlogging at the RD link depending on the number of backlogged queues. Due to space constraints, we do not consider in details the case corresponding to one backlogged queue. Summarizing that case, to avoid exceeding the delay constraint we introduce one more counter, besides L_D and L_R, that tracks D traffic needed to be departed from the D queue. We examine the more interesting case when both the D and R queues are backlogged. Let us consider an arbitrary packet p from the D queue. We

assume that p arrives at the D queue at time t_a and departs form the D queue at time t_d. Let us suppose that:

$$\frac{L_D(t_a)}{L_R(t_a)} = \alpha + \delta(t_a) \tag{13}$$

$$\frac{L_D(t_d)}{L_R(t_d)} = \alpha + \delta(t_d) \tag{14}$$

where $L_R(t_a) > 0$ and $L_R(t_d) > 0$. We consider the scenario where both the D and R queues are backlogged during time period $[t_a; t_d]$. We will refer to the traffic sent from the D and R queues during time period $[t_a; t_d]$ as δL_D and δL_R, $\delta L_R > 0$, respectively. We can distinguish two cases of the RD buffer configuration. The first one corresponds to a buffer of zero size, and, therefore, gives no queuing delay. The second case reflects a non-zero buffer, i.e., $d - w > 0$, where d is the delay constraint, w is defined by equation 9. Our analysis considers the second case. Let us prove the following:

Theorem 1: Maximum queuing delay d-w is supported for any packet p within any traffic pattern if:

$$\frac{\delta L_D}{\delta L_R} \geq \alpha \tag{15}$$

Proof: Indeed, if inequality (15) takes place, then $\frac{R_D}{R_R} \geq \alpha$ during $[t_a; t_d]$. From $B_D = \alpha R_R(d - w)$ we conclude that the maximum packet delay does not exceed $d - w$. ■

As $\delta L_D = L_D(t_d) - L_D(t_a)$, $\delta L_R = L_R(t_d) - L_R(t_a)$, we can rewrite inequality (15) as follows:

$$\frac{L_D(t_d) - L_D(t_a)}{L_R(t_d) - L_R(t_a)} \geq \alpha \tag{16}$$

Let us denote the left part of inequality (16) as γ. Then, using equations (13) and (14) and performing a simple transformation, we establish that:

$$\gamma = \alpha + \left(\delta(t_d) + \frac{L_R(t_a)}{\delta L_R}(\delta(t_d) - \delta(t_a))\right) \tag{17}$$

Therefore, inequality (15) takes place if and only if:

$$\delta(t_d) + \frac{L_R(t_a)}{\delta L_R}(\delta(t_d) - \delta(t_a)) \geq 0 \tag{18}$$

Let us now prove the following:

Theorem 2: $\frac{\delta L_D}{\delta L_R} \geq \alpha$ is supported for any traffic pattern and any packet p if and only if:

$$\delta(t_a) \leq 0, \quad \delta(t_d) \geq 0 \tag{19}$$

Proof: First, let us prove that it is a sufficient condition. Indeed, if $\delta(t_a) \leq 0$, $\delta(t_d) \geq 0$, then inequality (18) is true for any values of $L_R(t_a)$ and δL_R, i.e., for any traffic pattern and any packet p.

Second, let us prove that it is a required condition. Suppose that it is not true. We need to consider all such possible cases:

Case 1: $\delta(t_a) \leq 0$, $\delta(t_d) < 0$. Then from inequality (18) we have that:

$$\frac{\delta L_R}{L_R(t_a)} + 1 \leq \frac{\delta(t_a)}{\delta(t_d)} \tag{20}$$

As the left part of inequality (20) is larger than 1, and its right part can be smaller than 1, we have a contradiction.

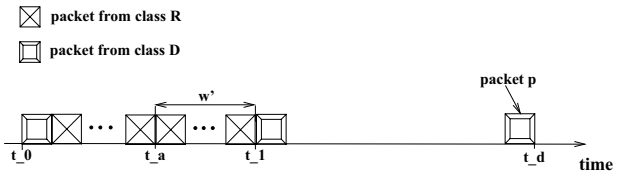


Fig. 7. Schedule of packet departures when $\delta(t_a) > 0$, $\delta(t_d) \geq 0$.

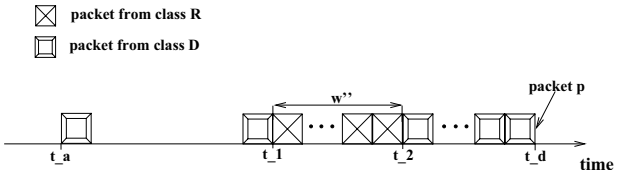


Fig. 8. Schedule of packet departures when $\delta(t_a) \leq 0$, $\delta(t_d) < 0$.

Case 2: $\delta(t_a) > 0$, $\delta(t_d) \geq 0$. Then from inequality (18) we derive that:

$$\frac{\delta L_R}{L_R(t_a)} + 1 \geq \frac{\delta(t_a)}{\delta(t_d)} \tag{21}$$

As there exists a traffic pattern and packet p such that the left part of inequality (21) is smaller than 2, whereas the right part of inequality (21) is bigger than 2, we have a contradiction.

Case 3: $\delta(t_a) > 0$, $\delta(t_d) < 0$. Inequality (18) leads us to:

$$\frac{\delta L_R}{L_R(t_a)} + 1 \leq \frac{\delta(t_a)}{\delta(t_d)} \tag{22}$$

As the left part of inequality (22) is bigger than 0, and its right part is smaller than 0, we have a contradiction.

Thus, we have shown that our assumption is not true, which means than (19) is a required condition. ■

From Theorem 1 and Theorem 2 we conclude that (19) expresses a sufficient condition for supporting queueing delay of at most $d - w$ for any packet with an arbitrary traffic pattern at the RD link. Let us now consider packet p that fills up buffer of the D queue, i.e., the enqueing of that packet satisfies $q_D = B_D$.

Theorem 3: Maximum queuing delay d-w is supported for p within any traffic pattern if and only if $\frac{\delta L_D}{\delta L_R} \geq \alpha$.

Proof: The sufficiency is following from Theorem 1. Let us now prove the necessity. Indeed, if $\frac{\delta L_D}{\delta L_R} < \alpha$, then $\frac{R_D}{R_R} < \alpha$. From the fact that $B_D = \alpha R_R(d - w)$ we conclude that the maximum packet delay is bigger than $d - w$. ■

Let us now consider all possible cases when the packet delay may exceed $d - w$ for such packet p.

Case 1: $\delta(t_a) > 0$, $\delta(t_d) \geq 0$. In Figure 7 we show the schedule of packet departures in the considered case. According to Theorems 2 and 3, if packet p arrived at time t_1 and departed at t_d, then its queuing delay would not exceed $d - w$ as $\delta(t_1) < 0$, $\delta(t_d) \geq 0$. As in the interval $[t_a; t_1]$ there is no potential arrival time t'_a of packet p at which $\delta(t'_a) < 0$, queuing delay of packet p might be exceeded by the interval $[t_a; t_1]$. We will refer to the length of that interval as w', and to the amount of R traffic sent during this interval w' as to X. Suppose that a D packet departing at time t_0 has size S_D,

then the following inequalities take place:

$$L_D(t_0) \leq \alpha L_R(t_0) \tag{23}$$

$$L_D(t_0) + S_D > \alpha L_R(t_0) \tag{24}$$

$$L_D(t_0) + S_D \leq \alpha L_R(t_0) + X\alpha \tag{25}$$

Lemma 1: The worst-case solution to inequalities 23,24,25 is at most:

$$X = \frac{S_D^{max}}{\alpha} + S_R^{max}. \tag{26}$$

Proof: Indeed, X defined by equation (26) is a solution to the considered three inequalities. Next, we need to show that there is no smaller solution. Let us suppose that there exists X':

$$X' = X - \delta X \tag{27}$$

where X is defined by equation (26), $\delta X > 0$, $\delta X < X$, δX is an integer, i.e., there exists δX such that X' satisfies inequalities (23),(24),(25). Let us assume that the traffic scenario is such that inequality (23) becomes equality. Then from inequality (25) and $L_D(t_0) = \alpha L_R(t_0)$ we derive that:

$$S_D \leq S_D^{max} + \alpha S_R^{max} - \alpha \delta X \tag{28}$$

Assuming that $S_D = S_D^{max}$ and $S_R^{max} < \delta X$, we have that inequality (28) is not valid. As it means that there exists a traffic scenario such that inequality (25) is not valid, we have a contradiction. Finally, we mention that S_R^{max} in (26) reflects that traffic is in packets, i.e., not fluid. ■

From Lemma 1 we conclude that the maximum queuing delay in excess of $d - w$ in the considered case is as follows:

$$w' = \frac{1}{C}\Big(\frac{S_D^{max}}{\alpha} + S_R^{max}\Big) \tag{29}$$

Case 2: $\delta(t_a) \leq 0$, $\delta(t_d) < 0$. In Figure 8 we demonstrate how the packets are scheduled for this case. If during time interval $[t_1; t_2]$, instead of packets from class R the link continued to serve the D queue up to packet p, then, according to Theorems 2 and 3, queuing delay of packet p would not exceed $d - w$ as $\delta(t_a) < 0$, $\delta(t'_d) \geq 0$, where t'_d would be its departure time. Therefore, queuing delay of packet p can exceed the delay constraint by the interval $[t_1; t_2]$. We refer to the length of that interval as w'', and to the amount of D traffic sent during this time interval as Y. As in Case 1, Y is defined by the right part of equation (26). Therefore, w'' equals to w' defined by equation (29).

Case 3: $\delta(t_a) > 0$, $\delta(t_d) < 0$. As this case is a combination of the two previous ones, the maximum queuing delay in excess of $d - w$ is the sum of w' and w'':

$$w = \frac{2}{C}\Big(\frac{S_D^{max}}{\alpha} + S_R^{max}\Big) \tag{30}$$

As we have not used the information that p fills up the buffer of queue D while considering the three possible cases of the exceeding the $d - w$ delay, we have proved the following:

Theorem 4: Sizing the D buffer according to equations (8) and (9) ensures that the RD router algorithm supports maximum queuing delay d.

VII. Performance Evaluation of the RD services

In this section, we evaluate performance of the RD services through simulations using version 2.29 of ns-2 [30]. Unless explicitly stated otherwise, all flows employ TCP NewReno [18] and data packets of size 1 KB. Each link buffer is configured to $B = B_{max} = C \cdot 250$ ms where C is the capacity of the link. Every experiment lasts 60 s and is repeated five times for each of the considered parameter settings. The default settings include k = 2, d = 10 ms, b = 18,000, T = 400 ms, E = 1 s, T_{avg} = 200 ms, and T_q = 10 ms, where T_{avg} refers to the averaging interval for the bottleneck link utilization and loss rate, and T_q denotes the averaging interval for queuing delay. We also average the utilization and loss rate over the whole experiment with exclusion of its first five seconds.

