A Framework for Resilient Internet Routing Protocols

Dan Pei, UCLA; Daniel Massey, USC/ISI; Lixia Zhang, UCLA

Technical Report TR-030052
UCLA Computer Science Department
November 13th, 2003

***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 various scale and severity occur from time to time. In this article we survey the research efforts over the years aiming 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 the known threats and 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 it can defend against. 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, protocol semantics checking and reaction to build a multi-fence defense system.**

I. Introduction

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 various scale and severity occur from time to time at various locations. Measurements show that in one major ISP 20% of the links have a mean time to failure of less than 1 day and 70% of the links less than 10 days [2]. Internet backbone paths exhibit a mean time to fail-over (due to either physical failure or policy changes) of roughly 2 days and only roughly 20% of paths stayed unchanged in five days [3]. Furthermore, 0.2 to 1% of the entries in the global Internet routing table suffered from operator misconfigurations [4] . Traffic overload due to large scale virus attacks has also added stress to the routing protocol's operations [5]. Assuring the dependability of global packet delivery service has been a long term objective, however different research efforts have focused on different aspects of the problem. In early packet switched network designs, the focus was on handling 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:

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 the DARPA, NSF, or Cisco Systems. A shorter version of this paper will appear in IEEE Network [1].

- Protecting routing protocols against outsider attacks through the use of simple passwords, keyed MD5 authentication in OSPF2[6], and TCP MD-5 to protect BGP sessions [7].
- Protecting routing protocols from certain types of insider attacks through the use of digitally signed link-state update messages [8], [9].
- Providing Byzantine robustness in routing protocols through the use of cryptographic mechanisms [10].
- Securing BGP routing update exchanges through encryption of neighbor-to-neighbor communication channels, authorization of origin information, and authorization of AS path data [11], [12], [13].
- Exploiting protocol and network properties to detect faults without using cryptographic mechanisms [14], [15].

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 routing infrastructure, Section III introduces a threat model based on the known (either existing or potential) threats, and Section IV sketches out a defense framework that embraces all major efforts in defending against faults and attacks. Sections V, VI, VII, and VIII use this framework to review existing work. Section IX reviews some additional techniques that react to faults in a way that allows packet forwarding to continute(if it is possible). 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 multi-fence defense system that integrates techniques ranging from cryptographic protection mechanism to statistical anomaly detection, protocol syntax checking, protocol semantics checking, and reaction in order to provide the highest possible dependability.

II. Background

The fundamental functionality provided by the Internet routing infrastructure is packet delivery. Other factors, 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

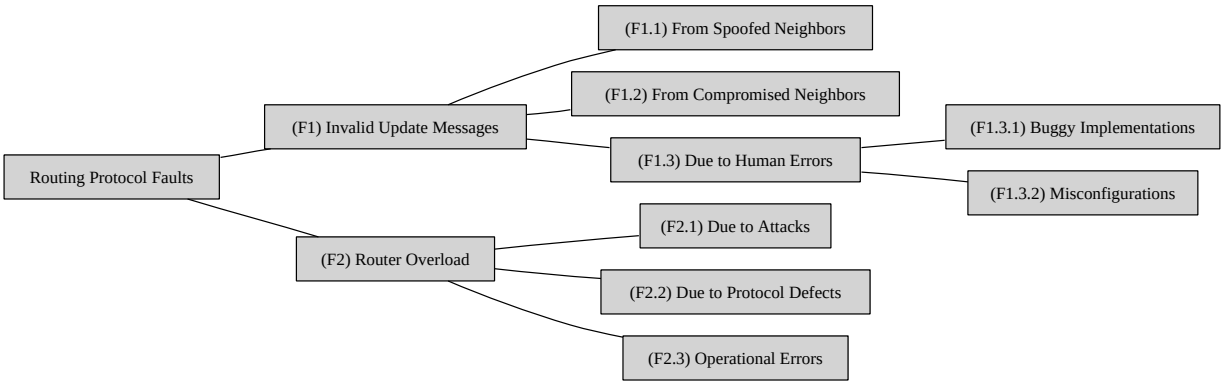


Fig. 1. A Fault Tree for Internet Routing

exists. This article provides a survey of research and development efforts aimed at enhancing the resiliency of the *network routing protocols* component.

A. A Brief Introduction to Network Routing Protocols

At the routing protocol level, the Internet is composed of thousands of Autonomous Systems (ASes), loosely defined as networks and routers under the same administrative control. BGP [16] is the *de facto* inter-AS routing protocol. The routing protocol running within an AS is called IGP (interior Gateway Protocol), typically one of OSPF[6], IS-IS[17], RIPv2 [18], or IGRP [17]. These various routing protocols can be divided into 3 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) globally to every other router in the same system. Each router collects the updates, builds the complete network topology, and uses this topology to compute paths to all destinations. Each node has knowledge of the full topology and there is minimal dependency between nodes in the routing computation; thus link-state routing protocols are generally considered most promising for detecting faults [10].

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 the network topology beyond its immediate neighbors and its shortest path computation is based on distances reported by neighbors. [10] 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 and this partial information marks a fundamental difference between path vector protocols and distance vector protocols. Although path information is not sufficient to construct the complete topological connectivity, Section VIII will show the path information can be used effectively for fault detection. Due to its critical role in routing packets across loosely coupled ASes in a global scale, majority of the research efforts cited in this survey are related to BGP resiliency.

III. 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, which can be caused by software viruses, may also affect routing operation due to the in-line signaling nature of Internet routing protocols. Furthermore, attacks can be directly aimed at routers and routing protocols. [1]

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 IETF Routing Protocol Security Working Group (RPSEC, http://www.rpsec.org) is carrying out a number of on-going efforts to construct threat models for global Internet routing and the threat model sketched here incorporates some of that work, together with inputs 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 Figure 1. Each node represents the potential cause of faults and any of the sub-faults 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 structure allows one to sort faults into different classes, to assess the severity of faults, and to associate defense mechanisms with the fault(s) it can effectively fence off.

Starting from the root we sort faults into two broad 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 including 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

[1]This article takes into account the attacks that result in an inability to exchange *routing data*, but the general problem of controlling traffic classes in the face of an attack is not strictly a routing protocol issue and we do not review it here.

Fig. 2. A Multi-Fence Defense Framework for Routing Protocols, where ($F.*$) in each leaf node indicating the faults that it can help guard against.

level. Nodes are numbered as shown in Figure 1 for reference in Figure 2.

