NetEx: Cost-effective Bulk Data Transfers for Cloud Computing

Massimiliano Marcon
MPI-SWS

Nuno Santos
MPI-SWS

Krishna P. Gummadi
MPI-SWS

Nikolaos Laoutaris
Telefonica Research

Pablo Rodriguez
Telefonica Research

Amin Vahdat
UCSD

Technical Report: MPI-SWS-2011-005

ABSTRACT

Cloud computing offers individuals and organizations the potential for reducing their IT costs. In this paper, we focus on the problem of high bandwidth prices charged by cloud providers for customers' data uploads and downloads. The cost of such data transfers can become prohibitive when the volume of data transferred is large. The high price of data transfers reflect the cost of raw bandwidth that cloud providers pay to transit ISPs. Raw bandwidth is expensive because ISPs need to overprovision their networks for peak utilization. In this paper, we propose that ISPs use the spare capacity on their backbone links to deliver bulk data. Since ISPs make more effective utilization of otherwise unused bandwidth, they can offer this service at lower prices, which will benefit cloud providers and cloud users. Cloud users could use this service to ship delay-tolerant data, e.g., data backups, software distributions, and large data sets. We present NetEx, a bulk transfer system that opportunistically exploits the excess capacities of network links to deliver bulk content cheaply and efficiently. NetEx uses bandwidth-aware routing, which adapts to dynamically changing available bandwidth across potentially multiple paths between the source and the destination of a bulk transfer. Because NetEx works in the background scavenging unused bandwidth, ISPs can easily deploy it over routers that support simple priority queueing without affecting existing Internet traffic or routing. We evaluated NetEx using data gathered from the backbone of a large commercial Tier-1 ISP. Our results show that NetEx achieves near-optimal utilization of spare link bandwidth and that ISPs can use it to deliver 60% to 170% more data than what they transfer today.

1. INTRODUCTION

Cloud computing can bring its users significant benefits, especially economical ones. Services offered by cloud providers like Amazon, Google and Microsoft, allow their users to outsource storage and computation for which they pay based on the "pay as you go" model. This model is very attractive for cloud users (both companies and private users), since it eliminates upfront costs for hardware, software and IT. Nevertheless, cloud users must decide whether it is economically viable to move to the cloud before doing so. This decision greatly depends on the workload they expect to outsource and the prices charged by cloud providers.

In particular, the cost of data transfers may soon constitute an economic bottleneck for cloud users [5]. Cloud providers charge their cloud users for bandwidth utilization. These costs may become relevant when the size of data transfers is high, or when the transfers are frequent. For example, as of April 2011, Amazon charges $0.1 per Gbyte transferred in-/out- of the cloud [4]. With these prices, a cloud user pays $100 for uploading a 1TB large data backup or map-reduce job to the cloud. The same amount is payed for serving 100K downloads of a small 10MB file e.g., a software package. Lowering bandwidth costs is fundamental to make cloud computing more attractive.

To a large extent, the bandwidth costs charged by cloud providers are conditioned by the raw bandwidth costs in the Internet. In the Internet, these bandwidth costs are largely dictated by tier-1 ISPs, which carry network traffic across cities, countries or continents. Tier-1 ISPs are provider-free, meaning that they don't pay any other ISP for connectivity. Instead, they charge their customers (either other ISPs or companies) high bandwidth prices based on the peak utilization of their access links [29]. Tier-1 ISPs charge based on peak utilization in order to discourage congestion and avoid high latency or packet losses. Because cloud providers generally connect their datacenters to tier-1 transit ISPs, they pay for bandwidth based on peak utilization, both for traffic between cloud users and datacenters and between datacenters in different locations. The high bandwidth costs paid by cloud providers end up inflating the bandwidth costs paid by cloud users.

One way to reduce data transfer costs is to ship data stored on physical media (e.g., hard disks) via postal services [3, 20], thus delivering very large volumes of data (on the order of terabytes) at high throughput with low per-byte cost. However, this is not a viable solution for small to large transfers (on the order of gigabytes), or when data has be delivered within a few hours or a day [29]. A more radical approach is to reduce the cost of ISPs' networking hardware, for example by building routers from commodity components [14, 32]. However, these solutions require radical changes to ISP backbones, and therefore are less likely to be deployed in the short term.

Our idea is to use the spare capacity of ISP backbone links to reduce the costs of bulk data transfers in cloud computing. Our focus is on bulk data transfers that are delay-tolerant at the packet level, for example file backups, software downloads, or data transferred for data mining and analytics. Because individual packets within a bulk data transfer are not latency-sensitive, packets can be delayed and sent only when spare bandwidth is available. Also, it is known that ISPs have significant reserves of spare capacity on their backbones caused by overprovisioninig and diurnal patterns [36]. Thus, ISPs could exploit this unused link capacity to deliver higher volumes of bulk data without having to upgrade their backbone links, thus lowering the cost of data transfers.

Using spare capacity for bulk transfers raises several important questions: (i) What incentives do ISPs have for using the spare capacity in their backbone links? (ii) How could an ISP offer a service that enables bulk data transfers in cloud computing? (iii) How much spare capacity could actually be used in a real scenario? To address these questions, this paper makes three main contributions.

First, we propose a service model that gives ISPs an economic incentive to exploit their spare network capacity. Our model comprises two classes of services: the standard best-effort service used for latency-sensitive traffic, and a bulk service that opportunistically uses the left over capacity for latency-tolerant bulk content. ISPs can charge their customers lower prices for the bulk service, and yet, depending on the expected growth of bulk demand, increase revenue.

Second, we present a networked system called NetEx that enables an ISP to provide a bulk transfer service. NetEx efficiently exploits the spare resources of the ISP backbone links while preventing bulk traffic from interfering with best-effort traffic. To achieve this, NetEx uses bandwidth-aware routing which intelligently selects paths within the ISP backbone with the most available spare capacity. At the routers, NetEx differentiates traffic into best-effort and bulk traffic classes, and forwards bulk traffic with strictly lower priority than best-effort traffic. To identify bulk traffic, NetEx relies on the cloud service to mark bulk packets according to the semantic of the service and preferences of cloud users.

Finally, we used real data gathered from an intercontinental commercial Tier-1 ISP backbone and from a national research backbone to study how well NetEx would perform if it were deployed in tier-1 ISPs. Our evaluation shows that tier-1 ISPs can use NetEx to deliver 60% to 170% more data than what they transfer today. Compared to traditional routing schemes, NetEx's bandwidth-aware routing increases the amount of bulk data NetEx can deliver by a factor of two, and achieves near-optimal utilization of spare resources.

Overall, our solution is practical and has the potential to reduce the bandwidth costs payed by cloud users. NetEx can be easily deployed by tier-1 ISPs since it does not require changes to router hardware, and does not require changes in the way network operators deal with best-effort traffic today. Although NetEx can be deployed independently by a single ISP, its benefits can increase as multiple ISPs incrementally deploy it. Cloud providers conntected through ISPs that offer NetEx can enjoy lower bandwidth costs and therefore charge their users lower prices.

2. LOW-COST BULK DATA TRANSFERS

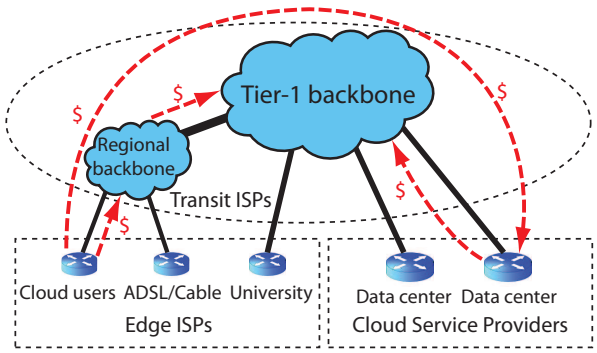


Figure 1: High-level structure of the Internet: Edge ISPs produce and consume data which is transferred by transit ISPs

In this section, we first argue why it makes sense for Tier-1 ISPs to offer a low-cost bulk data transit service to cloud providers. We then present our proposal for such a service, and finally discuss the potential benefits for the involved parties.

2.1 Transit ISPs in the cloud ecosystem

The focus of this paper are Internet's *transit* ISPs, and especially, Tier-1 ISPs. These are the long-haul networks that transfer data across large distances (e.g., inter-city, trans- and inter-continental distances). Transit ISPs interconnect multiple players in the cloud ecosystem, and set the base price for bandwidth.