Section VII-A evaluates the RD services in a wide variety of scenarios that include long-lived and short-lived traffic, diverse bottleneck link capacities, various settings for the delay constraint of the D service, Exponential flow interarrival times, and sudden changes in the numbers of R and D flows. Section VII-B continues the assessment in multi-ISP topologies and, in particular, examines whether the RD services are deployable despite the continued presence of legacy ISPs and without penalizing legacy traffic.

A. Basic properties

To understand basic properties of the RD services, this section experiments in a traditional dumbbell topology where the core bottleneck and access links have capacities 100 Mbps and 200 Mbps respectively. The bottleneck link carries 100 R flows and 100 D flows in both directions and has propagation delay 50 ms. We choose propagation delays for the access links so that propagation RTT (Round-Trip Time) for the flows is uniformly distributed between 104 ms and 300 ms.

1) Illustrative behavior: In this section, we illustrate how the RD design performs when the D flows employ TCP NewReno [18]. All flows stay throughout the experiment. With $k = 2$ and equal numbers of R and D flows, we expect the R and D services to utilize the bottleneck link capacity fully with the 2:1 ratio. Figure 9 mostly confirms this expectation and also plots queuing delay for D service. For the R service, maximum queuing delay is about 375 ms, as expected for the link that allocates two thirds of its capacity C to the R flows and has the buffer sized to the product of C and 250 ms. Queuing delay for the D service fluctuates between 0 and d = 10 ms.

2) Sudden changes in the numbers of flows: To investigate how the RD services react to sudden changes in the numbers of R and D flows, we experiment with the following traffic. 100 R flows start at time 0. 50 D flows join them 20 s later. 50 additional D flows arrive at time 40 s and thereby equalize the flow counts for the two services at 100. At time 60 s, 80 D flows finish. 80 other D flows arrive at time 80 s. All R flows leave at time 100 s but 20 new R flows start 40 s later. Finally, 80 extra R flows arrive at time 160 s and reestablish the parity in the numbers of R and D flows. Figure 10 shows that the RD design responds to the changes promptly and appropriately:

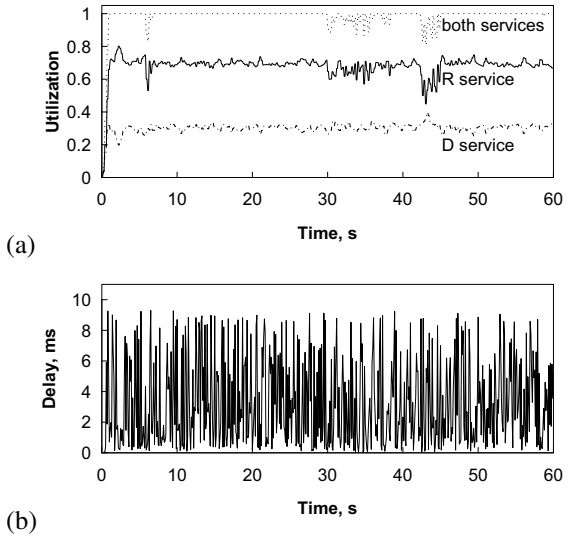


Fig. 9. Using TCP NewReno for D flows: (a) bottleneck link utilization; (b) queuing delay for D packets.

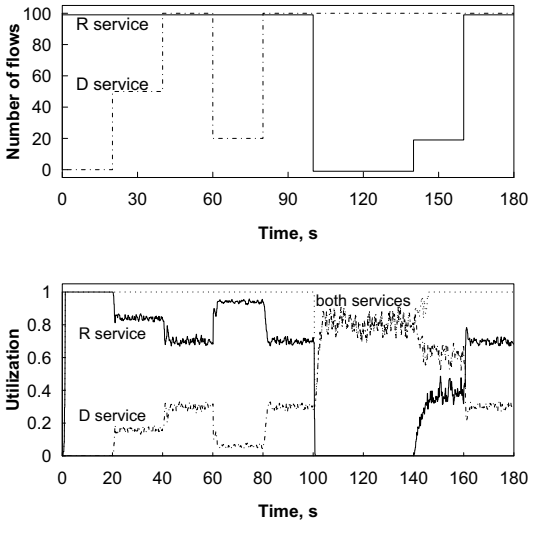


Fig. 10. The performance of the RD services when the number of flows of the classes changes suddenly: (a) dynamics of the number of flows; (b) utilization of the bottleneck link.

reflecting the current ratio of the flow counts, the per-flow rate ratio for R and D flows becomes 4:1 at time 20 s, reduces to 2:1 at time 40 s, grows to 10:1 at time 60 s, and returns to 2:1 at times 80 s and 160 s, at the latter time by reverting from 1:2. The RD services utilize the bottleneck link fully except between 100 s and 140 s. During that interval, the link carries only D flows and is underutilized due to the small size of the D buffer. The maximum queuing delay for D class does not exceed 10 ms as expected.

3) Influence of short-lived flows: To see how short-lived flows affect the RD services, we enhance the traffic mix on the bottleneck link in this and subsequent three experimental series with web-like flows from two sources: one source generates R flows, and the other transmits D flows. The sizes of the web-like flows are Pareto-distributed with the average of 30 packets and shape index of 1.3. The flows arrive according to a Poisson process. In the experiments of this section, the average arrival rate varies from 1 flow per sec (fps) to 400 fps. When the flows arrive more frequently, the traffic mix becomes burstier and imposes higher load on the bottleneck link. As expected, these factors drive up the loss rate for the D service. Figure 11 reveals that despite the increasing losses, the RD services closely maintain the intended 2:1 per-flow rate ratio for the R and D flows. The maximum queuing delay for class D does not exceed the delay constraint for all the values of the varied parameter.

4) Link capacity scalability: In this series of experiments, we vary the bottleneck link capacity from 1 Mbps to 1 Gbps while keeping the access link capacities twice as large. The average arrival rate for the web-like flows in this and next sections stays at 50 fps. Figure 12 shows that the rates of the R and D flows deviate from the intended 2:1 ratio significantly only for the lowest examined capacities close to 1 Mbps. The deviation occurs due to the extremely small buffering available for D packets in those settings. In particular, satisfying the 10-ms delay constraint at the 1-Mbps bottleneck link reduces the D buffer to about one packet, and the minimal buffering causes heavy losses and effectively shuts down the D service. As the bottleneck link capacity grows, the loss rate for the D flows decreases exponentially. Moreover, the delay constraint is supported for all the values of the bottleneck link capacity.

5) Sensitivity to the delay constraint: To examine sensitivity of the RD services to d, we vary the delay constraint of the D service from 3 ms to 15 ms. Figure 13 demonstrates that the per-flow rate ratio for the R and D flows stays close to the intended 2:1. As d increases, the loss rate for the D service decreases from about 8.4% to about 5.8% due to the increasing size of the D buffer.

B. Performance in multi-ISP topologies

Our investigation of the RD services proceeds by examining their incremental deployability and other properties in topologies where multiple ISPs own the infrastructure. Figure 14 depicts the settings shared by the multi-ISP topologies. The network core belongs to ISP Z and ISP Y. Routers y1 and y2 of ISP Y offer the RD services with $k = 2$ and $d = 15$ ms. Backbone link z2-y1 connects the two ISPs and provides universal connectivity for all users. The users form five pools H, J, K, F, and G. Each user accesses his or her ISP through a personal link with capacity 100 Mbps. Every user from pools H, J, K, and F transmits a long-lived flow to a separate user in pool G. Hence, while the flows from K and F traverse the infrastructure that belongs only to ISP Y, both ISPs serve the flows from pools H and J. We choose propagation delays for the access links so that propagation RTT for the flows is uniformly distributed between 64 ms and M. In particular, propagation delay for both access links of each flow from pool H or J is chosen between 1 ms and $\frac{M}{4} - 15$ ms, and both access-link propagation delays for a flow from pool K

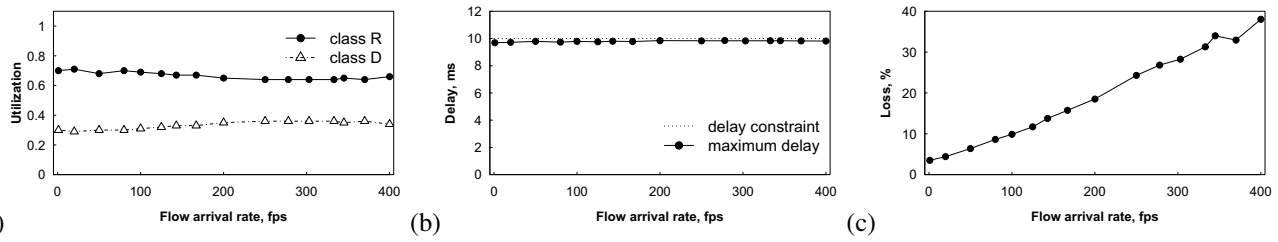


Fig. 11. Influence of short-lived traffic. Performance characteristics for different arrival rates of web-like flows: (a) average utilizations of classes R and D; (b) maximum queuing delay of class D; (c) average loss rate of class D.

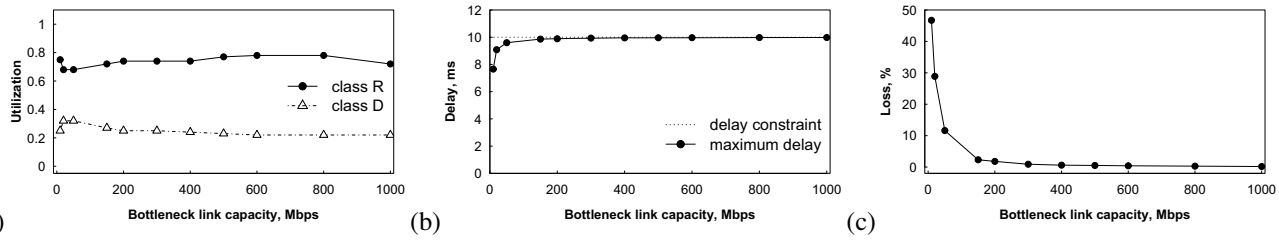


Fig. 12. Scalability of the RD services concerning bottleneck link capacity: (a) average utilizations of classes R and D; (b) maximum queuing delay of class D; (c) average loss rate of class D.