IV. A Framework for Resilient Internet Routing

Figure 2 sketches a basic framework covering the components in resilient Internet routing and shows the effect of each proposed defense as well as relations between defenses. The faults each defense mechanism guards against are also shown in each leaf node, and they are suggestive rather than definitive; the mapping of the defense fences against the fault types may vary depending on the specifics of individual faults and defense mechanisms. Note that (1) each individual fence by itself can achieve only limited effectiveness in preventing or detecting certain types of faults, (2) no single fence covers all faults, and (3) some faults are covered by more than one fence. One should also note that any of these defensive fences may itself fail. The nodes are labeled (D*) 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 at locking out *external* attackers. However, such protection mechanisms alone are insufficient [19]. In large scale distributed systems such as the Internet routing infrastructure, it is impossible to build perfect protection and inevitable imperfect components can be compromised, opening holes in the system. Anomaly detection, syntax and semantics checking primarily provide essential detection mechanisms that notice when prevention has failed and also when unexpected faults occurs. Once a fault has been detected, reaction fences allows packet forwarding to continute(if it is possible). Each class of fences is then reviewed in subsequent sections.

As faults occur in the system, correspondingly defenses have been built at different levels, each covering part of the broader

solution space. Most, if not all, of the fences were built in response to faults experienced in the Internet and Figure 2 may serve as a useful guide to see each piece in the proper context.

V. Cryptographic Protection Schemes

The framework in Figure 2 outlines three types of cryptographic protection schemes: 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* is to distinguish between a valid router and an outside entity from imitating a legitimate router. However even with perfect secure communication and authentication, a legitimate router may still take incorrect actions such as advertising addresses it does not own or reporting false path/link information. The third type of protection adds *authorization* to the routing message exchange and is intended to restrict the actions of a legitimate router so that it can only originate routes to address blocks it owns and can 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 [19]. In a typical PKI, an *authorized third party* issues certificates according to some well defined hierarchal 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 [10] designed two early network layer protocols that rely on cryptographic protection techniques: Byzantine Robust Flooding, and Robust Link State Routing. But these protocols are primarily theoretical and do not scale to Internet topologies. The remainder of this section reviews the existing work on cryptographic protection fences.

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

OSPF2 [6] provides secure neighbor to neighbor communication using simple passwords and keyed MD5 authentication. Simple passwords are vulnerable to eavesdropping, but keyed MD5 authentication is effective in protecting the neighbor to neighbor protocol exchanges. However, it is not effective against insider faults, i.e. any faulty router involved in the flooding of a link state packet may modify its content. To address some insider faults, Murphy *et al.* proposed an approach in which the originator of a link state packet digitally signs this packet using its private key, an approach similar to that in [10]. 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 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 decide the "trusted entities", is not discussed.

But digital signatures alone cannot achieve perfect protection. There are still residual vulnerabilities, e.g. misconfiguration or compromised originating routers/private keys. Furthermore, its protection power is degraded when only partially deployed. Generating and verifying the digital signatures adds performance overhead and potential complexity and affects the deployment of such approaches. [20] proposed some mechanisms to reduce the cost of cryptographic protection for link state routing.

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

Smith *et al.* [11] modify BGP to provide secure neighbor-to-neighbor encryption using a session key and message sequence number. To protect against some insider faults, the proposal also includes an originating UPDATE sequence number (set by the origin AS) to protect against replayed UPDATE messages and a new attribute called *predecesor*(second AS on the AS path) is added to the route by the origin AS. This predecessor and originating UPDATE sequence number are signed by the private key of the origin AS. Some other fixed attributes such as ORIGIN, and AGGREGATOR can be also signed by the origin AS. Given the signed predecessor, one can then use path finding algorithms to reconstruct and verify the route to the destination [21]. Details of the reconstruction can be considered a form of semantics checking and are discussed in Section VIII.

However, this approach treats each AS an individual node in the graph and assumes an AS has unique and consistent predecessor information. In fact, an AS may exhibit more complex behavior and single AS may in fact use and advertise multiple distinct routes to a single destination (defeating the path finding algorithm). In addition, the approach leaves the public key distribution as a separate problem.

C. Origination Authorization: Secure Origin BGP(D1)

Secure Origin BGP (SoBGP) [13] was introduced by Ng and adds a protective fence to verify the origin of route advertisements and prevent the advertisement of unauthorized prefixes. In addition, some partial verification of AS path information is offered. A new type of BGP message, the SECURITY message, is used 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. Then Entity Certificate is signed by some authorities such as registries or other entities whose public keys has been pre-configured in the router. Once the public key in the Entity Certificate has been authenticated, Authorization Certificates are used to verify an AS is authorized to advertise an address block. A BGP update with an unauthorized origin AS is discarded and thus prevents a router from accepting updates with an unauthorized origin AS. Finally to protect the path, Policy Certificates contain a list of attached ASes and security policy options that allow a router to sanity check at least part of the AS path. Details of the checking can also be considered a form of semantics checking and are discussed in Section VIII.

D. Secure-BGP(D1)

Secure-BGP (S-BGP)[12], [22] by Kent *et al.* provides a comprehensive protective fence for BGP. IP security (IPsec) is used to secure the neighbor-to-neighbor communication between BGP routers, For authentication and authorization, S-BGP defines a detailed PKI. An Address Allocation PKI specifies the assignment

Work/Approach	Secure n2n communication (D1.1)	Authenticated information (D1.2)	Authorized information(D1.3)	PKI distribution	Key
OSPF with digital signature[8], [9]	MD5	LSA	no	no	special LSA
Origination and Predecessor [11]	session key, sequence number	origin, predecessor	no	assumed	assumed
SoBGP[13]	MD-5	origin, peering	address ownership	web of trust	Security msg
S-BGP[12]	IPSec	all BGP path attributes	ASpath announce, address ownership	follow address/AS allocation/delegation	out of band

Fig. 3. Comparison of Cryptographic Protection Schemes

of address blocks to organizations and binds address block(s) to a public key belonging to the corresponding organization. Another PKI is used for Assignment of AS Numbers and Router Associations and binds an organization's public key with its assigned AS number(s) as well as binding a router's public key with its ID, ASN and DNS name. These PKIs follow the existing hierarchy used to assign addresses and AS numbers.

S-BGP argues that sending certificates along with updates is not only bandwidth-wasteful, but also difficult if not impossible given the large number of certificates needed for each update and the current maximum BGP update length of 4096 bytes and distributing the certificates through a new BGP message, such as the SoBGP Security message, is considered not backward compatible by the S-BGP designers. Instead, S-BGP introduces repositories where the certificate database and revocation list are downloaded and distributed to the routers.

