FINAL REPORT DARPA ATO FTN PROJECT

SCALABLE PROTECTION AGAINST DDOS AND WORM ATTACKS

PI: KIHONG PARK

RESEARCH TEAM: HYOJEONG KIM, BHAGYA BETHALA, ALI SELCUK

NETWORK SYSTEMS LAB
DEPARTMENT OF COMPUTER SCIENCES
PURDUE UNIVERSITY
WEST LAFAYETTE, IN 47907

This document is the final project report of DARPA ATO FTN project "Toward Scalable Solutions for Distributed Denial of Service Attack Prevention" (Program Manager: Dr. Douglas Maughan) under AFRL contract F30602-01-2-0530 (Agent: Alan Akins). Questions and comments should be directed to Kihong Park (park@cs.purdue.edu).

ABSTRACT

We address two pressing challenges facing network security today: distributed denial of service (DDoS) and worm attacks. We advance new solutions aimed at providing scalable defenses against these potentially debilitating cyber threats. We achieve two complementary modes of protection: one, proactive protection that prevents attacks from imparting harm in the first place, and two, reactive protection that locates the physical source of an attack and adapts to unforseen vulnerabilities.

The solutions are based on a new approach to network security—distributed packet filtering (DPF)—that casts a "filter net" over the network system which stops attack traffic in its tracks. Scalability is afforded by the small size of the filter net: with only 15% deployment for DDoS and 4% for worm, DPF is able to achieve overwhelming protection. Efficacy under partial deployment, a key requirement of any viable solution, is made possible by the recently discovered power-law connectivity of the Internet.

Performance evaluation of DPF using large-scale Internet topologies is carried out with DaSSF-Turbo, a scalable network simulation environment developed as part of the project. DaSSF-Turbo is a performance-oriented extension of DaSSF and facilitates Internet-scale benchmarking through automated network configuration, performance monitoring, and power-law partitioning.

ACKNOWLEDGMENT

A number of people have contributed to the project. The principal contributors are: Hyojeong Kim, Bhagya Bethala, and Ali Selcuk. Humayun Khan contributed to the BGP port of DaSSF-Turbo from SSFNet. Heejo Lee contributed to the early foundational part preceding the DARPA project. CERIAS, Purdue's security center, augmented the funding from DARPA through an internal grant. Xerox aided the PI's security and QoS efforts through gift grants, and Intel donated a 7-node IXP1200 network processor environment for DPF prototyping research. The hardware testbed for large-scale simulation benchmarking was procured using an NSF RI grant. ETRI is a sponsor of the PI's active security framework.

The project has benefited from collaboration with Cole Smith (Univ. of Arizona) on the complexity of optimal filter placement and David Nicol's (Dartmouth College/Univ. of Illinois) DaSSF simulator support.

TABLE OF CONTENTS

ABSTRACT **2**

ACKNOWLEDGMENT **3**

LIST OF FIGURES **6**

PROJECT SYNOPSIS **8**

KEY CHALLENGES OF NETWORK SECURITY 8
REQUIREMENTS FOR A VIABLE SOLUTION 8
DISTRIBUTED PACKET FILTERING: A NEW PARADIGM 9
MISSING LINK: POWER-LAW CONNECTIVITY 9
OBJECTIVE: SCALABLE NETWORK SECURITY 10
NEW CONTRIBUTION 11
Scalable DDoS Attack Protection *11*
Scalable Worm Attack Protection *12*
Performance Evaluation Tools *13*

TECHNICAL CONTRIBUTION: DDOS PROTECTION **16**

WHAT IS ROUTE-BASED PACKET FILTERING 16
DISTRIBUTED PACKET FILTERING & DDOS PROTECTION 17
DDoS Performance Metrics *17*
Route-Based DPF Performance *17*
POWER-LAW NETWORKS AND FILTER PLACEMENT 18
Power-Law Connectivity *18*
Filter Placement Strategy: Vertex Cover *19*
SCALABLE FILTER DEPLOYMENT 19
Partial Deployment: Power-Law Vs. Random Topology *19*
Incremental Filter Deployment *20*
SCALABLE PROACTIVE PROTECTION: CONTAINMENT 21
Containment: Power-Law Vs. Random Topology *21*
Maximal Filtering and Non-Shortest-Path Routing *22*
SCALABLE REACTIVE PROTECTION: TRACEBACK 22
Traceback: Power-Law Vs. Random Topology *22*
Maximal Filtering and Non-Shortest-Path Routing *23*
PERFORMANCE RESULTS: ROBUSTNESS 24
Other Internet Measurement Graphs: CAIDA, RIPE, USC/ISI, UMich *24*
Artificial Internet Topology: Inet *25*
OPTIMIZATION: REDUCED FILTER DEPLOYMENT 27
ROUTE-BASED DPF PROTOCOL 27
Route-Based Filter Table Look-Up *28*
Route-Based Filter Table Update: Issues *28*
Route-Based Filter Table Update: Protocols *29*
INFRASTRUCTURE PROTECTION AND RESILIENCE 34
Infrastructure Attack *34*
BGP Convergence Dynamics *35*
Route-Based DPF Resilience: Safety *36*
Route-Based DPF Resilience: Staleness *36*
Route-based DPF Resilience: Containment and Traceback *37*