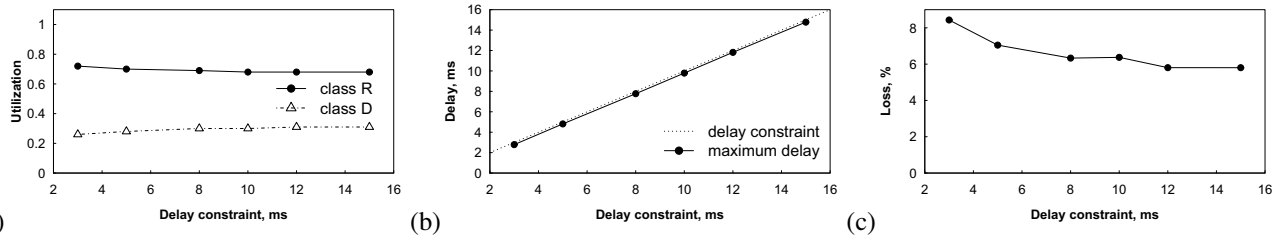


Fig. 13. Sensitivity of the RD services to delay constraint: (a) average utilizations of classes R and D; (b) maximum queuing delay of class D; (c) average loss rate of class D.

or F are selected between 11 ms and $\frac{M}{4} - 5$ ms. The default setting for the maximum propagation RTT is $M = 300$ ms. The flows arrive according to a Poisson process. The average arrival rate is set by default to 100 fps for creating a confident expectation that all the flows arrive before the measurement stage of the experiment.

1) Incremental deployability: Our design principles in Section II prescribe that a new service should attract adopters despite continued presence of legacy ISPs and without penalizing legacy traffic. This section experimentally verifies whether the RD services fulfill these design aspirations. Unlike ISP Y, ISP Z does not support the RD services and treats all traffic with the legacy service. 500 flows traverse the network: 125 flows come from pool H, other 125 flows originate at pool J, and the remaining 250 flows enter from pools K or F. Link z1-z2 has capacity 55 Mbps making link y1-y2 a bottleneck for all the flows. We vary ρ, the percentage of D flows. The other $1-\rho$ flows are either legacy or R flows. More specifically, $\lceil 125\rho \rceil$ D flows come from pool H, $\lceil 125\rho \rceil$ D flows originate at pool J, all $2 \cdot \lceil 125\rho \rceil$ flows from pool F indicate their preference for the D service, and the rest of the traffic consists of legacy and R flows.

Figure 15a plots the per-flow rates achieved by the legacy and R flows and D flows at link y1-y2 of ISP Y. As those legacy flows that are interested in low delay opt for the D service and thereby increase the percentage of D flows, the per-flow rate for the remaining legacy flows consistently improves even though some of them enter the network through the legacy ISP Z. Hence, the legacy traffic not only avoids being penalized by the adopters of the D service in accordance with Principle 2 but also benefits itself by becoming able to communicate at higher rates. Besides, Figure 15 reveals that adoption of the RD services yields a win-win outcome for all users: as ρ grows, the per-flow rate increases for the D flows as well, and the increasing size of the D buffer reduces the loss rate of the D service. Therefore, whereas a user opts for the D service to acquire low delay, future adoptions of the D service by other legacy users make the service even more valuable, facilitating the virulent deployment of the RD services. Besides, for 5% of the adopting flows there is a significant increase of loss rate up to 25%. The increase happens because a buffer size is less than three packets. It also explains the drop of throughput of class D for 5% of adopting flows. In addition, we observe that the delay guarantees for D class are supported for all the values of the varied parameter.

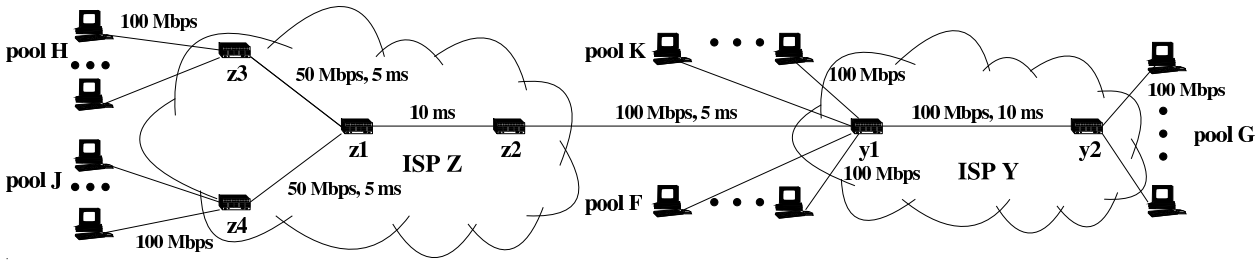


Fig. 14. Complex topology used in the simulations

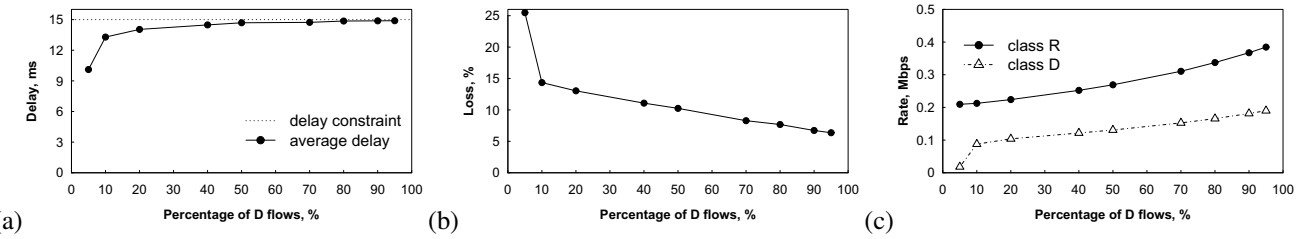


Fig. 15. Performance with the incremental deployment of the RD design on ISP Y: (a) maximum queuing delay of class D; (b) average loss rate of class D; (c) average per-flow rates of classes R and D.

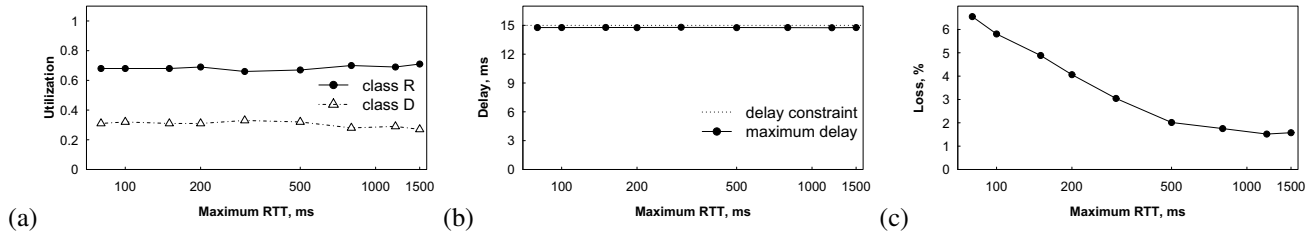


Fig. 16. Influence of the propagation RTT diversity on link $y1y2$: (a) average utilizations of classes R and D; (b) maximum queuing delay for class D; (c) average loss rate of class D.

2) Influence of propagation RTTs: From now on, we consider topologies where both ISPs espouse the RD services. ISP Z configures all its four routers to offer the rate-delay differentiation with $k = 2$ and $d = 10$ ms. The long-lived traffic includes 25 R flows and 25 D flows from pool H, 25 R flows and 25 D flows from pool J, 50 R flows from pool K, and 50 D flows from pool F. For each of the flows, the reverse direction of its path carries another long-lived flow of the same class. Also, two sources in pool H transmit web-like R and D flows to pool G. The web-like traffic has the same characteristics as in Sections VII-A4 and VII-A5. The capacity of link z1-z2 is set to 100 Mbps. Thus, the network contains two bottleneck links: z1-z2 and y1-y2.

To study the impact of propagation RTT on the RD services, we vary M from 80 ms to 1.5 s. As the maximum propagation RTT grows, the per-flow amount of packets inside the network increases. Consequently, the TCP flows enjoy lower loss rates. Figure 16 confirms this expectation, shows that the RD services consistently support the intended 2:1 per-flow rate ratio for the R and D flows, and demonstrates that the queueing delay for D class does not exceed the delay constraint at link $y1y2$. In addition, link $z1z2$ reveals similar behavior concerning throughput differentiation and holding the delay constraint.

3) Population scalability of the RD services: We also explore population scalability of the RD services, i.e., examine how their performance scales when the numbers of R and D flows change. First, we use a scaling factor σ to modify the traffic mix as follows: the population of the long-lived flows includes 25 R flows and $\lceil 25\sigma \rceil$ D flows from pool H, 25 R flows and $\lceil 25\sigma \rceil$ D flows from pool J, 50 R flows from pool K, and $2 \cdot \lceil 25\sigma \rceil$ D flows from pool F. To preserve the expectation that all the long-lived flows arrive before the measurement stage of the experiment, we reduce average interarrival time to 3 ms for $\sigma > 3$. The long-lived traffic in the reverse direction mirrors again the forward-direction arrangement.

For either of bottleneck links z1-z2 and y1-y2, Figure 17 shows that increasing the number of long-lived D flows redistributes some of the link capacity from the R service to the D service. Due to the presence of the web-like flows, the redistribution depends on σ non-linearly. Also, since links z1-z2 and y1-y2 serve different numbers of flows, the D service gains parity with the R service in utilizing link z1-z2 with a larger scaling factor than for link y1-y2. As σ grows, the per-flow rates of the R and D flows decrease, and the loss rates

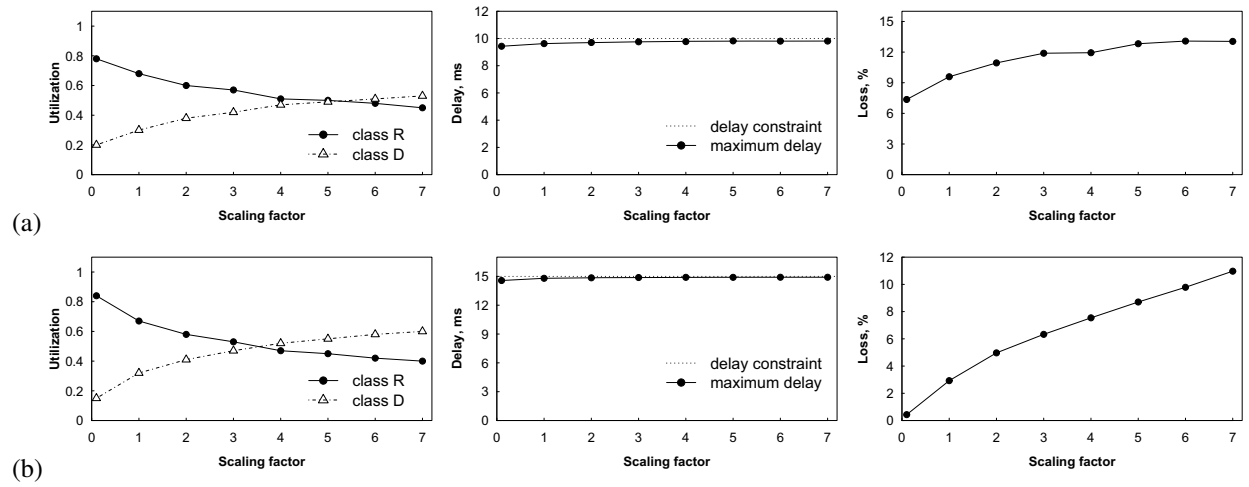


Fig. 17. Scalability concerning the number of long-lived D flows. Average utilizations of classes R and D, maximum queuing delay of class D, and average loss rate of class D: (a) link $z1z2$; (b) link $y1y2$.