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

Using a daily average BGP update rate, Kent *et al.* [23] showed 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 BGP router, this overhead presents a concern. Furthermore, performance should also be evaluated using peak update rate since BGP suffers from updates storms due to session reset, and the peak update rate can vary dramatically from the daily average. Various performance improvement mechanisms, such as caching of validation results, delaying validation of backup paths, background validation of backup paths, and special cryptographic hardware running S-BGP, might be used to counter these problems. However, at this time, no detailed study presents these possible improvements. Finally, incremental deployment remains an open challenge for S-BGP. In order for an AS to verify the AS path using attestations, all the ASes in the path must have deployed S-BGP and local administrators are left to set the security policy for handling updates when some route attestations are missing or when some certificates are not available.

E. Summary of Cryptographic Protection Schemes

Both SoBGP and S-BGP are promising for adding protective fences to the current BGP routing infrastructure. S-BGP is the more comprehensive of the two, but pays a cost of more overhead and does not address incremental deployment. Figure 3 summarizes the different cryptographic approaches to adding protective fences.

All these approaches introduce protective fences, but also introduce new vulnerabilities. For example, private keys could be lost or stolen, or the registration and signing performed by an authorized third party can be manipulated. In general insider faults remain a challenge and many faults are not addressed by these protective fences. If viewed as complete solution, none of the approaches would provide a truly resilient routing infrastructure. However, the cryptographic protection fences clearly address some faults and are intended to form only part of the overall multi-fence approach.

VI. Statistical Anomaly Detection

Statistical anomaly detection is based on behavior profiles where a router or an auxiliary device keeps a statistical profile of the routing update messages. If the newly observed statistics don't fit the expected profile, alarms are raised. This type of fence depends heavily on the ability to devise a useful statistical profile. Figure 4 summarizes the reviewed work and places the work in the multi-fence defense framework shown in Figure 2.

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

Qu *et al.* [24] measure the inter-arrival time of all OSPF packets received by a router, the distribution of OSPF packet types, and the "age" of the LSA packets and then applies a statistical intrusion detection algorithm. Testbed experiments show this approach was very effective in detecting three known link attacks, the "seq++ attack", "Maximum Age attack", and "Maximum Sequence Number attack" identified by [26].

B. RIP Update Count Monitoring(D2.1)

Mittal *et al.* [25] use 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 sensor-based approach also employs protocol syntax checking (reviewed in Section VII) and protocol semantics checking (reviewed in Section VIII).

Work/Approach	Message Timing(D2.1)	Topo. Properties(D2.2)	Message Content(D2.3)
LS Anomaly Detection [24] (D2.1,D2.3)	message inter-arrival time	no no	message types ages of LSA
Sensor-Based [25](D2.1)	RIP update count	no	no
Path Filtering [15](D2.2,D2.3)	no	path stability of top level DNS servers	paths

Fig. 4. Comparison of Statistical Anomaly Detection.

C. Path-filtering Using Topology Properties(D2.2,D2.3)

Wang *et al.* [15] protect the routes to the critical top level DNS servers by restricting route changes to within a set of established paths, based on statistical analysis over history. The resulting path-filter exploits the observation that top level DNS servers are well connected via stable routes and also exploits the high degree of redundancy in top level DNS servers. Heuristics derived from routing operations are used to adjust the potential routes over time. The path-filter design was tested against BGP routing logs to show the design can effectively ensure correct routes to top level DNS servers without impacting DNS service availability.

VII. Protocol Syntax Checking

The routing protocol syntax defines the legitimate sequence of messages and is used to reject invalid messages. A common approach taken by routing protocols is to use heart beat messages to detect whether a neighbor is reachable (i.e. Hello in OSPF, Keep-Alive in BGP). BGP neighbor to neighbor peering also uses extensive syntax checking. However, broader syntax checking involving more than peer to peer communication is seldom used in a distributed system such as a routing protocol.

A. Finite State Machine for RIP and BGP(D3.1)

If the protocol syntax is well defined, syntax checking can be very effective at detecting hardware faults, implementation bugs, and so forth. In [28] and [27], Pei *et al.* use the protocol specification to construct finite state machines for RIP and BGP (respectively). This approach formalizes the protocol specification and only legitimate sequence of routing update messages are allowed by the corresponding state machine. Illegitimate message sequences resulting from implementation bugs, hardware faults can be detected by violations of the finite state machine transitions.

For example, the BGP protocol [16] uses a withdrawal update message to notify neighbors when a prefix is no longer reachable. Intuitively, a router should not withdraw a route it has never advertised. However, [31], [3] measured BGP updates and observed that a majority of BGP updates in the Internet were pathological update such as withdrawal messages for routes that were never advertised. Furthermore, a popular router implementation would further propagate the withdrawal messages. The Finite State Machine checking in [27] would immediately identify this type of illegitimate syntax and would have detected these pathological updates.

Furthermore, formal methods such as a state machine help the development of the protocol specification and eliminate ambiguities. For example, [28] identifies an ambiguity in current RIP protocol specification[18], RIP routers send both periodic announcements (every 30 seconds) and triggered update(when there is a change). After a triggered update is sent, a timer is started with a value between 1 second and 5 seconds. Before it expires, no other triggered update can be sent. On face value, this seems clear and unambiguous. However, the RIP standard does not specify the transition if a periodical announcement is due before this triggered update timer expires. Intuitively, RIP should clear the timer otherwise an unnecessary triggered update be sent shortly after the periodic announcement. The formal state machine helps identifies and resolve ambiguities that might result in implementation bugs or unintended behaviors.

B. Extended Timed Finite State Machines For Link-State Protocols(D3.1)

The $JiNao$ [29] architecture by Chang *et al.* provides real-time intrusion-detection in link state routing protocols such as OSPF. JiNao uses timed 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 an FSM for each known attack pattern work collectively to determine the state of the OSPF. Known attacks, such as the “seq++ attack”, “Maximum Age attack”, and ”Maximum Sequence Number attack” [26], are detected by FSMs using pattern matching. FSMs for newly discovered attacks can be added as the attacks are discovered.

C. BGP TTL Security Hack(BTSH)(D3.2)

Gill *et al.* [30] extend the BGP syntax by having each router check the TTL of BGP update messages and drop messages if the TTL is not in a legitimate range. External BGP peers are normally adjacent and if BGP routers configure the initial TTL to be 255, then received update message should have a TTL no less than 254. This extremely simple procedure is very effective in detecting false messages from more than one hop away. Each non-faulty router decreases the TTL by one and the false update (with TTL lower than 254) will be dropped by the intended target. This feature is especially effective in countering Denial-of-Service attacks against the BGP TCP port.

