Towards an Internet Connectivity Market

Vytautas Valancius*, Srikanth Sundaresan*, Umayr Hassan*, Nick Feamster*,
Ramesh Johari†, Vijay Vazirani*
*Georgia Tech †Stanford University

ABSTRACT

Today's Internet achieves end-to-end connectivity through bilateral contracts between neighboring networks; unfortunately, this "one size fits all" connectivity results in less efficient paths, unsold capacity *and* unmet demand, and sometimes catastrophic market failures that result in global disconnectivity. This paper presents the design and evaluation of *MINT*, a Market for Internet Transit. MINT is a connectivity market and corresponding set of protocols that allows ISPs to offer path segments on an open market. Edge networks bid for end-to-end paths, and a mediator matches bids for paths to collections of path segments that form end-to-end paths. MINT can be deployed using protocols that are present in today's routers, and it operates in parallel with the existing routing infrastructure and connectivity market. We present MINT's market model and protocol design; evaluate how MINT improves efficiency, the utility of edge networks, and the profits of transit networks; and how MINT can operate at Internet scale.

1. Introduction

The Internet's growth and the demands of a wide variety of applications and traffic types has created a need for the network to provide different levels of connectivity—and different guarantees—to different pairs of sources and destinations. Once an experimental research network, the Internet now hosts applications ranging from video conferencing to bulk file transfer. IP has provided a ripe substrate for building many applications; unfortunately, today's routing infrastructure is often inadequate for supporting this diverse set of applications. With so many requirements, it is increasingly difficult for a single routing infrastructure, federated across tens of thousands of independently operated networks, to provide efficient, reliable connectivity to every application.

Today's Internet routing protocol, Border Gateway Protocol [1], allows independently operated networks to exchange traffic and express preferences according to certain types of business relationships. Unfortunately, edge networks are typically constrained to "one size fits all" contracts, where an edge network pays a single rate for connectivity to all destinations, and relies on the existence of pairwise contracts between transit service providers to ensure the existence of reliable and efficient paths. This market structure is inherently inefficient because edge networks cannot pay for end-to-end paths in accordance with how much they value them. The failings of the current market structure is most evident in "depeering" incidents, where edge networks become partitioned from one another when bilateral contracts in the middle of the network dissolve [2–4]. However, bilateral contracts create inefficiencies even in normal operation, because the existence and nature of connectivity between two ASes depends on myopic business decisions between those two ASes. The value of a path is only indirectly transferred from the edge networks to the transit ASes in the middle of the network. Bilateral agreements make the Internet less reliable too: many paths that exists in the underlying graph would never get exposed, even in extreme failure scenarios.

In this paper, we propose a *Market for Internet Transit (MINT)*. MINT is a connectivity market and set of supporting protocols that moves away from proprietary, bilateral contracting towards an open system to enable the "invisible hand" of competition. MINT allows networks to buy end-to-end *paths* from transit ISPs, which advertise *path segments* between intermediate exchange points at various prices to a *mediator*, who matches bids for end-to-end paths to collections of offers for path segments. MINT operates in parallel to the existing BGP routing infrastructure—as such, it is not intended to supplant BGP routing, but rather to provide an alternative means for ISPs to sell spare capacity and edge networks to buy connectivity to specific destinations. We show that MINT improves the efficiency of resource use and increases both utility of edge networks and profits for ISP networks, and also that MINT can operate at Internet-scale.

The design of MINT poses significant challenges. MINT must provide a mechanism for matching collections of path segments that are offered by ISPs to requests for end-to-end paths from buyers (*i.e.*, edge networks). To do so, MINT's protocols implement a first-price path auction. First, the mechanism must provide *incentives* for each party to participate in MINT. Specifically, we must show that ISPs will garner more profit by participating in MINT than they would if they just routed all of their traffic over BGP alone. Second, MINT must scale well with the growth of the network, with the rate of requests for paths, and with the overall link failure rate. In addition to the basic economic efficiency and scalability concerns, MINT poses a number of secondary challenges as well. MINT's design must preserve the privacy of transit networks, while still providing a market over which these providers can sell their spare capacity. MINT must also potentially reconcile disparity of contracts—because edge networks buy end-to-end paths, the quality and guarantees associated with each advertised path segment must be standardized.

This paper makes the following contributions. First, we present MINT, a new market for buying and selling Internet transit. We design a market mechanism to allow edge net-

works to purchase end-to-end paths from collections of segments sold by transit ASes. This market allows for a more *direct transfer of value* from the edge networks that need connectivity to the transit networks, who need to be paid for providing capacity. Second, we present a detailed protocol design for MINT that is both scalable and compatible with protocols already implemented on today's routers. Third, we present a detailed two-part evaluation: (1) we show that MINT provides significant gains in surplus to both edge networks and transit ASes over today's BGP-based market; (2) we show that MINT can operate at Internet scale.

The rest of the paper is organized as follows. Section 2 presents design goals and a high-level overview of MINT. Section 3 describe the protocols that implement the market. In Section 4, we show that MINT provides significant surplus to both users and transit ISPs over BGP's bilateral contracts; Section 5 argues that MINT can operate at Internet scale, even as the Internet continues to grow. Section 6 presents related work, and Section 7 concludes.

2. Overview

In this section we outline the goals for our market design and provide a high-level overview on how we achieve these goals.

2.1 Design Goals

MINT's market should be open, profitable for all entities, and its implementation should be easy to manage. We first discuss the design goals for the market; then, we outline the design goals for supporting protocols.

2.1.1 Market

Open. The primary objective in our design is to ensure that the market for end-to-end paths is open. With BGP today, connectivity depends on bilateral contracts between upstream providers. In contrast, MINT provides a market where edge networks can buy paths if their valuation for those paths is greater than the total sum cost of the segments. Such a market structure would not only allow for prices to more directly reflect the value that edge networks have for each path, but it would also improve efficiency, since excess capacity could be sold on an open market to buyers; this model contrasts with today's structure, where an AS cannot sell excess capacity unless one of its immediate neighbors wants to buy it.

Profitable. To be sustainable, MINT must provide incentives for each party to participate. In other words, both the buyers and the sellers (defined in the next subsection) must have an incentive to participate: the costs of supporting MINT, as well as selling excess capacity at a potentially lower price than otherwise available in a less open market, must both be countered by the additional profits that an AS could garner by running MINT. Similarly, MINT must provide significant benefit to the edge networks over simply using today's BGP protocol and selecting the best available upstream provider.

Manageable. MINT should also be easily manageable: Network operators should retain control over how traffic traverses their networks, and it should be easy for operators to set and adjust prices for various segments. When a path segment is sold, MINT should provide mechanisms that make it easy for an operator to manage the establishment of new paths, as well as the connection of the path segment in the local network to the adjoining path segments along the end-to-end path.

2.1.2 Protocols

To implement market with the properties describe above, we must employ a set of protocols. Such protocols, if deployed in the Internet scale pose many design challenges. Here we introduce the main design goals for the protocol infrastructure supporting MINT system.

Scalability. First and foremost, MINT's control plane must support a potentially large number of buyers and sellers. The number of buyers grows with the number of edge networks that wish to buy and sell paths and with the total number of destinations (*i.e.*, IP prefixes). The number of sellers grows with the number of *segments*, which is essentially the number of paths between any pair of exchanges. This number is difficult to estimate in practice, but we present methods for estimating the total number of segments in Section 4; and likely grow trends in Section 5. In Section 5, we will also evaluate how MINT's control and data planes scale as the number of segments grows, and as the number of overall destinations grows.

Backwards compatibility. MINT must run on today's Internet infrastructure. In addition to implementing MINT with existing protocols, we must also ensure that the hardware that forwards best-effort traffic, could accommodate MINT. Additionally, to the extent possible, we aim design MINT's data and control planes using protocols that are already standardized and implemented on today's routers. Our contribution is thus not any single protocol, but the design of the overall framework that allows existing protocols to be used to implement MINT.

Containment. MINT should allow the network providers to advertise their services, while withholding business-sensitive details about the network that the provider does not wish to make public (*e.g.*, the network topology). Additionally, when local networks experience failures that do not affect the connectivity of the advertised path possible (note that this property differs markedly from BGP-based interdomain routing, where local changes can propagate globally).

Speed. The mechanism for establishing and tearing down paths must be lightweight enough to allow services to be efficiently sold. The mechanism for establishing the paths themselves should not significantly degrade the overall packet forwarding rate—routers should forward packets along MINT paths at line rate.

Accountability. MINT must provide enough accountability for contract enforcement for proper market operation. Existing mechanisms, such as clearing houses, reputation systems or accounting monitors [5] or other traffic auditing mechanisms [6, 7] might help MINT provide better accountability, thus it is not our focus in this paper. We leave the details of

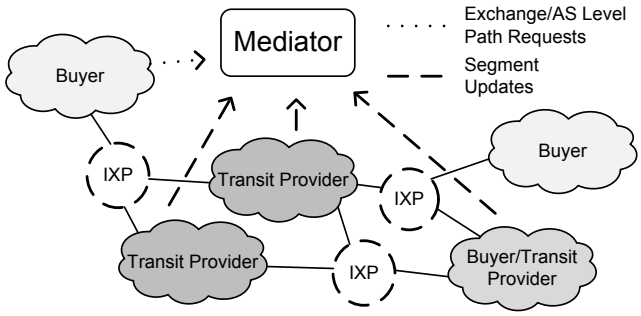


Figure 1: High-level MINT architecture. IXP - exchange point.

implementing accountability mechanisms for MINT to future work.