Figure 1 represents the cloud ecosystem with particular emphasis on transit ISPs, cloud providers (CPs), and cloud users. Transit ISPs operate the backbone networks that transfer data between *edge* ISPs (e.g., university, corporate, and residential broadband networks) located in different regions of the world, and cloud service providers (CPs) like Amazon or Google. Usually, CPs comprise multiple geographically dispersed datacenters, and rely on transit backbones provided by Tier-1 ISPs for connecting them. CPs provide online cloud services such as Amazon's EC2 or S3 services, which cloud users access from the network edge. Cloud users are often connected to the Internet through regional or access networks operated by Tier-2 ISPs.

In this scenario, cloud users pay cloud providers for cloud services (e.g. Amazon EC2) and also pay their ISPs (the regional backbone in Figure 1) for connectivity. Regional ISPs in turn need to connect their users to cloud providers, and therefore they pay transit ISPs for connectivity (Figure 1). Finally, cloud providers also pay transit ISPs for connectivity because they need to exchange data with their customers.

Because Tier-1 ISPs occupy the apex of the Internet ecosystem, they are in a privileged position and ultimately determine the raw cost of bandwidth. As we explain below, raw bandwidth costs are largely determined by peak load and the temporal characteristics of network traffic.

2.2 Transit ISPs are designed for and charge for peak load

The main reason for the high cost of transit bandwidth is that contracts between transit ISPs and their customers include SLAs [43] that specify the requirements for quick recovery of failures and performance guarantees on packet

delivery delays, jitter, and loss. Performance guarantees at the packet level are important for interactive applications, like Skype and Web browsing, which perform poorly even when a small fraction of packets is lost or delayed.

Transit ISPs today adopt a simple way of satisfying SLAs: overprovisioning their links and charging for the peak load. By designing their networks for peak load, they are likely to be able to sustain unexpected traffic spikes without breaking their SLAs. Because they design their network around peak traffic load, ISPs also charge their customers based on peak utilization, which is reflected in the widely used 9^{th}-percentile billing. In 95^{th}-percentile billing, a billing cycle (typically a month) is split in 5-minute intervals. At the end of the billing cycle the intervals are sorted based on the amount of traffic that was sent/received; the 95^{th}-percentile interval is then used to compute the bill.

In order to sustain traffic spikes and avoid congestions that would break their SLAs, transit ISPs overprovision their networks to carry a traffic load that is well above the expected traffic. This creates a lot of spare bandwidth that is not used outside peak periods. Further, the standardization of link technologies forces ISPs to increase bandwidth in large chunks, even if it leads to unused capacity. For example, an OC-12 (622 Mbps) link is typically upgraded to an OC-48 (2.5 Gbps) or OC-192 (10 Gbps) but nothing in between. In addition, to achieve quick recovery from link failures, most pairs of backbone routers have at least two disjoint paths between them [45]. Thus, if network links are lightly loaded due to overprovisioning, ISPs can quickly and transparently recover from a failure by rerouting traffic to another path [24]. Furthermore, network operators normally use rules of thumb like "upgrade at 40% or 50% utilization" or "at maximum 75% utilization under failure" [46] which may lead to overly conservative reserves of spare bandwidth in ISP backbones.

Another source of spare bandwidth are the diurnal patterns that result in many backbone links in ISPs exhibiting relatively low utilization when averaged over a day [7]. Studies of different backbones have consistently shown that the average usage of the network links tends to be significantly lower than their capacity [24, 36, 42].

2.3 Using spare bandwidth for bulk transfers

ISPs can improve the efficieny of their networks by leveraging the spare capacity of backbone links to deliver higher volumes of data, specifically *bulk data*. This is possible due to the unique characteristics of bulk traffic workload.

Many bulk transfers taking place in the cloud, such as file backups and scientific data transfers, have less stringent SLA requirements than typical interactive traffic. First, bulk transfers normally take a long time to complete and are less sensitive to delays, especially at the granularity of individual packets. Unlike interactive traffic, bulk traffic could be delayed to a later point of time, when the network is lightly loaded and there is more spare bandwidth (e.g., at nighttime). In addition, long-running bulk transfers can tolerate occasional periods of network congestion, and do not require the packet-level performance guarantees ISPs offer in their SLAs. Finally, bulk transfers are more tolerant to short term link or path failures, and ISPs do not have to overprovision backup paths a priori for quick failure recovery of bulk transfers.

Thus, ISPs can adapt their services to use network resources more efficiently. We propose that, in addition to standard *hard-SLAs* for traffic with strict packet-level requirements, ISPs provide a low priority *soft-SLA* service for bulk data transfers. Soft-SLA traffic does not interfere with hard-SLA traffic and only uses capacity that hard-SLA traffic doesn't currently use. This isolation properties, together with the flexibility of bulk traffic at the packet level, enable soft-SLAs traffic to fully saturate the link capacity without compromising the quality of transfers. This allows ISPs to increase the peak throughput, and maximize the usage of backbone links. Of course, interactive traffic like video streaming should not be sent as soft-SLA because this could degrade the quality of service.

Next, we discuss how to monetize the spare bandwidth in order to bring benefits for both ISPs and their customers.

2.4 Monetizing the spare bandwidth

Although ISPs have the potential to use the spare capacity of their backbones, they must have a clear economic incentive for doing so. In particular, an ISP must find a pricing model for soft-SLA traffic that can attract demand and consequently raise profit. In principle, since such a service would be limited to using the spare resources when available, an ISP should offer it at a lower cost than the current Internet service. However, pricing the low-cost service is not trivial: if the price is too low, customers might divert too much traffic from the high-cost to the low-cost class; conversely, if the price is too high, customers might refrain from using the new service. In both situations the ISP would be penalized.

Finding the right pricing model requires a thorough understanding of the market and workload trends, and therefore it is out of the scope of this paper. Such a task should be conducted by ISPs, which have entire departments that focus on pricing of their services with cumulated experience in pricing tiered services. Nevertheless, we elaborate a simplified pricing model to help grasp the tradeoffs and potential benefits for ISPs and their customers.

Our pricing model assumes that the introduction of the low-cost, soft-SLA bulk service will encourage more users to adopt cloud services and thus lead to an overall increase in the demand for bulk data.

We propose a double pricing scheme, where hard-SLA traffic is charged according to the usual 95th-percentile billing, while soft-SLA traffic is charged per byte. This scheme accommodates the requirements of hard-SLA traffic, and fairly reflects the opportunistic nature of the soft-SLA model. Thus, the potential increase in revenue for an ISP depends on the prices the ISP sets for hard and soft SLA traffic, as well as the growth in bulk traffic demand expected from the introduction of the new service. We express the increase in revenue r^T as the ratio of the revenue from the double pricing model R_2^T to the revenue from the single pricing model R_1^T during a billing period of length T:

$$r^T = \frac{R_2^T}{R_1^T} = \frac{p_{95}(A^T) \cdot h + B^T \cdot s}{p_{95}(A^T + B^T) \cdot h} \tag{2.1}$$

where A^T and B^T are the total amount of best-effort and bulk traffic expected during the time period of length T, respectively. The charging rates for hard-SLA and soft-SLA

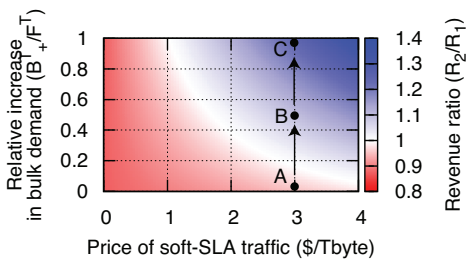


Figure 2: Revenue ratio r^T in the discussed example: The arrows represent the evolution in bulk demand from A ($r^T = 0.95$) to B ($r^T = 1.10$) to C ($r^T = 1.24$).

traffic are expressed by the variables h and s, respectively, and p_{95} detones the 95^{th}-percentile function.

In single pricing, all traffic is considered hard-SLA. Therefore, best effort and bulk traffic are charged based on the 95^{th}-percentile and the revenue R_1^T depends on the hard-SLA bandwidth price h (in \$/Mpbs) and the total 95^{th}-percentile bandwidth utilization. In double pricing on the other hand, the total revenue R_2^T is the sum of the soft and hard SLA bills computed independently. Hard-SLA traffic is billed as in single pricing (i.e. based on the 95^{th}-percentile), while soft-SLA traffic is billed based on its total amount at a rate of s \$/Mbit.