Mittal *et al.* proposed a similar approach [25] in which sensors sitting on a link check the link layer address and the TTL of the RIP update messages. (Semantics checking defined in the same work is reviewed in Section VIII.) In general, the simple TTL check demonstrates how protocol syntax checking can effectively counter any known or unknown DOS attack from more than one hop away.

D. Summary of Syntax Checking

Figure 5 summarizes the reviewed work and places the work in the multi-fence defense framework shown in Figure 2. Statistical anomaly detection can provide some clue about possible

Work/Approach	Protocol State Machine(D3.1)	Syntax Constrains from Operational Setting(D3.2)
BGP FSM(D3.1) [27]	derived from BGP specification	no
RIP FSM(D3.1) [28]	derived from RIP specification	no
Link State FSM(D3.1) [29]	Extended Timed FSM	no
BTSH (D3.2) [30]	no	TTL contrains of the message from BGP neighbors
Sensor-based(D3.2) [25]	no	TTL contrains of the message from RIP neighbors

Fig. 5. Comparison of Protocol Syntax Checking.

faults in message sequences, but these sequences might not be prohibited by the protocol specification. Protocol syntax checking is unique since no other fences can detect the illegitimate message sequences that should be prohibited according to the protocol specification (i.e. illegitimate message sequences caused by router implementation bugs). Furthermore, protocol syntax analysis using formal methods can help avoid bugs in the protocol design and help detect bugs in an existing protocol.

VIII. Protocol Semantics Checking

Protocol semantics checking uses the content of routing update messages to improve protocol behavior. In distributed routing protocols, one piece of routing information is often propagated throughout the network in multiple ways and one node may (explicitly or implicitly) receive multiple copies of the same information. Protocol semantics checking uses this information to derive properties from the protocol specification. Figure 6 summarizes the reviewed work and places the work in the multi-fence defense framework shown in Figure 2.

A. Assertions(D4.2, D4.3)

In his PhD thesis [32], Massey divides the Internet into compartments and faults are detected at the boundary of the compartments using pre-defined assertions. Assertions are conditions which must hold true if nothing goes wrong. The assertion approach is extensible since new assertions could be easily added. The technique is used to identify and detect a number of faults in distance vector multicast routing algorithms and some general framework is provided for arbitrary protocols.

The assertion approach was then applied to improve BGP's convergence in [33]. A consistency theorem is developed based on path vector's semantics, and says that two paths having one common node should have the consistent sub-paths to the destination. Two assertions(Route Withdrawal and Route Change) are derived from the consistency theorem in order to detect and outdated route information. To apply these assertions in BGP, some additional information are propagated to signal failure/policy withdrawal and address traffic engineering. Although this assertion approach cannot eliminate *all* the transient changes during BGP convergence, it reduces the convergence time substantially in simulation results.

B. Property Oriented Fault Detection(D4.3)

Wang *et al.* [34] apply an approach similar to assertions for Link State Routing Protocols. A centralized monitoring process p_0 collects all the routing message exchanges between routing processes and track the states kept in each routing process. p_0 detects problems based on a snapshot of a global state of the routing processes using a few properties that must hold true for global state in link state protocol. In the event a property does not hold, some more detailed properties can be used to diagnose where exactly the fault happened. The authors provide two case studies to detect the "seq++ attack" and "Maximum Sequence Number attack". However while Massey's assertion checking is done within each single compartment or via message exchanges between compartments, Wang's approach uses a centralized process and its applicability to a large network is limited due to its centralized design.

C. RIP-TP Triangle Checking (D4.3,4.4)

RIP with *T*riangle theorem checking and *P*robing message (RIP-TIP) [35] uses the very limited information exposed by RIP to check a simple triangle theorem. The theorem states that given a set of 3 nodes in a shortest path protocol, the distance between one pair of nodes should be always less than the sum of the distance of the other two pairs. However, delays in route convergence, message loss or update message delay, may cause a temporary violation of the triangle theorem. To distinguish temporary delays from faults, probing messages are sent to the destination to verify the suspicious routing update. Through simulations, the approach is shown to be effective in detecting false updates with low overhead. An additional advantage is that RIP-TP is incrementally deployable and any node that implements this approach can benefit without the support of any other nodes.

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

Smith *et al.* [11] add a new attribute called the predecessor (second AS in the AS path) to route updates. Section V discussed how signed predecessor data is used to prevent false routes. The predecessor also can be viewed as a type of semantics checking. Following a path-finding algorithm [21], each node in the network can learn and authenticate the path to a destination. This approach works well for a pure distance vector protocol where every node is a destination (such as RIP), but in BGP a link that is legitimate for one destination might be illegitimate for another destination due to routing policy and path finding does not fit well.

E. Sensor Monitoring With Global Knowledge(D4.1,D4.4)

Mittal *et al.* [25] detect faults in a RIP network using sensor, and its elements of anomaly detection and protocol syntax were discussed in previous sections. Sensors are placed on some (or all) of the links and each sensor is given the whole network topology as well as the positions of all other sensors. A sensor computes all the possible paths from each router to each subnet by essentially running a link state protocol on the manually configured topology. A sensor then analyzes the routing updates on its links and the update's semantics (i.e. distance) are checked against the sensor's set of all possible distances. If a distance is not in the legitimate

Work/Approach	Pre-configured (D4.1)	Newly Propagated (D4.2)	Utilizing existing Info. (D4.3)	Query (D4.4)
Assertion [32], [33]	no	policy withdrawal info. traffic engineering info.	assertions	yes
POD[34]	no	no	properties	no
RIP-TP [35]	no	no	triangle theorem	probing message
Predecessor [11]	no	predecessor	path-finding	no
Sensor [25]	global topology	no	no	between sensors
SoBGP[13]	no	peering relationship	no	no
MOAS [14]	no	MOAS list	piggybacked in updates	IRR/DNS-based
IRR	no	no	no	centralized database
IRV[36]	no	no	no	distributed IRV servers

Fig. 6. Comparison of Protocol Semantics Checking approaches

range, an alarm is raised. Otherwise, a query is sent to all the sensors along the possible path(s) that have this distance in order to verify the distance. This has major drawbacks for practical deployment since it implicitly requires static network topology, static sensor placements, and each sensor has to compute all the possible paths for each router to each destination. But this work introduces a mechanism which needs no modification to the router and is also a good example of multi-fence approach that includes statistical anomaly detection, syntax checking and semantics checking.