2.2 MINT Overview

In this section, we present an overview of MINT's market model. We briefly describe the participants, the auction-based structure of the market, and the mechanism that the mediator uses to construct end-to-end paths from segments.

MINT implements a continuous dynamic version of a *first-price path auction* [8]: Transit providers sell links, or path segments, at some price, and buyers pay for end-to-end paths. The amount that the buyer pays is equal to the sum of the advertised prices for each link along that path.

MINT has three classes of participants: buyers, sellers, and a mediator. *Buyers* may be either large Internet service providers (ISPs) or edge networks. *Sellers* are typically large transit ISPs, although they may also be edge networks that have more than one interdomain connection and spare capacity. The marketplace is implemented by a centralized *mediator*, which aggregates the information about the product to be traded (available path segments), and matches demand for end-to-end paths to supply of path segments. Figure 1 illustrates this high-level operation.

Basic operation Sellers advertise *path segments* to the mediator with multiple attributes, including available bandwidth and price. We call these advertisements *transit state updates*. A path segment is essentially a connection between two exchange points (or, more generally, a connection between an ingress and egress of a single ISP). For each path segment, a transit network uses distributed path computation to compute and establish *node-level* paths between the network's ingress and egress points. Updates about path segments only contain aggregate information; they do not reveal any information about the internal functioning of the network, thus preserving the privacy of the ISP's topology.

When a customer requires a path to another network, it issues a request with the path's endpoints (*i.e.*, the source and destination exchanges), as well as any constraints the path must meet, such as minimum bandwidth, maximum delay, and a list of networks to avoid. Each end-to-end path purchased by a buyer would typically comprise multiple segments. Sellers and buyers use the mediator to enable the exchange of goods by matching demand (*i.e.*, end-to-end paths) to supply (*i.e.*, collections of path segments). Note that a network can act as a buyer for some requests (if it acts as a request source) and as a seller for other requests (if it provides transit).

For each request, the mediator finds the lowest cost *exchange-level* path that satisfies the requested constraints. If such a path is found, the mediator returns this path (and the associated resources) to the buyer. For the purposes of this paper, we describe MINT as a mechanism for selling paths with desired capacities. In this case, the mediator can construct a path from segments greedily, by running a capacity constrained shortest paths algorithm using only paths that are lower than the price that the buyers are willing to pay. More generally, however, MINT could sell paths with other performance objectives as well (*e.g.*, loss or latency targets). If the mediator cannot find a collection of path segments that satisfies the buyer's bid, the mediator simply drops the bid. We assume that if a bid is not satisfied, the buyer will submit a re-valued bid.

Price adjustment and information In an ideal, competitive open market, sellers of goods can determine the equilibrium price for a good by adjusting price according to the user's demand. For goods that are perfect substitutes, and markets that are competitive, prices will converge to marginal cost. However, the MINT market is complicated by the fact that (1) sellers (*i.e.*, ISPs) do not know whether other ISPs are selling complementary goods (*i.e.*, path segments), and (2) they also do not know the prices at which these goods are being sold to other users. This makes it more difficult for a seller to determine exactly how it should set prices on any given segment. We assume that the seller uses a simple price adjustment model, whereby the seller sets a price for each segment, observes the utilization on the link after a fixed period of time, and raises the price on the segment if the utilization exceeds some threshold (and vice versa, lowers the price if utilization drops below a certain threshold).

Truthfulness in advertisements and bidding We note that market participants may have incentives to distort their characteristics to the mediator, leading to inefficient outcomes. Truthfulness in path auctions is a general concern: we aim to design auctions where buyers will bid their valuations and sellers do not overcharge. Related work studied VCG-based auctions, for example, and found that overpayment can be very high in these settings; other work suggests that first-price path auctions are not necessarily subject to the same overpayment [8].

It is worth noting that in MINT, there are legitimate reasons for sellers to avoid truthfully revealing their capacity when they advertise path segments. The connectivity information that an ISP announces may not necessarily reflect the exact state of the network but rather the state that the ISP is willing to disclose. For example, it might "under-promise and over-deliver", announcing less capacity than they might actually have in total. In Section 5, we examine the extent to which under-promising capacity in path segments can improve the overall scalability of the system. On the other hand, an ISP might also overbook its available capacity, selling more capacity than it is capable of carrying on various links.

3. MINT Protocols and Components

This section describe the protocols that we use to implement MINT. We first discuss the high-level requirements for MINT's protocols; we then describe MINT's control plane, which is the mechanism that implements the auction itself. Finally, we present MINT's data plane, which allows data packets to be forwarded along MINT paths.

MINT protocol design goals are outlined in Section 2. In this Section, show how MINT could be implemented in practice and deployed on the existing Internet, in parallel to the existing BGP-based infrastructure. Aside from the general requirements, MINT has a set of functional requirements. At a high level, ISPs need a mechanism to announce segment capacities and prices to a contract mediator, and edge networks that wish to obtain end-to-end paths must have the interface to request them (as shown in Figure 1). Finally, once the mediator sells a path, the ASes along the path must establish forwarding state for the path itself.

3.1 Control Plane

MINT control plane runs over a set of protocols between ISPs, mediators and clients. Each AS issues *transit state announcements*, which contain the price and parameters (*e.g.*, capacity, price) for each segment. A segment is defined by a pair of exchanges (with unique exchange IDs) that ISP connects. Each AS freely chooses which segments to announce to the mediator, as well as the parameters of each. Buyers issue bids for paths between a connecting exchange and another exchange. The bid contains the source and destination exchanges, path constraints, and maximum bid price. The mediator collects announcements from transit ISPs and matches them with bids. All control-plane mechanisms only require middle-box setup at sellers and buyers, without fundamental changes to current control protocols.

3.1.1 Sellers: Transit ASes

Monitoring. ASes that provide transit services employ *monitors* as an add-on network element that observes the internal network state between exchanges, accepts operator policies and reports transit-state updates between those exchanges to the mediator. These monitors can be implemented using standard link-state monitors [9] to collect network state information. Routing protocols such as OSPF [10] or IS-IS [11] support dissemination of additional information (*e.g.*, capacity) through ISIS-TE [12] and OSPF-TE [13] extensions. These extensions also support capacity updates on multiple traffic classes [14], enabling MINT to track and sell different levels of service. Various other optimizations (such as sending updates only when certain bandwidth thresholds are crossed) can also help reduce update traffic. Many large networks already employ these technologies for intra-domain traffic engineering.

Pricing. Operators, managing ISP networks and controlling segment sales, set the prices on entire segments. Operators could alternatively set prices on individual links, which subsequently get aggregated to segment prices before being exported to mediator. Although we assume that operators could set segment prices individually, in practice they could also be automated, leveraging prior work on market prediction and automated agent theory [15].

```
<update AS=10001>
  <state id=EFFEFFEE seq=1>
    <exchanges ex1=0xEFFE ex2=0xFFEE />
    <bandwidth bw=100 units='Mbps' />
    <mrb bw=10 units='Mbps' />
    <price hour=10 day=230 month=6500 />
  </state>
  <state id=FFEE3FFF seq=1>
    <exchanges ex1=0xFFEE ex2=0x3FFF />
    <bandwidth bw=300 units='Mbps' />
    <mrb bw=5 units='Mbps' />
    <ask hour=10 day=220 month=6000>
  </state>
</update>
```

Figure 2: Example transit-state update.

Announcement format. Exchanges announced in transit-state updates serve as the "glue" that connects multiple segments to form a complete end-to-end path; they must use uniform addressing to help identify these exchanges. Any naming system that does not allow duplicates can be used (*e.g.*, DNS names, or even MAC-like addresses). In our examples, we will use a 16 bit hexadecimal notation (*i.e. FFEE*).

The interface between the mediator and monitor can use any standard information transfer, or Web services protocol, such as SOAP, which uses HTTP for transport and XML for update formating. We use SOAP due to its ubiquitous deployment, and the fact that it has previously been shown to scale well [16]. In our examples, we encode updates in XML.

Figure 2 shows an example of a transit-state update. The update comes from AS 10001 and contains two transit states: the state between exchanges *EFFE* and *FFEE*, and the state between exchanges *FFEE* and *3FFF*. Transit-states resemble conventional link-states as used in link-state protocols: they have an identifier (*id*) and a sequence number (*seq*). The identifier must uniquely identify the set of exchanges within an ISP, while the sequence number is used to identify the latest update and invalidate older ones. In the example in Figure 2, *id* is formed by pairing exchange point identifiers, while *seq* indicates that this update is the first one to be announced.

