# A Framework for Resilient Internet Routing Protocols

**Dan Pei and Lixia Zhang, UCLA**
**Dan Massey, USC/ISI**

## Abstract

At a fundamental level, all Internet-based applications rely on a dependable packet delivery service provided by the Internet routing infrastructure. However, the Internet is a large-scale, complex, loosely coupled distributed system made of many imperfect components. Faults of varying scale and severity occur from time to time. In this article we survey the research efforts over the years aimed at enhancing the dependability of the routing infrastructure. To provide a comprehensive overview of the various efforts, we first introduce a threat model based on known threats, then sketch out a defense framework, and put each of the existing efforts at appropriate places in the framework based on the faults and attacks against which it can defend. Our analysis shows that although individual defense mechanisms may effectively guard against specific faults, no single fence can counter all faults. Thus, a resilient Internet routing infrastructure calls for integrating techniques from cryptographic protection mechanism, statistical anomaly detection, protocol syntax checking, and protocol semantics checking to build a multifence defense system.

Internet technology advances have benefited society and increased our productivity, but have also made us critically dependent on the reliability of Internet services. At a fundamental level, all applications rely on a dependable packet delivery service provided by the Internet routing infrastructure. However, the Internet is a large-scale, complex, loosely coupled distributed system made of many imperfect components. Faults of varying scale and severity occur from time to time at various locations. Measurements show that for one major Internet service provider (ISP) 20 percent of the links have a mean time to failure of less than one day, and 70 percent of the links less than 10 days. Internet backbone paths exhibit a mean time to failover, due to either physical failure or policy changes, of roughly two days, and only roughly 20 percent of paths stayed unchanged in five days. Furthermore, 0.2–1 percent of the entries in the global Internet routing table suffered from operator misconfigurations. Traffic overload due to large-scale virus attacks has also added stress to the routing protocol's operations.[1]

Ensuring the dependability of global packet delivery service has been a long-term research objective; however, different efforts have focused on different aspects of the problem. In early packet-switched network designs, the focus was on adapting to physical failures, such as link or node failure. More recently, the focus has shifted toward protecting routing protocols against more complex faults. Existing efforts include:

- Protecting routing protocols against outsider attacks through the use of plaintext passwords and keyed MD5 authentication in Open Shortest Path First 2 (OSPF2), and TCP MD5 to protect Border Gateway Protocol (BGP) sessions [2]
- Protecting routing protocols from certain types of insider attacks through the use of digitally signed link state update messages [2]
- Providing Byzantine robustness in routing protocols through the use of cryptographic mechanisms [3]
- Securing BGP routing update exchanges through encryption of neighbor-to-neighbor communication channels, authorization of origin information, and authorization of autonomous system (AS) path data [2, 4, 5]
- Exploiting protocol and network properties to detect faults without using cryptographic mechanisms [6, 7]

Despite all the efforts thus far, routing faults still occur now and then, and result in interrupted packet delivery. To assess the defense strength of a routing infrastructure, we introduce a threat model based on known (either existing or potential) threats, and sketch out a defense framework that embraces all major efforts in defending against faults and attacks. We use this framework to review existing work. Our analysis shows that although individual efforts can effectively guard against specific faults, no single effort can counter all faults. A resilient Internet routing infrastructure calls for a multifence defense system that integrates techniques ranging from cryptographic protection mechanisms to statistical anomaly detection, protocol syntax checking, as well as protocol semantics checking in order to provide the highest possible dependability.

*This work is partially supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. DABT63-00-C-1027, by National Science Foundation (NSF) under Contract No. ANI-0221453, and by a research grant from Cisco Systems. Any opinions, findings. and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of DARPA, NSF, or Cisco Systems.*

[1] *A complete list of references can be found in [1].*

0890-8044/04/$20.00 © 2004 IEEE

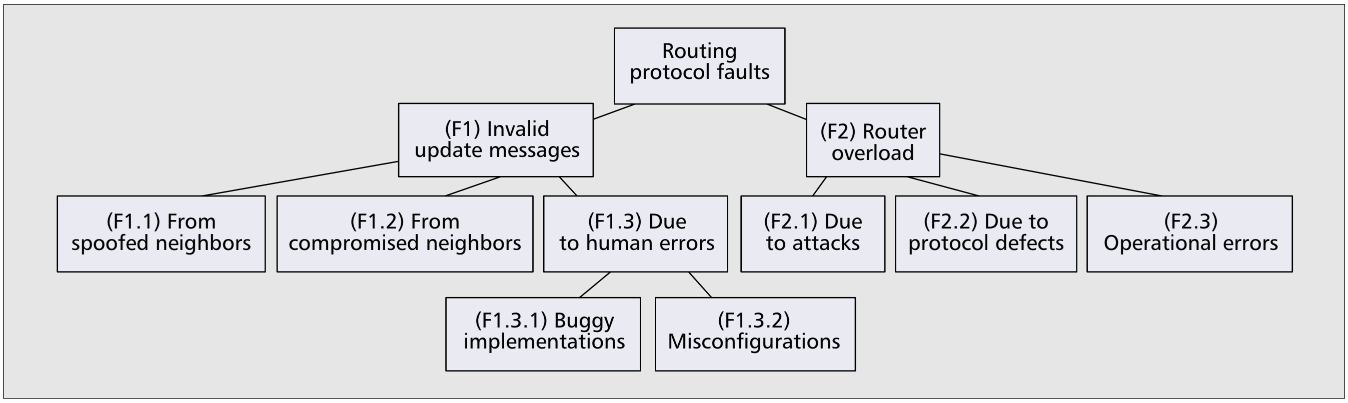

Routing protocol faults
(F1) Invalid update messages
(F2) Router overload
(F1.1) From spoofed neighbors
(F1.2) From compromised neighbors
(F1.3) Due to human errors
(F2.1) Due to attacks
(F2.2) Due to protocol defects
(F2.3) Operational errors
(F1.3.1) Buggy implementations
(F1.3.2) Misconfigurations

Figure 1. *A fault tree for Internet routing.*

## Background

The fundamental functionality provided by the Internet routing infrastructure is packet delivery. Other measures, such as delay and jitter, are meaningful only when a packet is delivered to its destination.