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

Section V discussed how SoBGP uses signed Policy Certificate to detect false paths, but the information can be viewed as a form of semantics checking. Every AS lists its AS-level neighbors through the Policy Certificate and these certificates can be used to build a directed graph that provides a view of the Internet. If a link in the received update message does not exist in the directed graph, the update will be considered invalid. Note that, even if each link in the update exists in the directed graph, the update could still be invalid since this link might not be available for the specific prefix in the update. In other words, SoBGP constructs a superset of possible AS level links and then check any routing update against this superset. But some ASes may not be willing to announce their local connectivity through Policy Certificates since this information may be considered confidential in the current Internet. The resulting set of links at each node is no longer a superset and some valid links might be considered invalid. To cope with this, SoBGP introduces a security preference, giving the update that failed the check a lower preference rather than excluding the route.

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

Zhao *et al.* [14] present a non-cryptographic approach to protect BGP against route origin spoofing. In BGP, a destination may appear to have multiple origin AS (MOAS) due to multi-homing, misconfigurations, or even attacks. [14] adds a MOAS attribute that contains the complete list of legitimate origins and this attribute is attached whenever an origin AS announces the route. There is no cryptography so the MOAS list can be altered by faults or an invalid origin may attach an arbitrary MOAS list. However, the rich network connectivity makes it very difficult to block all the correct MOAS lists from reaching a particular node. At each node, the MOAS lists learned from different peers are compared and all the MOAS lists should be same (if no faults occurred). Simulation results in realistic topologies shows this non-cryptographic result is very effective in detecting the MOAS conflicts. After detection of a MOAS conflicts, [14] needs some further information in order to know which of the conflicting MOAS lists is actually correct. This solution does not impose a fixed verification solution but rather list a few possible solutions such as consulting the IRR (http://www.irr.net) or using DNS-based origin verification [37].

H. The Internet Routing Registry(IRR)(D4.4)

The Internet Routing Registry(IRR, http://www.irr.net) places policy data collected from ASes into a small number of databases. BGP routers can then compare the received routes to the information listed in the database and any conflicting routes can be discarded. But as with SoBGP, ISPs may not be willing to publish their ASes' policy information and this limits the IRR's effectiveness. Also, an administrator may forget to update the database so existing data can be obsolete. The Routing Policy System Security[38] was proposed to make IRR more secure, but the information stored in IRR is still not digitally signed. Therefore, data in IRR itself is vulnerable to malicious attacks or simply human errors.

I. Inter-domain Routing Validation (D4.4)

The Inter-domain Routing Validation(IRV) [36] protocol by Goodell *et al.* can be used in conjunction with BGP. IRV provides a way to solicit information for semantics checking, and could be used together with approaches such as the MOAS list approach or could be used to help with partially deployment problems in SoBGP and S-BGP. Each participating AS designates an IRV server that answers queries regarding the AS routing policy, whether it originates a particular prefix, a report the BGP updates recently received from its neighbors, current BGP routes, and BGP updates sent to neighbors. The query results could be used for validating a routing policy or confirming the originator of a prefix or AS path information. IRV does not define when queries are to be sent and some control mechanism has to be devised to avoid problems such as a large number of ASes sending queries simultaneously to the same IRV server. Furthermore, a query may be based on data older than those in IRV, therefore mechanisms have to developed 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.

IX. Reaction Fences

Given the sheer scale and imperfect components of the Internet, faults are inevitable. Once a fault has been detected, a resilient protocol should react to faults in a way that allows packet forwarding to continue (if this is possible). Early protocol designs assumed a simple fail-stop fault model. In this model, fault detection fences provide an unambiguous statement that some component has failed and the corresponding reaction was clear: route around the failed component. But in a more complex environment, detection may be less precise and the corresponding reaction is no longer obvious. For example, an alarm raised by an anomaly detection system may not indicate an obvious reaction. In this section, we review some new approaches to building reaction fences. At all the times, the main objective of a resilient routing system is to react in way that allows packet forwarding to continue.

A. Forwarding Packets During Routing Faults(D5.1)

In perhaps the simplest example of reaction, an OSPF router detects the failure of a link through the lack of periodic "hello" exchanges and reacts to this fault by routing the around the failure. However, if the "failed" router has only stopped its routing process, forwarding through the link could continue and a protocol that reacts by declaring the router down may trigger an unnecessary period of route convergence. In this case, the route recomputation of caused by the node "failure" and subsequent node "addition" is unnecessary since the routing state before and after the reboot is the same and forwarding stills functions during the reboot. Rather than declaring the router down, packet forwarding might be better served by continuing to forward packets to the "failed" router.

[39] and [40] define more complex reactions that allow a router to continue forwarding packets even when its routing process is restarting. A router signals that its routing process will be temporarily unavailable, but forwarding can continue. Since the router cannot participate the routing messages exchanges and thus cannot update its forwarding table, it is possible routing loops may occur if there are topological changes during the restart period. [39] proposes a conservative reaction to avoid the loops during this period. If there are no other topological changes in the network, the restarting router's neighbor will continue using the restarting router. If changes occur, the neighbors will send out link-state messages to notify the rest of the network that the link to the restarting router is not available. [40] proposes a similar but less conservative approach. Neighbors of the restarting router decide whether the changes in the network will lead to loops. If so, they will send out link-state messages to notify the rest of the network that router has failed.

Similarly, BGP reacts to the loss of a peering session by assuming the neighboring router has failed. The temporary loss of peering session can thus trigger a large number of routing changes, even if the peering session has merely experience a transient failure. BGP graceful restart[41], [42] add mechanisms to notify the neighbors that a router can continue with its forwarding functionality while the routing protocol software is unavailable and avoid having to withdraw and then re-announce all BGP routes exchanged with the peer.