The *bandwidth* parameter specifies how much bandwidth is available for purchase, while *mrb* designates the *minimum reservable bandwidth*. Minimum reservable bandwidth is the lowest quantum of bandwidth that is available for purchase. This setting enables data plane scaling if only a limited number of transit paths are supported. The price in the update is indicated with the *ask* parameter. ISPs can set different rates for different timescales. It is likely that longer contracts will mandate lower prices. This schema can be easily expanded as needed for adding new features (*e.g.*, capping minimum or maximum contract duration) in future.

Scaling. Any AS that connects n exchanges and is willing to provide transit between any or all of them might need to send up to $O(n^2)$ updates to the mediator. In practice, large transit ISPs might participate in thousands of exchanges, thus the number of transit-state updates can become prohibitive. Fortunately, such amount of updates are not necessary; recent work on *pathlet routing* [17] suggests using *virtual nodes*

```
<request AS=10002 id=1>
  <start ex=0xEFFE />
  <finish ex=0x3FFF AS=10003 payload='IP' />
  <bandwidth bw=10 units='Mbps' />
  <duration units='day' len=2 />
  <bid max=200 />
</request>
```

Figure 3: Example bid for an end-to-end path.

(exchanges, in our case) to express the topology and pricing of paths within an ISP. The mediator can encourage such segment reductions by charging ISPs for each state that they announce.

3.1.2 Buyers: Edge Networks

Buyers could use a schema similar to sellers to communicate with the mediator. Figure 3 shows an example request. The request indicates an AS requesting a path between an entry and exit exchange; in addition to the exit exchange, the destination AS is also noted. The update contains the required bandwidth and duration for the path. The *bid* variable indicates the maximum price that the buyer is willing to pay for the path. The property *payload* of parameter *finish* is shown to highlight that our mechanism can be used for any payload (more on this parameter in data plane section). This schema can also be easily extended to support arbitrary parameters for path computation in the mediator (*e.g.*, requests for backup paths, exchange-diverse paths, avoiding certain exchanges).

Network operators could issue requests when business policies change, for specific applications that requires on-demand bandwidth (*e.g.*, video conference), or when traffic from some applications increase. Request generation could be automated; network management systems could issue path requests when traffic to given destinations cross predefined thresholds.

Matching paths can be computationally intensive. In the next section, we explore how to minimize unnecessary computations. The mediator could also moderate its load by charging a nominal fee for each new request. Such a fee, the "markup", should be high enough to cover the mediator's costs in running the clearinghouse and also make a profit. This transaction fee is similar to that in stock market trading systems and provides an incentive to run a mediator.

3.1.3 Mediator: Matching buyers and sellers

The mediator has two components: the transit state database, and the computational component. The transit state database processes and stores all the transit-state updates; any received transit-state update causes a change to this database. The computational element is triggered only by path computation requests. The computation element operates using a read-only snapshot of the transit state database, which helps achieve better scaling and consistency.

Matching segments to paths. The computation component in the mediator runs the matching algorithms specific to the good being traded. If only one additive constraint is used (*e.g.*, cost) in combination with one or more limiting constraints (*e.g.*, bandwidth), the simple constrained shortest path first algorithm will find the best match in $O(E + Vlog(V))$ where E is the number of edges and V is the number of vertices.

Scaling computation with lazy recovery. Conventional link-state protocols scale only to thousands of links, so how can MINT, which is essentially a centralized link-state protocol, scale to millions of segments on the Internet? It can, by using a different approach to update propagation and fault detection. Conventional link-state-like protocols control plane recompute all destinations every time there is a change in the network topology; the control plane in such protocols must ensure consistency in hop-by-hop forwarding networks. In MINT we apply a *lazy recovery* approach to convergence. Unlike in conventional protocols, MINT relies on the head-end of each path to detect problems with a path and request re-computation if necessary.

Lazy recovery provides several benefits. No state about active paths needs to be stored in the transit-state database (such state would be necessary to map the failure to the paths that failed). This simplification also reduces both storage and computing requirements; the mediator no longer needs to recompute all the paths or map updates to determine which paths might have failed. Lazy recovery also allows a simplified client-mediator interface, which can be initiated only by the client.

3.2 Data Plane

In this section, we describe MINT's path setup, encapsulation, and failure detection and recovery.

Path setup. Path setup could be performed in several ways: (1) the mediator triggers setup in each participating ISP, (2) the head-end (buyer) issues request to its neighbors or (3) some hybrid of (1) and (2), where both the head-end and mediator participate in path setup. Mediator-triggered path setup could speed up path establishment and might simplify accountability primitives, but it increases strain on a centralized system and limits decentralization capabilities. It would also require a new path setup protocol. The head-end based path-set up, on the other hand, is lightweight and, as described further, does not need any new protocols. Hence, MINT's current design calls for the buyer to explicitly set up paths.

Path requests sent to the mediator (Figure 1) result in path responses that contain a AS-level Explicit-Route-Object (ERO) [18]. The ERO in the response contains enough information for the source to initiate the reservation procedure along the path. Although security and accountability is not our main focus, the XML schema can be easily expanded to include security measures, such as the mediator's signature for each path; such signatures could be used to authenticate path establishment. Figure 4 shows an example ERO.

When an exchange level ERO is received, domain to domain path setup is initiated. Within a domain, the ingress router needs to compute node level path as shown in Figure 5. A Path Computation Element (PCE) [19] is used to offload such requests. Note that PCEs can be co-located with the monitor described in Section *3.1.1* to aid in network resource management and monitoring.

Encapsulation and path stitching. MINT utilizes flow-

```
<response AS=10002 id=1>
  <price match=30>
  <ero>
    <exchange1 id=0xEFFE>
    <exchange2 id=0x333F via=10005 />
    <exchange3 id=0x33FF via=10006 />
    <excahnge4 id=0x3FFF via=10007 />
    <destination id=0x3FFF AS=10003 />
  </ero>
</response>
```

Figure 4: Example explicit route object (ERO) response.

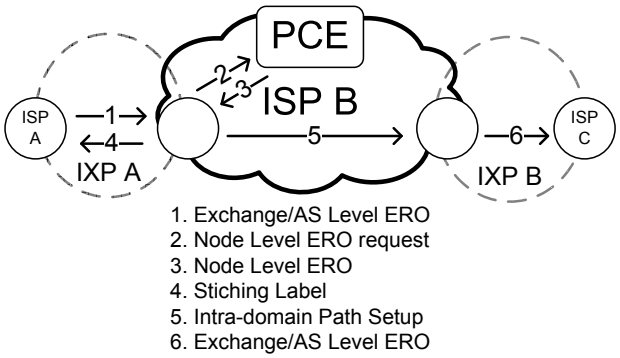


Figure 5: Path setup process in a transit ISP. Inter-domain path stitching.

based forwarding for end-to-end paths. Flow forwarding can be achieved with either hop-by-hop IP-in-IP tunneling or label switching (*e.g.*, MPLS [20]). Label switching allows inter-domain paths to carry any payload. In the simplest case such a payload would be plain IP packets between the source and the destination domains. The source AS would establish its own routing rules to route IP traffic though established paths; unmodified packets would enter the destination AS border router for further treatment. Aside from simple IP, MINT also supports other types of encapsulation, such as Ethernet frames, which would allow a source and destination to form a virtual interface, similar to what one would form in a switched exchange. Such peers could even establish a BGP session over such virtual interfaces and exchange traffic subject to BGP policies, allowing for a "virtual BGP session", even when two ASes are not physically present at the same exchange.

As shown in Figure 5, the first (*i.e.* ingress) node in the source network computes and establishes a label-switched path to the last (*i.e.* egress) node. The egress node contacts the ingress node in the next domain and requests a *stitching label* for proper forwarding. This stitching label allows the egress router to mark traffic traversing the exchange so that the next-hop ingress router can match it to the appropriate path. The process is continued until the last AS (noted in the *destination* parameter in the path request in Figure 4) is reached and a path is established to its border router. Such path stitching protocols for interdomain paths are already supported by many vendors with most high-end routing equipment; multiple standards already ensure compatibility across ASes and router vendors [21]. These protocols are primarily used today in networks that comprise several autonomous systems; MINT enables the use of this technology in the Internet at large by offering a transparent segment trading platform.

MINT provides better network resource utilization. Compared to a BGP-based market, MINT increases the average resource utilization by a factor of 2.	Figure 13(a)
MINT increases the total welfare. MINT increases the total welfare by 116% compared to a system based on BGP.	Figure 15
Opening up the market increases demand and supply matching. Using MINT about 25% more requests are matched than using BGP paths for matching.	Figure 14(a).

Table 1: Summary of main results.

Fault detection in lazy recovery setup. MINT detects faults in band, at the node where the path starts. Depending on the encapsulation mechanism, different fault detection mechanisms are possible. For example, if an Ethernet-like virtual link is used, the remote nodes can use high-performance fault detection mechanisms such as Bidirectional Forwarding Detection (BFD) [22]. For important paths, the edge may pre-compute backup paths; fail-over time does not exceed hundreds of milliseconds in these cases (which would be faster than a BGP fail-over by several orders of magnitude). For less important paths, the edge simply sends a new computation request to the mediator to perform the whole path setup again.

