Feedback Based Routing *

Dapeng Zhu dapengz@cs.stanford.edu
Mark Gritter mgritter@cs.stanford.edu
David R. Cheriton cheriton@cs.stanford.edu

Department of Computer Science
Stanford University
Stanford, CA 94305

ABSTRACT

Inter-domain routing entails coordinating the routers of many administratively independent ISPs to agree on operational routes such that packets are delivered reliably and efficiently. With the world-wide scale of the Internet, it is infeasible to assume that all such ISPs can be fully trusted or that all links and routers remain operational. Links and routers can fail or be compromised by attackers. At the same time, the increasing dependence on Internet-based applications calls for better robustness in the routing system. A corporate mission-critical VPN must remain operational regardless of individual router failures and compromises, and VoIP over such a VPN calls for low-latency fail-over. BGP is the current inter-domain routing protocol in the Internet yet it is susceptible to large-scale failures if a router is compromised and long service interruptions when links or routers fail.

In this paper, we explore what we call *Feedback Based Routing* as an approach to making inter-domain routing resistant to attacks and byzantine failures. Together with source-routing at the autonomous routing domain granularity, we show how the Internet can achieve much higher availability. Our evaluation indicates that feedback-based routing is far more scalable than BGP and can be incrementally deployed on the Internet.

1. INTRODUCTION

Inter-domain routing entails coordinating the routers of many administratively independent ISPs to agree on operational routes such that packets are delivered reliably and efficiently. Today, Border Gateway Protocol version 4 (BGPv4) [32] provides this coordination, allowing routers to form peering sessions and exchange reachability information about edge networks. By advertising an AS path to a prefix, a router is claiming it *can* and *will* forward packets to the specified prefix by the advertised next hop. Each BGP router relies on this information to calculate a route to every prefix. However, if the reachability information is incorrect, packet delivery and thus some portion of the Internet can fail.

With the world-wide scale of the Internet, it is infeasible to assume that all ISPs can be fully trusted or that all links and routers remain operational. Links can fail and routers can fail or be compromised by attackers. Both of these events can cause the reachability information provided to a router through BGP to be inaccurate, which in turn means that packets are forwarded incorrectly and not delivered. The more extensively compromised the reachability information is, the larger the portion of the Internet is rendered inoperable.

The vulnerability of the Internet to incorrect reachability information has been made frightfully evident by a number of incidents. For instance, in 1997, a misconfigured BGP router at a small Virginia ISP [4] advertised that it could provide a good route to every prefix in the Internet, causing other routes to divert traffic to this "blackhole". The operation of the Internet is disrupted for two hours. Researchers have shown [15] that smaller scale "blackhole" incidents are occurring on average 15 times per day. Although it is feasible to program inter-domain routers to reject reachability information that is clearly wrong, such as one might expect from accidental misconfiguration, the real challenge is extending inter-domain routing to defend against incorrect reachability information which is maliciously generated by knowledgeable attackers designed specifically to pass local credibility checks yet still compromise Internet packet delivery. Even just corrupting reachability information so as to significantly increase the delay of packet delivery, either by directing the traffic to suboptimal paths or orchestrating hot spot congestion, is sufficient to disrupt many applications and thus the dependent organizations.

With the deployment of very high-speed routers and the availability of long-distance fiber, the Internet is performance-capable of handling a wide range of important applications, including VoIP and enterprise mission-critical VPN service. However, the inter-domain routing vulnerability precludes the use of the Internet for certain critical distributed applications [20] and places organizations that do depend on the Internet at significant risk. Arguably, the lack of attack-resistance in the wide-area routing may be the key limiting factor in the utility of the Internet at the present and

*This project is supported by the US Defense Advanced Research Projects Agency (DARPA) under contract number MDA972-99-C-0024.

conversely the greatest risk/concern to Internet-dependent organizations.

In this paper, we explore the use of what we call *Feedback Based Routing (FBR)* to make inter-domain routing resistant to attacks and Byzantine failures. We argue this technology is complementary to the work on secure BGP. Together with source-routing at the equivalent of autonomous system level, we show how critical communication can achieve zero fail-over time with high probability. Our evaluation indicates that feedback-based routing is far more scalable than BGP and can be incrementally deployed on the Internet.

The next section describes the feedback-based routing approach we propose. Section 3 evaluates its performance and scalability. We briefly discuss the deployment path of FBR in Section 4. Related work is described in section 5.

2. FEEDBACK BASED ROUTING

2.1 Overview

It is standard engineering practice to design dynamic systems with feedback because they can be both simple and robust, following a straight-forward loop of estimate, try, measure and correct. We are applying this same logic to routing, itself a highly dynamic system. The key idea with feedback-based routing is for a router to monitor packet traffic on its routes and use this as feedback to determine the usability of the routes. A router accepts reachability information from other routers as *hints*. It does not have to trust this information because it selects which path to use based on how each path actually performs. By treating reachability information as hints and having an additional feedback mechanism, inter-domain routing can be made far more attack resistant than the "shared world view" approach at the core of BGP and most routing protocols.

However, the experience with the early ARPANet [12] has shown the instability problem that arises from every router making routing decisions based on dynamic metric. Therefore, FBR restricts the use of feedback to the first hop inter-domain router (*access router*), which reduces the instability of the system.

2.2 Terminology

An *Autonomous Routing Domain (ARD)* is a network whose internal packet forwarding and topology is not visible to the outside world, similar to a BGP autonomous system. (We use this separate term to allow us to define it independent of the connotations of *autonomous system*.) An ISP who does not wish to disclose its peering relationships forms an ARD with other ISPs.

An ARD has business relationship with other ARDs or *edge networks* to forward packets for each other. We call such a business relationship a *link*. The edge networks are represented by prefixes. We call the border routers of an ARD *transit routers*. Each edge network has one *access router* that forwards packet to the upstream providers. The components of the inter-domain routing system are illustrated in Figure 1.

2.3 ARD Topology

An access router has knowledge of the Internet topology at the ARD granularity, as illustrated by Figure 2. It learns the structural information using the ARD Topology Protocol (ATP). ATP is similar to a link state routing protocol, such as OSPF [23]. However, the end points of a link in ATP are autonomous routing domains or edge networks. An ARD is identified by a 32-bit number. An edge network is identified by the prefix and the IP address of its access router.

A link between A and B represents a business contract that A can forward packets to B and B can forward packets to A. The end points of a link are responsible for advertising the link periodically until the business contract ends. Each link also has a Time-To-Live (TTL) value. The business contract is assumed to terminate when the TTL expires. The link advertisements are propagated to all the access routers. If two transit routers peer with each other, they exchange link information. This is similar to the reachability information exchange of two peering BGP routers. Each transit router has a fixed size link repository. When a transit router hears a link advertisement from its peer, the repository is used to check whether it knows about the link. If the link is new, it is advertised to the transit router's peers, except the one from which it hears about this link. The new link is also inserted into this repository. If no more space is available, the new link replaces the oldest link in the repository. The link repository at a transit router is an optimization to prevent the transit router from advertising the same link to the same peer more than once.