The following basic components make up the Internet routing infrastructure:

- Physical network connectivity: At the IP level, this connectivity consists of routers and physical links that connect hosts to routers and routers with each other.
- Network routing protocols: Routers run routing protocols among themselves to distribute reachability information to various destinations and dynamically adjust the paths based on topological and other kinds of changes.
- Hop-by-hop forwarding: Routers accept packets from hosts and neighboring routers and forward the packets to next-hop router along the path toward the destinations.

A truly resilient routing infrastructure should be able to deliver packets as long as any legitimate physical path to the destinations exists. This article provides a survey of research and development efforts aimed at enhancing the resiliency of the *network routing protocols* component.

### A Brief Introduction to Network Routing Protocols

At the routing protocol level, the Internet is composed of thousands of ASs, loosely defined as networks and routers under the same administrative control. BGP is the de facto inter-AS routing protocol. The routing protocol running within an AS is called Interior Gateway Protocol (IGP), typically OSPF, Intermediate System to Intermediate System (IS-IS), Routing Information Protocol v. 2 (RIPv2), or Interior Gateway Routing Protocol (IGRP). These routing protocols can be divided into three general classes: distance vector protocols, link state protocols, and path vector protocols.

In a *link state protocol* (e.g., OSPF and IS-IS), each router floods its local connectivity information (i.e., link state) to every other router in the same network. Each router collects the updates, builds a complete network topology, and uses this topology to compute paths to all destinations. Because each node has knowledge of the full topology, there is minimal dependence between nodes in the routing computation; thus, link state routing protocols are generally considered most promising for detecting faults [3].

In a *distance vector protocol* (e.g., RIP or IGRP), each router advertises its shortest distance to all destinations. Based on the distance information learned from its neighbors, a router selects the neighbor that yields the shortest distance to each destination as the next hop. A distance vector router has no direct information regarding network topology beyond its immediate neighbors, and its shortest path computation is based on distances reported by neighbors. Reference [3] argues that distance vector protocols are poor candidates for detecting faults because a router has no way to verify the validity of the distance information.

In a *path vector protocol* (e.g., BGP), a router announces the full path to each destination. Path information provides each router with partial information regarding topological connectivity; this partial information makes a fundamental difference between path vector and distance vector protocols. Although path information is not sufficient to construct complete topological connectivity, as we show later it can be used effectively for fault detection. Due to BGP's critical role in routing packets across loosely coupled ASs in the global Internet, the majority of the research efforts cited in this survey are related to BGP resiliency.

## A Routing Protocol Threat Model

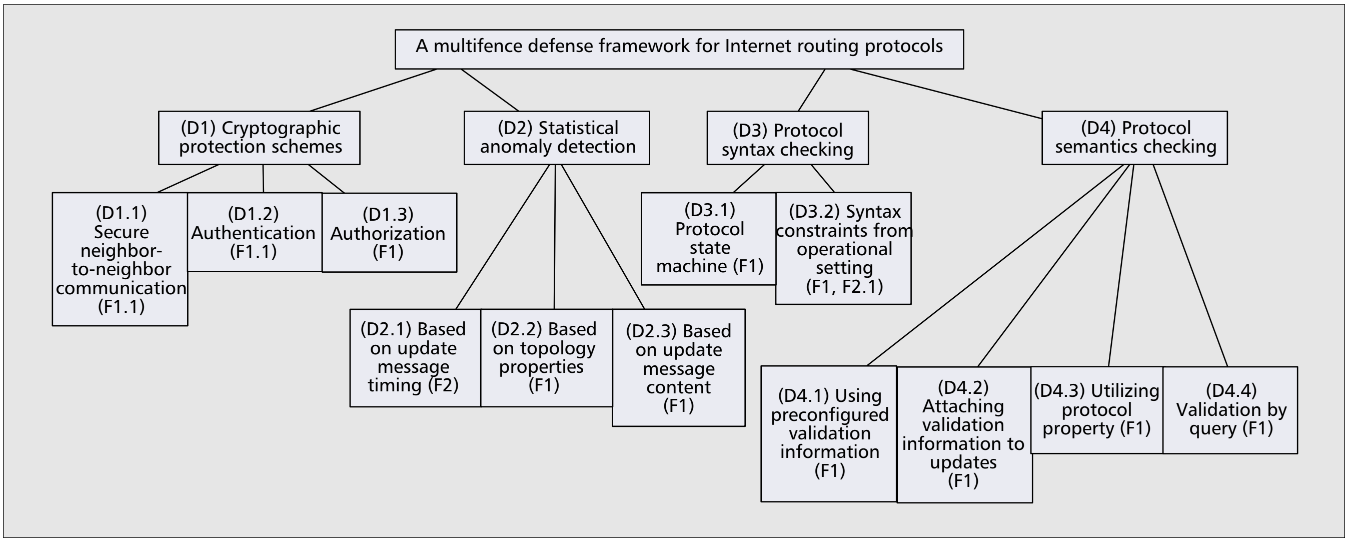
Routing can be interrupted by physical failures, operational faults, bugs in routing protocol implementations, unforeseen defects in the routing protocol designs, and other faults. Network resource exhaustion may also affect routing operation due to the in-line signaling nature of Internet routing protocols. Furthermore, there have been malicious attacks that were directly aimed at disabling routers and routing protocols.[2]