TECHNICAL CONTRIBUTION: WORM PROTECTION **39**

CONTENT-BASED DISTRIBUTED PACKET FILTERING 39

SPARSE FILTER PLACEMENT 40
SCALABLE CONTAINMENT: CRITICAL FILTER DENSITY 40
PROACTIVE PROTECTION: ROBUSTNESS 42
REACTIVE PROTECTION: TRACEBACK 42
PROTECTION UNDER ROUTING REACHABILITY 43
FINITE TIME HORIZON DYNAMICS 44
Worm Propagation Dynamics *44*
Finite Time Critical Filter Density *45*
REFERENCES AND RELATED WORK 47
REPORTS AND PUBLICATIONS FROM THE PROJECT 47
Published Papers and Reports *47*
Papers in Preparation *47*
POWER-LAW NETWORKS 48
Power-Law Network Modeling and Analysis *48*
Power-Law Measurement Topology *49*
DDOS REFERENCES 49
WORM REFERENCES 51
SCALABLE SIMULATION REFERENCES 53

LIST OF FIGURES

FIGURE 1 LAKE WOBEGON UNDER LOCAL FILTERING (LEFT) AND DISTRIBUTED FILTERING (RIGHT). 9
FIGURE 2 LEFT: A 300-NODE INTERNET DOMAIN GRAPH. RIGHT: A 300-NODE RANDOM GRAPH. 10
FIGURE 3 OVERALL ARCHITECTURE OF DASSF-TURBO. 15
FIGURE 4 SPOOFED DOS ATTACK FROM A TO H. LEFT: SEMI-MAXIMAL RF. RIGHT: MAXIMAL RF..... 16
FIGURE 5 FILTER NET IMPLEMENTING ROUTE-BASED DPF IN POWER-LAW NETWORK. 18
FIGURE 6 INGREDIENTS OF POWER-LAW MOLECULAR STEW. 18
FIGURE 7 LEFT: INTERNET AS GRAPH EVOLUTION 1997-2002. RIGHT: VERTEX COVER SIZE (%). 20
FIGURE 8 VERTEX COVER SIZE (%) IN RANDOM GRAPHS OF THE SAME SIZE AND EDGE DENSITY. 20
FIGURE 9 YEAR-TO-YEAR CHANGES IN VC MEMBERSHIP AS A FUNCTION OF DEGREE. 21
FIGURE 10 CONTAINMENT INDEX WITH VC-BASED FILTER DEPLOYMENT. LEFT: INTERNET AS TOPOLOGY. RIGHT: RANDOM GRAPHS. 21
FIGURE 11 CONTAINMENT PROTECTION. LEFT: MAXIMAL FILTERING. RIGHT: RANDOM ROUTING. 22
FIGURE 12 TRACEBACK PERFORMANCE WITH VC-BASED FILTER DEPLOYMENT. LEFT: INTERNET AS TOPOLOGY. RIGHT: RANDOM GRAPHS. 23
FIGURE 13 TRACEBACK RESOLUTION UNDER SEMI-RANDOM ROUTING. LEFT: SEMI-MAXIMAL FILTERING. RIGHT: MAXIMAL FILTERING. 23
FIGURE 14 TRACEBACK RESOLUTION UNDER MAXIMAL VS. SEMI-MAXIMAL FILTERING. LEFT: SHORTEST-PATH ROUTING. RIGHT: NON-SHORTEST-PATH ROUTING. 24
FIGURE 15 EXPANDED INTERNET AS BENCHMARK SUITE. LEFT: NETWORK SIZE. RIGHT: VC SIZE. 25
FIGURE 16 EXPANDED INTERNET AS BENCHMARK SUITE. LEFT: CONTAINMENT INDEX. RIGHT: TRACEBACK RESOLUTION. 25
FIGURE 17 VC SIZE OF INET GENERATED POWER-LAW TOPOLOGIES. LEFT: INET-2.2. RIGHT: INET-3.0. 26
FIGURE 18 ROUTE-BASED DPF PROTECTION IN INET-3.0 GRAPHS. LEFT: CONTAINMENT. RIGHT: TRACEBACK. 26
FIGURE 19 ROUTE-BASED DPF PERFORMANCE UNDER REDUCED FILTER DEPLOYMENT. LEFT: CONTAINMENT. RIGHT: TRACEBACK. 27
FIGURE 20 FILTER TABLE SIZE DISTRIBUTION. LEFT: CUMULATIVE ROUTE-BASED FILTER SIZE DISTRIBUTION IN 2002 NLANR TOPOLOGY. RIGHT: AVERAGE FILTER SIZE AT AS RANKED BY AS DEGREE. 28
FIGURE 21 LIMITATION OF ROUTE INFERENCE IMPOSED BY BGP'S DESTINATION-ROOTED ROUTE EXPANSION AND ROUTE ASYMMETRY. 29
FIGURE 22 LEFT: GENERATION OF STALENESS AT FILTER AS D. RIGHT: REMOVAL OF STALENESS THROUGH BY-PASS ACTION AT 30
FIGURE 23 PERFORMANCE OF VC-BASED PARTIAL INGRESS FILTERING. LEFT: CONTAINMENT. RIGHT: TRACEBACK. 31
FIGURE 24 FILTER-ENCLOSED POCKETS: MULTI-HOMED STUB, TRANSIT AS WITH TWO SINGLE-HOMED STUBS, AND GENERAL TRANSIT AS NETWORK. 31
FIGURE 25 AS ROUTING TREE ROOTED AT J. A, D, G, I ARE FILTER NODES, AND H IS A TRANSIT POCKET. .. 32
FIGURE 26 POCKET ADDRESS IDENTIFICATION. AS-PATH ADVERTISEMENT ARRIVING AT FILTER AS A IN THE POCKET CORRAL IS VOID OF FILTER AS NUMBER IF IT ORIGINATES FROM INTERNAL AS C. 34
FIGURE 27 BGP CONVERGENCE AT 300-SECOND AS FAILURE. LEFT: STUB AS. RIGHT: TRANSIT AS. 35
FIGURE 28 SAFETY VIOLATION DURING TRANSIENT PERIOD DUE TO AS FAILURE. LEFT: STUB AS. RIGHT: TRANSIT AS. 36
FIGURE 29 STALENESS DURING TRANSIENT PERIOD DUE TO AS FAILURE. LEFT: STUB AS. RIGHT: TRANSIT AS. 37
FIGURE 30 CONTAINMENT AND TRACEBACK PERFORMANCE DURING TRANSIENT PERIOD DUE TO AS FAILURE. LEFT: STUB AS. RIGHT: TRANSIT AS. 37
FIGURE 31 POCKETS IN 300-NODE INTERNET AS TOPOLOGY. LEFT: 25-NODE FILTER NET. RIGHT: 15-NODE FILTER NET. 39
FIGURE 32 CRITICAL FILTER DENSITY. SIZE OF GIANT COMPONENT AS A FUNCTION OF FILTER DENSITY FOR 12,517-NODE INTERNET AS TOPOLOGY AND CORRESPONDING RANDOM TOPOLOGY. 41