Scalability. Both path computation and maintenance must operate at Internet-scale. The main challenges are: (1) finding node-level paths is a computationally intensive task and it risks overloading the ingress routers, (2) the forwarding tables in routers must be large enough to fit all the flows traversing them, and (3) the routers with many traversing paths must be able to sustain high path establishment and tear down rates.

The Path Computation Element (PCE) [19] presents a possibility for setting up a node-level path. PCE is both a name for the protocol framework and a name for the element in that framework that computes constrained shortest paths. In the simplest case, the computation in PCE is triggered by the ingress router; more sophisticated scenarios allows PCEs from different domains to interact to pick the best egress-ingress router pairs at inter-domain boundary to balance load [23]. Dedicated path computation elements can both reduce the computational load and balance load across multiple ingress nodes.

Routing tables in today's high-end routers can support millions of label-switched paths at line-rate [24]. As far as path setup is concerned reservations can be performed at the rate of hundreds of thousands paths per second if necessary [18]. If for some reason the routing infrastructure cannot sustain the rate of path establishment and tear down, operators can resort to using minimum-reservable-bandwidth (*3.1.1*) transit state announcements to reduce the maximum number of paths that can be reserved on any given segment.

4. Economic Evaluation

We evaluate MINT economic performance evaluation metrics as well as scaling properties. This section deals with the former while Section 5 deals with the latter. Economic evaluation seeks to study resource utilization, utility to end networks, and revenue/profits of ISPs. We compare these metrics in networks served by MINT to the same networks running conventional path-vector routing protocols.

While MINT can be used for a variety of traded commodities, we choose network bandwidth as the commodity for evaluation in this section. A network bandwidth market is well suited for our evaluation because the commodity is a rivalrous good (*i.e.*, a reservation prevents other reservations for the same bandwidth); it can be linearly priced (*i.e.*, prices are set per megabit/second); and the bids for guaranteed bandwidth can be easily matched with available segments using simple shortest path algorithms. We summarize the main results of this evaluation in Table 1.

4.1 Evaluation Metrics

We focus on three metrics for economic performance evaluation: (1) user utility (or surplus); (2) provider profit; and (3) resource utilization.

User surplus. A successful bidder b's utility is the difference between its valuation v_b for a path, and the final price p_b it pays for the path; note that this price is the sum of segment prices that comprise the path. If a bidder is unsuccessful, i.e., if they are not allocated a path, we define their utility as zero. We define *user surplus* as the sum of all user utilities. In other words, if B^* denotes the set of all successfully accepted bids, then user surplus is: $\sum_{b \in B^*} v_b - p_b$. Typically in our analysis, we will consider user surplus per successful bid; in this case we normalize the user surplus by the size of the set B^*.

ISP profit. In principle, an ISP's profit is the difference between its total revenue from participation in MINT, and the costs incurred in running their network. Since costs are exogenously determined constants, for the purposes of our simulation, we set the cost to zero; thus we compute only ISP revenue. Thus the *total ISP profit* is the sum of all payments to ISPs made when bids are accepted. (Note that ISPs are paid exactly the segment price they posted.)

Network resource utilization. Finally, to gain insight into the extent to which MINT discovers and allocates unused capacity, we also study average resource utilization across the network. In particular, for each segment j, let ℓ_j be the total utilized capacity on that segment; and let c_j be the capacity of that segment. Then the *network resource utilization* is $\sum_j \ell_j / \sum_j c_j$. Note that this is not a perfect metric; for example, a more efficient routing of the same traffic matrix could cause this metric to decrease. However, as the results demonstrate, this metric provides good insight into the relative benefits of MINT compared to traditional BGP-based protocols.

4.2 Simulation Setup

We developed a custom, event-driven simulator to evaluate different market mechanisms. A high-level simulator diagram is shown in Figure 6. The main inputs to the market simulator are the *topology* and the *demand*. The topology is a graph of exchanges connected with segments as links. Each segment has a capacity and a price associated with it; both these parameters change as the simulation progresses. The demand is determined by a bid arrival process, where each bid has an associated maximum valuation, capacity constraint, and desired duration; we assume the demand process statistics are stationary for the lifetime of the simulation. As bids arrive, the market attempts to immediately match the bid to a path given the current topology; if this is not possible, the bid is blocked. As capacities change, the pricing of the segments is updated; see Section 4.3 for further details on our price update model.

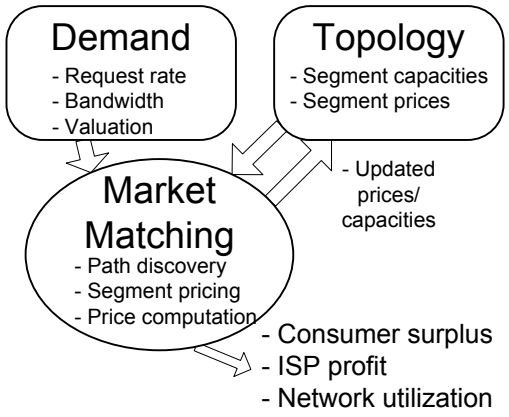


Figure 6: High-level evaluation model. The market matches the demand given the current topology. Demand is exogenous to the model, while the topology (including segment prices and segment capacities) changes as the simulation is running.

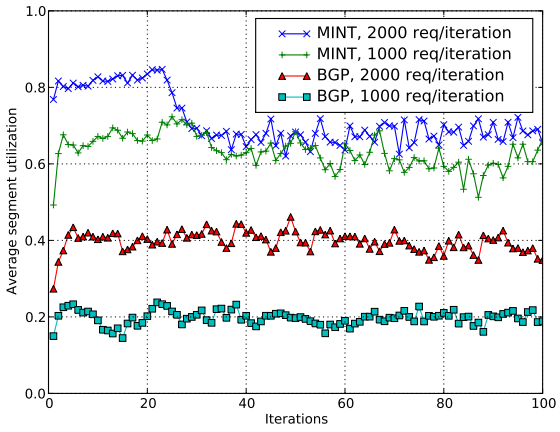


Figure 7: Average utilization convergence over the duration of simulation.

Each simulation run represents 200 iterations of market operation; ISPs issue price updates at the end of every iteration cycle. The chosen length of simulation was generally sufficient to observe market convergece to equilibrium. As seen in Figure 7, in most cases segment utilization and subsequently prices converge within 50 price updates.

4.2.1 Network topology

Network	Exchanges	Segments
PeeringDB	201	15439
Orbis-24	24	218
Orbis-48	48	1242

Table 2: Network topologies used for market evaluation. PeeringDB topology is derived directly from the peering database. Orbis-24 and Orbis-48 are extrapolated from the PeeringDB topology to have 24 and 48 exchanges accordingly.

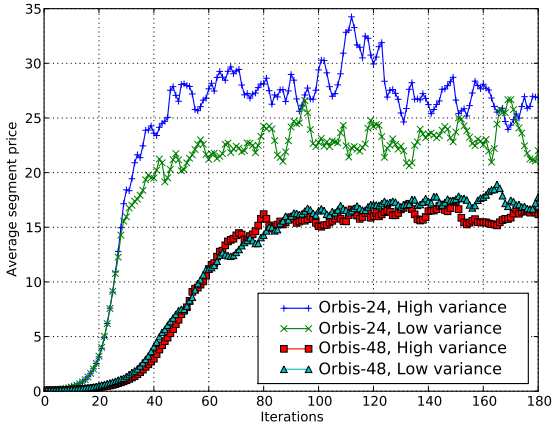


Figure 8: Simulation convergence to average segment price. MINT market model, 1000 requests per second.

To create a realistic exchange-level topology we used the PeeringDB [25] database. It contains public and private exchange points, along with a list of domains that participate in such exchanges. The seed topology contains 201 exchanges and 1116 domains. Many domains participate in more than one exchange; we noted 3804 unique domain presences in all exchanges. We assumed that when a domain participates in more than one exchange, it can provide segments between such exchanges. This approach yielded 15439 possible segments.

We also generated two smaller test topologies for faster experimentation. We used the Orbis [26] topology generation tool for new topology generator. Orbis takes a graph as an input, and produces a graph with given size and similar properties. We used our PeeringDB topology as a seed, and used Orbis to generate one topology with 24 exchanges and another with 48 exchanges. Table 2 shows the properties of the topologies we used in our evaluation.

We used the Orbis-24 topology as the reference testing topology. The Orbis-48 topology was used to rerun some experiments to confirm if similar trends hold as the graph size changes. Figure 8 shows that the larger topology takes slightly more time to converge. The larger network also results in a lower average price, due to a higher number of underutilized segments.