This simplified model can help ISPs find a price that is attractive for customers and is likely to increase revenue. To illustrate this, we use a hypothetical example targeting a single 1Gbps access link, and estimate the potential revenue under some reasonable assumptions. Assume that the average link utilization is 40% (i.e. the free capacity on the link is $F^T = 0.6Gbps \cdot 1month \approx 194Tbyte$) , with bulk traffic accounting for 40% of the total traffic ($B^T = 0.16Gbps \cdot 1month \approx 52Tbyte$), and the 95^{th}-percentile traffic load amounts to 80% of the link capacity. Using the bandwidth pricing tables of 2008 [29], the monthly bandwidth cost at the 95^{th}-percentile is \$20,000 for 1Gbps ($h = \$20/Mpbs$). We conservatively assume that the customer will divert all its bulk traffic to the cheaper soft-SLA class as soon as the ISP enables double pricing. As a consequence, we assume that the 95^{th}-percentile of hard-SLA traffic will drop from 80% to 70% since cloud providers already shift a significant portion of the bulk traffic from peaks to the troughs, normally over night [11].

Under these conditions, Figure 2 plots the revenue ratio for different values of s (on the x-axis) and increases in bulk traffic demand B_+^T relative to the total amount of spare capacity F^T (y-axis). A value of 1 on the y-axis means that all the spare capacity on the link is used. It is annotated with points A, B, and C showing the increase in revenue for a soft-SLA price of $\$3/TByte$ and different increases in bulk traffic demand. With a soft-SLA price of $\$3/TByte$, ISPs will lose 5% of their revenue if bulk traffic demand doesn't increase (A). However, if bulk traffic demand increases so that half of the link's spare capacity is used, the revenue of the ISP increases by 10% (B). Finally, if bulk traffic uses all the spare capacity, the ISP gains rise to 24% (C). Notice that today, without ISP support, cloud services cannot drive their access link usage to such high levels without increasing their 95^{th}-percentile traffic and thus incurring considerable additional charges.

In conclusion, ISPs have a lot of spare capacity in their backbones that they can exploit and monetize. They have

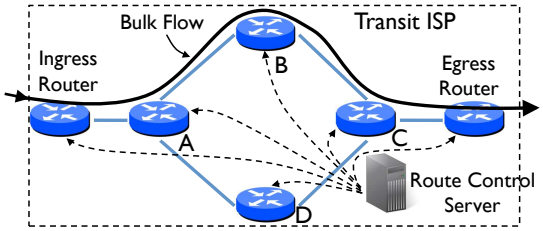


Figure 3: Deployment of NetEx in a Tier-1 ISP: Bulk transfers are routed through the ISP following the routes computed by the Route Control Server (RCS).

economic incentives to take advantage of this spare bandwidth while offering their customers a cheaper service for carrying bulk data. Next, we show how can ISPs offer this service in practice.

3. NETEX DESIGN

NetEx is a system that Tier-1 ISPs can deploy to provide an opportunistic bulk data transfer service. Its design was guided by two goals. First, NetEx should allow ISPs to achieve efficient use of spare bandwidth in their backbone links. Second, the design should be easy to deploy in practice and must offer immediate benefits to ISPs that deploy NetEx. When faced with design choices that deliver similar performance gains, we opted for choices that require fewer changes to existing network infrastructure, or facilitate the maintenance of the system by the network operators.

3.1 Overview

Essentially, NetEx design uses two main techniques: *traffic differentiation* and *bandwidth-aware routing*. The first allows bulk transfers to exploit spare bandwidth without interfering with existing traffic, while the second achieves efficient use of the spare resources.

1. Traffic differentiation: Traffic differentiation is necessary to allow bulk transfers to use left-over bandwidth without affecting best-effort traffic. NetEx separates traffic into best-effort traffic that is delay sensitive and bulk traffic that is delay tolerant. Best-effort traffic is forwarded without any change, while bulk traffic is forwarded with strictly lower priority, i.e., bulk traffic is sent only when there is no best-effort traffic waiting to be sent.

2. Bandwidth-aware routing: To achieve efficient use of spare resources, transit ISPs would have to modify the default routing within their networks. Intra-domain routing today is not optimized for bandwidth: ISPs do not use all possible paths between a pair of nodes, they do not necessarily pick the paths with the most available bandwidth, and they do not adapt their paths dynamically as the available bandwidths on the links vary. NetEx addresses these limitations by employing bandwidth-aware routing that is optimized to pick potentially multiple paths between a pair of nodes to deliver the most data, independent of the paths' latencies. NetEx also periodically recomputes its routes to account for changes in network conditions.

In the rest of this section, we first describe the architecture of NetEx. Next, we clarify how cloud services can use NetEx for bulk data transfers. Later we discuss how NetEx can be deployed by ISPs.

3.2 System architecture

Figure 3 illustrates the components of NetEx when deployed in the backbone of a transit ISP. NetEx comprises a dedicated controller called *Route Control Server* (RCS) which coordinates routing of bulk traffic in the backbone, and a router firmware extension that implements the forwarding protocols for bulk traffic. The ISP routers need to be upgraded with this firmware extension.

At a high-level, the system works as follows. The border routers of the ISP identify incoming packets as bulk or best-effort traffic. The bulk traffic is then forwarded within the ISP backbone according to routing tables computed by the RCS and distributed to the routers. The RCS computes the routes using NetEx's bandwidth-aware routing algorithm. It uses the recent state of the network links to estimate spare capacity and bulk traffic load in the near future [28, 29]. When forwarding packets along the routes, NetEx routers forward bulk traffic at a strictly lower priority than best-effort Internet traffic.

To identify whether packets are bulk, border routers check if the DSCP [34] bit field in their IP headers is set to a specific value[1]. NetEx requires packets to be classified as bulk before they reach the ISP's border routers. We now discuss how packets can be classified.

3.3 Classifying network traffic as bulk

Classifying a network packet as bulk requires knowledge of its application payload. Therefore, the application that generates the traffic is in the best position to classify traffic. In the case of cloud services, CPs must modify the protocol used by their cloud services to mark bulk traffic accordingly. For example, in Amazon S3, the protocol between the cloud servers and the client application running on the user's side needs to be modified to mark bulk traffic with the appropriate DSCP value that NetEx honors.

We propose two methods by which CPs can adapt their current protocols to use NetEx. One way is for the client-side software to explicitly mark its packets. In S3, this would mean that the client-side software must be modified to set the DSCP bit of the packets it generates. The alternative is to use a proxy responsible for setting the packets' DSCP bit on behalf of the sender. This proxy could be deployed by the ISP or by the cloud user in its own private network. Client requests can be transparently redirected to the closest proxy in a way similar to how requests for content served by Akamai are currently redirected to the closest Akamai proxy cache. For example, in the case of S3, a cloud user resolves an Amazon URL, and Amazon's DNS server redirects the request to a second DNS server located in and controlled by the ISP. Then, this second DNS server returns the IP of the most appropriate ISP-managed NetEx proxy. Because NetEx is a service offered by ISPs, they have an incentive to deploy and manage proxies and DNS servers to assist users.

Whereas the method based on modifying client-side software is simpler, it is only feasible if the CP controls the client-side software. If this is not the case, for example because the client-side software is a standard Web browser, a proxy-based solution is preferable.

3.4 Deploying NetEx in ISPs

As explained below, NetEx can be independently deployed by single or multiple ISPs without significant barriers.

[1]DSCP was previously termed ToS [1]

Deployment within a single ISP: Bandwidth-aware routing can be deployed by an ISP today without changes to router hardware. Many Internet routers support traffic prioritization [12] as well as software and firmware upgrading. The hardware investment in the RCS is negligible.

Deployment across multiple ISPs: NetEx does not require any changes to inter-domain routing protocols. Inter-domain bulk transfers are routed along the same inter-AS paths computed by BGP today. Although there may be additional benefits to be gained by having a cross-ISP implementation of NetEx, this would require major changes to BGP, a complex protocol with many specialized policies and mechanisms. Moreover, to deploy NetEx across multiple networks, ISPs would have to disclose information about their topologies and traffic matrices, something that ISPs have been reluctant to do in the past. We therefore suggest that ISPs independently and incrementally deploy NetEx, and route inter-domain bulk transfers on standard BGP paths.

As we showed in Section 2.4, whenever a transit ISP deploys NetEx, its direct customers can benefit from a reduction in costs. Thus, NetEx also provides immediate incremental deployment benefits to each adopting ISP and its customers. CPs peering with these ISPs can also set more competitive bandwidth prices for their services.

4. DETAILED DESIGN

In this section we present NetEx in detail. We first describe how the Route Control Server (RCS) computes and disseminates routes. Then, we show how routers route and forward bulk packets and handle congestion. Finally, we discuss the scalability and availability of the system.