FIGURE 33 POCKET SIZE DISTRIBUTION UNDER VARYING FILTER DENSITY FOR 12,517-NODE INTERNET AS TOPOLOGY RANKED BY SIZE. 41
FIGURE 34 CRITICAL FILTER DENSITY. LEFT: SIZE OF GIANT COMPONENT AS A FUNCTION OF FILTER DENSITY FOR CAIDA, RIPE, USC/ISI, UMICH AS TOPOLOGIES. RIGHT: CORRESPONDING RESULTS FOR INET-3.0. 42
FIGURE 35 AVERAGE AND WORST-CASE TRACEBACK RESOLUTION AS A FUNCTION OF FILTER DENSITY..... 43
FIGURE 36 CRITICAL FILTER DENSITY UNDER SHORTEST-PATH, SEMI-RANDOM, AND RANDOM ROUTING..... 43
FIGURE 37 WORM PROPAGATION DYNAMICS WITHOUT CONTENT-BASED FILTERING. LEFT: RANDOM SCANNING. RIGHT: LOCAL SCANNING. 44
FIGURE 38 WORM PROPAGATION DYNAMICS WITH CONTENT-BASED FILTERING. LEFT: RANDOM SCANNING. RIGHT: LOCAL SCANNING. 45
FIGURE 39 FINITE TIME CRITICAL FILTER DENSITY. LEFT: RANDOM SCANNING. RIGHT: LOCAL SCANNING. 46

PROJECT SYNOPSIS

KEY CHALLENGES OF NETWORK SECURITY

Network security threats, including DDoS and worm attacks, present a clear-and-present danger to Internet and parts of DoD security. There is a pressing need to curtail and manage the disruptive and potentially debilitating effect of cyber attacks on e-commerce, everyday societal activity, and national security. Two challenges have stood in the way of effective, deployable solutions:

- PARTIAL VS. FULL DEPLOYMENT: Any serious solution must consider partial deployment a fundamental maxim of Internet vulnerability and demonstrate efficacy in spite of it. Epidemiology teaches us why security patches, when partially deployed, are unable to contain the spread of worms.

- LOCAL VS. GLOBAL PROTECTION: Firewalls epitomize local protection whose aim is to shield an entity from adverse outside influence. Amazon.com may succeed in shielding itself against a worm attack, however, if a significant fraction of its customer base is disabled by the same attack, the impact is ultimately shared.

The limitation of local protection under partial deployment extends to DDoS. A service provider may protect its internal network against external DDoS threats using state-of-the-art firewalls, but as long as legitimate clients mixed in with DDoS traffic are prevented from making full use of its services, the benefit derived from local protection only goes so far.