B. Limiting Updates Dynamics(D5.2)

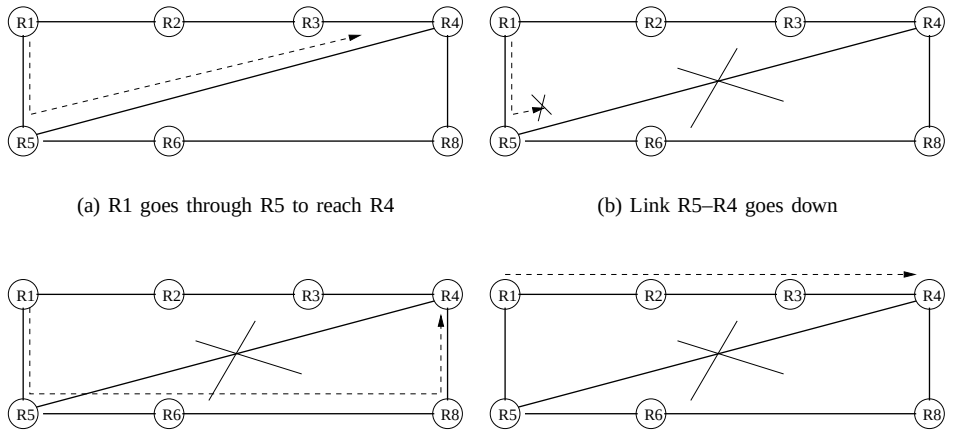
In BGP a single routing updating can trigger a slow convergence. [43] found that route failovers and route failures resulted in a significant delay. This delay averaged three minutes and some changes took up to 15 minutes in the current Internet. This work also showed that, in the theoretically worst case, a fully connected n-AS system might explore all ($n!$) possible paths before BGP converged on a new set of stable routes. During a slow convergence period, BGP routers may exchange a large number of updates and packets may be dropped due to loops or invalid transient routers. To counter these problems, BGP adds two reactive fences that limit update dynamics on different time scales.

At the scale of seconds, the BGP specification [16] requires that a minimum amount of time, denoted *MinRouteAdver*, must elapse between route advertisements for a specific destination. The design assumes that during convergence, a changed route is likely to change again in a brief interval [17] and rate-limiting allows BGP routers to "pack" consecutive updates. If a route changes multiple times during a $MinRouteAdver$ period, the updates are packed together and only the last change should be announced. Simulation results by Griffin *et al* [44] illustrated the necessity of having the $MinRouteAdver$ timer: eliminating the $MinRouteAdver$ timer would lead to not only unacceptably large number of updates but also unacceptably long convergence time.

The route flap damping mechanism defined in RFC 2439 [45] acts at a time scale of minutes and is designed to limit the global impact of an unstable link. Routers maintain a penalty value for each prefix learned from a peer. Route announcements increase the penalty value and if the value exceeds a threshold, route announcements for the prefix from that peer are surpressed. The penalty value decays using a half-life value. Although *MinRouteAdver* and route flap damping operate on different time scales, Griffin *et al* [44] pointed out the need to understand if route damping is invoked due to slow convergence. Mao *et al* showed that such behavior actually occurred in their testbed[46] and also showed that using the flap damping parameters recommended by RIPE routing Working Group[47], a route to a prefix that is first withdrawn and then re-announced 500 seconds later could be suppressed for up to an hour. This work illustrates that one should be careful when designing new reactive mechanisms in order to avoiding adding conditions where the new mechanism interacts badly with other existing mechanisms.

C. Utilizing Rich Connectivity(D5.3)

The primary goal of any resilient routing design is to continue forwarding packets despite faults. In response to a fault, the set of possible reactions and the effectiveness of these reactions is often determined by the underlying network topology. Over the last few years, there has been a rapid decrease in bandwidth cost and increase in network speed. The cost reduction had led to richer and richer Internet connectivity[50]. The increased connectivity offers



(a) R1 goes through R5 to reach R4

(b) Link R5–R4 goes down

(c) R5 switches to R6 to reach R4

(d) R1 switches to R2 to reach R4

Fig. 7. Packet could still be delivered during convergence

Work/Approach	Forwarding Despite Faults(D5.1)	Limiting Update Dynamics(D5.2)	Utilizing Rich Connectivity(D5.3)
Hitless OSPF Restart(D5.1) [39]	terminate hitless restart when topology changes	no	no
Avoiding OSPF instability(D5.1) [40]	terminate hitless restart if there is danger of loops	no	no
BGP Graceful Restart(D5.1) [41]	avoid withdrawing and re-announing	no	no
MinRouteAdver Timer(D5.2) [16]	no no	one update per prefix per 30 seconds	no
BGP route damping(D5.2) [45]	no	penalize flapping routes	no
Fast Re-route in LS(D5.3) [48]	no	no	backup-path stored in line-card
Packet Dynamics[49]	no	no	keep backup path propagate new info. ASAP

Fig. 8. Comparison of Reaction Fences.

multiple alternative paths in case of any component failure. At the same time, studies also show that rich network connectivity may make the current routing protocol BGP take longer to converge [51] and higher link bandwidths mean more packets "on the fly" at any given time. The continued growth of the network makes the frequency of failures higher and convergence time longer, but rich connectivity can improve the protocols ability to react to a fault.

Figure 7 show how rich connectivity allows a protocol to react to a fault and continue forwarding packets during protocol convergence. In this figure, each link has a unit cost and the data forwarding path between R1 and R4 is shown in dashed lines. Initially in Figure 7(a), R1 sends packets to R4 along the correct shortest distance route. In Figure 7(b), the link R5-R4 goes down. R1 continues to forward packets to R5 and packets are lost as R5 routes the packets over the failed link. In Figure 7(c), R5 detects the link failure and switches to the new shortest path by forwarding packets to R6. R1 has yet to learn of the change and continues to forward packets to R5. The packets follow a non-shortest path but still reach the destination. Finally, in Figure 7(d), R1 converged to the new shortest path, and forwards packets to R2. Note that packets are only dropped in time period after the link failure occurred and before R5 switched to a next hop of R6 (Figure 7(b)). During some of the convergence period (Figure 7(c)) packets follow a non-converged path route but still reach the destination.

In the above example, packets can continue to be forwarded during a fault, but a router cannot forward any packets after the previous next-hop is removed and before a new next-hop is computed. In [48], Alaettinoglu *et al* proposed to compute and install one backup next-hop in the line-card of the router and the backup next hop is used when the primary next hop is removed from the line card. This allows the router to have a backup readily available and the backup is present in the line card as well as being present in the routing process.

In [49], Pei *et al*. performed protocol analysis and simulations to examine how well existing routing protocols deliver packets during convergence and identify the protocol design issues that maximize packet delivery during convergence. The authors simulate three routing protocols, RIP, Distriubted Bellman-Ford

algorithm, and BGP in networks with different topological connectivity. The study shows that, in order to assure reliable packet delivery, the network must have adequate physical redundancy and the packet delivery ratio improves as the network connectivity becomes richer with all the routing protocols examined.

The study also shows that reactive fences can be better designed to fully utilize network redundancy in face of component failures. Among the three routing protocols studied, some reactions take better advantage of the rich connectivity than the others and two factors had the most impact on packet delivery performance during routing convergence. First, in addition to a best path, a router should also keep information on alternative paths to each destination. Even if an alternate is not the best available path after a failure, some alternative path should be readily available after detecting the failure of the best path. The study notes that increases in network connectivity reduce the probability that the alternate path will include the failed link. Second, once a change of connectivity is detected, the routing protocol should propagate the new information as fast as possible. This not only helps minimize the convergence time, it also helps improve the chance of delivery for those packets that are en-route at the time of failure.