Segment capacities. We derive segment capacities in the PeeringDB topology from the declared speeds at the exchanges. Each participant in the database documents the speed of its connection to the exchange. When we form a segment (link) between two exchanges, we take the minimum of the two declared speeds. While this can overestimate real capacities and does not account for competing requests that might be using an overlapping set of links at the network core, it provides a good starting point for market evaluation. The simulated Orbis topologies derive their capacities from the capacity distribution in the PeeringDB topology. The capacities of each segment change during the simulation as paths are reserved and released.

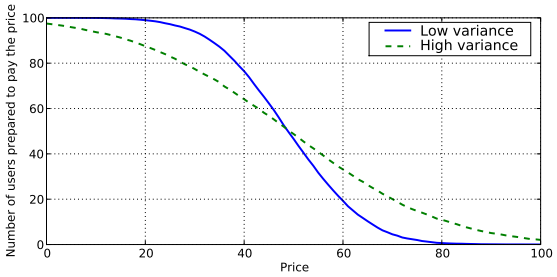


Figure 9: We use two aggregate demand curves to simulate path valuation: one with high variance in valuations, and another with low variance in valuations. X-axis is price, y-axis is the number of users in the population willing to pay that price.

4.2.2 Demand

The demand from end users is defined by four distinct parameters: (1) bid arrival rate (in number of bids per iteration), (2) desired reservation time for each bid (in number of iterations) (3) requested capacity, and (4) bid valuation.

Requested capacity. From the topology and the capacities described in Section *4.2.1* we generate a traffic matrix that nominally maximizes network resource utilization. This traffic matrix gives the mean requested capacity per iteration.

Formally, let f^{sd} denote the mean requested data rate between source s and destination d. We define a *weight* α^{sd} as the logarithm of the shortest path length from s to d. Our nominal traffic matrix then maximizes the weighted sum $\sum_{s,d} \alpha^{sd} f^{sd}$, subject to the exogenously specified capacity constraints in the network.

This nominal traffic matrix achieves two goals. First, it ensures that our arriving demands are well matched to network capacity, and will saturate some subset of segments in the network. Second, it ensures that we have a good mix of traffic between long and short path length flows in the network.

Request rates and reservation time. To achieve liquidity, we dynamically issue multiple requests between each source and destination that has a non-zero mean requested capacity f^{sd}. Requests arrive as a Poisson process at a rate of λ requests per interval; given the mean flow f^{sd} between source s and destination d, the size of each request follows a Pareto distribution with tail exponent 2, and mean $\mu^{sd} = f^{sd}/\lambda$. We vary the parameter λ in our simulations to study the effects of increasing liquidity.

The reservation time determines how long the successful request keeps the bandwidth. We draw reservation time from a Pareto distribution with mean equal to 3 iterations, and tail exponent 2.

Request valuation. Each request has an associated val-

Protocol	MINT	Path-Vector(BGP)
Discovery	Full view of the network	Only paths offered by upstreams
Purchase	Any path	Valley-free paths
Pricing	Segment pricing	Segment pricing

Table 3: Two market models and their differences. We assume segment pricing for BGP.

uation per unit of bandwidth. For evaluation purposes, we use two different demand curves, shown in Figure 9. The x-axis shows the price P per bandwidth unit, while the y-axis shows the number of users Q in our simulated population prepared to pay that price. We draw prices from a *Gaussian* distribution using a mean of 50 price units and a standard deviation of 12.5 (resp., 25) for low variance (resp., high variance) demand. Note that the y-axis in our case represents a proportion of the population willing to pay at least any given price. For example, in low variance demand, a price of 40 units will be affordable for 75% of users, while in the high variance demand model only 65% of users are prepared to pay the same price for a unit of bandwidth. Both demand curves are normalized: two equally large populations drawn from these two curves will produce the same aggregate valuation—only the variance will differ.

4.3 Market Formation

We compare two market models in our paper: the market formed by the MINT protocol, and the market formed by path-vector routing using BGP. These two models differ in resource discovery and in demand-to-supply matching.

The difference between the markets is outlined in Table 3. When a new bid for bandwidth arrives at the network, it must be matched to the existing topology accordingly to the market model in use. The following subsections describe how discovery and purchase choices are made in each model.

MINT discovery and request matching. Supply and demand matching in MINT is done sequentially using the constrained shortest path algorithm. The mediator has complete information about the advertised segments. First, the algorithm prunes edges that do not match the capacity requirements and then runs Dijkstra's algorithm on the resulting graph using price as the distance metric. The resulting path is the lowest price path that contains enough bandwidth for the request (if such a path exists).

The matching, in essence, performs a reverse combinatorial auction. There are several ways to price the end path. In our model, we set the end-to-end path price as the sum of segment prices. This is a version of a first-price auction, though it is carried out dynamically and continuously. It is also possible to implement alternative designs; for example, the Vickrey-Clarke-Groves (VCG) [8] auction is a more sophisticated auction design where the winning seller gets a payment indicated by the second best offer. In the case of MINT, such an arrangement could be implemented by defining the effective price of each segment j in a path (found by the constrained shortest path algorithm) as the difference between the best end-to-end price in a graph *with* segment j, and the best end-to-end price in a graph *without* segment j.

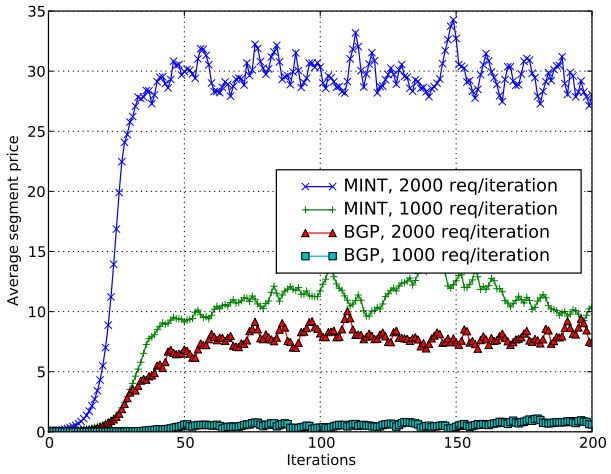


Figure 10: Average segment price for different loads as time progresses. Higher load produces higher prices.

BGP discovery and request matching. BGP is a path-vector protocol. Each participant in the protocol selects one best path and forwards announcements about such paths downstream. Thus, the number of paths any source can choose is limited to the number of neighbors it has. Furthermore, paths propagate subject to policy constraints. The policy constraints enforce valley-free property [27].

We implement a version of BGP routing that can be directly compared to MINT. When requests arrive from source s, we construct a spanning BGP tree for each neighbor $n \in N_s$ (where N_s is a set of s neighbors) starting with n as root and ending at all other nodes. We prefer shorter paths over longer paths, breaking ties to prefer paths with lower total price. While constructing such trees we avoid the non-valley-free paths. To find valley-free paths we infer the relationships between participating autonomous systems using a simple heuristic. For two ASes i and j, we say that i is a service provider for j if $N_i > N_j$, and vice versa; in case $N_i = N_j$ we break the tie randomly. Such a heuristic allows us to build a hypothetical hierarchical structure that can be explored to build valley-free paths.

For each constructed BGP tree starting at each neighbor of source s, we find the path to destination d. We eliminate paths that cannot provide enough bandwidth and pick the lowest priced path among the remaining paths.

Note that our algorithm matches BGP in its path discovery; however, in contrast to the use of BGP in the current Internet, our BGP-based model prices path segments. In the current Internet, the interconnections are priced on long timescales, instead of fast timescale pricing of segments. Similarly, end users on the Internet pay directly only for their immediate upstream connection, while in our BGP-based model buyers pay each ISP on the allocated path for use of their segment. Although unusual, this model allows us to directly evaluate the effectiveness of MINT in exploiting available capacity that lies dormant under standard BGP

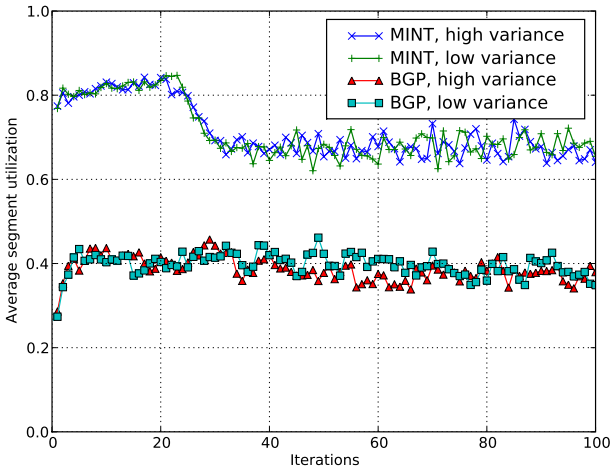


Figure 11: Average utilization convergence over time. Despite different demand profiles, the utilization is moderated around the same level.