Establishing a threat model introduces a structure among different classes of faults based on the types of damage each may cause. One can then assess the goals and effectiveness of various existing countermeasures in defending against the faults. The Internet Engineering Task Force (IETF) Routing Protocol Security Working Group (RPSEC, http://www.rpsec.org) is carrying out a number of ongoing efforts to construct threat models for global Internet routing. The threat model sketched here incorporates some of that work, together with input collected from other sources. Our threat model is presented at a high level, leaving out details on precisely how potential faults and attacks might be carried out and instead focusing on the results of the faults.

Our threat model is represented by a fault tree, shown in Fig. 1. Each node in the tree represents the potential cause of faults; any of the subfaults can lead to the parent fault. The specific faults can adversely affect any component on the paths from data sources to destinations or could be against any of the routers and links between the sources and destinations. This fault tree allows one to sort faults into different classes, assess the severity of faults, and associate defense mechanisms with the fault(s) they can effectively fence off.

Starting from the root node we sort faults into two broad

[2] *This article takes into account the attacks that result in an inability to exchange routing data. However, the general problem of denial of service attacks is not strictly a routing protocol issue, and we do not review it here.*


A multifence defense framework for Internet routing protocols
(D1) Cryptographic protection schemes
(D2) Statistical anomaly detection
(D3) Protocol syntax checking
(D4) Protocol semantics checking
(D1.1) Secure neighbor-to-neighbor communication (F1.1)
(D1.2) Authentication (F1.1)
(D1.3) Authorization (F1)
(D2.1) Based on update message timing (F2)
(D2.2) Based on topology properties (F1)
(D2.3) Based on update message content (F1)
(D3.1) Protocol state machine (F1)
(D3.2) Syntax constraints from operational setting (F1, F2.1)
(D4.1) Using preconfigured validation information (F1)
(D4.2) Attaching validation information to updates (F1)
(D4.3) Utilizing protocol property (F1)
(D4.4) Validation by query (F1)

■ Figure 2. *A multifence defense framework for routing protocols, where (F.*) in each leaf node indicates the faults it can help guard against.*

classes: faults that result in invalid update messages and faults that disable the router by overload. Invalid update messages can be further sorted into three categories: those from spoofed neighbors, those from compromised neighbors, and those due to human errors such as buggy implementations and misconfigurations. Similarly, router overload may be due to malicious attacks, protocol defects, or operational errors. We intentionally limit the depth of the fault tree since our objective is to present threats at an abstract level. Nodes in Fig. 1 are numbered for ease of reference in Fig. 2.

## A Framework for Resilient Internet Routing

Figure 2 sketches a basic framework covering various components in adding resiliency to Internet routing. It also shows the effect of each proposed defense as well as relations between different solutions. The faults each defense mechanism guards against are also shown in each leaf node; they are suggestive rather than definitive. The mapping of the defenses against the fault types may vary depending on the specifics of individual faults and defense mechanisms. Note that:

- Each individual defense fence by itself can achieve only limited effectiveness in preventing or detecting certain types of faults.
- No single fence covers all faults.
- Some faults are covered by more than one fence.

One should also be aware that any of these defensive fences may fail itself. The nodes are labeled for reference in the rest of this survey.

The framework divides defenses into several classes: cryptographic schemes, statistical anomaly detection, protocol syntax checking, and protocol semantics checking. Cryptographic mechanisms primarily aim to lock out *external* attackers. However, such protection mechanisms alone are insufficient [8]. In large-scale distributed systems such as the Internet routing infrastructure, it is impossible to build perfect protection; inevitably imperfect components can be compromised, opening holes in the system. Anomaly detection, syntax checking, and semantics checking play an essential role in detecting faults when the prevention fence has failed or unexpected faults occur. Note that reaction mechanisms that aim at minimizing or repairing the damage caused by the faults can be viewed as part of the resiliency framework. For brevity, reaction fences are not discussed here, but can be found in the extended version of this article [1].

As faults occur in the system, correspondingly defenses are built at different levels, each covering part of the broader solution space. Most, if not all, of the fences have been built in response to faults experienced in the Internet; Fig. 2 may serve as a useful guide to see each piece in the proper context. We review each class of defense mechanisms in subsequent sections.

## Cryptographic Protection Schemes

The framework in Fig. 2 lists three types of cryptographic protection scheme: secure neighbor-to-neighbor communication, authentication, and authorization. Every routing protocol requires communication between neighboring routers, and *secure neighbor-to-neighbor communication* is designed to prevent an outside entity from modifying, deleting, or adding messages exchanged between routers. *Authentication* aims to prevent an outside entity from imitating a legitimate router. However, even when perfect secure communication and authentication are in place, faults can still occur when implementation bugs or operational errors cause a legitimate router to advertise addresses it does not own or report false path/link information. The third type of protection restricts the actions of a legitimate router by applying *authorization* to the content of routing message exchanges, so each router can only originate routes to address blocks it owns and only include legitimate inter-AS links in routing paths.