REQUIREMENTS FOR A VIABLE SOLUTION

The preceding discussion shows that security-wise the fates of the protected and unprotected are intertwined. A merchant without customers, or a customer without merchants, is not a useful state-of-affairs. Unprotected systems break neutrality and become a staging ground for cyber attacks on the healthy: severity is determined by the size of the unprotected network. Firewalls, a first line of defense in today's networks, are necessary but not sufficient to combat Internet cyber threats. To overcome the limitation of local protection under partial deployment, we require a new dictum:

(i) Protective action must be centered at transit points, not end systems, to exploit checkpoint screening and containment afforded by transportation networks.

(ii) Transit points must cooperate to affect global protection; a locally protected network system is inherently vulnerable to DDoS and worm attack.

(iii) The collective action of a few, under partial deployment, must yield an overwhelming synergistic effect that protects the whole.

In general, realizing a solution that satisfies all three conditions is a tall order. None of the conditions, however, are dispensable if finding an effective solution is the goal. Local (and selfish) protection under partial deployment, which leads to a partially protected system, does not work. Global (and cooperative) protection—modulo the specifics of what and how—can be instituted to satisfy (i) and (ii). The most difficult part is property (iii). Without it, cooperative protection is as limited as selfish protection.

DISTRIBUTED PACKET FILTERING: A NEW PARADIGM

Distributed packet filtering (DPF) is a new approach to network security aimed at providing scalable protection against cyber threats by achieving properties (i)-(iii). To illustrate the notion of distributed filtering, suppose Lake Wobegon is being polluted by elements carrying out distributed water contamination attacks. Contaminated water affects fish and wildlife, and eventually threatens towns and cities' water supplies. Local filtering (i.e., setting up a firewall) cordons off a shore segment and purifies the water therein for human consumption and some beach activity.

Inter-city commerce, mediated by ships and boats on water routes, dwindles for fear of admitting contaminated water particles. The fishing and sea food industry is shot. Cities that did not heed precaution turn into ghost towns. Figure 1 (left) depicts the grim state-of-affairs in Lake Wobegon.

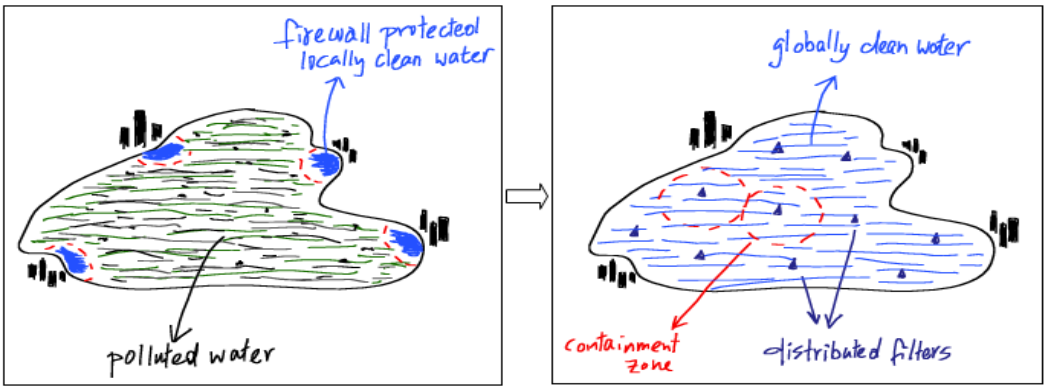

Figure 1 Lake Wobegon under local filtering (left) and distributed filtering (right).

Under distributed filtering, a number of cities band together and install water filters across the whole lake. Individual filters in the filter net are not aimed at protecting a specific town or city but the system as a whole. Any one community has little incentive to install a filter outside its immediate living space: a filter or two in the middle of the lake would not do much good anyway. It is only when a sufficient number is deployed that contamination introduced anywhere on the lake gets trapped by one of the filters and further spread contained.

A by-product of this local self-healing property is that the location of an attacker is revealed through the activated filters: containment assures that filters—treated as sensors—can be used for approximate localization. The improved state-of-affairs under distributed filtering is depicted in Figure 1 (right).

MISSING LINK: POWER-LAW CONNECTIVITY

Distributed packet filtering, as illustrated above, only satisfies properties (i) and (ii): cooperative protective action carried out at transit points for the good of the whole. Supposing Lake Wobegon is super-sized to Internet scale, without property (iii) that predicates a small filter deployment with a big bite, DPF would not be viable. Only when an economic filter placement achieves decisive protection can the necessary transit parties be brought together and induced to form a coalition for the greater good. The larger the required coalition, the less chance this has of succeeding.

In a transportation network, the paths that an entity can take are constrained by the underlying connectivity structure. In communication networks routing further limits the route that a packet can

take. In road systems, checkpoints at border crossings and major arteries have been used to shield countries and cities from unauthorized access. These gateways, when engaged as distributed filters, have been less effective at apprehending fugitives of justice: under partial deployment, there are simply too many ways for a person to travel from point *A* to point *B*.