An access router receives ARD topology information from the transit routers it peers with. The access router itself does not advertise any link information. Based on the topology information, an access router builds a graph representation of the Internet. The vertices of the graph are ARDs and prefixes. An edge corresponds to a link. In the following discussion, we use the term *link* and *edge* interchangeably.

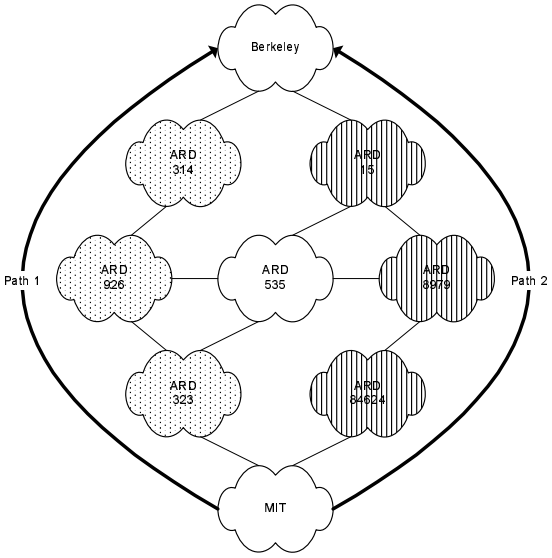

Figure 2: ARD Topology

The protocol is designed to propagate only topology infor-

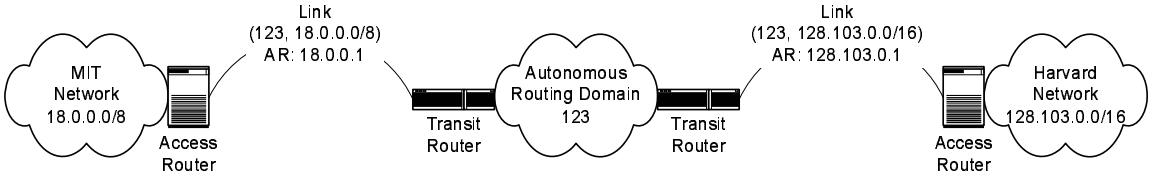

Figure 1: Terminology

mation. The operational state of the link is not propagated. This design reduces the number of routing messages significantly. Another consequence of the design is that there is no "withdraw" message in the system. This prevents a malicious router from removing valid topology information.

2.4 Access Router Algorithm

2.4.1 Source Routing

Each access router specifies the path to a given destination that its packets are to take using "source routing" techniques. That is, it inserts "routing directive" into the packet. The transit routers along the path look at the routing directive inside a packet and determines how the packet travels inside the ARD in order to reach the next hop ARD. The access router of the destination network removes the routing directive. This source routing is carried out at the level of autonomous routing domains. This avoids conflict with ISPs that do not want outsiders to dictate how a packet travels through its network, for technical and/or business reasons. It also reduces the amount of topology information that each access router needs to know. Source routing and its associated security problem is discussed in section 2.5. A couple of mechanisms to implement source routing is discussed in section 4.

2.4.2 Monitor Route Quality

We define a route to be *unusable* if no packet can flow through it or if its latency, loss rate, or throughput does not meet the standard set by the administrator. An access router monitors the following three metrics to determine if a route is usable: the round trip time (RTT), loss rate, and throughput. These metrics are compared with the tolerable range set by the administrator to determine whether a route is usable.

An access router samples the round trip time (RTT) to a destination D by measuring the time between forwarding a TCP SYN packet and receiving the corresponding SYN ACK packet. It also monitors the time between forwarding a DNS name request and getting the response. The access router keeps a running average of the round trip time, similar to TCP [30]. The learning parameter is tunable by the administrator. As Zhang et al. has shown, various methods to predict round trip time based on past experience perform almost identically [38].

To measure the loss rate, the access router monitors the retransmission of TCP SYN packets. If a TCP SYN packet is retransmitted, it means either the SYN packet gets lost, or the SYN ACK gets lost. If p is the loss rate of the route, then the probability that either SYN or SYN ACK gets lost is $1-(1-p)^2$. The access router keeps track of the most recent k SYN ACKs received, as well as the number of SYN packets sent N. N does not include the SYN packets for which there are no corresponding SYN ACKs. We estimate the loss rate of the route p to be:

$$p = 1 - \sqrt{(1 - \frac{N-k}{N})} \quad (1)$$

This method may underestimate the loss rate because we do not count the SYN packets if there is no corresponding SYN ACK. There are a variety of methods to measure the loss rate on a route [8, 10]. It is on our research agenda to study the effect of these estimators.

Using the measured round trip time and loss rate, an access router calculates the expected throughput of a route because throughput is a function of RTT and loss rate [17, 27]. An access router can also measure the bottleneck bandwidth using Packet Pair techniques [14]. However, such a technique adds overhead to the system that is not proportional to the amount of useful traffic. Therefore we choose to use it only to identify the bottleneck in case of failure.

An access router also monitors the quality of routes using probes. It maintains a counter of the number of flows to a prefix. It generates a probe packet to the access router of the prefix whenever the number of new flows since the last probe has reached the *probing threshold*. The probe is sent using all the routes the access router knows about the destination. The same algorithm is used to sample the RTT and loss rate.

This monitoring function can be performed without degrading router performance by using a flow monitoring mechanism such as Cisco Netflow [7], as implemented in hardware by, for instance, the Cisco Catalyst 6000 switch-routers. Using this mechanism, high-speed flows can be monitored and sampled according to specifications in access control lists and the results forwarded to software for processing. Software routers have similar Netflow capabilities. The router software must be programmed to sample at a rate so as not to overload the router processor. Alternatively, the samples can be redirected to a feature line card with its own processor that handles this router quality monitoring.

2.4.3 Probe

A probe packet has the same format as a TCP SYN packet with a random source and destination port number. An access router treats any TCP connection setup request as a probe except for management traffic, which only comes from a small set of addresses. Upon receiving a probe packet, an access router returns a SYN ACK packet. The response packet is sent using the reverse route of the SYN packet. TCP SYN packets are used for the probes because we want

the probe packets to follow the same processing path as regular packets. Because an ARD has total autonomy in handling packets inside its network, it might choose to forward ICMP packets in a different manner than regular TCP packets. For instance, if the ARD chooses to use TCP Switching [22], a TCP SYN packet causes a circuit setup while an ICMP packet does not. Another benefit of using TCP SYN packets as probes is that if an malicious transit router wants to provide misleading feedback, it has to respond to all TCP connection setup requests.

2.4.4 Failure Detection

The access router uses timed-out TCP sessions as a hint of a route failure. The timeout value is specified by the administrator. When a timeout event occurs, an access router compares the time T of the last packet received in that session with the "last known working time"(LKWT) of the route. LKWT is the latest time when the access router receives a packet using that route. If T is later than LKWT, the access router initiates a series of probes to the corresponding access router using the suspicious route. If all probes are unanswered, the access router changes the status of the route to unusable. The last probe times out after a multiple α of the round trip time. Suppose N probes are sent with an interval k, the failure detection time FDT is

$$FDT = k(N-1) + \alpha RTT + T. \quad (2)$$

The timeout may be the result of an idle TCP connection, such as Telnet. It may also be the result of remote host crashing. To reduce the number of extraneous probes, the administrator is likely to use a timeout value of tens of seconds. A faster failure detection can be achieved by always doing background probes, say every 50 milliseconds. If 3 lost probes are used to indicate a route failure with an interval of sending of 50 ms, the failure detection time can be as low as $RTT + 100ms$. However, if such a mechanism is used, the amount of probe traffic will not be proportional to the amount of useful traffic. Therefore there are serious scalability issues. For example, if every access router starts probing `Google.com` at high frequency, the access router for `Google.com` might not be able to handle it.