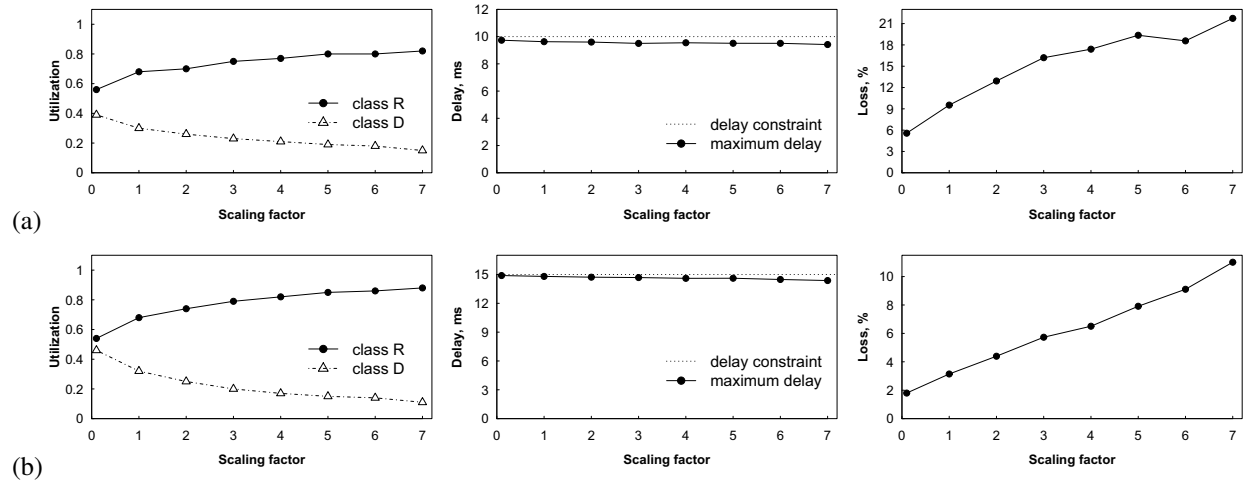


Fig. 18. Scalability concerning the number of long-lived R flows. Average utilizations of classes R and D, maximum queuing delay of class D, and average loss rate of class D: (a) link $z1z2$; (b) link $y1y2$.

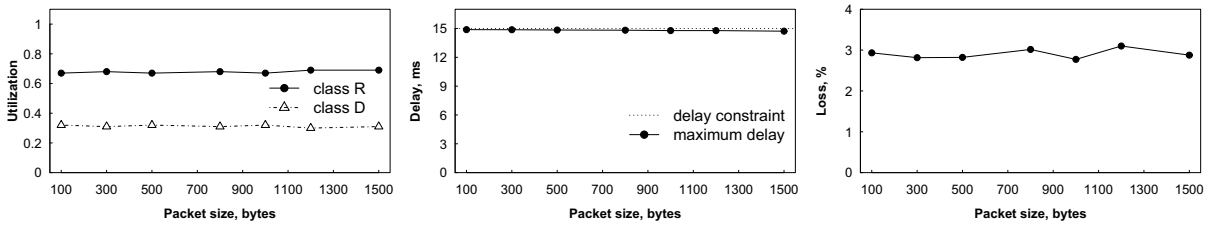
of the services increase accordingly.

Finally, we conduct a similar study for scalability of the RD services with respect to the number of R flows. Once again, the long-lived traffic arrangement is symmetrical in the forward and reverse directions. In the forward direction, the long-lived traffic includes $\lceil 25\sigma \rceil$ R flows and 25 D flows from pool H, $\lceil 25\sigma \rceil$ R flows and 25 D flows from pool J, $2 \cdot \lceil 25\sigma \rceil$ R flows from pool K, and 50 D flows from pool F. Figure 18 plots utilization and loss rates for links z1-z2 and y1-y2. The analytical rationale for the observed performance profiles is the same as the above explanations for the scaling of the D population.

4) Impact of packet sizes: To estimate the influence of the packet size on the design performance, we vary the sizes of packets in both the classes. We run two sets of the experiments. In each set, we fix the packet size for one class to 1000 bytes and vary the packet size for the other class in the range between 100 bytes and 1500 bytes. The long-lived traffic includes 25 R flows and 25 D flows from pool H, 25 R flows and 25 D flows from pool J, 50 R flows from pool K, and 50 D flows from pool F. For each of the flows, the reverse direction of its path carries another long-lived flow of the same class. Figures 19 and 20 depict that the delay constraint is kept over the whole range of packet size from class R and class D, respectively. Whereas the size of packets from class R does not affect significantly the loss rate of class D, the loss rate of class D increases monotonically with the size of packets from class D, which is explained by the bigger influence of part $S_D^{max}\frac{kn_R}{n_D}$ than S_R^{max} in equation (9) for adjusting the D buffer size. As expected, the ratio 2:1 between throughputs of classes R and D is supported for the whole range of the packet size if varying the size of a packet from each of the classes.

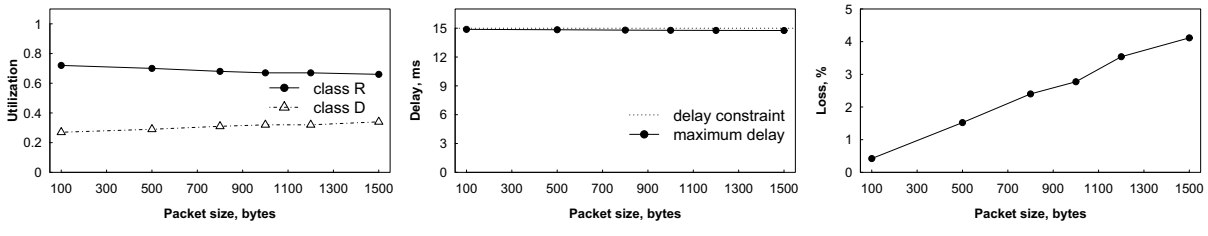
C. Impact of the packet discard policy

To ensure the strict queuing delay constraint for D class, the design employs a packet discard mechanism desrcibed in Section V-B3. Without flushing D packets, if the service rate



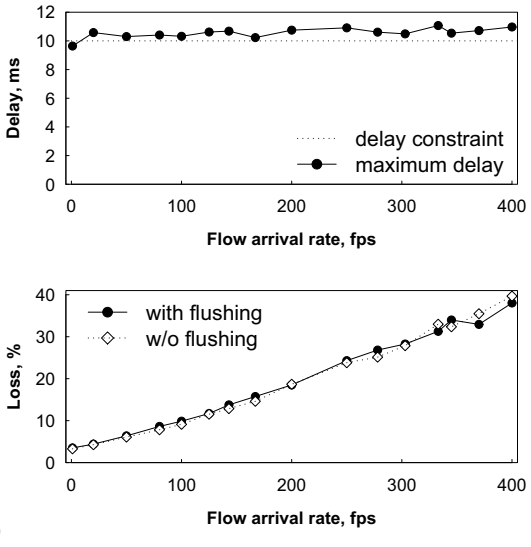
(a)

Fig. 19. Sensitivity of the RD algorithm to the size of packets from class R on link $y1y2$: (a) average utilizations of classes R and D; (b) maximum queuing delay for class D; (c) average loss rate of class D.



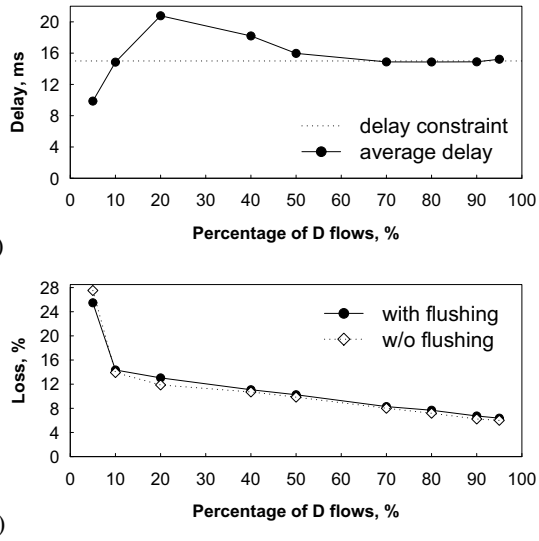
(a)

Fig. 20. Sensitivity of the RD algorithm to the size of packets from class D on link $y1y2$: (a) average utilizations of classes R and D; (b) maximum queuing delay for class D; (c) average loss rate of class D.



(a) (b)

Fig. 21. Influence of avoiding the "flushing" of the D queue with different intensities of web-like traffic: (a) maximum queuing delay for class D; (b) loss rate of class D.



(a) (b)

Fig. 22. Influence of avoiding the "flushing" of the D queue with the incremental deployment of the RD design: (a) maximum queuing delay for class D; (b) loss rate of class D.

of D queue becomes smaller after periodical recalculation on timeout, the maximum queueing delay of D packets can exceed the delay constraint. On the other hand, no forced flushing can potentially lead to a smaller loss rate. The purpose of this section is to explore how the discard of all packets from the D queue at the moment of the recalculation of control parameters affects the loss rate of class D. As an accurate theoretical analysis of the influence of the flushing is difficult, we apply an experimental approach and conduct two sets of simulations.

In Figure 21 we report the results for the same experimental setting as in Section VII-A3, influence of web-like traffic, with and without the flushing from the D queue. As expected the maximum queuing delay of class D exceeds the delay constraint for all the values of arrival rate of web-like flows except for 1 fps. The loss rate of class D is approximately the same as with the flushing.

Figure 22 shows the results for the same experimental setting as in Section VII-B1, which illustrates incremental deployment, with and without the flushing from the D queue. We observe that the maximum queuing delay of class D is bigger than the delay constraint for the half of the values of the percentage of D flows. The loss rate of class D is approximately the same as with the flushing except for the

R-factor range	MOS	Category of voice transmission quality	User satisfaction
90 - 100	4.34 - 4.50	Best	Very satisfied
80 - 90	4.03 - 4.34	High	Satisfied
70 - 80	3.60 - 4.03	Medium	Some users dissatisfied
60 - 70	3.10 - 3.60	Low	Many users dissatisfied
50 - 60	2.58 - 3.10	Poor	Nearly all users dissatisfied

Fig. 23. Categories of voice transmission quality

traffic with 5% of D flows, for which the loss rate increases by 3%.