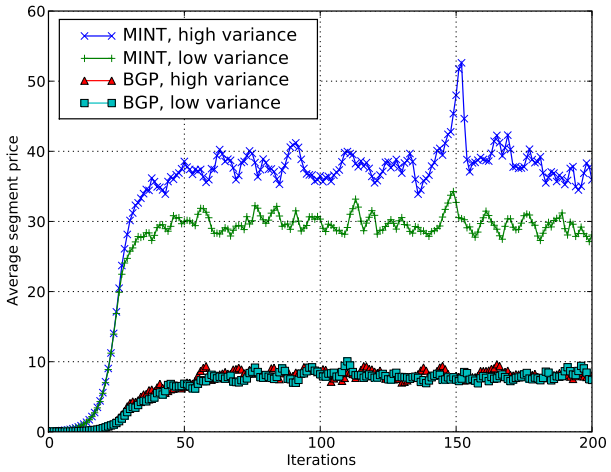


Figure 12: Average price convergence over time given different demand. Due to limited path discovery, demand variance doesn't significantly affect the BGP market.

path discovery mechanisms; our use of prices in both models controls for the effects of introducing a market model.

Segment pricing. We apply a simple segment pricing heuristic that seeks to maximize the resource utilization of each ISP without selling segments or connections at a loss. Resource utilization maximization is justifiable because the marginal cost of allocating available capacity (until a desired utilization threshold is reached) is nearly zero.

Each ISP updates prices for their segments independently, as follows. Initially, the price of each segment is set to a small value. The price is then updated according to the load. If load is below a fraction τ of the link capacity, we reduce the price by a multiplicative factor β; on the other hand, if load is above a fraction τ of the link capacity, we increase the price by a multiplicative factor β. We use a factor $\tau = 0.8$ in our simulations. In our experiments, we found that $\beta = 2$ provided a reasonable convergence time. For example, Figure 10 shows prices converging within 50 iterations.

4.4 Results

Figures 7, 10, 11 and 12 show the convergence properties of the market models. There are two dimensions to convergence: convergence of resource utilization, and convergence of prices. The average utilization as reported in the figures, corresponds to the reservation load on all the segments of the network, averaged by the number of such segments. Similarly, the average price is the sum of all segment prices normalized by number of segments.

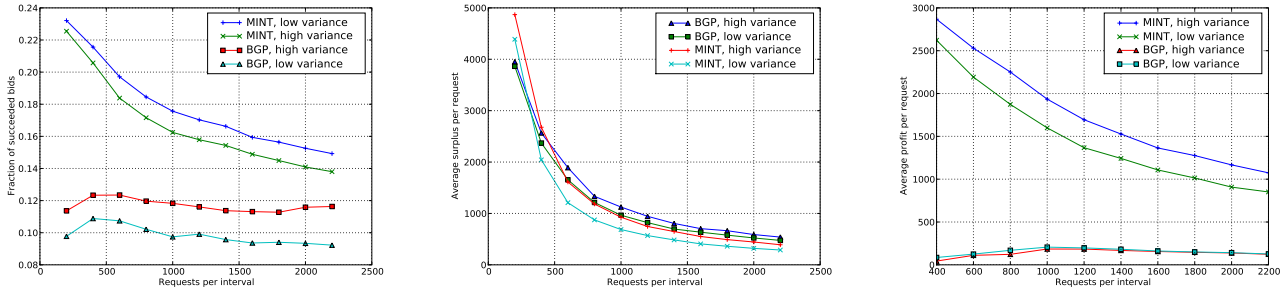
We study the impact of convergence as we vary request granularity and the valuation profile. Request granularity is altered by changing the arrival rate λ; note that in our model, this still leaves the mean source-destination flow unchanged (cf. Section *4.2.2*). The high and low variance demand profiles are described in Section *4.2.2*; note that total user valuation grows with number of requests.

Figures 7 and 11 show that utilization peaks initially and then, after approximately 30 iterations is moderated by the price. The effect is less pronounced in case of BGP, because most of the of the network is underutilized. We can correlate that with the price plots (Figures 10 and 12), where we see that prices reach their equilibrium around at around 30-50 iterations. In case of the BGP market, the demand variance does not considerably affect the market due to a limited ability to explore more paths. The few paths that are available get utilized from the very start independent of the demand variance.

Figure 13 shows the price and utilization trends as we increase the number of requests in the system. As expected, in case of MINT, the utilization reaches a steady level when the number of requests is enough to saturate the capacity computed by our maximum-flow formulation. At 100, 200 and 400 requests per iteration, the network is not fully utilized under MINT, as the granularity of the requests often results in denied capacity. As we increase the requests, more users contend for the same bandwidth and at the rate of 600 requests per iteration it achieves constant average utilization. The BGP-based market takes more requests to find a stable resource utilization. Most importantly, we find that MINT improves resource utilization over BGP by approximately 100%.

Figure 13(b) shows the average price trends as we increase number of requests for different simulations. As MINT loads more segments, the average price over the network is much higher. The unutilized segments in the BGP market drive the average segment price down.

Figure 14 summarizes the main results of our simulation. First we can see in Figure 14(a) that MINT attains a higher successful bid rate in comparison to the BGP-based market (MINT has approximately 25% more successful bids). As we compare the user surpluse per successful bid in the BGP-based market and MINT, we observe that the difference is negligible—in fact BGP has slight a edge. Marginally higher surplus in BGP can be explained by the fact that only the bids with the highest valuations managed to secure reservations.



(a) Success rate. (b) Average per-bid surplus. (c) Average per-bid profit.

Figure 14: Bid success rate and the surplus/profit per-successful-bid as we vary the request rate.

The most interesting trend is in Figure 14(c), where it is seen that ISP profits in MINT are much higher.

Finally, Figure 15 shows the total welfare (user surplus plus ISP profit) the markets produce. Not surprisingly, MINT produces approximately 116% greater welfare in comparison to the BGP-based market. This result is a combination of higher resource utilization, and a greater percentage of successful bids.

5. Scalability

To be feasible, MINT has to scale to the size of the current and future Internet. In this section, we evaluate the scaling performance of MINT. We examine how much memory the mediator needs to store the state of advertised path segments, both for the current AS-level topology and in the future. We also examine mediator computational power requirements.

5.1 Memory Requirements

The number of ASes in the Internet is growing all the time–the number of paths MINT needs to serve will grow as well. We show the amount of storage today is manageable and that the growth rate is manageable given projected growth in semiconductor technology. There are two primary components in MINT that where we study memory requirements: at the mediator, and at routers on a forwarding path.

Memory overhead at the mediator. The mediator must store the state of each segment. We pick an approximate storage size for each update with the following parameter assumptions: 32 bytes for segment end-point identification; 16 bytes for available bandwidth; and 24 bytes for price information—a total of 72 bytes. Table 4 shows the projected size of the Internet in 2020 [28] and the corresponding size and cost of the mediator memory to maintain state. To compute the expected number of segments, we use the extrapolated number of ASes and generate a graph using Orbis [26], which allows arbitrary scaling of Internet topologies. Table 4 shows that MINT is technically feasible; we see that the growth in the number of segments (and hence the memory the mediator requires) is outpaced by the expected drop in DRAM prices. Thus, even as storage required for MINT segments increases, the overall cost of memory required to support the necessary storage will fall.

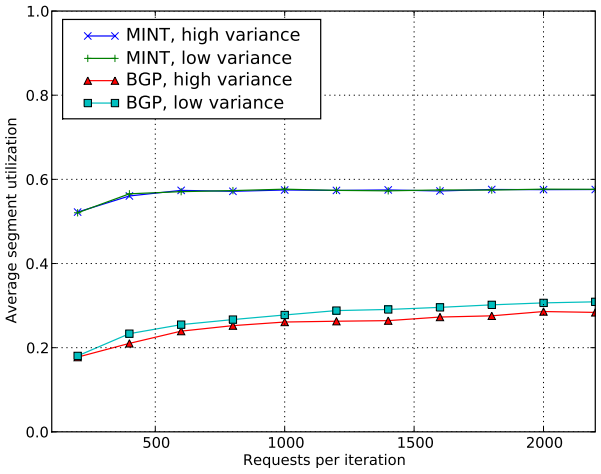
Component	2007	2020
DRAM $/Gbit	9.6	0.3
CPU Frequency (GHz)	4.7	12.4
Internet(AS)	25,000	60,000
Segments	68M	400M
Memory (Cost)	4.6GB ($353)	27GB ($64)

Table 4: Growth projections for the Internet, the memory cost and the CPU speeds [29].