A common technique used to achieve secure neighbor-to-neighbor communication, authentication, and authorization is public key cryptography. A corresponding public key infrastructure (PKI) reduces the problem of verifying everyone's public key to verifying just one (or a few) public keys [8]. In a typical PKI, an *authorized third party* issues certificates according to some well defined hierarchical structure. A certificate binds an entity with its public key and is signed with the authorized third party's own private key. The recipient of a certificate can use the authorized third party's public key to authenticate the certificate. Perlman designed two early network layer protocols that rely on cryptographic protection techniques: Byzantine Robust Flooding and Robust Link State Routing [3]. Unfortunately, these protocols do not scale to the size of today's Internet topology. The remainder of this section reviews the existing work on cryptographic protection fences.

### OSPF with Digital Signature (D1.1, D1.2)

OSPF2 [2] provides secure neighbor-to-neighbor communication using plaintext passwords and keyed MD5 authentication. Plaintext passwords are vulnerable to eavesdropping. Keyed MD5 authentication can effectively protect neighbor-to-neighbor protocol exchanges but is ineffective in the presence of insider faults; a faulty router may modify the content of any link state packet passing through it. To address such insider faults, Murphy *et al.* proposed an approach [2] in which the originator of each link state packet digitally signs the packet using its private key, an approach similar to that in [3]. The receiver of a signed link state packet can verify its authenticity using the originator's public key. The public key of each router is flooded using a specialized link state packet, called public key LSA (PKLSA). This PKLSA contains the public key of the router, and signatures of one or more *trusted entities* to verify this public key. However, the details of the public key infrastructure (i.e., exactly how to choose the trusted entities) are not discussed.

Unfortunately, digital signatures alone do not provide perfect protection. There exist residual vulnerabilities, such as misconfigurations, compromised originating routers, and compromised private keys. Furthermore, generating and verifying digital signatures add performance overhead and potential complexity; such factors all impact the deployment of such approaches. Reference [9] proposed some mechanisms to reduce the cost of cryptographic protection for link state routing.

### Origination and Predecessor (D1.1, D1.2)

Smith *et al.* [4] proposed to provide secure neighbor-to-neighbor encryption in BGP by using a session key and message sequence number (used only between direct neighbors). The proposal also includes an originating UPDATE sequence number (set by the origin AS) to protect against replaying UPDATE messages. In addition, a new attribute called the *predecesor* (the second AS on the AS path) is added to the route by the origin AS. The origin AS uses its private key to sign this predecessor and originating UPDATE sequence number, as well as some other fixed attributes in a BGP update message such as ORIGIN and AGGREGATOR. One can use the signed predecessor information to reconstruct and verify the entire path to the destination [10]. The specific steps in path reconstruction can be considered a form of semantics checking and are discussed later.

This approach treats each AS as an individual node in the topology, and assumes each AS has a unique and consistent predecessor. In reality, however, an AS may exhibit more complex behavior, and may use and advertise multiple distinct routes to a single destination, defeating the path finding algorithm. In addition, the approach leaves out public key distribution as a separate problem.

### Origination Authorization: Secure Origin BGP (D1)

Secure Origin BGP (SoBGP) [2] was introduced by Ng to verify the origin of route advertisements and prevent the advertisement of unauthorized prefixes. In addition, it offers some partial verification of AS paths. SoBGP uses a new type of BGP message, the SECURITY message, to distribute three types of certificates. The *Entity Certificate* is used to distribute public keys associated with entities such as an AS and provides in-band BGP method for changing the public keys. It is signed by some well-known authorities such as registries or other entities whose public keys have been preconfigured in the router. Once the Entity Certificate has been authenticated, *Authorization Certificates* are used to verify whether an AS is authorized to advertise an address block. A BGP update with an unauthorized origin AS is discarded. Finally, to protect the path, *Policy Certificates* contain a list of attached ASs and security policy options that allow a router to sanity check at least part of the AS path. These checking procedures can be considered a form of semantics checking and are discussed later.

### Secure-BGP (D1)

Secure BGP (S-BGP) [5] by Kent *et al.* provides comprehensive protection for BGP. IP security (IPsec) is used to secure neighbor-to-neighbor communication between BGP routers. For authentication and authorization, S-BGP defines a detailed PKI. An address allocation PKI specifies the assignment of address blocks to organizations and binds address block(s) to a public key belonging to the corresponding organization. Another PKI is used to bind an organization's public key with its assigned AS number(s), and bind a router's public key with its ID, AS number, and DNS name. These PKIs follow the existing Internet registry hierarchy that assigns IP addresses and AS numbers.

S-BGP argues that, given the large number of certificates needed for each update and the current maximum BGP update length of 4096 bytes, it is not only bandwidth-wasteful but also difficult, if not impossible, to send certificates along with updates, and it would not be backward compatible to distribute the certificates through a new BGP message. Instead, S-BGP introduces repositories from which routers can download the certificate database and revocation list.

S-BGP introduced a new type of BGP route attribute, an attestation, and defined two types of attestations. An address attestation (AA) is similar to the authorization certificate used in SoBGP; a recipient AS uses the origin AS's public key to verify that the origin AS has been authorized to advertise a prefix. A route attestation (RA) is signed by a router's private key. A route attestation signed by $AS_x$ indicates that $AS_x$ authorized $AS_{x+1}$ to advertise the path of ($AS_x$, $AS_{x-1}$, …, $AS_0$). The recipient AS uses the public key of each router along the path to verify the corresponding link in the AS path. In other words, when $AS_{x+1}$ receives from $AS_x$ the path ($AS_x$, $AS_{x-1}$, …, $AS_0$), it will verify the attestation of each of $AS_i$, $0 \le i \le x - 1$.