4.1 Route computation and dissemination

In NetEx, the routing of bulk transfers is globally coordinated by the Route Control Server (RCS). To adapt to constantly changing available bandwidths, the RCS periodically recomputes the routes using NetEx's bandwidth-aware routing algorithm. Time is divided into time intervals. At the end of each interval, the RCS uses link utilization and traffic demands observed during the previous interval as inputs. The RCS periodically collects this information directly from the ISP routers. To improve the flexibility of NetEx, the RCS can also be configured by the network operator with alternative routing algorithms. We demonstrate this feature when we evaluate the performance of NetEx in Section 6.

Once new paths are computed, the RCS disseminates the routing information to all routers in the ISP, ensuring that the new routes are used during the next time interval. Because the propagation of tables to all routers is not globally synchronized, there may be periods of time when the lookup tables at different routers are inconsistent. To mitigate this problem, the RCS generates unique and monotonically increasing labels for the new paths it computes, and routers discard old labels when they are updated. Routers simply drop packets tagged with old labels they do not recognize. Thus, routing inconsistencies during the short period of time when routes are updated would lead to dropped packets. Our evaluation in Section 6 shows that routes only need to updated once every 30 minutes, which suggests that the potential impact of updates on flows is small.

Bandwidth-aware routing algorithm: The goal of bandwidth-aware routing is to maximize the amount of bulk

data that can be delivered using the spare bandwidth on network links. We cast the routing problem as a maximum concurrent flow problem [39] for which there are efficient solving techniques. The inputs to the optimization problem are (a) the network topology, (b) the available link bandwidths, and (c) a matrix containing the commodity (data) transfer demands between any pair of routers in the network. The goal of the optimization problem is to maximize the minimal fraction of the flow of each commodity to its demand. This is equivalent to finding (potentially multi-path) routes that maximize the amount of data that can be delivered while conserving the demand matrix. To solve the optimization problem we use CPLEX [13], a commercial optimization software. The output from the algorithm specifies, for each router in the network, the fraction of a given source-destination transfer that should be routed on a given outgoing link. This information is used to generate the paths for the entire network. Appendix A describes the linear problem formulation.

4.2 Path assignment at the border routers

To identify a bulk data packet, ingress border routers (i.e., the routers through which traffic enters the ISP) check whether the DSCP bit field in the IP headers of its packets is set. We explain in Section 3.3 how the DSCP field can be set. After identifying a bulk data packet by observing the appropriate DSCP value, an ingress router assigns the packet a path along which it should be routed within the ISP.

The ingress router determines the right path by looking up two routing tables. First, the router looks up the inter-domain BGP routing tables to determine the egress border router (i.e., the router through which traffic must exit the ISP). Second, after determining the egress router, the ingress router consults the NetEx routing table, which is installed by the central RCS route control server. This table contains the set of MPLS paths [38] currently used by NetEx to route bulk traffic between each ingress and egress router. Each MPLS path is also associated with a weight, which is proportional to the fraction of bulk traffic between the ingress and egress router that the path must carry. By splitting traffic along the paths in proportion to their weights, the router ensures that traffic is balanced according to the strategy that best uses the spare capacity of links.

Because out-of-order delivery can hinder the performance of TCP flows, it is important to route all packets belonging to the same flow on the same path. To ensure this without keeping per-flow state at routers, we use the packet-header based hashing technique described in [23].

4.3 Packet forwarding and congestion handling

To forward bulk packets within the ISP network, NetEx establishes MPLS tunnels between the ingress and the egress routers along the paths determined by the RCS. To signal the path along which a bulk packet must be routed, ingress routers augment the packet with an MPLS header containing the appropriate MPLS label corresponding to the path. Each router along the path keeps a simple forwarding table that maps each MPLS label to the next downstream router and the outgoing interface number. MPLS packets are forwarded hop-by-hop using these tables until they reach the egress router, where the packet is decapsulated and routed to the adjacent ISP.

When forwarding packets, NetEx routers transmit bulk traffic with a strictly lower priority than best-effort traffic. NetEx routers use two queues, one for best-effort traffic and one for bulk traffic. Packets from the bulk traffic queue are sent only when there are no packets in the best-effort traffic queue. By giving strict higher priority to best-effort traffic, NetEx ensures that bulk traffic does not affect best-effort traffic and that bulk traffic only uses spare bandwidth. Many Internet routers already support multi-queue scheduling [12] and ISPs can prioritize traffic by simply enabling such scheduling. Note that traffic prioritization doesn't require any changes at end hosts.

During link congestion, the router drops all the incoming bulk packets while the bulk traffic queue is full. An alternative design is having the router buffer bulk packets in memory or local storage while the bulk traffic queue is full. Although this approach reduces packet retransmissions and therefore saves spare bandwidth, it requires fundamental changes to existing networks. We believe that such clean-slate designs are warranted only when they bring considerable performance benefits. However, our evaluation shows that bandwidth-aware routing by itself achieves near-optimal performance. Thus, in order to preserve deployability in today's networks, NetEx does not store bulk traffic when the bulk traffic queue is full.

4.4 Scalability and availability

The centralized RCS could constitute a scalability and availability bottleneck. In particular, the complexity of the optimization problem solved by the RCS could introduce a computational bottleneck. From our experience, however, the optimization problem can be solved efficiently, and modern hardware can easily solve even larger networks than the ones we have studied (see Section 6.3).

Regarding availability, NetEx may be affected if the RCS goes offline for a duration that exceeds the routing update period. Since the RCS is stateless, tolerating these situations is straightforward. The ISP can keep secondary RCS replicas in standby, which can quickly take over if the primary RCS fails. To reduce the risk of correlated failures, multiple replicas should be located in different PoPs.

5. IMPLEMENTATION

To ensure that we have not ignored any detail or subtle issues in our design, we implemented a fully functional prototype of NetEx and verified that all the forwarding and routing techniques work as expected.

Our NetEx prototype was implemented on Linux. The prototype consists of a software router and a *route control server* (RCS) process. To implement the software router, we used the Click [26] modular router running in kernel mode. We based our software router on the Click reference specification of a standard IPv4 router [26], which we extended to provide the additional functionality required by NetEx.

The RCS is implemented as a user-level daemon that periodically collects traffic information from the routers, computes the new NetEx routing tables and distributes them to the software routers. The RCS and the software routers communicate using TCP connections.

In addition to the specification of the software router in the Click declarative language, we implemented 5 new Click

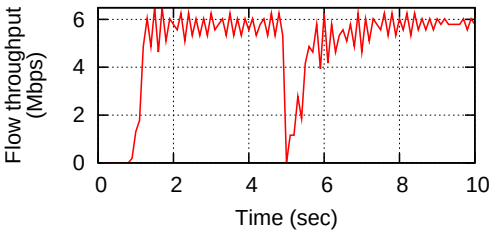


Figure 4: Effect of path rerouting on the throughput of bulk flows: A bulk flow takes approx. 1 second to fully recover from a change in the routing path.

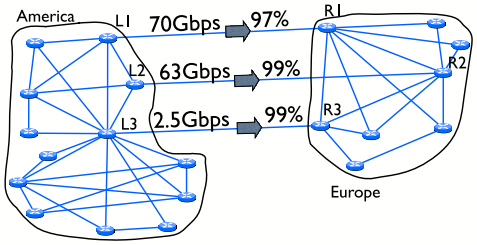


Figure 5: Tier-1 topology used to evaluate NetEx: A large commercial Tier-1 ISP operating in three continents

elements in C++ (2347 lines of code). The RCS was implemented in Perl (2403 lines of code).

6. NETEX EVALUATION

In this section, we study how well NetEx would perform when deployed within a single large transit ISP. In particular, we are interested in answering three questions: (a) How much more bulk data can NetEx send compared to current shortest path routing? (b) How well do bulk flows perform in terms of throughput? and (c) Which aspects of NetEx, the workload, and the topology used are more important for shaping the final gains?

6.1 Prototype evaluation

We deployed the prototype on Emulab to study how well various routing and forwarding techniques work. Here we present an example result that shows the effect periodic routing changes in NetEx have on the performance of bulk TCP flows. We emulated the network topology of Figure 3 using the Emulab [15] testbed. Specifically, we used machines with 850 MHz Intel Pentium III processors having 512 MB of physical memory and 5 10/100 Mbps Ethernet ports. Links have 10 Mpbs capacity and 10 ms delay. The NetEx router is deployed on all machines, and the RCS is deployed on the ingress router.