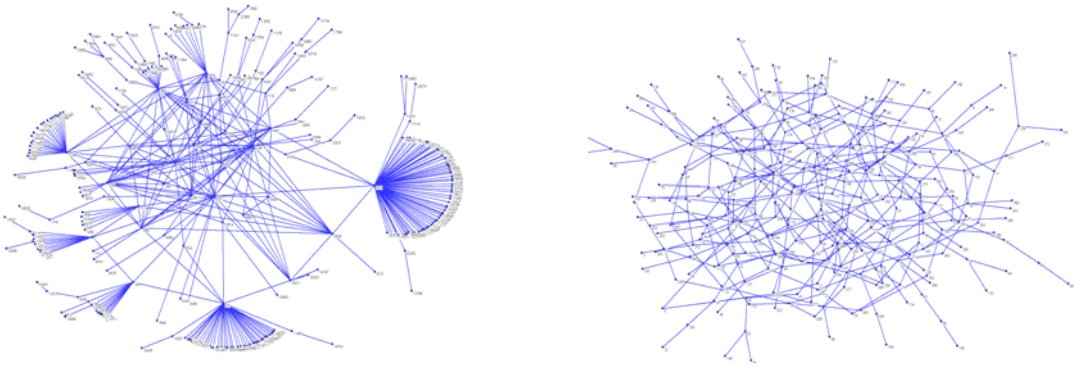

Figure 2 Left: A 300-node Internet domain graph. Right: A 300-node random graph.

The recently discovered power-law connectivity of the Internet is the missing link that facilitates property (iii). Figure 2 (left) shows a 300-node subgraph of the Internet domain graph where nodes are administrative domains and edges denote peering relations. Figure 2 (right) shows a 300-node random graph with the same edge density. The connectivity structure of the two graphs is markedly different. Whereas random graphs are regular and look locally about the same, power-law graphs possess "hubs" of varying sizes—power-law refers to a trend in the size distribution—that are connected through a "backbone." The highly nonuniform nature of Internet connectivity admits sparse, strategic placement of distributed filters that stops attack packets in their track.

OBJECTIVE: SCALABLE NETWORK SECURITY

The objective of distributed packet filtering for network security is three-fold:

- PROACTIVE PROTECTION: Prevent attack packets from reaching their targets and affecting harm in the first place. For DDoS, this means that attack packets are discarded before they can converge at the victim. For worms, attack packets carrying malware are discarded before they can spread to other systems.

- REACTIVE PROTECTION: One, localize the physical source of an attack for attribution and network management. For DDoS and worm attacks, this means that the physical locations of partaking systems are identified. Two, adapt to new forms of attack. This means detecting and responding to new attack signatures.

- SCALABILITY: The two modes of protection—proactive and reactive—must be achieved with small partial deployment in very large, i.e., Internet-scale, network systems. As the system size grows, the deployment level required for decisive global protection must stay invariant.

Proactive protection is achieved through containment by the filter net's sparse but effective coverage. We therefore refer to this as the containment effect, or simply containment. A dual, and complementary, effect of containment is source localization or traceback, which, in turn, aids

prevention through attribution's deterrent effect. We refer to this reactive protection mode as traceback. Detection and response to new attacks is called adaptation.

NEW CONTRIBUTION

The technical contribution of this research is comprised of three parts: performance evaluation of route-based DPF for DDoS attack, performance evaluation of content-based DPF for worm attack, and the building of requisite software tools.

Scalable DDoS Attack Protection

We introduce route-based DPF as an approach for achieving scalable protection against spoofed DDoS attack in power-law networks. Our focus area is inter-domain routing governed by BGP, but applies to other routing contexts including intra-domain and ad hoc routing. Route-based filtering at transit points determines whether packets lie about their true origin, and if so, discards them. The decision is based on routing information against which a packet's source and destination addresses are checked for consistency. A router, when performing packet forwarding, asks "where are you headed" (quo vadis) of the packet; we add "where are you coming from" (unde venis).

Conceptually, route-based filtering (RF) can be viewed as a generalization of ingress filtering which only works when neighbors are stubs—non-transit domains—whose allotted address space is well-defined. When neighbors are transit domains, ingress traffic may include neighbors' transit traffic which renders ingress filtering (IF) infeasible. RF, by computing the transitive closure of the transit address space, enables safe filtering: only spoofed packets are discarded.

The key properties of route-based DPF for DDoS protection are:

- PROACTIVE PROTECTION: We show that proactive protection using DPF is achieved for spoofed DDoS attack. With 15% deployment, 99% of all Internet domains are contained with respect to spoofed DDoS attack: no spoofed packet emanating from these domains can reach its target wherever it may be.

- REACTIVE PROTECTION: With 15% deployment, DDoS attack—spoofed or unspoofed—is localized to within 4 possible domains out of a total of 12,000+ (Jan. 2002). For the majority of cases, traceback involves just 2 choices inclusive the unspoofed sender's location.

- SCALABLE PROTECTION: Scalability crucially depends on the filter placement economy afforded by Internet's power-law connectivity. We use vertex cover (VC) as the filter placement strategy: a set of nodes is a VC if all edges are incident on one of the nodes in the set. 15% corresponds to a "minimal" VC.