We explain the loss rate behavior as following. Although with the flushing there are induced losses of class D, the dropping of packets creates the room in the buffer for new packets from class D, and this compensates for losses caused by the flushing. Thus, based on the results of the experiments, we assert that flushing the D queue for ensuring the delay constraint does not lead to the growth of the loss rate.

VIII. Internet Telephony

A. Application and its Needs

To evaluate the quality of the delivered service for VoIP, we use Mean Opinion Score (MOS) [3], a subjective score for voice quality ranging from 1, unacceptable, to 5, excellent. To estimate a MOS score through network characteristics, we employ the E-Model [5], which assesses VoIP quality accounting network characteristics like loss and delay. The E-Model uses the R-factor that is computed as a function of all of the impairments occurring with the voice signal, and is ranged from 0 to 100. The relationship between R-factor and MOS score can be described through the following equation [5]:

$$MOS = 1 + 0.035R + 7 * 10^{-6}R(R-60)(100-R) \quad (31)$$

According to E-Model, R-factor is calculated as follows:

$$R = R_0 - I_s - I_{e,eff} - I_d + A \quad (32)$$

where R_0 captures the basic signal-to-noise ratio, I_s accounts the impairments occurring with the voice signal and does not depend on the transmission over the network, $I_{e,eff}$ describes impairments related to data loss and low rate codecs, I_d specifies the impairments induced by delay and echo, and A, "advantage factor", compensates the above impairments taking into account that a user may tolerate some decrease of voice quality in exchange for access advantage. For example, whereas for a wired phone A equals to zero, A becomes equal to ten for cellular in a moving vehicle. Table 23 maps the values of R-factor into MOS, the category of voice transmission quality, and user satisfaction. We should notice that connections with R-factor below 50 are not recommended [5].

B. Evaluation Methodology

To generate VoIP traffic and perform measurements of voice quality, we use the tool developed in [6], an additional module of the network simulator ns2 [30]. We use the same network topology as in Section VII-A3 with the same traffic from R class and web-like traffic from both the classes, but the bottleneck link delay is 10 ms. Instead of long-lived D flows, there are 100 VoIP flows with the same propagation RTTs of 150 ms. The value of d is 50 ms. In addition, there is one web server and one web traffic receiver connected to the bottleneck link for classes R and D. Web flows arrive with the intensity of 50 fps. We perform five experiments for each settings, and each experiment lasts for 70 sec. To encode the speech, we employ AMR (Adaptive Multi-Rate) Audio Codec [4] operating at audio bitrate of 12.2 kbps. In some experiments we also use G.711 [1] and G.729A [2] codecs with audio bitrates 64 kbps and 8 kbps, respectively.

The parameters we measure are average Mean Opinion Score (MOS) and the average utilization of class R. While measuring MOS, first ten seconds of the experiment are neglected. All flows join the network during the first second. We compare the performance of the RD Network Services with the performance of the DropTail link.

C. Experimental Results

1) Transient behavior: In this experiment, VoIP flows join the network during the whole experiment lasting for 600 sec. There are 500 VoIP flows that start coming to the network from the beginning. The arrival process is described by Exponential distribution with the average 1 fps. Whereas the average MOS with the DropTail link is 2.97, MOS with the RD Network Services is 4.16. The utilization of class D is 84.45% and 83.85% with the RD Network Services and DropTail link, respectively. Thus, the RD Network Services deliver better service for VoIP in the considered dynamic scenario.

2) Influence of propagation RTT: To explore how propagation RTT affects the quality of VoIP, we modify the propagation RTT of VoIP flows in the interval between 30 ms and 800 ms. We run the experiments with three different codecs: AMR, G.711, and G.729A. In Figure 24, we notice that the RD Network Services reveal better performance for VoIP over the whole range of the varied parameter with all the codecs. In particular, at least medium quality of voice is supported for the propagation RTTs up to 300 ms, 400 ms,

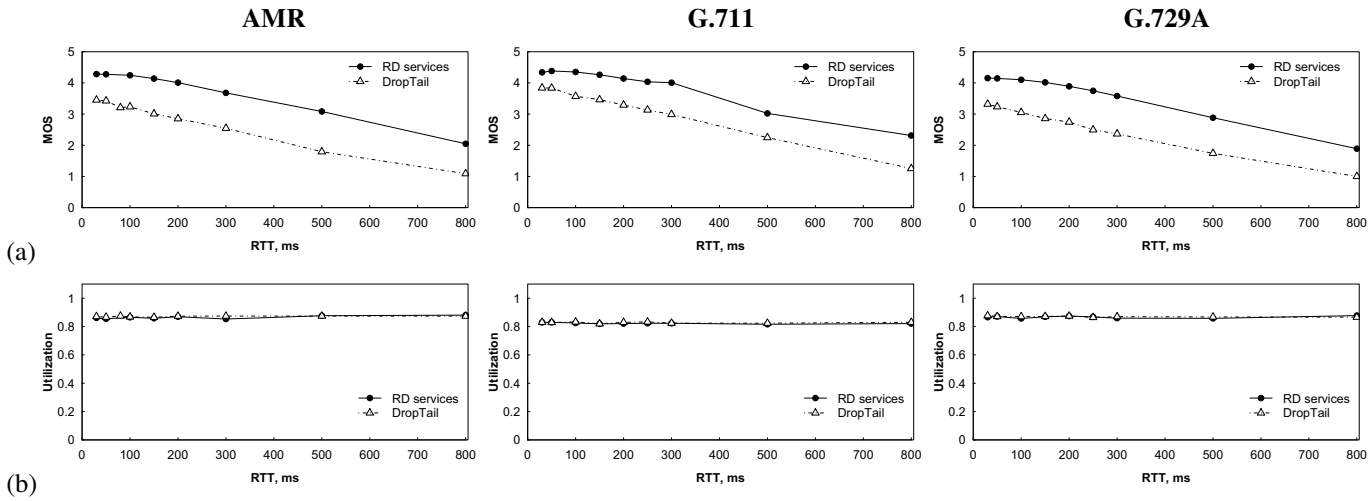


Fig. 24. Influence of the propagation RTT for AMR, G.711, and G.729A codecs: (a) Average MOS; (b) average utilization of class R.

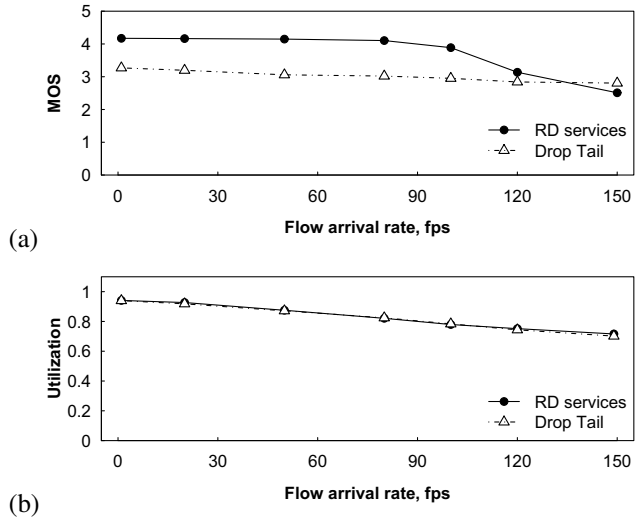


Fig. 25. Influence of the intensity of the web-like flows: (a) Average MOS; (b) average utilization of class R; (c) average throughput of the "misbehaving" throughput-greedy flow.

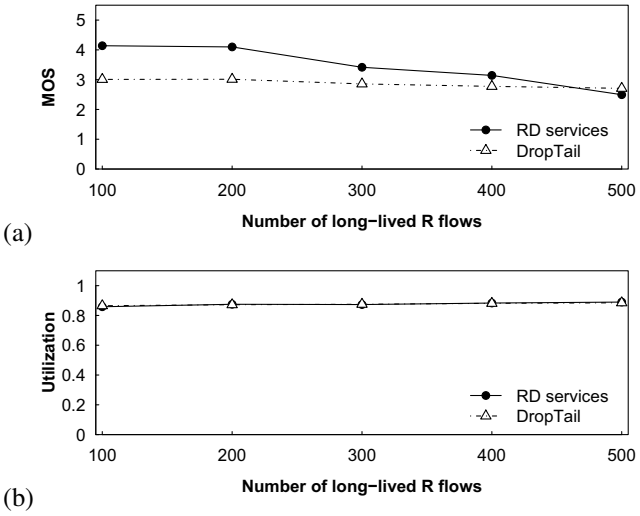


Fig. 26. Influence of the number of R flows: (a) Average MOS; (b) average utilization of class R.

and 300 ms for AMR, G.711, and G.729A, respectively, with the RD link. On the other hand, the DropTail link can support the same quality of voice for the propagation RTTs no more than 50 ms and 100 ms for AMR and G.711, respectively, and is unable to support that for G.729A. Besides, R traffic gets the same bottleneck link utilization and loss rate with both the schemes.

3) Influence of the web-like traffic: To study the influence of the web-like flows, we change the intensity of the web-like flows in the interval between 1 fps and 150 fps. In Figure 25, we observe that the RD Network Services demonstrate better performance for VoIP over almost the whole range of the varied parameter, whereas the R traffic gets the same bottleneck link utilization comparing to the DropTail link.

4) Influence of the long-lived R Flows: In this experiment, we change the number of the long-lived R flows in the interval between 100 and 500. Figure 26 shows that VoIP flows receive better service comparing to the DropTail link through almost the whole range of the varied parameter, whereas R flows reveal the same performance concerning link utilization. The deterioration of voice quality with the increase of the number of R flows, when MOS reduces from 4.14 till 2.5, is because the decrease of the D buffer size increases the loss rate.