We initiated a single TCP transfer from the ingress to the egress router. The initial path traverses nodes A, B and C. After five seconds, the flow is rerouted through node D. Figure 4 shows the evolution of the flow's throughput during the first ten seconds. As we can see, changing the routing path causes a sharp drop in throughput, from which the flow recovers completely after approximately 1 second. Given that NetEx routing paths need to be recomputed only once every 30 minutes (see Section 6), such a sporadic drop in throughput is negligible and doesn't affect the performance of long-lived bulk flows.

6.2 Methodology

It is hard to scale our prototype implementation deployed on Emulab to the number of routers and links in a large ISP network. Therefore, we reimplemented NetEx in the ns-2 network simulator and used it for our evaluation.

Even when using ns-2, simulating high-speed multi-gigabit backbone links is computationally very expensive. Therefore, in our simulations, we use a well-known technique [37] to scale down the network capacity, link usage, and the traffic matrices by a factor of at least 1000 [2]. While scaling down the traffic matrices and link usage, we preserve their relative proportions as well as the observed diurnal patterns, thus allowing the results to be scaled back to the original network capacities and traffic demands.

Ideally we would have liked to run actual TCP flows for both best-effort and bulk traffic. However, our nodes ran out of memory while simulating the hundreds of thousands of TCP flows that run simultaneously in large ISPs. To make simulations more efficient, we therefore (a) don't simulate the large number of best-effort network flows; instead, we traffic-shape the bulk flows on a link to the spare bandwidth left unused by the best-effort flows, and (b) don't simulate bulk flows smaller than a certain size (4GB for Abilene and 40 GB for the Tier-1 ISP due to the higher scaling factor used); instead we simulate larger flows that comprise several smaller flows. At the end of the simulation, we divide the throughput attained by the large flows equally among the smaller TCP flows that constitute them. [3]

6.2.1 *Data from real-world ISPs*

The input to our simulations are the network topology and traffic matrix of an ISP. We use data from two large backbone networks: the Abilene research backbone [2] and a large commercial Tier-1 AS offering transit service to access ISPs in multiple continents.

Topologies: Figure5 shows the topology of our Tier-1 backbone. The Tier-1 network offers transit services to more than 40 access ISPs most in Europe and in the Americas, where it also peers with 200 other Tier-1/Tier-2 networks and major distributors of content. The backbone PoPs in the Tier-1 ISP are interconnected using one or multiple 10 or 40 Gbps links. We also used data from the Abilene network (not showed), whose links are OC-192 optical fibers with 10 Gbps of capacity.

Traffic matrices: To derive the traffic matrices for Abilene, we used its NetFlow data to compute the intra-domain traffic entering and leaving at each Abilene router. We applied the simple gravity model [22] to estimate the traffic matrix. Our data comprises all traffic sent in Abilene during the week starting on January 8^{th}, 2007. For the Tier-1 ISP, we also obtained 5-minute load aggregates for the traffic entering and exiting the backbone at each one of its PoPs. Our measurements reflect real loads during February 2009. As before we computed the network's traffic matrix using a gravity model.

6.2.2 *Description of experiments*

[2]The factor is 1000 for Abilene, and 10000 for the Tier-1 network because of its higher capacity

[3]In this case, we assume that bulk TCP flows that are larger than a few MBs but smaller than 4 GB would share bandwidth fairly. We believe that this is a reasonable assumption given our difficulties in simulating such small flows.

For each of the two networks we conducted the following experiment.

Simulating best-effort traffic: We used ns-2 to implement the topology, the routing, and the best-effort traffic as described before. For the Tier-1 AS we used our traffic aggregates to model the best-effort traffic load. Because Abilene is a research network, its real usage levels are very low (around 3%). It is obvious that in such over-provisioned research network there exists plenty of spare capacity that can be used for bulk data transfers. To make our evaluation more realistic, we therefore chose to scale up the load on Abilene links to what one would find in Tier-1 ISPs.

Simulating bulk traffic: We attached to each PoP an additional source generating bulk traffic according to the traffic matrix. The bulk sources are connected with links of infinite capacity, i.e., they can use all the available bandwidth given to them. We produce the trace describing the arrival times for bulk data transfers by simulating a Poisson process with rates varying over time according to the diurnal traffic patterns. We generated flows of different sizes according to a distribution observed in the Abilene's backbone.

Evaluated bulk data workloads: We evaluated the performance of NetEx for two different workloads. Each workload uses different traffic matrices to generate the bulk traffic. Every traffic matrix specifies the traffic demands between each pair of PoPs in the ISP. For each workload, we compute the maximum amount of bulk data that NetEx can deliver while preserving the ratio of traffic demands between the different PoPs. The maximum is computed by scaling up the traffic matrix and repeating the simulations at every step until no more data can be delivered.

1. Native workloads: The first workload directly uses the traffic matrices corresponding to the real traffic demands as measured in the ISP. We believe that this workload is not only the most realistic but also the most challenging. This is because NetEx could easily be used to transfer additional traffic that falls outside the existing traffic matrix, and in this case NetEx could drive the network utilization to 100% by sending bulk traffic over every link whose best-effort load leaves any free capacity. However, such hypothetical bulk traffic matrix is not representative of any real traffic matrix.

2. Intra-datacenter workloads: The second workload aims to evaluate the effectiveness of NetEx in delivering traffic between dataceters. Intra-datacenter traffic accounts for a significant fraction of the Internet traffic generated by cloud computing [11]. For this purpose, we colocated a virtual datacenter with 5 of the 8 PoPs in the European subtopology of the Tier-1 ISP (Figure 5). We chose the European topology because it's where the Tier-1 ISP has better presence, and selected the 5 best connected PoPs in that topology. The traffic matrices for this workload are generated by selecting sender and receiver datacenters and having the senders send as much data as possible to each receiver.

Routing: We evaluated numerous routing algorithms in NetEx. They fall into three broad categories. *1. Static routing:* We simulated static least-weight path routing with different weights for the links: geographical distance (DS), simple hop-count (HC), and the real routing weights used in the studied topologies (WE). *2. Greedy routing:* We simulated a greedy widest path algorithm [40] where each data source selects the path with the most available spare capacity, independent of other sources. The performance of greedy routing reflects the performance of overlay routing schemes where routes are selected without coordination between different flows. *3. Traffic Engineering:* In stark contrast to greedy routing, traffic engineering computes routes taking the global demand of the network into account. NetEx's bandwidth-aware routing, described in Section 4.1, falls into this category. We simulated bandwidth-aware routing using two different traffic engineering objectives; one that optimizes for maximum bulk traffic delivery and a second one that attempts to balance traffic and minimize the peak load across different links.

6.3 Overhead of NetEx

The overhead of NetEx's bandwidth-aware routing algorithm is broken down in 3 components:

Route computation cost at the RCS: In our simulations, computation of routes for the larger Tier-1 topology using a linear solver took on average 0.1 seconds, and never more than 1.14 seconds[4]. This shows that route computation on a well-provisioned RCS should scale well even to larger topologies. For very large topologies where linear solvers may become inefficient, algorithms do exist that approximate a solution efficiently [25], with complexity polylogarithmic in the number of edges and nodes of the topology.

Bandwidth costs: To computer routes, the RCS has to fetch information on load and traffic demands from the routers and distribute the routing information back to the routers. If we encode both link loads and elements of the demand matrix with 16 byte fields, the RCS needs to receive an aggregate of 7.5 Kbytes at every routing update for out Tier-1 topology. The routing information produced by the RCS for the Tier-1 topology was never more than 10 Kbytes, and the information shipped to any single router never more than 1.5 Kbytes. Since routes are required to change only every half an hour or more (see Section 6.6), these values result in very modest bandwidth requirements.

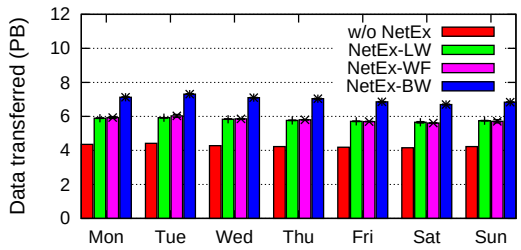
Increase in routing table size: In our large Tier-1 topology, the maximum total size of the NetEx routing tables (see Section 4.2) and the MPLS lookup tables ever dispatched to a single router was 122 entries, corresponding to 1.5 Kbytes.