2.4.5 Route Computation

Based on the topology information it has, an access router computes two routes to every prefix. The two routes are chosen to be as independent as possible. In the ideal case, they are *ARD disjoint*. This is done by a maximum flow like algorithm. The access router treats a link between two ARDs as bi-directional and link between an ARD and a prefix as one directional, since a prefix does not forward packets for other prefixes.

There is enough redundancy on the Internet today that two ARD disjoint routes can usually be found, as shown by an investigation of the Internet topology. In particular, by processing the Route View data [19] from April 1st, 2002, we found that there are 4730 multi-homed edge autonomous systems. There are two AS disjoint routes between 4715 of them.

An access router monitors the usability of the two routes. When a route becomes unusable, an access router tries to identify the culprit. If no packet is getting through, the access router uses a traceroute-like process to identify the last ARD on the path that is still working properly. It assumes that the problem is caused by both the link and the next hop ARD. If the throughput is too low, an access router uses a combination of packet pair [14] and traceroute to detect the bottleneck. Once the cause of the problem is identified, an access router computes a route that is disjoint from the usable route and the problematic ARD/link. If the problem can not be identified, the access router randomly excludes a set of ARDs from the unusable route and computes a new route that is disjoint from the working route. After a new route is computed, probes are sent to determine its usability.

There is enough time for the identification of failed component and route computation to take place after a failure. If we assume the two ARD disjoint routes are failure independent and suppose the probability of failure for one route is π, then the probability of both routes failing simultaneously is π^2. Some researchers report [21] that the probability of a route failure is about 0.84% on the current Internet. The probability of both ARD disjoint routes failing simultaneously is 7×10^{-5}. Other researchers report [33] that routes to popular destinations usually have fail-over times of dozens of days.

2.4.6 Path Selection

An access router maintains a table of TCP sessions that pass through it. Upon receiving an outgoing TCP packet, the source and destination addresses as well as source and destination ports are used to match it to a session. If it belongs to a session that is initiated by a remote host, it is sent by reversing the route that the incoming packets of the same session take. If the packet belongs to an existing session, its current route is used unless it is not usable any more. In that case, a usable route is selected. If the packet does not belong to any existing session, the route with the shortest round trip time is used, and a new session record is inserted into the table. The session records are removed from the table when a FIN packet is received, or if a timeout value has been reached since the last packet of that session.

Feedback Based Routing system, or any other inter-domain routing system, does not try to provide optimal performance to any pair of source and destination. FBR only provides an adequate level of service as defined by the administrator. The first reason for not chasing the best path is that there is no single definition of optimal. Some applications want short RTT, while others want low loss rate. The second reason is that there may be thousands of routes between the source and destination. Unless one has perfect information about the behavior of every router, link, and end host, there is no way to predict which route is going to have the best performance. Savage et al. [35] showed that majority of the paths chosen by BGP are not optimal. The goal of FBR is to provide a reasonable level of performance and to prevent bad events from happening.

The dynamic path selection is not expected to degrade end-to-end performance from path oscillation or self-synchronization because the randomness in the path computation algorithms and the diverse range of tolerable threshold set by the network administrators. It is difficult to quantify the result

using a small scale testbed. For a large-scale, realistic example, we looked at deployments of RouteScience's PathControl [34] product. It uses a feedback-based algorithm to select an upstream ISP to forward a packet. Based on customer experience, RouteScience has observed that any potential for flapping in the Internet due to routing decisions is countered by appropriate policy, cost and threshold controls. Overall, the company has not seen any negative impact on end-to-end performance because of oscillations [3]. RouteScience counts Google, Qualcomm, and RealNetworks among its customers [34].

2.4.7 Simultaneous Transmission

If an access router switches to an alternative route once it detects that the current route has become unusable, there is a time window in which packet forwarding stops or suffers significantly. We call this time window the *fail-over time*. The route switching is called *explicit fail-over*. The fail-over time can be reduced to zero using a proactive approach: "simultaneous transmission". In this scheme, an access router duplicates the packets going to certain prefixes and/or using certain protocols, and sends them using different routes. The administrator is responsible for configuring the class of packets that get duplicated.

As we have shown in section 2.4.5, the probability of the two ARD disjoint routes failing at the same time is 7×10^{-5}, , which means there is a 99.993% probability that one of the routes is working. Such availability is very close to the five-nine requirement of many mission critical applications. The availability can be further improved with using three ARD disjoint routes.

A variety of mechanisms can be used to deal with the duplicate packets resulting from the simultaneous transmission. For instance, the virtual link might be part of a Virtual Private Network (VPN) using IPsec. As a result, the VPN gateways drop the duplicate packets as specified in RFC 2085 [26].

At first glance, transmitting each packet twice over the wide-area appears impractical because it doubles the packet traffic overhead. However, if there is capacity to deal with the packet traffic on a separate path once the other path has failed, the capacity must be there and available during normal operation so it can just as well be used for duplicate traffic and eliminate the fail-over time. Moreover, researchers have shown that the utilization of Internet backbone links is 10% to 15% [25]. With the ready availability of fiber and failing cost of high speed routers relative to customer value, the link utilization is getting even lower. Therefore it is economically feasible to provide these duplicate capacity for important Internet customers.

2.5 Source Routing

Source routing allows each access router to cause a packet to take a new path that it selected without it having to communicate this change to every router along the path using the control plane, namely the routing system. Communicating each such path change would require the routers to communicate them at a rate matching the path changes. Even ignoring the current reality that normal BGP protocol processing is a substantial burden for each router, this approach places no real limit on the rate at which a (compromised) access router could try to propagate path selection updates. Using source routing, the overhead is limited to a percentage of what is considered useful traffic.

Source routing is recognized as providing the capability to attack a network, by spoofing the true source of a packet. The attack works in the following way. The attacker sends a source routed packet with a trusted source address to the victim. The attacker also inserts a routing tag in the packet indicating the packet is routed through the attacker. An innocent victim then responds by sending a source routed packet with the attacker as an intermediate routing hop.

In FBR, source routing is restricted to the ARD level. End hosts do not see source routed packets. A transit router drops any packet whose origin is not one of its customers. The access router on the receiving side should not only filter on source address; it should also filter on the upstream ARD of the source.

Given these measures, an attacker can still launch an attack as described above if she manages to compromise some transit routers inside an ARD. In such an attack, the attacker generates packets with the trusted source address and first hop ARD. By injecting itself into the middle of the route, the response packets go through the compromised ARD. For example, if the attacker compromised ARD 2 and if she knows the source address `10.0.0.10` and first hop ARD 1 is trusted by the victim, she generates a packet with source address `10.0.0.10` and route `[1 2 3]`. The victim responds by sending a packet back using route `[3 2 1]`. The attacker then captures the packet when it is in ARD 2.