Using a daily average BGP update rate, Kent *et al.* showed that S-BGP added 139.9 minutes of CPU processing overhead per day per BGP session, and required a factor of 2 increase in storage overhead per BGP session. Given the large number of sessions present in a typical backbone BGP router, this overhead raises a concern. Furthermore, since BGP suffers from update storms due to session reset, and the peak update rate can vary dramatically from the daily average, performance evaluation based on daily average load is inadequate. Various performance improvement mechanisms, such as caching of validation results, delaying validation of backup paths, background validation of backup paths, and using special cryptographic hardware to run S-BGP, might help reduce the overhead to an acceptable level. However, at this time, there is no quantitative evaluation on these proposed improvements. Finally, incremental deployment remains an open challenge for S-BGP. An AS cannot use the attestations to verify a received path unless the PKI is deployed and all the ASs in the path have deployed S-BGP; when some route attestations are missing or some certificates unavailable, local administrators have to set security policies to handle such difficult cases.

### Summary of Cryptographic Protection Schemes

Both SoBGP and S-BGP seem promising for adding protective fences to the current BGP routing infrastructure. S-BGP is the more comprehensive of the two, but pays a much higher overhead cost. Table 1 summarizes the different cryptographic mechanisms proposed for the routing infrastructure.

All these approaches add protections, but also introduce new vulnerabilities at the same time. For example, private keys

| Work/approach | Secure n2n communication | Authenticated information | Authorized information | PKI | Key distribution |
|---|---|---|---|---|---|
| OSPF with digital signature [2] | MD5 | LSA | No | No | Special LSA |
| Origination and Predecessor [4] | Session key, sequence number | Origin, predecessor | No | Assumed | Assumed |
| SoBGP [2] | MD-5 | Origin, peering | Address ownership | Web of trust | Security msg |
| S-BGP [5] | IPSec | All BGP path attributes | AS path announcement, address ownership | Follow address/AS allocation/delegation | Out of band |

■ Table 1. *A comparison of cryptographic protection schemes.*

could be lost or stolen, or the registration and signing performed by an authorized third party could be manipulated. Generally speaking, insider faults remain a challenge, and many faults are not addressed by these protective fences. If judged as a complete solution, none of the approaches could lead to a truly resilient routing infrastructure. Nevertheless, cryptography-based protections are clearly effective against certain faults and can serve as part of the overall multifence approach.

## Statistical Anomaly Detection

Statistical anomaly detection is based on behavior profiles. A router or an auxiliary device keeps a statistical profile of the routing update messages; it rings an alarm if newly observed routing update statistics fail to fit the expected profile. The effectiveness of such detections largely depends on the ability to devise a useful statistical profile.

### LS Anomaly Detection (D2.1, D2.3)

In [11] Qu *et al.* measure the interarrival time of all OSPF packets received by a router, the distribution of OSPF packet types, and the age of the packets, and apply a statistical intrusion detection algorithm against the collected statistics. Testbed experiments show that this approach was effective in detecting three specific and known types of link state protocol attacks, the seq++ attack, Maximum Age attack, and Maximum Sequence Number attack.

### RIP Update Count Monitoring (D2.1)

Mittal *et al.* [12] take a similar approach and install sensors to detect faults in a RIP network. A sensor on a link counts the number of updates sent by a router. Upper and lower bounds are determined statistically and experimentally, and are used to detect possible faults. This approach also employs protocol syntax checking (reviewed next) and protocol semantics checking (reviewed later).

### Path Filtering Using Topology Properties (D2.2)

Wang *et al.* [7] protect the routes to the critical top-level Domain Name Service (DNS) servers by restricting route changes to within a set of established paths based on statistical analysis over history. The design exploits the observation that top-level DNS servers are well connected via stable routes and the high degree of redundancy in top-level DNS servers. Heuristics derived from routing operations are used to adjust the valid route set over time. The path filter design was tested against BGP routing logs; the results show that the design can effectively filter out bogus routes without impacting DNS service availability.

## Protocol Syntax Checking

The routing protocol syntax defines the legitimate sequence of messages and can be used to reject invalid messages. A common approach taken by routing protocols is to use heartbeat messages to detect whether a neighbor is reachable (i.e., Hello in OSPF, Keep-Alive in BGP). BGP also uses extensive syntax checking in message exchanges between peering routers. However, syntax checking is seldom used in communications beyond neighbor-to-neighbor exchanges.

### Extended Timed Finite State Machines for Link State Protocols (D3.1)

The *JiNao* [13] architecture by Chang *et al.* provides real-time intrusion detection for link state routing protocols such as OSPF. JiNao uses extended finite state machines (FSMs) where each state maintains the time of the first transition into this state, the last transition into this state, the current event time, and a few other state variables. An FSM for normal behavior and one for each known attack pattern are used collectively to determine the state of the OSPF. Known attacks, such as the seq++, Maximum Age, and Maximum Sequence Number attacks, are detected by FSMs using pattern matching. FSMs for newly discovered attacks can be added as the attacks are identified.