6.4 How much bulk data can NetEx deliver?

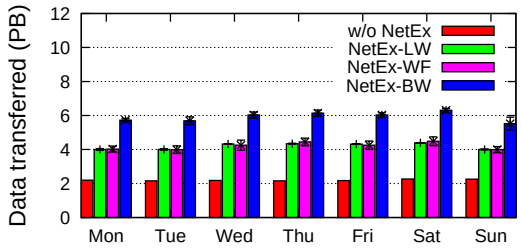
We start by evaluating the aggregate amount of data, both best-effort and bulk, that can be carried using the routing algorithms described in Section 6.2.1. Figure 6 shows the aggregate bulk data delivered by NetEx during each day of the week using the different routing categories in the entire Tier-1 topology and its European subgraph. For each of the three routing categories, static, greedy, and traffic engineering (TE), we only show results for the routing algorithm that performed best in the category.

Additional bulk data transfer capacity: We see that with traffic engineering schemes NetEx can transfer 60 - 170% more data than what is being delivered today without NetEx. In Abilene (not shown in Figure 6) the increase was around 120%. The amount of extra bulk data that can be delivered is almost 3 PBytes per day in the case of the Tier-1 AS. To put things in perspective, such a volume is almost 100 times greater than the raw amount of data produced each day by the Large Hadron Collider of CERN, one of the

[4]We used CPLEX [13] running on a 2.5Ghz Linux AMD processor



(a) Tier-1 ISP



(b) Tier-1 ISP (Europe)

Figure 6: Additional data transferred by NetEx: NetEx bandwidth-aware traffic engineering (NetEx-TE) attains the best utilization of spare capacity, increasing the ISP's throughput by up to 170%

biggest individual producers of data in the world (27 Tbytes per day [29]).

Impact of ISP topologies: We find that NetEx gains are higher within continental backbones (e.g., Abilene and European sub-graph of Tier-1 ISP) compare to inter-continental backbones. This is because PoPs within continental backbones are more densely connected, thus offering more alternate routes to exploit for traffic engineering. We also ran additional experiments in two other Tier-1 continental subgraphs (North and South America). We found even larger benefits in South American backbone (+300% more data). In the North American backbone, however, NetEx wasn't able to increase the network traffic. This is because of the low capacity of links connecting a PoP in that subgraph to the rest of the topology.

Bandwidth-aware traffic engineering versus static routing: The difference between the performance of NetEx-TE and static routing shows the extent to which traditional inter-domain routing algorithms limit the efficient use of spare bandwidth. In Abilene, NetEx-TE delivers 30% more bulk data than static routing. In the Tier-1 ISP, NetEx-TE delivers at least 1 Pbyte of additional bulk data every day.

Traffic engineering versus greedy routing: The difference between NetEx-TE and NetEx-greedy shows the efficiency loss that NetEx incurs when each bulk flow is routed greedily without any coordination with other flows. On average, greedy routing delivers 20% less data than traffic engineering. This result hints at the potential limitations of overlay routing schemes at the application-layer where each flow tries to find paths that optimize its throughput independently of other flows.

Comparing different traffic engineering objectives: As said before, we evaluated two different traffic engineering objectives, one maximizing data delivery and another balancing load across the different links. We found that both traffic engineering objectives perform very similarly (not shown in the Figures above) and deliver similar amounts of data. As we show later in Section 6.6, both schemes achieve near-optimal performance by saturating the links that form a min-cut within the network.

Traffic matrix	Daily data sent Min.	Avg.
Single source	427TB	484TB
Full mesh	106TB	108TB

Table 1: Performance of NetEx with intra-datacenter workload: Amount of daily data delivered when only one datacenter acts as data source (single source) and when all datacenters act as data sources (full mesh). Minimum and average are computed across all datacenters and simulated days

In summary, bandwidth-aware traffic engineering, NetEx-TE, performs considerably better than all other routing schemes. Since the overhead of implementing bandwidth-aware routing is relatively small (see Section 6.3), NetEx-TE stands out as the most favorable routing scheme. Next, we evaluate how NetEx-TE increases the performance of inter-datacenters bulk transfers.

6.4.1 Evaluation with intra-datacenter workloads

So far we evaluated NetEx using as input the native traffic demands of the ISP, which reflect more closely the traffic between cloud users and datacenters. We now evaluate how much data NetEx can deliver in the case of the inter-datacenter workload described in Section 6.2.2.

Our evaluation uses two traffic matrices where: (a) a single one datacenter sends data to each of the remaining datacenters (single source) , and (b) every data center concurrently sends data to all other datacenters (full mesh). In both cases, we assume that each datacenter has the same amount of data to send and we measure the maximum amount of data that can be delivered on each day. To efficiently distribute the load, we assume that all datacenters form a swarm and cooperate in distributing the data, with data sources acting as seeders.

The results are summarized in Table 1. In the full mesh scenario, every data center can deliver at least 100 TBytes of daily data to all other datacenters. In the case where there is only a single datacenter acting as sender, this number increases to nearly 430 TBytes a day. To put this into perspective, this is more than 100 times the data generated by Facebook picture uploads every day [17]. These results suggest that NetEx has a lot of potential for serving the increasing traffic generated by applications hosted in datacenters.

6.5 How well do NetEx flows perform?

The previous section showed that NetEx can indeed deliver substantial amounts of additional bulk data. A natural next question is how well individual NetEx flows perform and whether their performance is acceptable to different types of delay-tolerant applications.

To quantify the performance that different applications achieve when their traffic is routed using NetEx, we generated a trace of flows of different sizes, corresponding to a set of popular bulk transfer sizes, as illustrated in Table 2.

Completion times of bulk flows: Figure 7 shows the time these flows take to complete in the Tier-1 topology when routed through NetEx. For delay-tolerant applications like

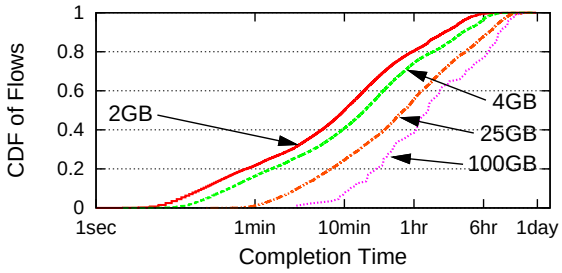


Figure 7: Completion time of flows of different sizes routed through NetEx: Most flows achieve good completion times

Size	Example
2GB	iTunes VoD
4GB	DVD
25GB	Blu-ray disc
100GB	Data backup

Table 2: Application examples used in our analysis: Each flow size corresponds to popular bulk data application

online backups and DVD downloads, NetEx provides good completion times and performance: a 100GB backup rarely takes longer than 1 day, nearly 80% of 4GB DVD movie downloads take less than 1 hour, and the median download time for a large 25GB Blu-ray disc is 1 hour.

Our results show that NetEx flows achieve good performance despite being routed using spare bandwidth. However, because it cannot offer any bandwidth guarantees over short time scales, NetEx is not suitable for real-time or highly interactive applications like video conferencing, online games or VoIP. For example, we found that (results not shown) 30% of the time NetEx is not able to sustain a 200 Kbps data rate (low definition YouTube video) without buffering for several seconds.

6.6 How close to optimal is NetEx?

In the previous sections we have shown that NetEx can transfer substantially more bulk data than what is currently being transferred in the network and that the data transfers achieve good performance. In this section we show that NetEx is actually very close to optimal in terms of the maximum volume that can be transferred in the examined topologies under the given traffic matrices. We will establish this optimality by showing that NetEx almost saturates a cut in each network (a cut is set of links that partitions the graph).

Figure 5 shows the topology of the Tier-1 backbone. For the Tier-1 ISP, NetEx-TE saturates the cut comprising the transatlantic links (L1,R1), (L2,R3), and (L3,R3). When using NetEx-TE, the utilizations of the links in the cut are, 97%, 99%, and 99%, respectively. For Abilene (not shown), the cut has two links, both of which are driven to 99% utilization by NetEx-TE. Any routing algorithm attempting to deliver more data under the given traffic matrices will be bounded by the capacity of these minimum cuts. This also explains why both our traffic engineering algorithms optimizing for different objectives achieve the same performance. Both saturate the min cut and are ultimately limited by the cut's capacity.