From a cryptographic perspective, route-based DPF can be viewed as providing source address authentication using networking techniques as opposed to cryptographic techniques. The assurance, 99% containment under 15% deployment, is decisive but not perfect: a consequence of the trade-off, albeit small, necessary to achieve very sparse filter deployment. This imperfection is complemented by the traceback effect.

If the Internet domain graph were not power-law, property (iii) would not hold. In random graphs of the same edge density, vertex covers are significantly larger (55%) but their proactive and

reactive protection small (62% for proactive and 25-site uncertainty for traceback). Without power-law connectivity, route-based DPF would not be viable.

Small VCs induce two key properties—preference for large transit nodes and uniform filter density along routes—both of which are critical to achieving proactive and reactive protection. Since a large transit domain has many border routers at which RF must be installed, one may argue that in terms of router deployment the cost is high. However, the principal barrier to achieving cooperative network control on the Internet—be they for security or QoS—are policy barriers across administrative domains, not deployment at border routers within a domain. As such, cost accounting must occur at the granularity of autonomous systems, i.e., administrative domains.

An important aspect of spoofed DDoS protection using route-based DPF is that attack detection is obviated: filtering is applied to all packets. For unspoofed DDoS attack, an attack detection subsystem is required.

Mobile IP makes use of source address spoofing for mobility management. Presently, Mobile IP is not widely used on the Internet, inclusive Wi-Fi hot spots, and a straightforward modification exists that avoids resorting to IP spoofing. Source address spoofing is a practice that is unnecessary; protocol designers should be discouraged from employing it.

Scalable Worm Attack Protection

We use content-based DPF for scalable worm filtering in power-law networks. Content-based filtering is a well-known technique used by firewalls to weed out known computer worms. Whereas table look-up in route-based filtering is efficient and occurs at the time scale of route table look-up, content-based filtering (CF) is inherently more complex. This is further discussed below.

The key properties of content-based DPF for worm protection are:

- PROACTIVE PROTECTION: With 4% deployment, worm attacks with known signatures are trapped in small, constant size pockets and the spread of worms is contained. An interesting discovery is the existence of a phase transition: there is a critical filter density (~4%) such that filter deployment below the critical level is ineffective and deployment above is superfluous.

- REACTIVE PROTECTION: Since proactive protection at the critical filter density is perfect, traceback is not needed; if engaged, it is of a trivial form (treat filters as "sensors"; traceback for route-based DPF is much more subtle). Adaptation to new worms entails detection, computation of signatures, and updating of content-based filters in the filter net.

- SCALABLE PROTECTION: As in route-based DPF, power-law connectivity is crucial to affecting small deployment. In content-based DPF, the VC property can be relaxed (4% is too small to yield a VC) without incurring a commensurate sacrifice in protective performance.

A phase transition in proactive protection with respect to filter density also exists in random graphs. The critical filter density, however, is significantly higher (~30%). The engineering implication of the phase transition is: without knowledge of the critical filter density, instituting content-based DPF is likely to lead to either impotent protection or unnecessary waste.

In worm attacks, as in DDoS attacks, time scale is relevant. Perfect containment with 4% deployment holds under the worst-case assumption of eventual infection. If we consider the spread of infection and resulting damage after finite time, we find that there remains a phase transition around a time-bounded critical filter density. Moreover, the deployment level can be further reduced.

Worm attacks that exploit new system vulnerabilities are reactively defended against through 3-step adaptation: new worm detection, signature computation, and filter update with new signatures. Since worm filtering at transit points must occur at line rate, signature matching with a large signature base is a potentially debilitating bottleneck. The current state-of-the-art in content-based filtering is especially vulnerable to worm mutation. A clever attacker can easily obfuscate and create polymorphic worm variants that are all aimed, say, at triggering buffer overflow in Windows IIS. Polymorphic worm detection is at its early stages and part of our future work.

Performance Evaluation Tools

To facilitate performance evaluation of route-based DPF and content-based DPF on large-scale Internet domain topologies, we built two software tools. The static simulation tool was released to DARPA in the summer of 2002. The dynamic simulation tool is in preparation for public release.

Scalable Static Simulation: Route-based DPF Simulator

The static route-based DPF simulator in an optimized performance evaluation tool that allows exact computation of containment, traceback, and secondary performance metrics for route-based DPF in large-scale network topologies. Benchmark topologies include Internet domain topologies from NLANR, CAIDA, RIPE, Univ. of Michigan, and USC/ISI. The optimized tool—a complete revamping of an earlier tool—reduced the time complexity from cubic to quadratic. For example, a 12,000+ node Internet autonomous system (AS) topology requires 60 minutes or less on a single processor workstation. Previously the time requirement was 3-4 days.

The main static route-based DPF simulator features are:

- Routing. On-line shortest-path routing and external loading of user-specified routing tables is provided; randomized routing and multi-path routing support is also available.

- Vertex Cover. Support for filter net calculation using vertex cover—greedy heuristic and a factor-2 optimal algorithm—is provided. An arbitrary filter net can be specified by the user.