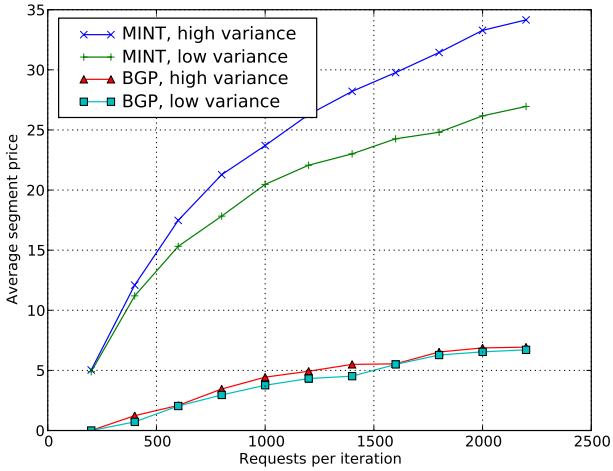
Memory requirements on routers A forwarding node has to be able to maintain information about all the paths that pass through it. Because MINT may be used in conjunction with BGP—as opposed to being an outright replacement—we don't expect bids for connecting every AS to every AS. Instead, we envision MINT being used to satisfy bids for paths with very specific purposes, such as establishing reliable paths to subsets of content providers. As a simple scenario, we consider a case where edge networks wish to establish reliable, high quality paths to a subset of destinations. We evaluate such a scenario on an AS-level graph from RouteViews [30] where 80% of total ASes would be willing to pay for a path to destinations, which currently form about 5% of the Internet [28]. The busiest AS in such case has a few million (5.5) paths passing through it. Since ASes have multiple border routers, each router will have to store a few hundred thousand paths in memory; this is feasible with current technology. Further, techniques such as [24] can be applied to reduce the router memory requirements to even lower levels.

5.2 Computational Requirements

In this section, we explore MINT's computational requirements, and how these requirements are likely to scale with both the growth of the Internet and various failure rates. We first examine how the growth of the Internet might affect the increase in the number of bids that the MINT mediator would have to satisfy. We then examine how increasing failure rates might affect the scaling of the mediator—both in terms of the number of updates that the mediator would need to process and in terms of the frequency with which the mediator would have to recompute new end-to-end paths as



(a) Utilization.



(b) Price

Figure 13: Utilization and price trends as we vary the request rate.

a result of changes in available segments and capacities.

Bid-induced computations To estimate the bid-arrival rate that the mediator would need to process we consider the same scenario as in the previous section, where 80% of ASes (stubs) request paths to 5% of destination ASes. We also assume that ASes periodically re-evaluate their path requirements and re-submit bids accordingly. Increasing either fraction (of destinations and stubs) would affect the bid rate linearly. Table 5 shows the estimated number of queries that the mediator has to process in such a scenario. We also show the case where the fraction of CPs is 10%, translating to a higher query rate (twice the rate) at the mediator. We see that the number of queries that the mediator has to process due to the larger number of ASes increases only by a factor of 5.5 by 2020. Improvements in chip density as well as chip frequencies in the same timeframe (Table 4) are expected to outpace this rate quite comfortably.

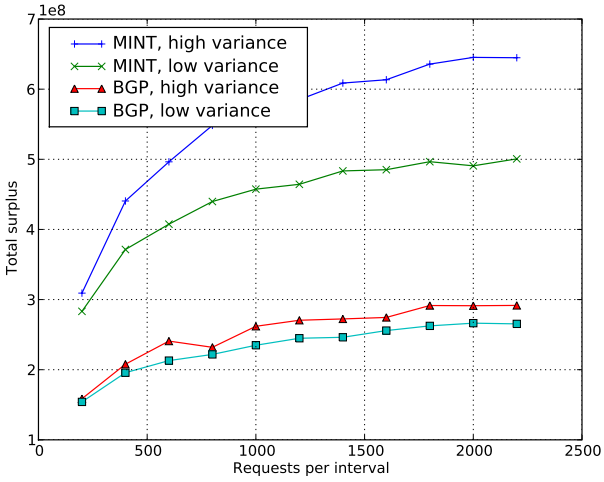


Figure 15: Total system surplus as we vary the request rate.

Periodicity	% of Destinations ASes	2007	2020
Weekly	5	41.3	238.1
	10	82.6	476.2
Monthly	5	9.7	55.6
	10	19.4	111.2

Table 5: Bids (queries/sec.) the MINT mediator must process.

Failure-induced churn In this section, we study the effects of link failure rates inside an AS on MINT. Link failures affect ASes in which they occur, by reducing the segment capacities that the AS might have advertised to the mediator, and hence also the mediator as it has to recompute paths and also re-assign affected paths.

We first explore how failure rates of links within each AS affect the rate of updates seen at the mediator. Let "segment failure" denote the case when a segment is either disconnected or has its capacity reduced as as result of link failure. The number of announcements is proportional to the number of segment failures. We would expect to see announcements scale linearly with failure rates. (note that in BGP, a single link failure can produce a non-linear amount of updates [31]). To avoid frequent churn-induced updates, ASes could have an incentive to under-advertise their links. We also study the effect under-advertisement has on the number of announcements.

We use the Rocketfuel Point-of-Presence (PoP) topologies for AS 1239 (Sprint) and AS 3356 (Level 3), assuming that the inferred links weights reflect actual link capacities [32]. For each topology, we initially allocate capacity for all segments to be the bandwidth of the largest bottleneck link in the segment. The capacity that an AS advertises to the mediator is assumed to be some fraction of this initial capacity, which we define as the *advertisement ratio*. We fail intra-AS links at random, recalculate segment capacities, and determine the resultant number of updates received at the media-

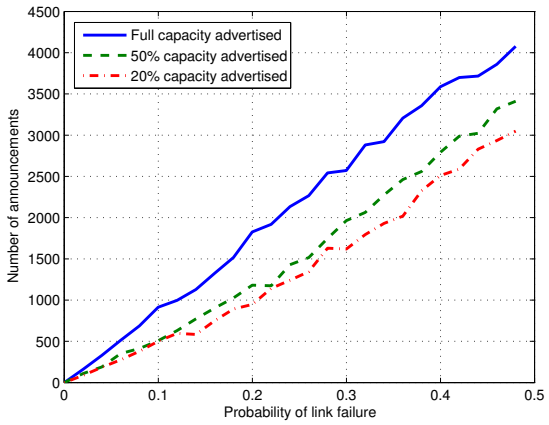


Figure 16: Number of updates resulting from churn inside AS 1239.

tor.

Figure 16 shows the number of announcements, for different failure rates, averaged over fifty trials. For AS 1239, advertising 50% of segment capacities reduce the number of announcements to the mediator by 32%, while advertising 20% of capacities reduces the number of announcements by 41%. Similarly, for AS 3356 (not shown due to space constraints), an 50% capacity advertisement reduces the number of announcements by 53% while 20% capacity adverisements reduces the number by 57%. The results confirm our expectation that update announcements increase linearly with failure rates. The mediator has to process these updates in order to recompute and reallocate paths, hence the computational requirements at the mediator also scale linearly.

6. Related Work

Bandwidth was recognized early as a potential market commodity. In 1999, Enron Communications proposed a bandwidth market for buying and selling bandwidth as an exchangeable commodity on the basis of a standard contract between the parties involved. The Bandwidth Trading Organization (BTO) [33] is similar to the mediator in that it serves as the central clearing house. They key difference is that the BTO is governed by physical traders of bandwidth while the mediator is an automated marketplace. Moreover, as in MINT, Pooling Point Administration, which is done by transit networks in MINT, is independent from the BTO. Exchanges like Arbinet [34] and Equinix [35] offer products and services that allow better network connectivity and route management. Unlike MINT, their market is not open and the services may be limited to the few QoS classes supported by the exchanges. Electronic bulletin boards like BuySellBandwidth [36] allow bandwidth buyers and sellers to post goods needed or sold but does not mediate the exchange of goods by enforcing any particular contract between parties involved. In contrast, MINT offers a market mechanism to profitably exchange goods.

The bandwidth exchange market, BAND-X [37] offers a clearing house for spot and future market operation. It also incorporates accounting and authorization mechanisms. Unlike MINT, the clearing house may simply be an electronic bulletin board like [36] and the customer may need to construct end-to-end path herself. M3I technology [38] offers multiple services using end-to-end session-oriented business models over a connectionless service. Bill-Pay [39] uses a micro-payment scheme to extend the bilateral contracts between service providers and encourage them to provide better services. In contrast, MINT works at a higher granularity and allows implementation of diverse business models while offering an open market for connectivity to foster competition and efficient resource utilization. The recent spate of de-peering [2–4], the availability of excess capacity [40] and evidence for increasing market competitiveness [41] seem to support the rationale for MINT in todays Internet.

Open market models have been recognized as possibly effective resource allocation schemes for large distributed systems [42]. While MINT does not propose entirely new market mechanisms, it may benefit from developments in multipart combinatorial auctions [43]. Moreover, we do not analyze price dynamics due geographical arbitrage and liquidity effects and market dynamics due to network failures. The work by Cheliotis et al. [44] provides direction for such analysis. Our work is also related to congestion pricing [45], which is a form of market segmentation according to user resource requirements and willingness-to-pay.

7. Discussion and Conclusion

This paper has presented MINT, a market structure and supporting set of protocols for enabling edge networks to purchase end-to-end paths from ISPs that sell path segments between exchange points. We have introduced a new market structure—and a set of supporting routing protocols—for buying and selling Internet paths that can co-exist with the existing interdomain routing framework and can operate using protocols that are already deployed on today's Internet routers. Given the recent network neutrality debates regarding the cost of access, MINT may also provide an alternative for transit and access ISPs to sell end-to-end paths to edge networks that might not otherwise be profitable. We briefly conclude with several interesting related issues and open directions.