### BGP TTL Security Hack (D3.2)

Gill *et al.* [2] extend the BGP syntax checking by having each router check the time to live (TTL) value of BGP update messages and drop those messages with TTL values not within a valid range. External BGP peers are normally adjacent. If BGP routers configure the initial TTL value to be 255, received update messages should have a TTL value no less than 254. Since non-faulty routers decrease the TTL of each received packet by one, this simple check is very effective in detecting false messages injected from more than one hop away, such as denial-of-service attacks against the BGP TCP port.

Mittal *et al.* proposed a similar approach [12] in which sensors sitting on a link check the link layer address and TTL of the RIP update messages (semantics checking developed in the same work is reviewed next). In general, the simple TTL check is an example of how protocol syntax checking can be used effectively in countering known or unknown faults in routing message exchanges.

### Summary of Syntax Checking

Statistical anomaly detection can provide hints about potential faults in message sequences, although these sequences might not be prohibited by the protocol specification. Protocol syntax checking may play a unique role in detecting false message sequences that are not explicitly stated in the protocol specification. Furthermore, protocol syntax analysis using formal methods can help identify bugs in the protocol design and detect defects in an existing protocol, as demonstrated in the extended version of this article [1] for RIP and BGP through some examples.

| Work/approach | Preconfigured (D4.1) | Newly propagated (D4.2) | Utilizing existing information (D4.3) | Query (D4.4) |
|---|---|---|---|---|
| Assertion [14] | No | No | Assertions | Yes |
| POD [15] | No | No | Properties | No |
| RIP-TP [16] | No | No | Triangle theorem | Probing message |
| Predecessor [4] | No | Predecessor | Path finding | No |
| Sensor [12] | Global topology | No | No | Between sensors |
| SoBGP [2] | No | Peering relationship | No | No |
| MOAS [6] | No | MOAS list | Piggybacked in updates | IRR/DNS-based |
| IRR | No | No | No | Centralized database |
| IRV [17] | No | No | No | Distributed IRV servers |

■ Table 2. *A comparison of protocol semantics checking approaches.*

## Protocol Semantics Checking

Protocol semantics checking uses the content of routing update messages to detect faults. In distributed routing protocols, one update message often propagates through the network along multiple paths. Thus, one node may, directly or indirectly, receive multiple copies of the same information. Protocol semantics checking utilizes the fact that multiple copies of the same information should be consistent with each other. Table 2 provides a brief summary of all the work reviewed in this section, positioning each according to the multifence defense framework.

### Assertions and Property-Oriented Fault Detection (D4.3)

In his Ph.D. thesis [14], Massey took the approach of dividing the Internet into compartments and detecting faults at compartment boundaries by using predefined assertions. Assertions are conditions that must hold true if each protocol functions correctly. The assertion approach is extensible since new assertions can easily be added. Reference [14] demonstrated how the technique is used to detect a number of faults in a distance vector multicast routing protocol, and provided a more general framework for arbitrary protocols.

Wang *et al.* [15] apply a similar approach to link state routing protocols. A centralized monitoring process collects all the routing messages exchanged between routing processes and keeps track of the state kept in each routing process. The monitoring process detects faults by taking snapshots of the global state made up of all the routing processes and using a few properties that must hold true for a link state protocol. In the event a property does not hold, more specific properties can be used to diagnose where exactly the fault happened. The authors provide two case studies of detecting the seq++ and Maximum Sequence Number attacks. While Massey's assertion checking is done by each compartment and via message exchanges between compartments, Wang's approach uses a centralized design that limits its applicability to large-scale networks.

### RIP-TP Triangle Checking (D4.3, D4.4)

*RIP* with triangle theorem checking and probing (RIP-TP) [16] uses limited information obtained from RIP update messages to detect potentially invalid routing updates. The triangle theorem states that, given a shortest path protocol and a set of three nodes, the distance between any pair of nodes should always be less than or equal to the sum of the distance of the other two pairs. However, delays in update message propagation or message losses may cause a temporary violation of the triangle theorem. To distinguish temporary delays from invalid messages, RIP-TP sends probing messages to the destination to verify the routing update in question. Simulation results show that RIP-TP is effective in detecting false updates and has low overhead. RIP-TP is also incrementally deployable; any node that implements RIP-TP can benefit independent of whether any other nodes have implemented RIP-TP.

### Propagating Predecessor Information (D4.2, D4.3)

Smith *et al.* [4] add a new attribute called the predecessor (second AS in the AS path) to route updates. Earlier we discussed how signed predecessor data can be used to detect false routes. This approach can be viewed as a type of semantics checking. Following a path-finding algorithm [10], each node in the network can learn and authenticate the path to a destination. This approach works well for a shortest path distance vector routing protocol such as RIP. However, in BGP a link that is legitimate for one destination might be illegitimate for another due to routing policy, and path finding does not fit well.

### Sensor Monitoring with Global Knowledge (D4.1,D4.4)

Mittal *et al.* [12] use sensors to detect faults in a RIP network; its elements for anomaly detection and protocol syntax checking were discussed earlier. Sensors are placed on some or all of the links, and each sensor is given the whole network topology as well as the locations of all other sensors. A sensor computes all the possible paths from each router to each destination by running a link state protocol over the given topology. A sensor then analyzes the routing updates observed on its link and checks their content (i.e., distance information) against its computed set of possible distances. If a distance is not in the valid set, an alarm is raised. To verify that a distance is in the valid set, a query is sent to all the sensors along the possible path(s) that have this distance. One major obstacle in deploying this design is that each sensor needs to be preconfigured with the entire network topology and sensor placement *in advance*. A major advantage of the proposed design is that it requires no modification to the existing routing protocol and routers. It is also a good example of integrating multiple detection mechanisms, including statistical anomaly detection, syntax checking, and semantics checking.