5) Impact of VoIP population size: To examine the scalability of the design concerning the population of VoIP flows, we vary the number of them in the interval between 100 and 500. Figure 27 reports that the number of VoIP flows does not affect the quality of VoIP. In particular, MOS with the RD Network Services is in the range between 4.14 and 4.16 whereas MOS with the DropTail link is between 3.01 and 3.13. The constant performance of VoIP over the whole range of the varied parameter is because a VoIP flow requires a relatively small connection throughput. On the other hand, R flows reveal

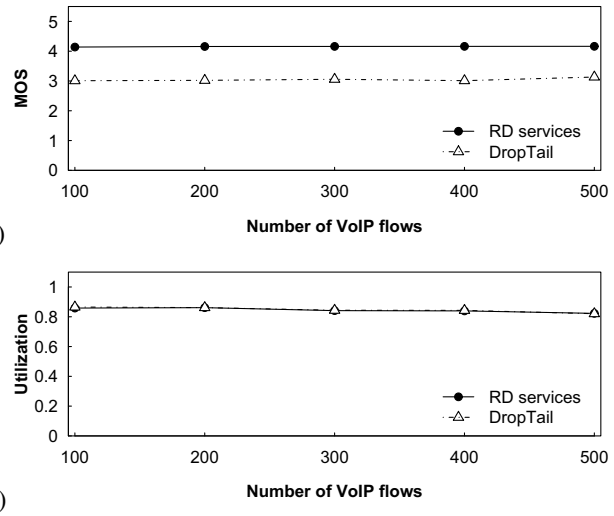


Fig. 27. Influence of the number of VoIP flows: (a) Average MOS; (b) Average utilization of class R.

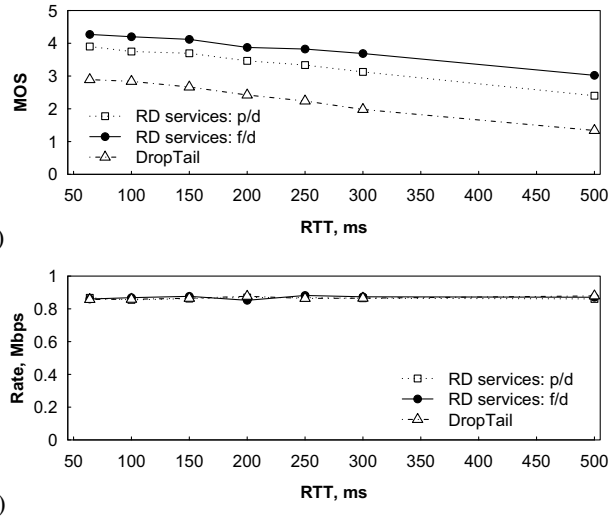


Fig. 28. Performance under the partial deployment for different propagation RTTs: (a) Average MOS; (b) average per-flow throughput of class R.

the same bottleneck link utilization, which is between 82% and 87%.

6) Partial deployment: In this experiment, we explore the situation when VoIP flows use the service provided by two different ISPs. There is one bottleneck within each ISP, 50 VoIP flows, 50 R flows going through both the ISPs, and two groups of 50 R flows each so that each group goes only through one ISP. In particular, one can consider two deployment scenarios. The first one assume that only one ISP has deployed the RD Network Services, whereas in the second one both the ISPs have adopted the considered architecture. Propagation RTT of VoIP flows is varied in the range between 64 ms and 500 ms. In Figure 28, we observe that even under the partial deployment of the RD Network Services, which is labeled as "p/d" in the graph, VoIP flows get better service. Moreover, the full deployment of the design labeled as "f/d" further improves the voice quality. More importantly, the improvements of VoIP quality do not affect the service delivered to the R class concerning the flow rates.

IX. Web

A. Application and its Needs

As the majority of the traffic generated by a web application consists of short-lived flows [42], Flow Completion Time is considered as the main performance characteristic for the web application flows [15]. FCT is defined as the interval between the initialization of a connection and the delivery of its last data packet. To evaluate the performance of a web application, which generates flows with different sizes, we calculate the average goodput of the web-like flows as the average of the goodput of each web-like flow. To calculate the goodput of a web-like flow, we compute the ratio between the flow size and its FCT.

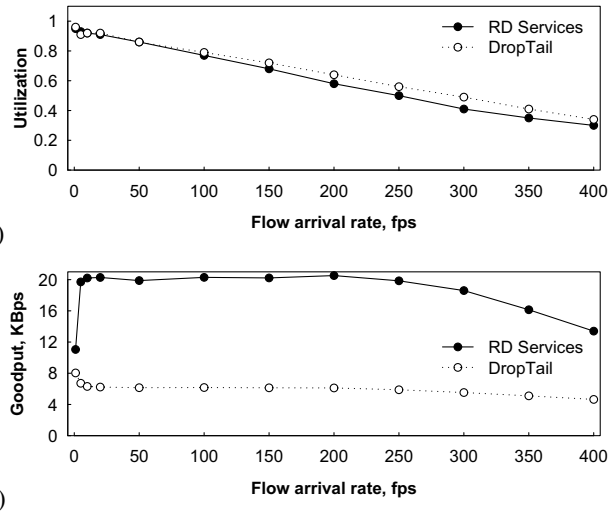


Fig. 29. Influence of the intensity of the web-like flows: (a) average utilization of the long-lived flows; (b) average goodput of the web-like flows.

B. Evaluation Methodology

In the experiments, we employ a dumbbell topology with the same experimental settings as in Section VIII. To compare the RD Network Services design, we also run the experiments under the same settings for the DropTail link. There are 100 long-lived flows in the forward and reverse directions that are served as class R. The R flows join the network during the first second of an experiment. The value of d is 50 ms. In addition, there is one web server and one web traffic receiver connected to the bottleneck link. The web server generates flows with the same parameters as in Section VII-A4, which are served as class D.

C. Experimental Results

1) Influence of the web-like traffic: We study the influence of the intensity of the web-like flows varying their arrival rate

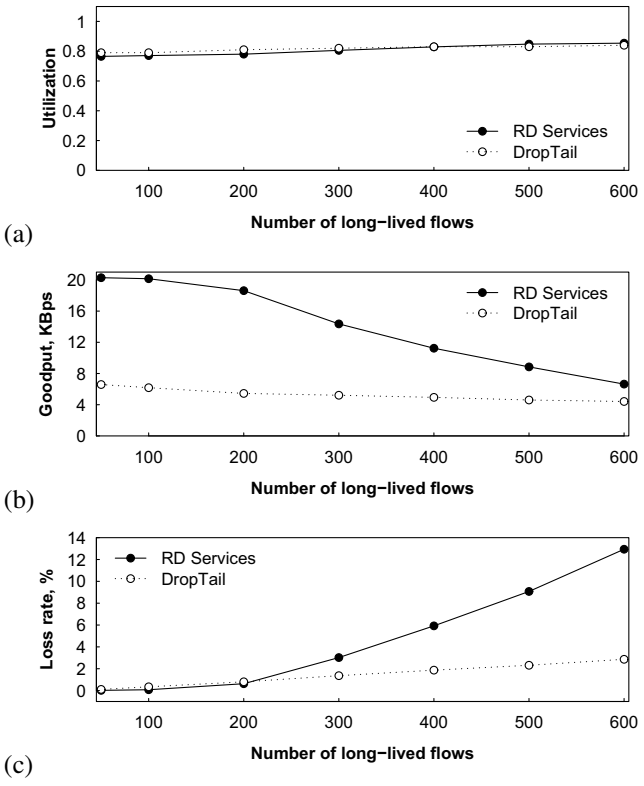


Fig. 30. Influence of the number of the long-lived flows: (a) average utilization of the long-lived flows; (b) average goodput of the web-like flows; (c) average loss rate for the web-like flows

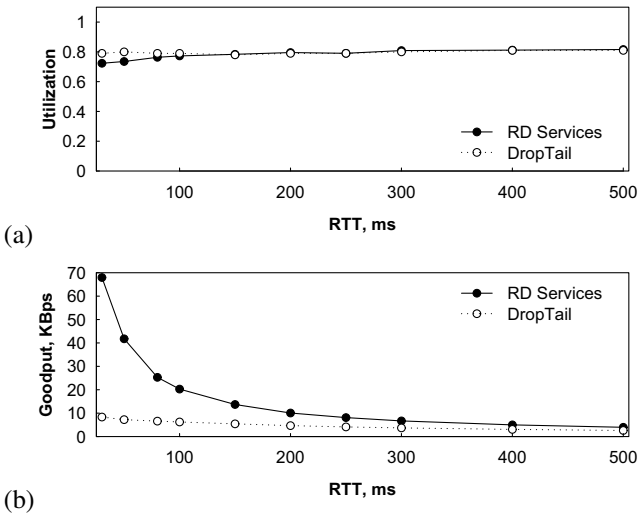


Fig. 31. Influence of propagation RTT of the web-like flows: (a) average utilization of the long-lived flows; (b) average goodput of the web-like flows.

in the interval between 1 s^{-1} and 400 s^{-1}. In Figure 29, we see that the RD link gives the improvement of the performance of the web-like flows over the whole range of the varied parameter. On the other hand, there is a deterioration of the performance of the long-lived flows for the web-like flows with the intensity larger than 100 s^{-1}. However, such scenarios lead to the utilization of the bottleneck link by the web-like flows larger than 20%, whereas, according to the measurements, the amount of the web-like flows in the Internet does not exceed 20% [42]. Therefore, those scenarios are not expected to be a common case. The decrease of the performance of web-like flows for their intensities bigger than 250 s^{-1} is because of the increased loss rate. Concerning the drastic drop of the goodput of the web-like flows for their intensity of 1 s^{-1}, we attribute it to the very small intensity of the web-like flows and plan to investigate that question in details in future.

2) Influence of the long-lived flows: To explore the population scalability, we vary the number of the long-lived flows between 50 and 600. The intensity of the web-like flows is 100 s^{-1}. Figure 30 shows that the web-like flows have a bigger loss rate with the RD Network Services than with the DropTail link if the number of the long-lived flows is bigger than 200. However, the RD Network Services demonstrate better performance of the web-like flows over the whole range of the varied parameter, whereas the long-lived flows have the same goodput with the RD Network Services and DropTail link. In particular, the former improves the goodput of the web-like flows by 50%-206%.

3) Influence of propagation RTT of the web-like traffic: In this experiments, we vary the propagation RTT of the web-like flows in the range between 30 ms and 500 ms. The number of the long-lived flows is 100. In Figure 31, we observe that the RD Network Services significantly improve the goodput of the web-like flows for small RTTs . Besides, the performance of the RD Network Services and DropTail link for the long-lived flows is similar except for RTTs less than 50 ms, for which the RD Network Services reveal slightly smaller bottleneck link utilization than the DropTail scheme. In addition, the performance of the web-like flows with the RD link becomes closer to one with the DropTail link with the increase of propagation RTT. We explain such a behavior that large propagation RTT gets the dominant factor in determining FCT.