However, such a compromise requires significantly more resource than it requires to launch the attack from a single computer. Furthermore, in the current BGP world without any source routing support, a compromised BGP router can inject bogus routing updates into the global routing table and spoof a connection with the victim. Therefore, if source spoofing is a real concern, application level authentication should be used, instead of source address based filtering.

3. EVALUATION

3.1 Testbed

The Feedback Based Routing System described above was implemented on Linux as 16000 lines of C++ code and another 24000 lines of support libraries. We used this implementation in a testbed depicted in Figure 3 to evaluate the effectiveness of FBR with respect to failures and attacks. In the testbed, there is one transit router per ARD. The peering relationships among them are statically configured. Although it is a very simple topology, the result is representative because the algorithms that deal with failure and attacks do not depend on the network topology.

3.1.1 Traffic Generator

The client in prefix 1 initiates a series of TCP sessions with the server in prefix 2. The flow length follows a uniform distribution from 1 KB to 100 KB. The session creation time also follows a uniform distribution from 100 milliseconds and 1 second.

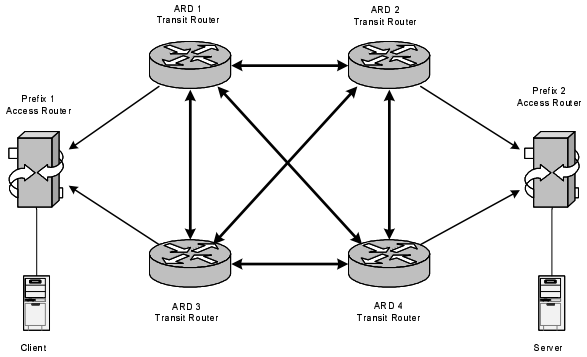

Figure 3: Testbed Configuration

3.1.2 Data Collection

We are interested in how the connectivity between the two prefixes is affected by failures and attacks. To measure that, the client uses ICMP echo requests to determine the connectivity to the server. A request is sent every 300 milliseconds. If a reply is not received within 1 second, the connectivity between the client and server at the sending time is marked as 0.

During the evaluation process, we inject various events to the system. These events include router state changes, link state changes, and bogus topology advertisements. When these events happen, the client is notified through a RPC mechanism. The client records the time of these events so that we can correlate the events with the connectivity.

3.2 Attack Resistance

There are two types of attacks that can be launched against the routing system. The first type of attack involves the propagation of bogus topology information with the goal of fooling some routers into using a link which either does not exist, or does not function well. FBR is resilient to such attacks because an access router does not use a route until it has evaluated its quality so a non-working or poor performing route is rejected.

To verify this claim, we conducted the following test. A

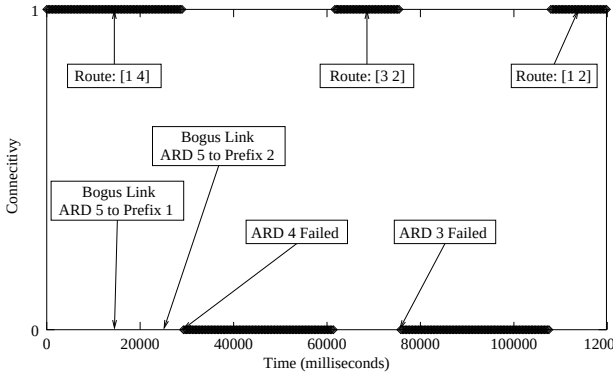

Figure 4: Attack Resistance

fictitious ARD, ARD 5 is created. It has two non-existent links to Prefix 1 and Prefix 2. In the real Internet, the two access routers would discard these two bogus links outright, because they should be statically configured for the peering relationships with the upstream providers. But in this test, they are modified to store the bogus links in their topology database. When they want to send a packet to ARD 5, the packets are silently dropped.

As illustrated in Figure 4, feedback-based routing is robust against this bogus topology information because it avoids selecting this incorrect route even when a valid link/ARD fails. The x-axis is the time in milliseconds, and the y-axis is the connectivity between the two prefixes.

The access router does have to compute and probe a new route after a route failure. If the majority of topology information an access router has is bogus, the new route is unlikely to work. If only one route has failed, the access router continues to forward packets since there is another working route. However, the situation becomes very dangerous when both routes fail. It takes a long time for the access router to find another working route in the worst case. However the window of vulnerability is very small. As we have shown in section 2.4.5, the probability of both routes failing is only 7×10^{-5}, which translates to 37 minutes every year. However, care has to be taken when providing an access router topology information when it boots up. If its initial database is filled with bogus information, it is unlikely to work at all.

The second type of attack involves the propagation of bogus topology information as well as providing misleading feedback to the access routers. For example, in order to hijack the connections to `Google.com`, an attacker has to compromise a transit router; advertise a bogus link between the transit router and `Google.com`; and answer TCP SYN packets to `Google.com` with SYN ACKs. If the malicious router does this quickly enough, some access routers will be fooled by this attack and choose a route that includes the compromised transit router to reach `Google.com`.

However, it takes bandwidth to provide bogus feedback. Let B_{bad} represent the bandwidth at an attacker's disposal and B_{good} represent the bandwidth of the legitimate network it is trying to impersonate. We assume these bandwidths are symmetric. The attacker responds to all SYN packets with SYN ACK packets. It also receives an ACK packet in response to the SYN ACK. Therefore, the attacker receives at least 80 bytes and sends 40 bytes. The incoming bandwidth is the dominating factor for the attacker. There might be more packets depending on the attacker's behavior. However, for the simplicity of our model, we do not count any other packets. For the target, it has to transfer some data to the client and bandwidth used by the transfer is the dominant factor for the victim. Suppose the length of the transfer is L bytes. We further assume that there is a linear relationship between the round trip time and the bandwidth utilization and we assume that everyone is at equal distance D from the attacker and the target.

$$RTT = k \times BandwidthUtilization + D \qquad (3)$$

In order for the attacker to fool p percent of the access routers to send packets to the attacker out of a total of S sessions, the RTT to the attacker should be smaller than

the RTT to the real destination. Therefore the following inequality holds:

$$k\frac{80pS}{B_{bad}} + D \leq k\frac{L(1-p)S}{B_{good}} + D \quad (4)$$

We have:

$$B_{bad} \geq \frac{p}{1-p}\frac{80}{L}B_{good} \quad (5)$$

The inequality shows that the attacker must have bandwidth proportional to the combined bandwidth of victims to successfully deceive a majority of the access routers. According to Thompson et al. [37], the average length of a TCP flow is 7KB. If $L = 7168$, then the attacker has to have 10% of the bandwidth of the target to fool 90% of the access routers. In reality, the ACK packet in response to the SYN ACK often contains payload, such as a HTTP request. If we assume a 500 bytes packet, an attacker has to have 68% of the bandwidth of the target network to fool 90% of the access routers. This simple calculation show that in order to effectively shut down the Internet, an attacker has to compromise a significant portion of transit routers on the Internet. However, if the target of the attack is an important part of the Internet, such as the root DNS name servers, the result is catastrophic. The concern is addressed by an approach similar to [16].