Multiple mediators. We briefly describe how MINT could be adapted to a setting with multiple mediators. In a distributed case, key ISPs (or any network participant with enough resources) can maintain copies of the transit-state database. At each independent participant, dedicated middle-boxes can form SOAP sessions with their neighbors to form a transit-state exchange network that keeps the database up-to-date. The neighbors in such a network would exchange updates periodically with update intervals long enough to process all the updates received during the exchange window. In facilitating non-hierarchical update distribution the network could run an OSPF-like flooding mechanism; unlike OSPF, however, the transit-states would not trigger the path computations (as described above) and would be first processed and delayed before forwarding. (In high performance OSPF-like protocols the priority is to for-

ward the updates.) Even with long update intervals (*e.g.*, on the order of several hours), if network participants establish a well connected update exchange graph, the updates could propagate through the network within a dozen or less exchange sessions. Stale information and blocked path requests are the main side effect of such decentralization; failure detection speed and recovery is not affected.

Contract enforcement. This paper has focused on a market for *establishing* contracts, but has not developed any mechanisms for actually enforcing those contracts. We believe, however, that MINT-style contracts may be more enforceable than the bilateral contracts that exist in today's Internet, primarily because either the mediator or the party that purchases the path has direct recourse when an end-to-end path is not performing as expected. Additionally, all traffic passes through exchanges, which could serve as natural places for monitoring traffic. In future work, we plan to examine how contracts in MINT could be enforced.

8. Additional Authors

REFERENCES

[1] Y. Rekhter, T. Li, and S. Hares. *A Border Gateway Protocol 4 (BGP-4)*. Internet Engineering Task Force, January 2006. RFC 4271.

[2] C&W Briefly Drops Peering Agreement with PSINet. `http://www.isp-planet.com/news/2001/psinet_cw.html`, June 2001.

[3] Alin Popescu and Todd Underwood. D(3) Peered: Just the Facts Maam. A Technical Review of Level (3)s Depeering of Cogent. In *NANOG 35*, October 2005.

[4] Martin A. Brown and Alin Popescu and Earl Zmijewski. Peering Wars: Lessons Learned from the Cogent-Telia De-peering. In *NANOG 43*, June 2008.

[5] Paul Laskowski and John Chuang. Network monitors and contracting systems: competition and innovation. In *Proc. ACM SIGCOMM*, Pisa, Italy, August 2006.

[6] Katerina Argyraki, Petros Maniatis, Olga Irzak, Subramanian Ashish, and Scott Shenker. Loss and Delay Accountability for the Internet. In *IEEE International Conference on Network Protocols (ICNP)*, Beijing, China, October 2007.

[7] Boaz Barak, Sharon Goldberg, and David Xiao. Protocols and lower bounds for failure localization in the internet. In *Proceedings of Eurocrypt 2008*, 2008.

[8] Nicole Immorlica, David Karger, Evdokia Nikolova, and Rahul Sami. First-price path auctions. In *EC '05: Proceedings of the 6th ACM conference on Electronic commerce*, 2005.

[9] Aman Shaikh and Albert Greenberg. OSPF Monitoring: Architecture, Design, and Deployment Experience. In *Proc. First Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, March 2004.

[10] J. Moy. *OSPF Version 2*, March 1994. RFC 1583.

[11] D. Oran. *OSI IS-IS intra-domain routing protocol*. Internet Engineering Task Force, February 1990. RFC 1142.

[12] H. Smith and T. Li. *Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE)*. Internet Engineering Task Force, June 2004. RFC 3784.

[13] D. Katc, K. Kompella, and D. Yeung. *Traffic Engineering (TE) Extensions to OSPF Version 2*. Internet Engineering Task Force, September 2003. RFC 3630.

[14] F. Le Faucheur. *Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering*. Internet Engineering Task Force, June 2005. RFC 4124.

[15] Yiling Chen. Prediction markets: economics, computation, and mechanism design. In *EC '07: Proceedings of the 8th ACM conference on Electronic commerce*, 2007.

[16] Qi Yu, Xumin Liu, Athman Bouguettaya, and Brahim Medjahed. Deploying and managing web services: issues, solutions, and directions. *The VLDB Journal*, 2008.

[17] Brighten Godfrey, Scott Shenker, and Ion Stoica. Pathlet routing. In *Proc. 7th ACM Workshop on Hot Topics in Networks (Hotnets-VII)*, Calgary, Alberta. Canada., October 2008.

[18] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. *RSVP-TE: Extensions to RSVP for LSP Tunnels*. Internet Engineering Task Force, December 2001. RFC 3209.

[19] A. Farrel, JP. Vasseur, and J. Ash. *A Path Computation Element (PCE)-Based Architecture*. Internet Engineering Task Force, August 2006. RFC 4655.

[20] Bruce Davie and Yakov Rekhter. *MPLS: Technology and Applications*. Academic Press, San Diego, CA, 2000.

[21] A. Ayyangar, K. Kompella, JP. Vasseur, and A. Farrel. *Label Switched Path Stitching with Generalized Multiprotocol Label Switching Traffic Engineering (GMPLS TE)*. Internet Engineering Task Force, February 2008. RFC 5150.

[22] D. Katz and D. Ward. *Bidirectional Forwarding Detection*. Internet Engineering Task Force, March 2008. Internet Draft draft-ietf-bfd-base-08.txt; work in progress.

[23] JP. Vasseur, R. Zhang, Bitar N., and JL. Le Roux. *A Backward Recursive PCE-based Computation (BRPC) Procedure To Compute Shortest Constrained Inter-domain Traffic Engineering Label Switched Paths*. Internet Engineering Task Force, April 2008. Internet Draft draft-ietf-pce-brpc-09.txt; work in progress.

[24] Hitesh Ballani, Paul Francis, Tuan Cao, and Jia Wang. ViAggre: Making Routers Last Longer! In *Proc. 7th ACM Workshop on Hot Topics in Networks (Hotnets-VII)*, Calgary, Alberta. Canada., October 2008.

[25] Peering Database. `http://www.peeringdb.com`.

[26] Mahadevan, P. and Hubble, C. and Krioukov, D. and Huffaker, B. and Vahdat, A. Orbis: rescaling degree correlations to generate annotated internet topologies. In *Proc. ACM SIGCOMM*, pages 325–336, Kyoto, Japan, August 2007.

[27] Lixin Gao. On inferring automonous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, December 2001.

[28] Amogh Dhamdhere and Constantine Dovrolis. Ten Years in the Evolution of the Internet Ecosystem. In *Proc. Internet Measurement Conference*, Vouliagmeni, Greece, October 2008.

[29] Semiconductor Roadmap. `http://www.itrs.net/`.

[30] University of Oregon. RouteViews. `http://www.routeviews.org/`.

[31] C. Labovitz, R. Malan, and F. Jahanian. Internet Routing Instability. *IEEE/ACM Transactions on Networking*, 6(5):515–526, 1998.

[32] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring ISP topologies with Rocketfuel. In *Proc. ACM SIGCOMM*, Pittsburgh, PA, August 2002.

[33] Andrew Schwartz. Enron's missed opportunity: Enron's refusal to build a collaborative market turned bandwidth trading into a disaster. Ucais berkeley roundtable on the international economy, working paper series, UCAIS Berkeley Roundtable on the International Economy, UC Berkeley, November 2003.

[34] Arbinet. `http://www.arbinet.com`.

[35] William Norton. Internet service providers and peering. `http://www.equinix.com/press/whtppr.htm`.

[36] BuySellBandwidth.com. `http://www.buysellbandwidth.com`.

[37] Band-X. `http://www.band-x.com`.

[38] Bob Briscoe. M3I Architecture PtII: Construction. Technical Report 2 PtII, M3I Eu Vth Framework Project IST-1999-11429, February 2002.

[39] Cristian Estan, Aditya Akella, and Suman Banerjee. Achieving Good End-to-End Service Using Bill-Pay. In *Proc. 5th ACM Workshop on Hot Topics in Networks (Hotnets-V)*, Irvine, CA, November 2006.

[40] Cisco Visual Networking Index. `http://newsroom.cisco.com/visualnetworkingindex/`.

[41] Emanuele Giovannetti and Cristiano A. Ristuccia. Estimating market power in the internet backbone. using the ip transit band-x database. *Telecommunications Policy*, May 2005.

[42] Donald F. Ferguson, Christos Nikolaou, Jakka Sairamesh, and Yechiam Yemini. Economic models for allocating resources in computer systems. In *Market Based Control of Distributed Systems. World Scientific*, 1996.

[43] Rahul Jain and Pravin Varaiya. The combinatorial sellers' bid double auction: An asymptotically efficient market mechanism. In *Journal of Economic Theory*, February 2006.

[44] Giorgos Cheliotis and Chris Kenyon. Dynamics of link failure events in network markets. *Netnomics*, 2002.

[45] Jeffrey K. MacKie-Mason and Hal R. Varian. Pricing the internet. 1995.