4) Multi-bottleneck topology: To explore the performance of our design under multi-bottleneck topology, we employ a parking lot topology shown in Figure 32. All access links are 200 Mbps. Propagation RTTs of the flows are uniformly distributed in the range between 74 ms and 300 ms. There are 20 long-lived flows going from pool P0 to P7, and 20 long-lived flows in the reverse direction. Each of the bottleneck links r1-r2, r2-r3, r3-r4, r4-r5, r5-r6 are shared by 20 long-lived flows starting from pools P1, P2, P3, P4, P5, and destining to pools P2, P3, P4, P5, P6, respectively. Besides, a web-server in pool P0 generates traffic destined to pool P7, which is described by the same parameters as in Section IX-C2. We vary the number of bottleneck links with the deployed RD scheme between 0 and 5. In Figure 33, we report the throughput of the long-lived flows going from pool P0 to P7 and the goodput of the web-like flows. We observe that wider deployment of the RD Network Services improves the performance of the web-like flows, and affects the performance of the long-lived flows negligibly. Moreover, the goodput of the web-like flows is similar to a power function of the number

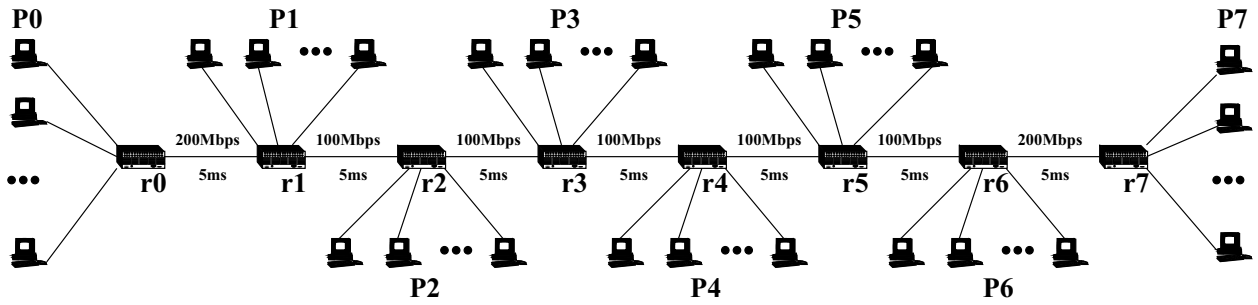


Fig. 32. Multi-bottleneck topology.

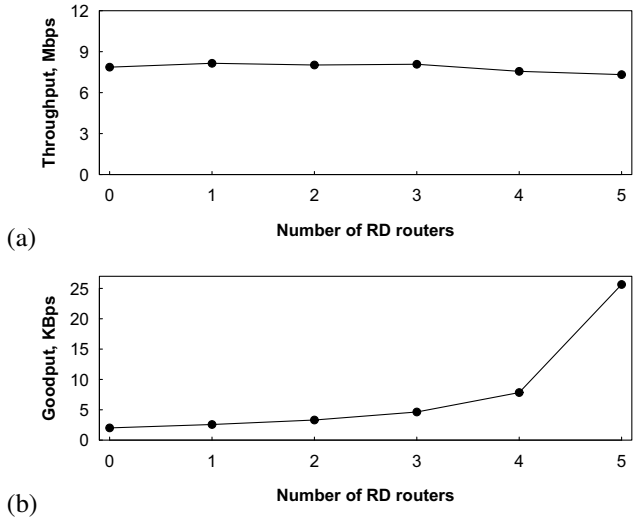


Fig. 33. Incremental deployment in the multi-bottleneck topology: (a) average throughput of the long-lived flows; (b) average goodput of the web-like flows.

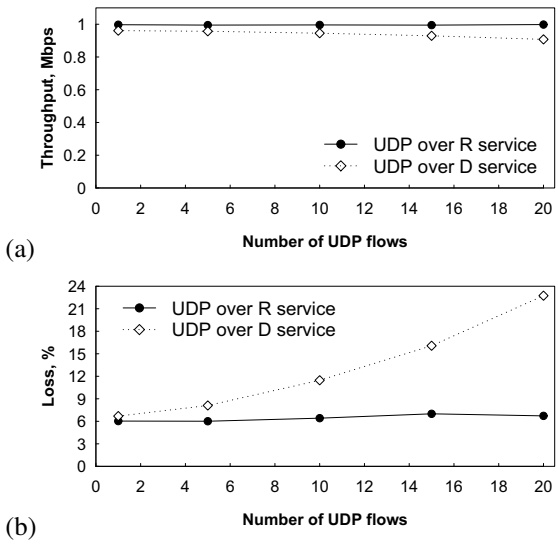


Fig. 34. Choosing the R versus D service for UDP transmission: (a) per-flow throughput of the UDP flows and (b) loss rate of the D service.

of the RD links. We explain that as follows. Assuming that the bottleneck links with the same deployed scheme, i.e., RD or DropTail, have the same loss rate for the web-like flows, we have that the probability of a successful delivery of D packet is a power function of the number of the RD links.

D. Security considerations

Whereas security of the RD architecture needs a separate future evaluation, this section experimentally examines few potential vulnerabilities of the RD design to sender misbehavior. We conduct the experiments in the same network topology and for the same traffic pattern as in Section VII-A3, except for the bottleneck link delay that we set to 10 ms.

First, we explore a scenario with throughput-greedy UDP senders where each of the UDP sources transmits at the constant rate of 1 Mbps. We vary the number of the UDP senders from 1 to 20. The intensity of the web cross traffic is 50 fps. Figure 34 reports the per-sender UDP throughput achieved when the UDP sources use either the R service or the D service. Consistently, the throughput is higher with the R service. Hence, in agreement with our incentive intentions, the RD design steers the throughput-greedy flows to the R service, rather than to the low-delay D service where the negative impact of the excessive UDP transmission on the loss rate would be greater.

Second, we assess an attempt of a throughput-greedy TCP sender to exploit the potentially low transmission rates of delay-sensitive D flows. The throughput-greedy TCP source might increase its throughput by switching from the intended R service to the D service if the legitimate D flows underutilize their share of the bottleneck link capacity. To create such an underutilization, our simulation setup replaces the long-lived D flows with 100 VoIP (Voice over the Internet Protocol) D flows that have the same propagation RTT of 150 ms. To generate the VoIP traffic, we use the tool developed in [6]. In addition to the VoIP flows, the bottleneck link serves web traffic as described in Section VII-A3. We vary the intensity of the web flow arrivals from 1 fps to 150 fps. Queuing delay bound d is set to 50 ms. Figure 35 reveals that the throughput-greedy TCP sender is indeed able to benefit from the misbehavior and attain a significantly higher throughput by switching to the D service. The switch also raises the loss rate of the D service, although the increase is not substantial.

The success of the above attack is not certain and depends on the traffic pattern of the legitimate flows. Now, we con-

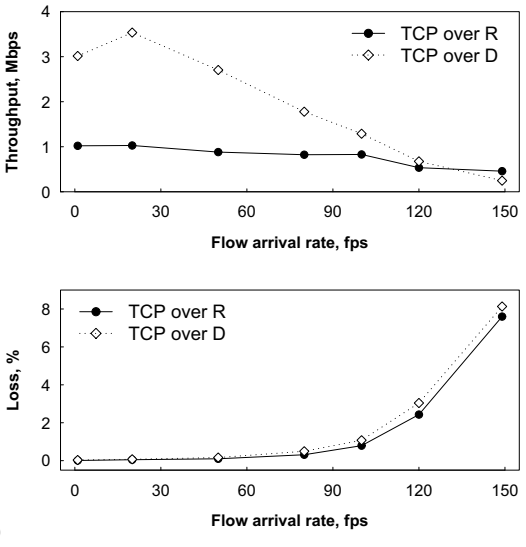


Fig. 35. Exploiting the low transmission rates of legitimate D flows: (a) throughput of the throughput-greedy TCP flow and (b) loss rate of the D service.

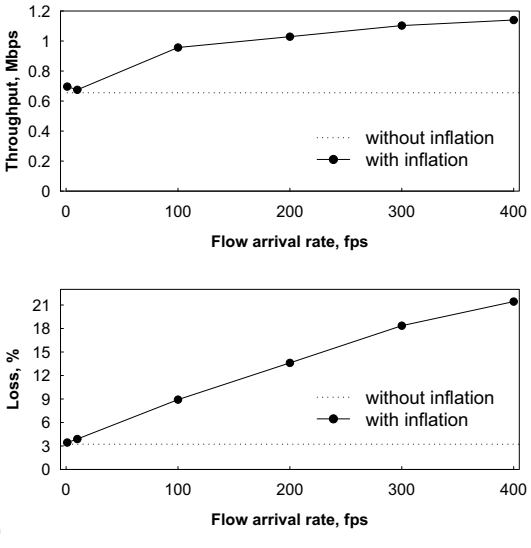


Fig. 36. Inflating the R flow count: (a) throughput of the throughput-greedy TCP flow and (b) loss rate of the D service.

sider an explicit attempt by a throughput-greedy R sender to manipulate the forwarding algorithm at the bottleneck link. More specifically, the throughput-greedy R sender inflates the count of R flows by generating dummy one-packet R flows. In its turn, the inflated flow count increases the bottleneck capacity share allocated to the R class, and this translates into personal throughput benefits for the misbehaving R sender. In our simulations of this scenario, we have no web cross traffic and vary the intensity of the dummy-flow arrivals from 1 fps to 400 fps. Figure 36 confirms that the misbehaving R sender succeeds in improving its throughput substantially. Also, the flow count inflation increases the loss rate for the suppressed D service.

The presented experiments show vulnerabilities of the RD forwarding algorithm to attacks on its flow-counting implementation. The attacks enable a misbehaving sender to acquire both high throughput and low queuing delay at the bottleneck link. While the incentive mechanism of the RD services is imperfect, there exists space for future RD-like designs that assure as large throughput with an R-like service as with a D-like service and as low queuing delay with the D-like service as with the R-like service.

X. Related Work

Network service differentiation has been a topic of extensive research, with the IntServ [9] and DiffServ [8] initiatives being prominent examples. The main feature that favorably distinguishes the RD services from the prior work is their incremental virulent deployability despite continued presence of legacy traffic and legacy service providers.

IntServ offers users an exciting possibility to receive absolute end-to-end rate and delay guarantees for individual flows. To provide the flexible but assured differentiation at the flow granularity, the best IntServ designs employ such complicated link scheduling algorithms as WFQ (Weighted Fair Queuing) [12], WF^2Q (Worst-case Fair Weighted Fair Queueing) [7], Start-time Fair Queueing (SFQ) [22], Virtual Clock (VC) [41], or Earliest Deadline First (EDF) [13] and restrict network access with distributed admission control [20], [29]. In contrast, RD routers maintain only two FIFO queues per output link and schedule the link capacity with the simple algorithm which is easy to implement even at high bitrates. Besides, the RD services exercise no admission control because the latter is ineffective under partial deployment where legacy ISPs keep providing users with unfettered access to shared bottleneck links of the network.