In conclusion, the Feedback Based Routing system is robust in the face of attack that only generates bogus topology information. The access routers are vulnerable to Denial-of-Service attacks that generate a massive amount of bogus topology information. But the window of vulnerability is small. FBR is vulnerable to attacks when attackers are also able to generate misleading feedback. However, such attacks require the attacker to have bandwidth that is proportional to the combined bandwidth of his victims.

3.3 Scalability

In this section, we show that FBR is scalable from the standpoint of resource demand on the control planes of transit and access routers as a function of Internet size.

3.3.1 Access Router Scalability

An access router has to calculate routes to every destination prefix on the Internet and has to monitor the performance of packet forwarding. To determine this route calculation overhead, we processed the April 1st, 2002 BGP routing table dumps from the Route View Project [19]. We found that there are 13041 autonomous systems, 124330 prefixes, and 155706 links. Among the links, 28851 are inter-AS links. While these numbers might not exactly describe the topology of the Internet, they do give us a sense of the amount of computation and storage we are dealing with. In our implementation, it takes less than 2 seconds to compute a shortest path to every prefix on a Duron 950Mhz processor. Although it takes about 30 minutes to find an alternative route for every prefix, the bootstrapping time is very short, because packet forwarding starts as soon as there is one route to a prefix.

Figure 5 shows the growth of memory and CPU power on typical desktop machines [5], and the number of prefixes on the Internet [1] from 1993 to 1999. All the values are normalized to 1 at 1993. Clearly, memory and CPU speed are growing much faster than the number of prefixes on the Internet. This graph indicates that an ISP does not have to upgrade an access router in processing and memory capacity unreasonably relative to the growth of the Internet, although some upgrade is required as the Internet grows to maintain constant service level. The two disjoint paths used in FBR also means that even if an access router falls behind in route calculation, it (and the rest of the Internet) continues to function because the probability of the both routes simultaneously failing is very low.

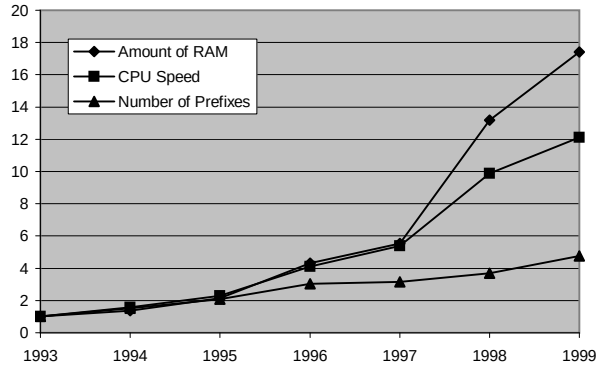

Figure 5: Growth of memory, CPU, Number of Prefixes

FBR also has the property that if an edge network fails to upgrade an access router, the only people who suffer are those whose traffic flows through this access router. The rest of the Internet is not affected. In contrast, BGP requires the cooperation of every BGP router. If some of the routers become overwhelmed with routing updates and fall behind in processing the updates, the entire routing system takes longer to converge. Therefore, no matter how fast an ISP upgrades its BGP routers, the service quality it provides is dependent on the actions of other ISPs. In FBR, if an ISP aggressively upgrades its access routers, its customers directly benefit from the better route selection and faster computation of additional paths.

3.3.2 Transit Router Scalability

A transit router is relatively simple. It does not do any route calculation, and only needs to keep track of recently heard links and send new link advertisements to its peers. The cost of doing this is low compared to the computing power of current desktop computers and substantially less than BGP. Moreover, even if for some reason, a transit router falls behind in propagating link advertisement, it is unlikely to affect end-to-end communication.

3.3.3 Attack Scalability

Another component of scalability is the amount of damage that an attacker can inflict on the Internet relative to the effort expended, as a function of Internet scale. We argue that a routing system is not scalable if the amount of damage an attacker can inflict for a given amount of effort grows more or less proportional to the size of the Internet. This is because attacking becomes increasingly attractive as the Internet grows, and defenders have to expend increasing

amounts of effort to protect the Internet. By this definition, the current BGP protocol is not scalable because the effort to compromise one BGP router can pay off in the ability to disrupt almost the entire Internet. In contrast, FBR is scalable because compromising a router only compromises the traffic going through this router and a number of victims proportional to the bandwidth this router has access to.

3.3.4 Packet Overhead

The FBR packet overhead consists of additional probe packets as well as overhead for source routing of packets. The percentage of the overhead out of all useful traffic does not grow when the size of the Internet grows, unless there is significant change in the topology such that the number of ARDs between two prefixes gets bigger. The overhead caused by fault isolation and failure handling is negligible, because in the real Internet, routes to popular destinations are very stable, usually with a fail-over time of dozens of days [33].

Let α be the probing threshold. Let P be the average number of packets in a TCP session from the session initiator. The percentage of the probe packets out of all outgoing packets is $\frac{2}{2+\alpha P}$. According to [37] and [18], the average number of packets in a TCP session is 16-20. Even if the packets from the initiator are only the initial SYN and the subsequent ACKs, the number of packets from the initiator is about 6, since most TCP implementations are derived from the BSD implementation, which sends an ACK for every two data packets received. Therefore, in the expected case, if TCP is used as the probing protocol, the percentage of probe packets out of all outgoing packets is $\frac{1}{1+3\alpha}$. With $\alpha = 50$, that is 0.6% out of the number outgoing packets.

The percentage of probe response packets out of all incoming packets is $\frac{2}{2+\alpha Q}$, where Q is the number of incoming packets in a TCP session. Using the above analysis, when all the incoming packets are ACKs, Q is 6 on average. Therefore, in the expected case, the percentage of probe packets out of all incoming packets is $\frac{1}{1+3\alpha}$. It is also worth noting that the probe packets and their responses are all minimum sized packets. Therefore the bandwidth overhead is much smaller than $\frac{1}{1+3\alpha}$.

Our analysis of BGP routing tables shows that there are less than 5 autonomous systems between most prefixes (even for the backup route). Therefore in most cases, less than 24 bytes are needed for most packets. We are still going to concentrate on TCP, which carries the majority of the Internet traffic. Assuming there is one ACK for two data packets and MTU discovery is used, TCP needs to send 3040 bytes of data to deliver 2920 bytes of payload. After inserting source routing information into the packets, TCP can deliver 2812 bytes of payload by sending 3064 bytes of traffic. The bandwidth utilization is reduced from 96.1% to 91.8%. Given the utilization of most backbone links is between 10% to 15% [25], the overhead is not a concern.