Impact on average link utilization: Notice that saturating the cut links does not imply that all other links of the network are fully utilized. In Figure 8 we plot for each day of a

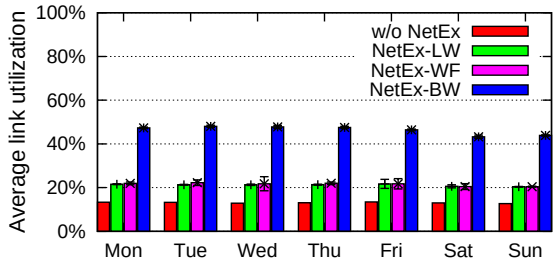


Figure 8: Average link utilization with NetEx: Results from Tier-1 ISP show that NetEx can increase the utilization of the ISP's links by a factor of 2 or more

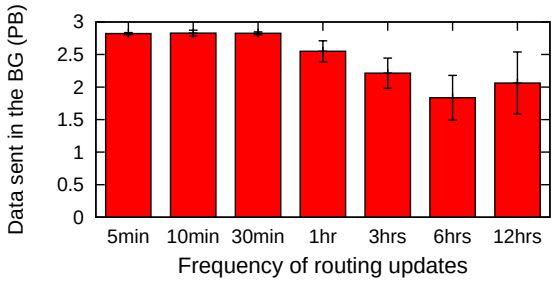


Figure 9: Amount of additional bulk data delivered when routes are recomputed with different frequencies: Results from Tier-1 ISP show that increasing the routing interval to up to 30 minutes does not cause a noticeable decrease in performance

week the average link utilization in our Tier-1 ISP. Even as NetEx-TE achieves optimal performance, the average utilization across all links is around 30%-40%. Note, however, that this represents a two to three fold increase with respect to the average link utilization levels before NetEx. We observed a similar increase in link utilization for the Abilene network. We believe that the increase in link utilization is sufficiently high to incentivize ISPs to deploy NetEx.

NetEx's bandwidth-aware traffic engineering achieves optimal performance by periodically recomputing routes and leveraging multiple paths between a source and a destination. In the rest of the section, we explore the role of these factors in more detail.

The role of dynamic route reconfiguration: NetEx-TE benefits from recomputing its paths periodically. In Figure 9 we plot the amount of additional bulk data transferred by NetEx-TE under different frequencies of route recomputation over the Tier-1 ISP. In general, NetEx benefits from recomputing paths periodically during the course of the day. In the Tier-1 AS, recomputations need to be performed at least once every 30 minutes. Less frequent recomputations lead to noticeable reductions in the transferred volume. However, the frequency with which the routes must be computed varies from one ISP to another. In Abilene (data not shown in the figure), we found that it is sufficient to compute routes every 3 hours; more frequent recomputations do not improve performance.

The role of multi-path routing: NetEx-TE leverages spare capacity along multiple paths between a source and a destination to deliver bulk data. We counted the number of total paths used over a day to route traffic between every pair of source-destination PoPs in both the Tier-1 ISP and Abilene. In Abilene, all source-destination pairs were using at most

6 paths. In the Tier-1 ISP, 75% of source-destination pairs were using 20 paths or less, but a few pairs were using as much as 40 paths.

These different paths can indeed vary considerably in terms of delay. To show this, we computed the ratio of the latency of the longest to the shortest path selected by NetEx during an entire day (latency stretch). The latency stretch can be as high as 10 for some source-destination pairs in our Tier-1 and Abilene networks. Only latency-insensitive bulk applications can afford such a high variability in end-to-end latency.

In summary, NetEx-TE routing selects multiple paths with variable latencies between pairs of PoPs, which suggests that NetEx-TE routing is not suitablefor interactive applications that require stable and predictable QoS.

7. RELATED WORK

On bulk transfer solutions at the application or transport layer:Previous attempts to exploit spare bandwidth include new transport protocols [47] and application-level solutions like overlays [10, 27, 50] or P2P file distribution [8]. Because all these approaches don't rely on any support from the network, they necessarily cannot use spare bandwidth as efficiently as network-level approaches like NetEx. In fact, recent proposals such as P4P [49] have attempted to provide network-level support for P2P protocols in order to increase their efficiency.

On scheduling of delay tolerant traffic: Previous work has attempted to reduce the cost of delay-tolerant bulk transfers by delaying them to a later time. Recently, intelligent application-level scheduling [29] or network-level traffic shaping [31] of bulk transfers have been proposed as a way for access ISPs to reduce their bandwidth costs under 95^{th}-percentile billing. These techniques are orthogonal to NetEx because they operate at the network edge and ignore spare capacity present in Tier-1 backbones, which is the focus of NetEx. In particular, they don't address the problem of routing traffic in the backbone to maximize the usage of spare capacity. Previously, delay tolerant networks (DTNs) [18, 48] based on store-and-forward have been proposed as a way to route delay-tolerant (but not necessarily bulk) traffic in intermittently-connected networks like satellite links. Although it also exploits the delay-tolerant nature of some traffic, NetEx targets network with continuous end-to-end connectivity with the goal of increasing their utilization.

On traffic engineering and multi-path routing in transit ISPs: There is a large body of work on traffic engineering to satisfy QoS guarantees by minimizing the amount of resource consumed in the network. This can involve multi-commodity flow problems [19, 21], using predictions on future network load [6] or combining flow commodity models with recent past network history [44]. As we showed before, NetEx uses existing traffic engineering techniques as part of its bandwidth-aware routing algorithm to make optimal use of spare capacity. This is possible because NetEx only operates on latency-insensitive bulk traffic and can thus safely ignore latency to aggressively use the spare capacity in the network. On the other hand, when traffic engineering operates on best-effort traffic, care must be taken to avoid long routes that increase latency and negatively impact QoS [21]. This ultimately limits the efficiency with which traditional traffic engineering without traffic differentiation can use spare bandwidth in the backbone. Similar observations can be made for multi-path routing algorithms like equal-cost multi-path [30, 33] that spread load across paths of equal cost and in general prefer shortest paths over longer ones. To the best of our knowledge, no network routing algorithm proposed for the Internet selects routes based solely on available bandwidth, as NetEx does. NetEx can ignore path length because it only deals with delay-tolerant bulk traffic.

On differentiated services in backbones: The Scavenger service of QBone [41] has been an early attempt to tap on these resources for the benefit of low priority bulk transfers. Unlike NetEx, Scavenger performs only traffic differentiation but no bandwidth-aware routing. Therefore, it is limited to using only the spare capacity available on the shortest path between each bulk data sender and receiver. On the other extreme, Shaikh and Rexford [40] have proposed using bandwidth-aware routing for long-lived flows but do not consider traffic differentiation or any coordinated optimization between concurrent bulk flows. One of the main stands of NetEx is that both traffic differentiation and coordinated bandwidth-aware routing are necessary to take full advantage of existing unutilized bandwidth without negatively impacting interactive traffic.

On QoS and differentiated services: Traffic differentiation has been largely studied in the context of quality of service (QoS). Diffserv [9] and the twobit [35] are examples of architectures that provide differentiated traffic services. Unlike most previous work, NetEx uses traffic differentiation to provide class of service that is lower than (as opposed to higher than) best-effort.

On multi-homing and capacity leasing to reduce bandwidth costs: Multi-homing and capacity leasing [16] are connectivity solutions that help edge networks dynamically select transit ISPs in order to reduce network costs. Although these solutions exploit differences in market prices among ISPs, they don't reduce the fundamental cost of bulk transfers and are therefore orthogonal to NetEx. In fact, if transit ISPs were to provide a low-cost bulk service, edge networks could use existing multi-homing and capacity leasing solutions to select the most convenient bulk transit rate at any time.

8. CONCLUSIONS

Today, cloud providers pay tier-1 ISPs high prices for the bandwidth generated by their cloud services. As a result, cloud providers are forced to charge their users significant bandwidth prices, thus creating an economic obstacle to the adoption of their services. To reduce bandwidth costs, we argue that Tier-1 ISPs have an incentive to exploit their spare network capacity and use it to offer a cheap and efficient bulk data service. By using this service, cloud providers could charge their customers lower prices for bandwidth, thus increasing the demand for cloud services.

In this paper, we propose NetEx, a system that exploits the abundant spare bandwidth resources available in ISPs' networks to deliver delay tolerant bulk transfers for cloud computing. To take full advantage of the spare resources, NetEx uses traffic differentiation and bandwidth-aware routing. Traffic differentiation allows NetEx to opportunistically use spare bandwidth resources while not interfering with existing traffic. Bandwidth-aware routing allows NetEx to dynamically adapt to changes in available link bandwidth

by routing around hot spots in the network. We evaluated NetEx using data from a commercial Tier-1 provider and the Abilene research backbone network. Our evaluation shows that NetEx achieves near optimal utilization of spare resources and that the bulk content delivered by NetEx can increase today's traffic by a factor of two or more.

9. REFERENCES

[1] Internet protocol, 1981.

[2] Abilene Backbone Network. http://abilene.internet2.edu/.