While early retrospectives attributed IntServ deployment failures to the overhead imposed on backbone routers by per-flow storage and processing, core-stateless versions of IntServ designs moved all per-flow state and operations to the network edges and scheduled the core link capacities with simpler algorithms such as Core-Stateless Fair Queuing (CSFQ) [36] or Core Jitter Virtual Clock (CJVC) [37]. The core-stateless IntServ designs put even more faith in access ISPs and also fail to realize the promise of guaranteed services under partial deployment.

DiffServ continued the above trend of focusing on scalability rather than incremental deployment. DiffServ distinguishes services not at the flow granularity but at a much coarser granularity of traffic classes [19]. Various DiffServ designs support either absolute guarantees or relative differentiation between the few traffic classes by employing such algorithmic frameworks as Expedited Forwarding (EF) [27], Assured Forwarding (AF) [10], [24], or Class Selector (CS) [14], [31]. The DiffServ schemes that offer absolute performance guarantees require admission control, e.g., the Premium service of the DiffServ EF designs assures low queuing delay only if the upstream ISPs enforce the maximum rate negotiated for the service [27]. The DiffServ schemes that support relative performance differentiation preserve the Internet openness but

serve one traffic class better than another. Such differentiation requires charging lower prices for worse services because all users would otherwise opt for the best service. Since either admission control or differentiated pricing is ineffective in the presence of legacy ISPs, incremental deployability of all the DiffServ schemes is poor as well. In comparison, the incentives for adopting the RD services are tied only to the performance itself, not the price. The added D service is neither better nor worse than the R service but is merely different, and the RD architecture gives each user complete freedom to select a higher rate or low queuing delay.

Among other proposals for service differentiation, Alternative Best Effort (ABE) [25] resembles the RD services by aspiring to diversify services without distinguishing their prices. In addition to a D-like low-delay green service, ABE offers a blue service with a smaller loss rate. The storage and processing overhead of ABE is substantially larger than for our RD design. Also, while ABE considers it normal for a flow to mark some packets blue and other packets green, potential negative impact of such practices on legacy traffic raises a concern that the ABE design does not incorporate a sound strategy for incremental deployment. Most importantly, the blue service does not consistently provide a larger rate, e.g., by transmitting more aggressively, the green users can enjoy both a higher rate and lower queuing delay than those of the blue users. The lack of explicit rate-delay differentiation significantly weakens incentives for adopting ABE. Best Effort Differentiated Services (BEDS) [17] are similar to ABE and suffer from similar limitations.

XI. Future Work

We believe that the approach of designing for deployability holds great promise for not only network service differentiation but also other types of networking problems. Even within the conceptual framework of rate-delay differentiation, we see numerous opportunities for further fruitful exploration. For example, whereas our strict enforcement of the delay constraint for the D service is a conscious attempt to encourage the service adoption only if the user is really interested in assuredly low queuing delay, it is worth to investigate whether delay should be allowed to spike occasionally as long as average low delay remains guaranteed.

Despite the above envisioned improvements of the RD design, a flow that opts for the D service will likely experience a larger loss rate. The significance of the heavier losses for applications is an interesting topic for future study. If the impact is tangible, we anticipate subsequent design efforts on transport protocols tailored for the D service.

A related issue is whether the RD architecture induces any unintended behavior of users who seek to improve own service or deliberately disrupt services for other users. Although the two-queue design alleviates some denial-of-service attacks, the RD architecture inherits most security problems of the Internet. Furthermore, our own limited experimental evidence indicates that the incentive mechanism of the RD services is imperfect. While securing the RD design is clearly an important area for future investigation, prior simple performance-based [21], [35] and other [39], [40] security proposals constitute promising starting points.

XII. Conclusion

We presented the RD network services, an architecture for rate-delay differentiation in a confederation of network domains owned and operated by multiple providers. Putting an emphasis on incentives for both end users and ISPs to adopt the new low-delay service despite its partial deployment, we designed and implemented the RD services that offer two best-effort services of low queuing delay or higher throughput. The RD router supports the services with two queues per output link, one queue per traffic class. The extensive evaluation revealed that the design supports the intended rate-delay differentiation in a wide variety of settings. Other contributions of the RD services include:

- incremental deployability within the current Internet;
- preservation of the current end-to-end transport protocols and IP datagram header structure;
- elimination of the billing and management problems of previous DiffServ designs.

Besides, our approach of designing for deployability holds promise for solving other types of networking problems.

References

[1] Pulse Code Modulation (PCM) of Voice Frequencies. ITU-T Recommendation G.711, November 1988.

[2] Coding of Speech at 8 kbps Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP). ITU-T Recommendation G.729, March 1996.

[3] Methods for Subjective Determination of Transmission Quality. ITU-T Recommendation P.800, August 1996.

[4] Wideband Coding of Speech at around 16 kbit/s Using Adaptive Multi-Rate Wideband (AMR-WB). ITU-T Recommendation G.722.2, July 2003.

[5] The E-model, a Computational Model for Use in Transmission Planning. ITU-T Recommendation G.107, June 2006.

[6] A. Bacioccola, C. Cicconetti, and G. Stea. User-level Performance Evaluation of VoIP Using ns-2. In *Proceedings NSTools 2007*, October 2007.

[7] J. Bennett and H. Zhang. WF2Q: Worse-case Fair Weighted Fair Queuing. In *Proceedings IEEE INFOCOM 1996*, March 1996.

[8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. IETF RFC 2475, December 1998.

[9] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. IETF RFC 1633, June 1994.

[10] D.D. Clark and W. Fang. Explicit Allocation of Best Effort Packet Delivery Service. *IEEE/ACM Transactions on Networking*, 6(4):362–373, August 1998.

[11] D.D. Clark, J. Wroclawski, K.R. Sollins, and R. Braden. Tussle in Cyberspace: Defining Tomorrow's Internet. In *Proceedings ACM SIGCOMM 2002*, August 2002.

[12] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queuing Algorithm. In *Proceedings ACM SIGCOMM 1989*, September 1989.

[13] M. Dertouzos. Control Robotics: The Procedural Control of Physical Processes. In *Proceedings IFIP Congress 1974*, August 1974.

[14] C. Dovrolis, D. Stiliadis, and P. Ramanathan. Proportional Differentialed Services: Delay Differentiation and Packet Scheduling. In *Proceedings ACM SIGCOMM 1999*, September 1999.

[15] N. Dukkipati and N. McKeown. Why Flow-Completion Time is the Right Metric for Congestion Control. *ACM SIGCOMM Computer Communication Review*, 36(1):59–62, 2006.

[16] C. Estan, G. Varghese, and M. Fisk. Bitmap algorithms for counting active flows on high speed links. *IEEE/ACM Transactions on Networking*, 14(5):925–937, October 2006.

[17] V. Firoiu, X. Zhang, and Y. Guo. Best Effort Differentiated Services: Tradeoff Service Differentiation for Elastic Applications. In *Proceedings IEEE ICT 2001*, June 2001.

[18] S. Floyd and T. Henderson. The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 2582, April 1999.

[19] S. Floyd and V. Jacobson. Link Sharing and Resource Management Models for Packet Networks. *ACM Transactions on Database Systems*, 3(4):365–386, August 1995.

[20] S. Gorinsky, S. Baruah, T. Marlowe, and A. Stoyenko. Exact and Efficient Analysis of Schedulability in Fixed-Packet Networks: A Generic Approach. In *Proceedings IEEE INFOCOM 1997*, April 1997.

[21] S. Gorinsky, S.Jain, H. Vin, and Y. Zhang. Design of Multicast Protocols Robust against Inflated Subscription. *IEEE/ACM Transactions on Networking*, 14(2):249–262, April 2006.

[22] P. Goyal, H. Vin, and H. Cheng. Start-time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks. In *Proceedings ACM SIGCOMM 1996*, August 1996.

[23] R. Guerin, S. Kamat, V. Peris, and R. Rajan. Scalable QoS Provision Through Buffer Management. In *Proceedings ACM SIGCOMM 1998*, September 1998.

[24] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. IETF RFC 2597, June 1999.

[25] P. Hurley, J.-Y. Le Boudec, P. Thiran, and M. Kara. ABE: Providing a Low-Delay Service within Best Effort. *IEEE Network*, 15(3):60–69, May/June 2001.

[26] V. Jacobson. Congestion Avoidance and Control. In *Proceedings ACM SIGCOMM 1988*, August 1988.

[27] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. IETF RFC 2598, June 1999.

[28] H.-A. Kim and D. O'Hallaron. Counting Network Flows in Real Time. In *Proceedings IEEE GLOBECOM 2003*, December 2003.

[29] J. Liebeherr, D. Wrege, and D. Ferrari. Exact Admission Control for Networks with a Bounded Delay Service. *IEEE/ACM Transactions on Networking*, 4(6):885–901, December 1996.

[30] S. McCanne and S. Floyd. *ns Network Simulator. http://www.isi.edu/nsnam/ns/.*

[31] K. Nichols, S. Blake, F. Baker, and D.L. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. IETF RFC 2474, December 1998.

[32] M. Podlesny and S. Gorinsky. RD Network Services: Differentiation through Performance Incentives. In *Proceedings ACM SIGCOMM 2008*, August 2008.

[33] J. Postel. Internet Protocol. DARPA Internet Program. Protocol Specification. IETF RFC 791, September 1981.

[34] K. K. Ramakrishnan and R. Jain. A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with a Connectionless Network Layer. In *Proceedings ACM SIGCOMM 1988*, August 1988.

[35] S. Savage, N. Cardwell, D. Wetherall, and T. Anderson. TCP Congestion Control with a Misbehaving Receiver. *ACM Computer Communications Review*, 29(5):71–78, October 1999.

[36] I. Stoica, S. Shenker, and H. Zhang. Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks. In *Proceedings ACM SIGCOMM 1998*, September 1998.

[37] I. Stoica and H. Zhang. Providing Guaranteed Services Without Per Flow Management. In *Proceedings ACM SIGCOMM 1999*, September 1999.

[38] C. Villamizar and C. Song. High performance TCP in ANSNET. *ACM SIGCOMM Computer Communication Review*, 24(5):45–60, October 1994.

[39] D.J. Wetherall, U. Legedza, and J. Guttag. Introducing New Internet Services: Why and How. *IEEE Network*, 12(3):12–19, May-June 1998.

[40] X. Yang, D. Wetherall, and T. Anderson. A DOS-limiting Network Architecture. In *Proceedings ACM SIGCOMM 2005*, August 2005.

[41] L. Zhang. A New Traffic Control Algorithm for Packet Switching Networks. In *Proceedings ACM SIGCOMM 1990*, August 1990.

[42] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker. On the Characteristics and Origins of Internet Flow Rates. In *Proceedings ACM SIGCOMM 2002*, August 2002.