### Propagating Peering Relationships with SoBGP (D4.2)

Earlier we discussed how SoBGP uses signed policy certificates to detect false paths. This detection step can also be viewed as a form of semantics checking. Each AS lists its AS-level neighbors in a policy certificate; these certificates can be used to build a directed graph of the Internet topology. If a received update message contains a link that does not exist in

| Work/approach | Cryptographic schemes | Statistical anomaly detection | Protocol syntax checking | Protocol semantics checking |
|---|---|---|---|---|
| OSPF with digital signature [2] (D1.1, D1.2) | Signed LSA<br>MD5 (D1.1, D1.2) | No | No | No |
| Origin, Predecessor [4] (D1.1, D1.2, D4.2, D4.3) | Origin and predecessor (D1.1, D1.2) | No | | Checking based on path-finding (D4.2, D4.3) |
| SoBGP [2] (D1, D4.2) | Address ownership (D1) | No | No | Peering map (D4.2) |
| S-BGP [5] (D1) | IPsec, AA, RA (D1) | No | No | No |
| LS Anomaly detection [11] (D2.1, D2.3) | No | Yes (D2.1, D2.3) | No | No |
| Path Filtering [7] (D2.1, D2.2) | No | Topology property (D2.1, D2.2) | No | No |
| LS FSM [13] (D3.1) | No | No | Known attacks (D3.1) | No |
| BTSH [2] (D3.2) | No | No | TTL (D3.2) | No |
| Assertion [14] (D4.3) | No | No | No | Assertions (D4.3) |
| POD [15] (D4.3) | No | No | No | Properties (D4.3) |
| RIP-TP [16] (D4.3, D4.4) | No | No | No | Triangle theorem probing message (D4.3, D4.4) |
| Sensor [12] no (D2.1, D3.2, D4.1, D4.4) | No | Update count (D2.1) | TTL, link layer address (D3.2) | With preconfigured global knowledge (D4.1, D4.4) |
| MOAS [6] (D4.2, D4.3, D4.4) | No | No | No | Checking MOAS list (D4.2, D4.3, D4.4) |
| IRR (D4.4) | No | No | No | Centralized database for query (D4.4) |
| IRV [17] (D4.4) | No | No | No | Distributed IRV servers for query (D4.4) |

■ Table 3. *Summary of reviewed work labeled with their defense fences.*

this directed graph, the update will be considered invalid. Note that SoBGP constructs a superset of possible AS-level links, but routing policies may prohibit certain prefixes from using certain links. Thus, even when every link in an update message exists in the directed graph, the update may still be invalid. Furthermore, some ASs may not be willing to announce the connectivity to all their neighbor ASs because such information may be considered confidential.

### Propagating MOAS Lists (D4.2, D4.3, D4.4)

Zhao *et al.* [6] proposed a non-cryptographic approach to protect BGP against route origin spoofing. In BGP, a destination may appear to have multiple origin ASs (MOAS) due to multihoming, misconfigurations, or even malicious attacks. Reference [6] adds a MOAS attribute to BGP update messages that contains a complete list of legitimate origins; this attribute is attached whenever an origin AS announces the route. The MOAS list can be altered by any router along the way, and an invalid origin may attach an arbitrary MOAS list. However, the rich connectivity in today's Internet makes it difficult to block all the updates carrying correct MOAS lists from propagating out. Each router in the network compares the MOAS lists received from different peers, and any difference in the MOAS lists will raise an alarm. Simulations with realistic topologies show that this non-cryptographic solution is effective in detecting the existence of false origin ASs.

### The Internet Routing Registry (D4.4)

The Internet Routing Registry (IRR, http://www.irr.net) places policy data collected from ASs into a small number of databases. BGP routers can then compare the received routes against the information listed in the database, and any conflicting routes can be discarded. However some ISPs may not be willing to publish their routing policy information, and network operators may not update the database promptly. As a result, the IRR database may be incomplete or contain obsolete information. Furthermore, the information stored in IRR is not digitally signed, so it is vulnerable to malicious attacks.

### Interdomain Routing Validation (D4.4)

The Interdomain Routing Validation (IRV) [17] protocol by Goodell *et al.* provides a way for BGP routers to solicit information for semantics checking. Each participating AS designates an IRV server that answers queries regarding the AS routing policy, whether it originates a particular prefix, the BGP updates recently received from its neighbors, its current BGP routes, and its BGP updates sent to neighbors. The query results can be used to validate a routing policy or confirm the origin of a prefix. IRV can be used to provide verification information for other proposed mechanisms, such as the MOAS list approach when conflicting MOAS lists are detected, or SoBGP and S-BGP in partial deployment. However, how to maintain the freshness of information at the IRV server, and the authenticity of the query and reply remains a challenging problem. Furthermore, a query may be based on data older than those in IRV, so mechanisms to distinguish this old (but valid) data from invalid data remains unsolved in the current solution. And of course, IRV also introduces its own vulnerabilities. For example, IRV configurations and policies could be incorrect or intentionally manipulated by a compromised IRV server.