3.4 Fail-over

3.4.1 Explicit Fail-over

We evaluated the effectiveness of our fail-over algorithms using the a test in which various transit routers fail. Figure 6 shows the result of the experiment when the access routers use explicit fail-over. A TCP session timeout value of 30

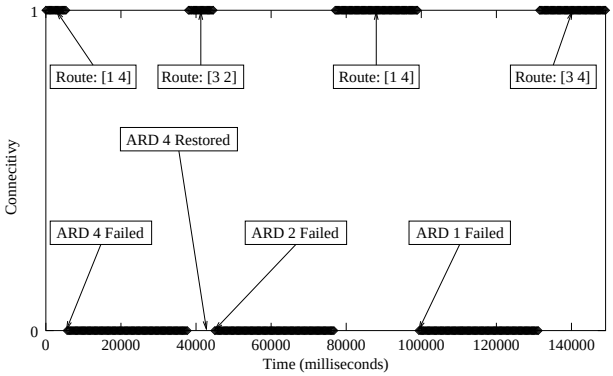

Figure 6: Explicit Fail-over

seconds is used. The access router sends 3 probes with 1 second interval after the timeout. It determines the route has failed after the probes are unanswered for $30+2+2RTT$ seconds. As we show in Figure 6, the connectivity between the client and the server is disrupted for about 32 seconds after each failure. The reason for a timeout value as long as 30 seconds is to avoid treating host crash as a route failure and sending unnecessary probes.

With the explicit fail-over mechanism, FBR is more scalable than BGP when dealing with failures. The only component of the FBR failure detection that depends on the size of the Internet is the expected round trip time. It is unlikely to dominate the failure detection time. In contrast, Labovitz et al. showed that BGP fail-over time is $30\theta(n)$ seconds, where n is the number of BGP routers on the *longest* path between the two hosts [13].

As we have discussed earlier, the failure detection time can be reduced even further if the access router uses background probes. However, instead of using a technique that generates an amount of traffic that is not proportional to the useful traffic, we focus on the simultaneous transmission technique, whose overhead is well understood.

3.4.2 Simultaneous Transmission

The effectiveness of the simultaneous transmission mechanism is shown in a test whose result is displayed in Figure 7. In this test, the access routers used an algorithm similar to IPsec's Replay Prevention [26] to suppress the duplicate packets. A series of router failures are injected into the testbed, yet there was no effect on end-to-end connectivity.

We have been assuming that each ARD is failure independent. Therefore the two ARD disjoint routes that an access router has are failure independent. This assumption is invalid when there are hidden dependencies among autonomous routing domains. For instance, two ARDs might be operated by the same company. If the company files for Chapter 7 bankruptcy, both ARDs will go down. It is also possible that disjoint routes go through the same physical wire, or the same wiring closet. It would be valuable to have

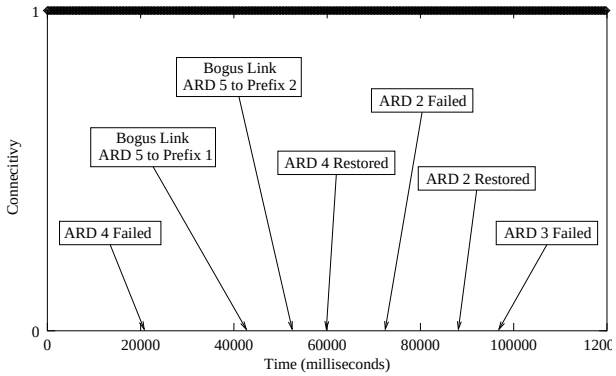

Figure 7: Zero Down Time

dependence information of this nature provided as part of the ARD topology. This is an area for future research.

Another problem with simultaneous transmission is that it might shield the end hosts from congestion signals on one route if unmodified TCP is used. This is because traditional TCP uses packet drops as congestion signals. One approach is to use Explicit Congestion Notification (ECN) [31] to relay the congestion signals to the end hosts explicitly. However, in reality, given the WAN capacity in most environments, it seems easier to just provision sufficiently for simultaneous transmission for customers that require this capability.

4. DEPLOYMENT

To deploy FBR, the Internet has to incrementally transition from the BGP model of a shared "world view" of Internet topology to that of individual access routers creating their own view of the Internet from feedback in conjunction with reachability information. Our approach is for ISPs to incrementally enable support for source routing and form peering relationships with other ISPs who have enabled source routing.

4.1 IP Loose Source Routing Option

FBR can be deployed with multiple source routing mechanisms, although a single suitable protocol would be preferred. IP Loose Source Routing Option [29] is an established standard. However, in many current routers, any IP option typically causes the packet to be transferred to the slow path through the router and may cause the packet to be dropped. Thus, deployment using LSRO would involve substantial hardware upgrade to existing WAN routers. Border routers and every router on a source-routed path must be upgraded so that packets with LSRO not addressed to them are forwarded at full wire-speed.

4.2 Wide-area Relay Addressing Protocol

A second approach is to use a protocol designed to support loose source routing over the WAN at high speeds: Wide-area Relay Addressing Protocol (WRAP). WRAP is a shim protocol that effectively inserts a source routing header between the IP layer and higher-layer protocol header, such as TCP, as illustrated in Figure 8. With WRAP, only the transit routers and access routers have to be upgraded to handle WRAP. Other routers see WRAP packets as normal option-free IPv4 packets, unlike with LSRO.

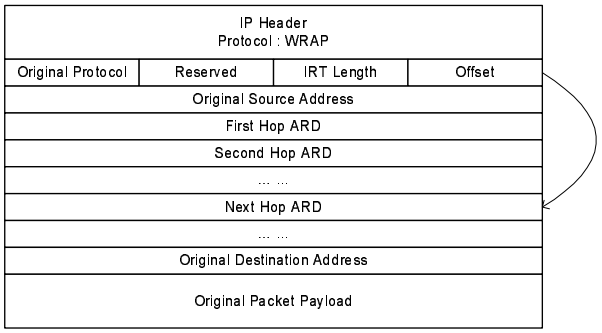

Figure 8: WRAP Packet Format

A WRAP packet contains an Internet Relay Token (IRT) which can be just an IP address or may be some identifier such as ARD identifier recognized by the receiving router. In either case, it indicates to which ARD the packet should be forwarded.

Figure 9 illustrates how WRAP works, showing the WRAP packet contents on each peering link. When an outgoing IP packet arrives at an access router, the packet is rewritten into a WRAP packet. Its IP source address remains the same, the destination address is changed to the address of the peering transit router. The real destination address is saved together with the list of ARDs the packet should go through.

When the WRAP packet arrives at a transit router, it is rewritten in the following way: the source address is replaced by the address of the current router, the destination address is replaced by the transit router of the next hop ARD (or the real destination, if this is the last ARD). The transit router determines the ARD from which the packet comes based on the incoming interface. It saves the information in the packet as well. The packet is then sent based on the intra-domain routing mechanism.

When the WRAP packet arrives at the destination prefix' access router, it is rewritten into a regular IP packet by removing the WRAP header and replacing the source address by the real source address.

4.3 Initial Deployment

In the initial phase of deployment, the legacy portion of the Internet that does not support source routing is treated like an autonomous routing domain with ARD number 0. Every FBR enabled ISP peers with it in the initial phase, as illustrated in Figure 10. Each ISP who joins the FBR camp is assigned an ARD number that is actually a "virtual" IP address. This address is reachable from the legacy Internet. An access routers can learn the existence of source routing enabled ISPs through a variety of methods, such as routing registries, or DNS. The access routers run the same FBR algorithm with a slight modification. If ARD 0 is on the route, it is not part of the routing information inserted into the packet. Upon receiving a packet, a transit router checks whether it has a direct peering relationship with the next hop ARD. If it has, packet forwarding proceeds normally.