D. Summary

Once the assumptions of a simple fail-stop model are no longer in use, reaction becomes a more complex problem. Protective fences help some faults from occurring in the first place, but it is not possible to lock at all faults. Detection helps identify faults, but in more complex systems the detection may not be unambiguous and the response is not immediately clear. Reactive fences use the input from detection and attempt to respond in way allows packet forwarding to continue. New approaches to reaction, such as continuing to forward packets through a router whose routing process in unavailable, attempt to continue packet delivery despite faults. Other reactive fences such as route damping react by attempting to limit the global impact of fault. The network topology often plays a critical role in determining how effective a reaction may be and designing protocols that make use of topological properties is a promising new area of research. Figure 8 summarizes the reviewed work in this section.

X. Summary

In this article we reviewed the various approaches to improving the resiliency of the Internet routing protocols. Figure 9 provides a summary of the approaches. By examining both the routing faults that occurred in operational Internet and the mechanisms to protect the routing infrastructure, we can 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, however they cannot detect or prevent all types of faults such as implementation bugs, configuration errors, or compromised routers. Furthermore, Cryptographic mechanisms themselves are 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 be used to provide a stronger overall system.

A. 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 not only increase the number of physical failure events, but will also increase the number of operational errors and unexpected faults. Furthermore, its importance in the society will attract more intentional attack. In such a complex and hostile environment, no single protection or detection mechanism can be adequate. Instead we must build a multi-fence defense system to assure a resilient Internet routing infrastructure.

At the same time, we recognize inevitable trade-offs. Any new piece we add to a system can introduce potential new vulnerabilities and new interferences. For example, public-key cryptographic mechanisms protect the protocol from outsider attacks, but also introduces a new dependency on PKI. Whether to add a new piece to the protocol is therefore a trade-off between the benefits and the new vulnerabilities. Another trade-off is fault detection capability versus performance scalability. Detection usually benefits from more information, especially more global information. However, propagating global information in a large system adds performance overhead. More performance overhead is usually unavoidable, but a good solution's performance overhead should be scalable as the system size increases. This is especially true for a system as large as the Internet, whose size keeps increasing over years. A final challenge is partial deployability of new protection and detection mechanisms since any new Internet fences will face partial deployment issues 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," *IEEE Network Special Issue on Protection, Restoration, and Disaster Recovery*, 2004.

[2] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, "Analysis of Link Failures in an IP Backbone," in *Proceedings of ACM IMW 2002*, October 2002.

[3] C. Labovitz, A. Ahuja, and F. Jahanian, "Experimental Study of Internet Stability and Wide-Area Network Failures," in *Proceedings of FTCS99*, June 1999.

[4] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding bgp misconfiguration," in *Proceedings of ACM Sigcomm*, August 2002.

[5] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. Wu, and L. Zhang, "Observation and Analysis of BGP Behavior under Stress," in *Proceedings of the ACM IMW 2002*, October 2002.

[6] J. Moy, "OSPF Version 2," SRI Network Information Center, RFC 2328, September 1998.

[7] A. Heffernan, "Protection of BGP Sessions via the TCP MD5 signature option," SRI Network Information Center, RFC 2385, August 1998.

[8] S. Murphy and M. Badger, "Digitial Signature Protection of the OSPF Routing Protocol," in *Symposium on Network and Distributed System Security*, 1996.

[9] S. Murphy, M. Badger, and B. Wellington, "OSPF with Digitial Signatures," September 1997.

[10] R. Perlman, "Network layer protocols with byzantine robustness," Ph.D. dissertation, MIT Lab. for Computer Science, 1988.

[11] B. R. Smith, S. Murphy, and J. J. Garcia-Luna-Aceves, "Securing the border gateway routing protocol," in *Global Internet'96*, November 1996.

[12] S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol (s-bgp)," *IEEE JSAC Special Issue on Network Security*, 2000.

[13] J. Ng, "Extensions to BGP to Support Secure Origin BGP," October 2002, http://www.ietf.org/internet-drafts/draft-ng-sobgp-extensions-00.txt.

Work/Approach	Cryptographic Schemes	Statistical Anomaly Detection	Protocol Syntax Checking	Protocol Semantics Checking	Reaction
OSPF with digital signature[8](D1.1,D1.2)	Signed LSA MD-5(D1.1,D1.2)	no	no	no	no
Origin,Predecessor [11] (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)	no
SoBGP[13](D1,D4.2)	address ownership(D1)	no	no	peering map(D4.2)	no
S-BGP[12](D1)	IPsec, AA, RA(D1)	no	no	no	no
LS Anomaly detection [24](D2.1,D2.3)	no	yes(D2.1,D2.3)	no	no	no
Path Filtering [15] (D2.2,D2.3)	no	topology property (D2.2,D2.3)	no	no	no
BGP FSM[27](D3.1)	no	no	from BGP spec	no	no
RIP FSM[28](D3.1)	no	no	from RIP spec	no	no
LS FSM [29](D3.1)	no	no	known attacks(D3.1)	no	no
BTSH[30](D3.2)	no	no	TTL(D3.2)	no	no
Assertion[32], [33] (D4.2,D4.3)	no	no	no	assertions(D4.2,D4.3)	no
POD[34](D4.3)	no	no	no	properties(D4.3)	no
RIP-TP [35] (D4.3,D4.4)	no	no	no	triangle theorem probing message(D4.3,D4.4)	no
Sensor [25] (D2.1,D3.2,D4.1,D4.4)	no	update count (D2.1)	TTL, Link Layer Address(D3.2)	with pre-configured global knowledge (D4.1,D4.4)	no
MOAS [14] (D4.2,D4.3,D4.4)	no	no	no	Checking MOAS list (D4.2,D4.3,D4.4)	no
IRR (D4.4)	no	no	no	centralized database for query(D4.4)	no
IRV[36] (D4.4)	no	no	no	distributed IRV servers for query(D4.4)	no
Hitless OSPF Restart in [39](D5.1)	no	no	no	no	terminate if topo changes
Hitless OSPF Restart in [40](D5.1)	no	no	no	no	terminate if loops is likely
BGP Graceful Restart(D5.1) [41]	no	no	no	no	avoid withdrawing/ re-announing
MinRouteAdver Timer(D5.2) [16]	no	no	no	no	one update per prefix per 30 seconds
BGP route damping(D5.2) [45]	no	no	no	no	penalize flapping routes
Fast Re-route in LS(D5.3) [48]	no	no	no	no	backup-path stored in line-card
Packet Dynamics[49]	no	no	no	no	keep backup path propagate new info. ASAP

Fig. 9: Summary of Reviewed Work Labeled With Their Defense Fences.