## Summary

In this article we review the various approaches to improving the resiliency of the Internet routing protocols. Table 3 provides a summary of the approaches. By examining both the routing faults that occur in the operational Internet and the mechanisms to protect the routing infrastructure, we make the following observations:

- In a system as large as today's Internet, faults are the norm rather than the exception.
- Cryptographic protection mechanisms can be effective in guarding against specific faults, but they cannot detect or prevent all types of faults, especially those due to implementation bugs, configuration errors, or compromised routers. Furthermore, cryptographic mechanisms themselves are also subject to faults.
- A number of detection mechanisms have been developed recently to detect faults in the Internet routing system. Although each has limited detection power, collectively they can provide a stronger overall protection against faults.

## Looking Forward

As the Internet continues to grow, it faces an increasingly hostile environment. The collection of imperfect components operated by different administrative entities will increase not only the frequency of physical failures, but also the number of operational errors and unexpected faults. Furthermore, the importance of the Internet in society will attract more intentional attacks. In such a complex and hostile environment, no single protection or detection mechanism can be adequate. Instead, we must build a multifence defense system to ensure a resilient Internet routing infrastructure.

At the same time, we recognize inevitable trade-offs. Any new piece we add to a system adds new overhead and can introduce potential new vulnerabilities. For example, public key cryptographic mechanisms protect the protocol from outsider attacks, but also introduce a new dependence on PKI. Whether to add a new piece into the system is therefore a trade-off between the benefits and the new vulnerabilities. Another trade-off is fault detection capability vs. performance scalability. Detection mechanisms usually benefit from more information. However, propagating information in a large system adds performance overhead. Although performance overhead is usually unavoidable, a good solution's performance overhead should be scalable as the system size increases. This is especially necessary for a system as large as the Internet, whose size keeps increasing over time. A final challenge is partial deployability of new protection and detection mechanisms, which must be taken into consideration in the design.

## References

[1] D. Pei, D. Massey, and L. Zhang, "A Framework for Resilient Internet Routing Protocols," UCLA, Tech. rep. CSD-TR-030052, 2003.

[2] Internet RFCs and Drafts, http://www.ietf.org.

[3] R. Perlman, "Network Layer Protocols with Byzantine Robustness," Ph.D. dissertation, MIT Laboratory for Computer Science, 1988.

[4] B. R. Smith, S. Murphy, and J. J. Garcia-Luna-Aceves, "Securing the Border Gateway Routing Protocol," *Global Internet '96*, Nov. 1996.

[5] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (SBGP)," *IEEE JSAC*, Special Issue on Network Security, 2000.

[6] X. Zhao *et al.*, "Detection of Invalid Routing Announcement in the Internet," *Proc. IEEE DSN 2002*, June 2002.

[7] L. Wang *et al.*, "Protecting BGP Routes to Top Level DNS Servers," *Proc. ICDCS 2003*, 2003.

[8] B. Schneier, *Secrets and Lies–Ditgital Security in a Networked World*, Wiley, 2000.

[9] R. Hauser, T. Przygienda, and G. Tsudik, "Reducing the Cost of Security of Link-State Routing," *Symp. Network and Distrib. Sys. Sec.*, 1997.

[10] J. J. Garcia-Lunes-Aceves and S. Murthy, "A Loop-Free Path-Finding Algorithm: Specification, Verification and Complexity," *Proc. IEEE INFOCOM*, Apr. 1995.

[11] D. Qu *et al.*, "Statistical Anomaly Detection of Link-state Routing Protocols," *Proc. ICNP*, Nov. 1998.

[12] V. Mittal and G. Vigna, "Sensor-Based Intrusion Detection for Intradomain Distance-Vector Routing," *ACM CCS 2002*, Nov. 2002.

[13] H. Chang, S. F. Wu, and Y. F. Jou, "Real-Time Protocol Analysis for Detecting Link-State Routing Protocol Attacks," *ACM Trans. Info. and Sys. Sec.*, vol. 4, no. 1, Feb. 2001, pp. 1–36.

[14] D. Massey, "Fault Detection and Security in Routing Protocols," Ph.D. dissertation, UCLA, 2000.

[15] F. Wang, F. Gong, and S. Wu, "A Property Oriented Fault Detection Approach for Link State Routing Protocol," *Proc. ICCCN 2000*, Oct. 2000.

[16] D. Pei, D. Massey, and L. Zhang, "Detection of False Routing Update in RIP," *IEEE GLOBECOM*, Dec. 2003.

[17] G. Goodell *et al.*, "Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing," *NDSS*, 2003.

## Biographies

DAN PEI (peidan@cs.ucla.edu) is currently a Ph.D. candidate at the University of California at Los Angeles (UCLA) Computer Science Department. His current research interests include the fault tolerance and performance of Internet routing protocols. He received his Bachelor's and Master's degrees from Tsinghua University.

DAN MASSEY [M] is a research assistant professor at the University of Southern California Information Sciences Institute and is currently the principal investigator on DARPA and NSF funded research projects investigating techniques for improving the Internet's DNS and BGP infrastructures. He received his doctorate from UCLA and is a member of the IEEE Communications and Computer Societies. His research interests include fault tolerance and security for large scale network infrastructures.

LIXIA ZHANG [SM'95, ACM'84] received her Ph.D. degree from the Massachusetts Institute of Technology. She was a member of the research staff at the Xerox Palo Alto Research Center before joining the faculty of UCLA's Computer Science Department in 1995. In the past she has served on the Internet Architecture Board, Co-Chair of IEEE Communications Society Internet Technical Committee, Vice Chair of ACM SIGCOMM, and as an editor for *IEEE/ACM Transactions on Networking*.