- Filter Type. There exist two filter types for route-based filtering: maximal (both source and destination addresses are used) and semi-maximal (only source address is used). Support is provided for both. Additionally, egress filtering is available.

- Performance Evaluation. A range of performance evaluation statistics can be selected at run-time. Scripts for data conversion, stub/transit classification, among others, are provided as part of the tool kit.

The first version was released in May 2002. A second enhanced version, completed in September 2002, is available at

http://www.cs.purdue.edu/nsl/DPFv2.tar.gz (~4.1 MB)

Documentation of the static route-based DPF simulator (also contained in the software release package) is available at

http://www.cs.purdue.edu/nsl/dpf-v2.pdf (~136 KB)

The first version of the tool has undergone Red Team testing by SRI. Suggestions for improvements have been incorporated in version 2.

Scalable Dynamic Simulation: DaSSF-Turbo

A limitation of the static route-based DPF simulator is the lack of time dynamics: performance evaluation—although exact—corresponds to a static, in-equilibrium scenario that is void of convergence and other time dependence issues. To affect dynamic performance evaluation of route-based and content-based DPF in Internet-scale networks, we built a simulation environment—DaSSF-Turbo—as an extension of DaSSF, the C++ based SSF realization for workstation clusters. SSF provides a process-driven simulation environment that is conducive to faithful, if not production code quality, parallel/distributed simulation. DML (Domain Modeling Language) provides user-level configuration support in DaSSF.

The key features of DaSSF-Turbo are:

- AUTOMATED NETWORK CONFIGURATION: Manually creating DML configuration files for large networks is infeasible. We provide Meta-DML, a network configuration tool, that allows automated generation of DML files for large-scale networks. Meta-DML supports specification of external network topology, protocol stack configuration, measurement configuration, and partitioning.

- RUN-TIME PERFORMANCE MONITORING: A key barrier to scalable network simulation is main memory: partitioning must ensure that the virtual memory (VM) subsystem does not come into play. Even if a single participating PC engages its VM, the whole simulation is dilated by the slowest component. DaSSF-Turbo provides a user configurable run-time measurement system encompassing events inside and outside the simulation kernel. The measurement system tracks dynamic memory consumption of transient and persistent events not accessible through static analysis but critical for memory balancing. Run-time monitoring is also used for partitioning when determining and verifying CPU/communication cost trade-off as well as diagnosis and analysis of the simulated network system.

- POWER-LAW PARTITIONING: Internet AS graphs, to some extent router graphs, Web graphs, and call graphs reaching sizes in the thousands of nodes obey power-law connectivity, which, in turn, has direct bearing on effective partitioning for CPU, communication, and memory load balancing in parallel/distributed simulation. Inefficient partitioning can result in marginal speed-up—even slow-down—that constrains scalability. Based on recent advances in power-law network partitioning, DaSSF-Turbo incorporates tools for automated partitioning that enhance scalability.

In addition to the three key features, we provide a comprehensive protocol suite spanning application, transport, network, and link layers. They include accurate link layer queueing, node/link failures, DPF-lookup, TCP/IP (essentially the same as DaSSF), UDP, DPF-update, BGP, traffic generators (CBR, Poisson, self-similar, trace-driven), and applications for DDoS and worm attack. Figure 3 depicts the overall structure of DaSSF-Turbo.

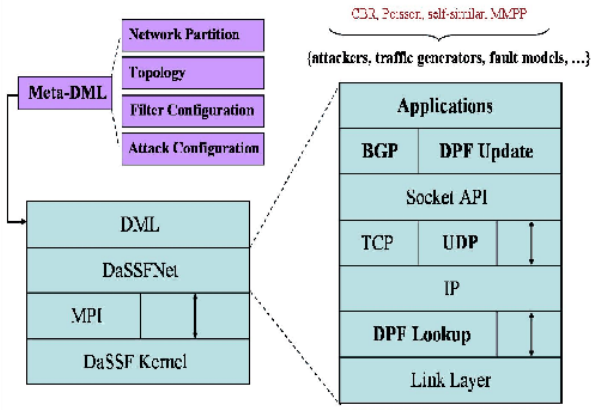

Figure 3 Overall architecture of DaSSF-Turbo.

The difference between DaSSF and DaSSF-Turbo may be viewed as a performance oriented version of the functional difference between TeX and LaTeX. LaTeX provides a macro-based abstraction and sugar coating necessary for wide-spread, non-expert use of TeX's powerful typesetting capabilities. DaSSF-Turbo provides support facilities necessary to affect scalable network simulation that fall outside the scope of DaSSF.

TECHNICAL CONTRIBUTION: DDOS PROTECTION

WHAT IS ROUTE-BASED PACKET FILTERING

A router asks of an arriving packet: where are you headed ("quo vadis")? In route-based packet filtering, we additionally ask: where do you hail from ("unde venis")? If, based on routing information, it can be unequivocally determined that a packet is lying—the source address is spoofed—then the packet is discarded.