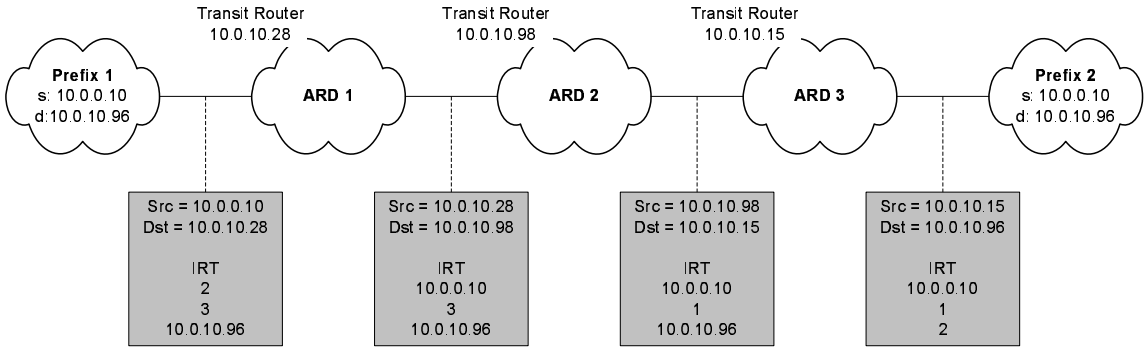

Figure 9: WRAP Example

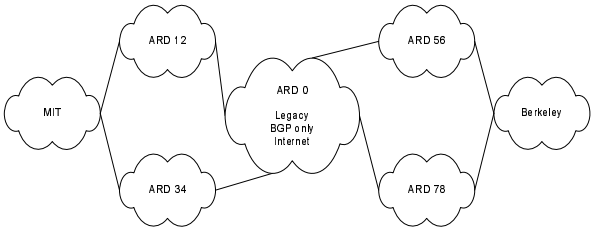

Figure 10: Initial Phase of Deployment

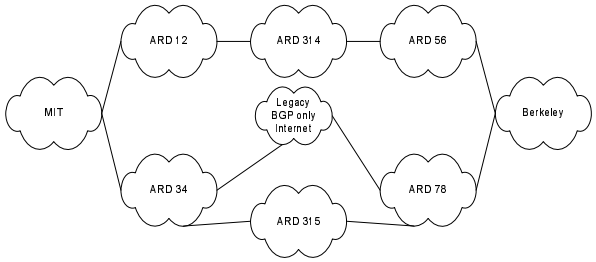

Figure 11: Second Phase of Deployment: Some ISPs stop peering with the portion of Internet that does not support source routing

Otherwise, it uses the next hop ARD number as the destination address and the packet is sent across the legacy Internet. For example, assuming the topology of Figure 10, if a host in MIT wants to send a packet to Berkeley using route `[12 0 78]`, the transit router of ARD 12 discovers the next hop ARD is 78. Because it does not have a direct peering relationship with ARD 78, it converts 78 to an IP address `78.0.0.0` and sends it out using the legacy Internet. It is worth noting the four ARDs depicted in Figure 10 do not have to be ISPs. ARD 12 and 34 can be degenerated into a single device that have two IP addresses.

The benefit of the initial deployment is the decrease in failover time between the two edge networks depicted in Figure 10. It is likely the route between ARD 12 and ARD 56 is different from the route between 34 and 78. Therefore the probability of both routes fail simultaneously is lower. One of our future research directions is to quantify the benefit. Such a deployment scenario is similar to what some current route control systems do [3] for certain customers.

4.4 Second Phase Deployment

Figure 11 illustrates the phase when more and more ISPs start supporting source routing. The portion of the Internet that does not support source routing has shrunk significantly. Some routes are free from the legacy Internet. These routes are not affected by the long BGP convergence time, thus motivating ISPs to stop peering with the portion that does not support source routing.

5. RELATED WORK

Labovitz et al. [13] show that the amount of routing messages a BGP router has to handle is growing exponentially as a result of the rapid growth in advertised prefixes [9]. This study also shows that BGP takes 3 minutes on average to converge after a routing failure and up to 30 minutes of convergence time has also been observed. The long fail over time is caused by certain BGP design decisions, the distributed decision making process inherent in the routing model, and the sheer size of the Internet. In general, Labovitz et al. showed that the time to regain connectivity between two end hosts is $30\theta(n)$ seconds, where n is the number of BGP routers on the longest path between the two hosts [13].

Secure BGP [11] addresses the routing security issues by adding signatures to route advertisements, so that reachability information can not be forged. Although this approach improves the security of the routing infrastructure by protecting against spoofing of routing messages, it does not handle the case of a compromised or misconfigured router. In this sense, secure BGP provides authentication whereas feedback-based routing complements it by providing robustness.

Routing registries have been proposed as a separate trusted source for routing information, allowing routers to filter out bogus updates from peers. However, in practice, information in the routing registries is often outdated and inaccurate. Mahajan et al. showed that 31% of the origins in the Internet Routing Registries (IRR) are inconsistent [15]. Filtering based on inaccurate information can lead to routing failures. The fundamental problem of this approach is that information is derived from unverifiable sources, which is what the feedback-based approach solves.

Much earlier work on "byzantine robustness" [28] address

some of the same security problems we do, but there are significant differences. First, transit routers are not required to know the network topology. This optimization makes the transit routers' control plane resource requirements almost independent of the growth of the Internet. Second, we use TCP SYN and SYN ACK packets for end-to-end performance measurement by observing that most network traffic is TCP, rather than inventing a separate network layer ACK protocol. Finally, all expensive route computation are moved out of the critical path of the algorithm, performed in parallel to packet forwarding.

The Nimrod Routing Architecture [6] is a proposal to build a scalable Internet wide link state based routing system. It is similar in spirit to the ARD topology protocol. Nimrod also proposed the use of source routing to forward packets. However, the path selection algorithm is not specified by the RFC.

The Resilient Overlay Network [2] takes the approach of building an overlay network that gets around routing failures and improve route performance. We choose to address the robustness problem at the routing layer.

There are several companies, such as RouteScience [34], SockEye Networks [36] and netVmg [24], which are building products that can be used in an edge network to dynamically choose better paths, based on the end-to-end performance measurement. These products provide performance enhancement and resiliency to the failure of upstream providers. Works of these companies show that route quality monitoring can be done fast; dynamic route selection does not cause harmful oscillations; and there is customer demand for better route selection than what is provided by BGP.

6. CONCLUDING REMARKS

Feedback-based routing (FBR) represents a significant departure from conventional routing approaches. Rather than trusting the reachability information received from its peers (and forwarded to them by their peers), a router treats this reachability information as *hints* and verifies the utility of the advertised paths using feedback. The feedback comes from monitoring the performance of normal customer traffic over paths it selects as well as generating a complementary amount of probe traffic, particularly for paths that it is considering but has not yet been selected for normal packet traffic. Using this feedback and the ability to specify its path selection using source routing techniques, a router starts to use a new path without waiting for other routers to converge to the same "world model" of the Internet topology, as required with BGP.