[14] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang, "Dection of Invalid Routing Announcement in the Internet," in *Proceedings of the IEEE DSN 2002*, June 2002.

[15] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. Wu, and L. Zhang, "Protecting BGP Routes to Top Level DNS Servers," in *Proceedings of the ICDCS 2003*, 2003.

[16] Y. Rekhter and T. Li, "Border Gateway Protocol 4," SRI Network Information Center, RFC 1771, July 1995.

[17] C. Huitema, *Routing in the Internet*. Prentice-Hall, 2000.

[18] G. Malkin, "Routing Information Protocol Version 2," SRI Network Information Center, RFC 2453, November 1998.

[19] B. Schneier, *Secrets and Lies–Ditgital Security in a Networked World*. John Wiley and Sons, Inc., 2000.

[20] R. Hauser, T. Przygienda, and G. Tsudik, "Reducing the Cost of Security of Link-State Routing," in *Symposium on Network and Distributed System Security*, 1997.

[21] J. J. Garcia-Lunes-Aceves and S. Murthy, "A Loop-Free Path-Finding Alogirthm: Specification, Verification and Complexity," in *Proceedings of the IEEE INFOCOM*, April 1995.

[22] C. Lynn, J. Mikkelson, and K. Seo, "Secure BGP (s-bgp)," October 2002, http://www.ietf.org/internet-drafts/draft-clynn-s-bgp-protocol-00a.txt.

[23] S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol (s-bgp)–real world performance and deployement issues," in *NDSS*, 2000.

[24] D. Qu, B. Vetter, F. Wang, R. Narayan, S. F. Wu, Y. F. Jou, F. Gong, and C. Sargor, "Statistical anomaly detection ofr link-state routing protocols," in *Proceedings of ICNP*, November 1998.

[25] V. Mittal and G. Vigna, "Sensor-based intrusion detection for intra-domain distance-vector routing," in *ACM CCS's 02*, November 2002.

[26] B. Vetter, F. Wang, and S. Wu, "An experimental study of insider attacks for the ospf routing protocol," in *Proceedings of the INCP 1997*, October 1997.

[27] D. Pei, D. Massey, and L. Zhang, "Finite State Machines for BGP Protocol," UCLA CSD, Tech. Rep., Feburary 2003.

[28] ——, "Formal Specificaiton of RIP Protocol," UCLA CSD, Tech. Rep., 2003.

[29] H. Chang, S. F. Wu, and Y. F. Jou, "Real-time protocol analysis for detecting link-state routing protocol attacks," *ACM Transactions on Infromation and System Security*, vol. 4, no. 1, pp. 1–36, Feburary 2001.

[30] V. Gill, J. Heasley, and D. Meyer, "The bgp ttl security hack (btsh)," December 2002, http://www.ietf.org/internet-drafts/draft-gill-btsh-01.txt.

[31] C. Labovitz, G. Malan, and F. Jahanian, "Internet Routing Instability," in *Proceedings of ACM Sigcomm*, September 1997.

[32] D. Massey, "Fault Detection and Security in Routing Protocols," Ph.D. dissertation, University of California, Los Angeles, 2000.

[33] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, F. S. Wu, and L. Zhang, "Improving BGP Convergence Through Assertions Approach," in *Proceedings of the IEEE INFOCOM*, June 2002.

[34] F. Wang, F. Gong, and S. Wu, "A property oriented fault detection approach for link state routing protocol," in *Proceedings of the ICCCN 200*, October 2000.

[35] D. Pei, D. Massey, and L. Zhang, "Detection of False Routing Update in RIP," in *IEEE Globecom*, December 2003.

[36] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin, "Working around bgp: An incremental approach to improving security and accuracy of interdomain routing," in *NDSS*, 2003.

[37] T. Bates, R. Bush, T. Li, and Y. Rekhter, "Dns-based nlri origin as verfification in bgp," 1998, http://www.nanog.org/mtg-9802.

[38] C. A. C. Villamizar, D. Meyer, and S. Murphy, "Routing Policy System Security," SRI Network Information Center, RFC 2725, December 1999.

[39] J. Moy, P. Padma, and A. Lindem, "Hitless OSPF Restart," October 2002, http://www.ietf.org/internet-drafts/draft-ietf-ospf-hitless-restart-04.txt.

[40] A. Shaikh, D. Dube, and A. Varma, "Avoiding Instability during Shutdown of OSPF," in *Proceedings of the IEEE INFOCOM*, June 2002.

[41] S. R. Sangli, Y. Rekhter, R. Fernando, J. Scudder, and E. Chen, "Graceful Restart Mechanism for BGP," October 2002, http://www.ietf.org/internet-drafts/draft-ietf-idr-restart-05.txt.

[42] I. Cowburn, "Toward a Hitless Network: BGP Graceful Restart," Riverstone Technology, Tech. Rep. 134, 2002, white Paper.

[43] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet Routing Convergence," in *Proceedings of ACM Sigcomm*, August 2000.

[44] T. Griffin and B. Premore, "An Experimental Analysis of BGP Convergence Time," in *Proceedings of ICNP*, November 2001.

[45] C. Villamizar, R. Chandra, and R. Govindan, "BGP Route Damping," SRI Network Information Center, RFC 2439, May 1998.

[46] Z. Mao, R. Govindan, G. Varghese, and R. Katz, "Route Flap Damping Exacerbates Internet Routing Convergence," in *Proceedings of ACM Sigcomm*, August 2002.

[47] C. Panigl, J. Schmitz, P. Smith, C. V. Chandra, and J. Scudder, "RIPE routing-gw recommendations for coordinated route-flap damping parameters," RIPE, Tech. Rep. 229, October 2001.

[48] C. Alaettino and A. Zinin, "Igp fast reroute," 2002, talk slides, http://www.packetdesign.com/publications.

[49] D. Pei, L. Wang, D. Massey, S. F. Wu, and L. Zhang, "A study of packet delivery performance during routing convergence," in *IEEE DSN*, June 2003.

[50] G. Huston, "The State of BGP Routing," http://www.ietf.org/proceedings/01mar/slides/plenary-2/index.html.

[51] R. Bush, T. Griffin, and Z. M. Mao, "Route Flap Damping: Harmful?" 2002, http://www.ripe.net/ripe/meetings/archive/ripe-43/presentations/ripe43-routing-flap.pdf.