[3] Amazon Import/Export. http://aws.amazon.com/importexport.

[4] Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/pricing/.

[5] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, UCB, 2009.

[6] A. Bestavros and I. M. B. University. Load profiling for efficient route selection in multi-class networks. In *Proc. of IEEE ICNP*, 1997.

[7] S. Bhattacharrya, C. Diot, J. Jetcheva, and N. Taft. Geographical and Temporal Characteristics of Inter-POP Flows: View from a Single POP. *European Transactions on Telecommunications*, February 2002.

[8] BitTorrent homepage. www.bittorrent.org.

[9] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Service. RFC 2475 (Informational), Dec. 1998. Updated by RFC 3260.

[10] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. SplitStream: high-bandwidth multicast in cooperative environments. In *Proc. of SOSP '03*.

[11] Y. Chen, S. Jain, V. K. Adhikari, Z.-L. Zhang, and K. Xu. A first look at inter-data center traffic characteristics through yahoo! dataset. In *Proc. of Infocom*, 2011.

[12] Cisco Systems. Cisco QoS. http://www.cisco.com/en/US/products/ps6558/products_ios_technology_home.html.

[13] ILOG CPLEX. http://www.ilog.com.

[14] M. Dobrescu, N. Egi, K. Argyraki, B.-G. Chun, K. Fall, G. Iannaccone, A. Knies, M. Manesh, and S. Ratnasamy. Routebricks: exploiting parallelism to scale software routers. In *Proc. of the SOSP*, 2009.

[15] Emulab - network emulation testbed. http://www.emulab.net.

[16] Equinix Direct. http://www.equinix.com/download.php?file=Equinix_Direct.pdf.

[17] Needle in a Haystack: Efficient Storage of Billions of Photos, 2009. Facebook Engineering Notes, http://tinyurl.com/cju2og.

[18] K. Fall. A Delay Tolerant Networking Architecture for Challenged Internets. In *Proc. of SIGCOMM*, 2003.

[19] B. Fortz and M. Thorup. Internet traffic engineering by optimizing ospf weights. In *Proc. of INFOCOM*, 2000.

[20] Google helps terabyte data swaps, Mar 2007. http://news.bbc.co.uk/2/hi/technology/6425975.stm.

[21] E. Gourdin and O. Klopfenstein. Comparison of different qos-oriented objectives for multicommodity flow routing optimization. In *Proc. of the International Conference on Telecommunications*, 2006.

[22] A. Gunnar, M. Johansson, and T. Telkamp. Traffic Matrix Estimation on a Large IP Backbone - A Comparison on Real Data. In *Proc. of IMC*, 2004.

[23] J. He and J. Rexford. Towards Internet-wide multipath routing. *IEEE Network Magazine*, 2008.

[24] S. Iyer, S. Bhattacharrya, N. Taft, and C. Diot. An Approach to Alleviate Link Overload as Observed on an IP Backbone. In *Proc. of Infocom*, San Francisco, March 2003.

[25] G. Karakostas. Faster Approximation Schemes for Fractional Multicommodity Flow Problems. In *Proc. of ACM/SIAM SODA 2002*.

[26] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The click modular router. *ACM Trans. Comput. Syst.*, 18(3):263–297, 2000.

[27] D. Kostic, A. Rodriguez, J. Albrecht, and A. Vahdat. Bullet: high bandwidth data dissemination using an overlay mesh. In *Proc. of SOSP '03*.

[28] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. 2004. Proc. of SIGMETRICS.

[29] N. Laoutaris, G. Smaragdakis, P. Rodriguez, and R. Sundaram. Delay tolerant bulk data transfers on the Internet. Proc. of SIGMETRICS'09.

[30] G. M. Lee and J. S. Choi. A survey of multipath routing for traffic engineering. Technical report, Information and Communications University, 2002.

[31] M. Marcon, M. Dischinger, K. P. Gummadi, and A. Vahdat. The Local and Global Effects of Traffic Shaping in the Internet. In *Proc. of COMSNETS*, 2011.

[32] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 2008.

[33] J. Moy. Ospf version 2, 1998.

[34] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers , 1998.

[35] K. Nichols, V. Jacobson, and L. Zhang. A Twobit Differentiated Services Architecture for the Internet, Dec 1997. Internet draft.

[36] A. Nucci, N. Taft, C. Barakat, and P. Thiran. Controlled Use of Excess Backbone Bandwidth for Providing New Services in IP-over-WDM Networks. *IEEE Journal on Selected Areas in Communications - Optical Communications and Networking series*, November 2004.

[37] R. Pan, B. Prabhakar, K. Psounis, and D. Wischik. SHRiNK: a method for enabling scaleable performance prediction and efficient network simulation. *IEEE Transactions on Networking*, 13, October 2005.

[38] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture, January 2001.

[39] F. Shahrokhi and D. W. Matula. The maximum concurrent flow problem. *Journal of the ACM*, 1990.

[40] A. Shaikh, J. Rexford, and K. G. Shin. Load-sensitive routing of long-lived ip flows. In *Proc. of SIGCOMM*, 1999.

[41] S. Shalunov and B. Teitelbaum. Qbone Scavenger Service (QBSS) Definition. Internet2 technical report, Mar. 2001.

[42] Sprint AR&ATL. http://research.sprintlabs.com.

[43] Sprint Internet Access SLA. http://www.sprint.com/business/resources/dedicated_internet_access.pdf.

[44] S. Suri, M. Waldvogel, D. Bauer, , and P. R. Warkhede. Profile-based routing and traffic engineering. *Computer Communications*, Mar. 2003.

[45] R. Teixeira, K. Marzullo, S. Savage, and G. Voelker. Characterizing and Measuring Path Diversity in Internet Topologies. In *Proc. of SIGMETRICS*, June 2003.

[46] T. Telkamp. Traffic Characteristics and Network Planning, 2002. www.caida.org/workshops/isma/0210/talks/thomas.pdf.

[47] A. Venkataramani, R. Kokku, and M. Dahlin. TCP-Nice: A Mechanism for Background Transfers. In *Proc. of OSDI02*, December 2002.

[48] R. Y. Wang, S. Sobti, N. Garg, E. Ziskind, J. Lai, and A. Krishnamurthy. Turning the Postal System into a Generic Digital Communication Mechanism. In *Proc. of SIGCOMM*, 2004.

[49] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz. P4p: provider portal for applications. In *Proc. of SIGCOMM '08*.

[50] Y. Zhu, C. Dovrolis, and M. Ammar. Dynamic overlay routing based on available bandwidth estimation. *Computer Networks Journal*, 2006.

APPENDIX

A. BANDWIDTH-AWARE ROUTING ALGORITHM

We describe the algorithm that is used to compute the routes that maximize the amount of data delivered. The problem is characterized as a multi-commodity flow problem and is solved using a linear solver.

In a network with k PoP pairs and n links, there will be k commodities $C = c_1, \ldots c_k$ to maximize and k demands $D = d_1, \ldots d_k$ corresponding to the bulk transfer demands. For each network link l_i in $L = l_1, \ldots l_n$, the associated spare capacity $sc(l_i)$ is the number of bytes that can be transferred using the free capacity on link l_i. Since the distribution of the demands is not necessarily uniform, we want the resulting solution to preserve the relative proportions in the traffic demands.

The solution to the problem will have to satisfy the following constraints:

1. Capacity constraints: the total bulk data routed through each link cannot exceed the available background capacity. Therefore:

$$\forall i \in [1, n] : \sum_{j \in [1,k]} f(c_j, l_i) \leq sc(l_i) \tag{A.1}$$

where $f(c_j, l_i)$ is the amount of commodity c_j routed through link l_i.

2. Flow conservation constraints: bulk traffic can only be generated at the sources and consumed at the destinations. These constraints are identical to those found in ordinary network flow problems.

3. Traffic matrix conservation constraints: commodities must be maximized while preserving the relative proportions between the original demands. We therefore identify the maximum demand $d_{max} = max(d_1, \ldots d_k)$, and add the following constraints:

$$\forall i \in [1, k] : c_i \geq \frac{d_i}{d_{max}} maxc \tag{A.2}$$

Where $maxc$ is a new variable identifying the maximum commodity. This makes sure that each commodity c_i will get a fraction of the available bandwidth *at least* as high as the ratio between the corresponding demand d_i and the maximum demand d_{max}.

Then the solver maximizes the maximum commodity $maxc$. Because of the constraints, all commodities will be maximized while preserving the rations between the original demands. The output of the solution is mapped into a *probabilistic routing table* for each router in the network.