We have shown that FBR is scalable, and in particular more scalable than BGP because:

- The damage an attacker can inflict is bounded in proportion to the number of routers and links it can compromise, whereas in BGP it is not.
- The service level provided by an FBR router to its directly attached customers is primarily determined by its own capacity and connectivity and largely independent of the scale of the Internet, whereas BGP convergence makes the performance dependent on the routing capacity of *all* routers.
- The resources required by an FBR router remain effectively constant as the Internet grows, given the historical improvement trends in processor and memory costs.

In particular, we show that FBR rejects incorrect reachability information that would have otherwise caused packet traffic to be mis-directed in a conventional routing system based on BGP or any similar "shared world view" system. We further show that an attacker requires bandwidth proportional to the combined bandwidth of the victims to generate sufficiently misleading feedback to disrupt packet traffic, so the level of disruption is limited by the number of routers that an attacker can compromise.

Limiting routing decision making to individual access routers means that the investment an ISP makes in such an access router is justified by being directly beneficial to its customers. An access router is not dependent on the speed with which other ISPs routers converge in their routing calculations. We note that FBR achieves scalability by giving control of route selection to the edge networks and freeing the core routers from maintaining a complete model of the Internet. This is in spirit of the Internet end-to-end principle, allowing the ends to implement the desired service.

FBR depends on a loose source routing mechanism. While standards solutions such as IP Loose Source Routing Option could be employed, we use a simple shim protocol called WRAP that only requires FBR-enabled routers to be upgraded, allowing smooth incremental deployment.

We showed that AS disjoint paths exist between most pairs of multi-homed autonomous systems on the current Internet. Using route quality monitoring, an access router switches between the two disjoint paths it has on failure within time largely determined by local network administrator policy rather than proportional to route computation and worldwide convergence, as with BGP. The fail-over time can be effectively reduced to zero by using simultaneous transmission. Thus, with FBR, the availability of communication to a destination is independent of the scale of the Internet.

In summary, the conventional "shared world model" approach to routing, the basis for all standard routing protocols including BGP, is not scalable to inter-domain routing, especially with the significant risk of malicious compromise of one or more routers. Thus, inter-domain routing calls for a new approach, not just incremental refinements on existing protocols. Our results indicate that Feedback Based Routing is a promising candidate to provide scalable attack-resistant inter-domain routing. As part of our future work, we plan to enable the incremental deployment we have outlined in this paper.

7. REFERENCES

[1] J. Aldridge. James aldridge's routing page. http://www.mcvax.org/ jhma/routing/.

[2] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proceedings of ACM SOSP*, October 2001.

[3] O. Baldonado, F. Siddiqi, and M. Karam. Private communication with routescience engineers.

[4] R. Barrett, S. Haar, and R. Whitestone. Routing snafu causes internet outage. *Interactive Week*, April 25 1997.

[5] E. Berndt, E. Dulberger, and N. Rappaport. Price and quality of desktop and mobile personal computers: A quarter century of history. http://www.nber.org/ confer/2000/si2000/berndt.pdf.

[6] I. Castineyra, N. Chiappa, and M. Steenstrup. The nimrod routing architecture. RFC 1992, August 1996.

[7] Cisco. Cisco ios technologes netflow. http://www.cisco.com/warp/732/netflow/, July, 1995.

[8] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. In *Proceedings of ACM SIGCOMM*, 2000.

[9] G. Huston. Commentary on inter-domain routing in the internet. RFC 3221, December 2001.

[10] V. Jacobson. Congestion avaoidance and control. In *Proceedings of ACM SIGCOMM*, 1988.

[11] S. Kent, C. Lynn, and K. Seo. Secure border gateway protocol. *IEEE J. Select. Areas Commun.*, 18(4), April 2000.

[12] A. Khanna and J. Zinky. The revised arpanet routing metric. In *Proceedings of SIGCOMM*, pages 45–56, September 1989.

[13] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed internet routing convergence. In *Proceedings of ACM SIGCOMM*, pages 175–187, 2000.

[14] K. Lai and M. Baker. Nettimer: A tool for measuring bottleneck link bandwidth. In *Proceedings of USITS*, March 2001.

[15] R. Mahajan, D. Wetherall, and T. Anderson. Understanding bgp misconfiguration. In *Proceedings of ACM SIGCOMM*, 2002.

[16] D. Massey, L. Wang, X. Zhao, D. Pei, R. Bush, A. Mankin, F. Wu, and L. Zhang. Protecting the bgp routes to top level dns servers. http://www.nanog.org/mtg-0206/bush.html.

[17] M. Mathis, J. Semke, and J. Mahdavi. The macroscopic behavior of the tcp congestion avoidance algorithm. *Computer Communication Review*, 27(3), July 1997.

[18] S. McCreary and k. claffy. Trends in wide area ip traffic patterns - a view from ames internet exchange. ITC Specialist Seminar, Monterey, CA, September 2000.

[19] D. Meyer. University of oregon route views project. http://antc.uoregon.edu/route-views/.

[20] P. Molinero-Fernadez, N. McKeown, and H. Zhang. Is ip going to take over the world (of communications)? In *Proceedings of HotNets-I*, October 2002.

[21] P. Molinero-Ferna'ndez and N. McKeown. Study of routing behavior through traffic analysis and traceroute measurements. http://klamath.stanford.edu/tools/Traceroute/.

[22] P. Molinero-Ferna'ndez and N. McKeown. Tcp switching: Exposing circuits to ip. *IEEE Micro*, January 2002.

[23] J. Moy. Ospf version 2. RFC 1583, March 1994.

[24] netVmg. netvmg. http://www.netvmg.com.

[25] A. Odlyzko. The internet and other networks: Utilization rates and their implications. citeseer.nj.nec.com/odlyzko98internet.html, 21, 1999.

[26] M. Oehler and R. Glenn. Hmac-md5 ip authentication with replay prevention. RFC 2085, Feburary 1997.

[27] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modelling tcp throughput: A simple model and its empirical validation. In *Proceedings of ACM SIGCOMM*, 1998.

[28] R. Perlman. Network layer protocols with byzantine robustness. Technical Report 429, MIT Laboratory for Computer Science, 1988.

[29] J. Postel. Internet protocol. RFC 791, September 1981.

[30] J. Postel. Transmission control protocol. RFC 793, September 1981.

[31] K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ecn) to ip. RFC 3168, September 2001.

[32] Y. Rekhter and T. Li. A border gateway protocol (bgp-4). RFC 1771, March 1995.

[33] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang. Bgp routing stability of popular destinations. In *ACM SIGCOMM IMW (Internet Measurement Workshop) 2002*, 2002.

[34] RouteScience. Routescience. http://www.routescience.com.

[35] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson. The end-to-end effects of internet path selection. In *Proceedings of SIGCOMM*, pages 289–299, September 1999.

[36] SockEye. Sockeye networks. http://www.sockeye.com.

[37] K. Thompson, G. Miller, and R. Wilder. Wide-area internet traffic patterns and characteristics. *IEEE Network*, November/December 1997.

[38] Y. Zhang, N. Du, V. Paxson, and S. Shenker. the constancy of internet path properties, 2001.