A route-based filter (RF) is called maximal if the verification is performed using both source and destination addresses in the packet header. It is called semi-maximal if only the source address is used. In general, maximal RF is more powerful: there are packets that a maximal filter may detect as being spoofed that a semi-maximal filter cannot. However, since the space complexity of maximal RF is quadratic, unless filtering performance degradation is significant, semi-maximal filtering is preferred due to its linear space complexity.

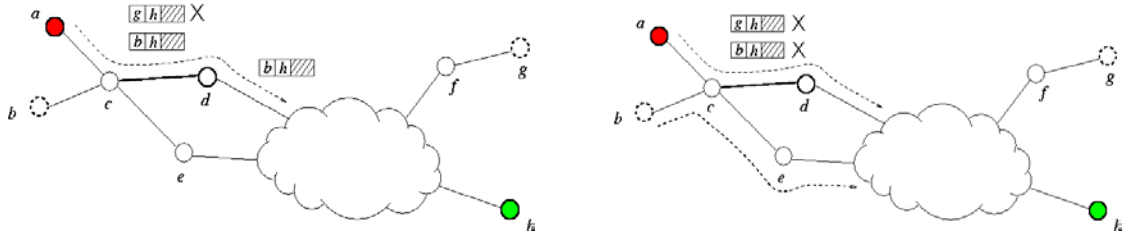

Figure 4 Spoofed DoS attack from a to h. Left: Semi-maximal RF. Right: Maximal RF.

Figure 4 depicts a network where node *a* attacks node *h* using source address spoofing. Route-based packet filtering is carried out at node *d* on interface (*c*,*d*). Figure 4 (left) shows that when *a* uses *g* as its spoof address, the packet is discarded at *d*. However, when *b* is used as the spoof address the packet must be let through under semi-maximal filtering since the packet could have come from node *b*. Figure 4 (right) shows that under maximal filtering, assuming a packet emanating from *b* destined for *h* is routed through *e*, the same spoofed packet arriving at *d* can be discarded.

Formally, let graph $G = (V,E)$ denote a network and let R denote a routing on G. R can be viewed as a set function that maps a source-destination pair (u,v) to a set of paths from u to v. In Internet routing, the path, at an instance, is unique. In general, R admits multi-path routing. A route-based filter F is defined at an edge $e = (x,y) \in E$ where, given a packet $M = (s,d)$ with source address s and destination address d arriving at node y from node x, the packet is discarded if and only if there does not exist a path from s to d that goes through e. This is denoted by $F_e(s,d) = 0$. F's value is 1 if the packet is let through.

In semi-maximal filters, only the source address is inspected by F; $F_e(s) = 1$ if and only if there is a destination z such that there is a path from s to z through e. By definition, route-based filters are safe: packets are only discarded if their source addresses are spoofed (no false positives).

DDoS Performance Metrics

A single route-based filter is of little use. We consider a set of route-based filters defined on a subset of edges $K \subseteq E$. For any path from node u to node v, let e be an edge on the path that precedes another edge e' on the same path. Suppose both edges are in K. Then for any packet $M = (s,v)$ traversing the path, we have

$$F_e(s,v) \preceq F_{e'}(s,v).$$

That is, M will not be discarded by a later filter (downstream) if it is not discarded by an earlier (upstream) filter. The same condition, without y, holds for semi-maximal filters. We call $u \in V$ a filter node if all edges incident on u are equipped with filters. We call a subset $L \in V$ of filter nodes a filter net.

The two primary measures of DDoS protection under spoofed attacks are containment and traceback. They are defined as follows:

- CONTAINMENT: Given graph $G = (V,E)$ and filter net L, $u \in V \setminus L$ is called innocuous if for all destination nodes $v \in V$, a spoofed packet emanating from u is discarded by a filter node before it can reach its target. The fraction of innocuous nodes, φ, is called the containment index.

- TRACEBACK: In the same setting, a node $v \in V$ is called k-traceable if for all source addresses s, a packet with source-destination address (s,v) arriving at v can be localized to within k possible nodes. If all nodes are k-traceable, we say that the network with the given filter net has traceback resolution k or is k-traceable.

There is a family of performance metrics with varying relevance to spoofed DDoS protection performance. Containment and traceback are the two principal metrics.

Route-Based DPF Performance

A natural optimization problem for containment is as follows: Given $G = (V,E)$ and real number $0 \le r \le 1$, find a minimum filter net L such that its containment index φ is greater than r. An analogous optimization problem exists for traceback. For general networks, the optimal filter placement problem is combinatorially hard. For example, in the case of $r = 1$, i.e., perfect containment, we can prove that optimal filter placement is NP-hard (joint work with Cole Smith and Benjamin Armbruster). The proof goes by reduction from vertex cover. Notwithstanding its general difficulty, in power-law networks, vertex covers are both small and can be easily approximated. A rigorous proof of the latter is a theoretical challenge and part of future work.

The goal of route-based DPF is to find a small filter net that achieves decisive containment ($\varphi \approx 1$) and traceback (k-traceability with k a small constant). The same filter net protects all. Figure 5 depicts a segment of a power-law network and deployed filter net.

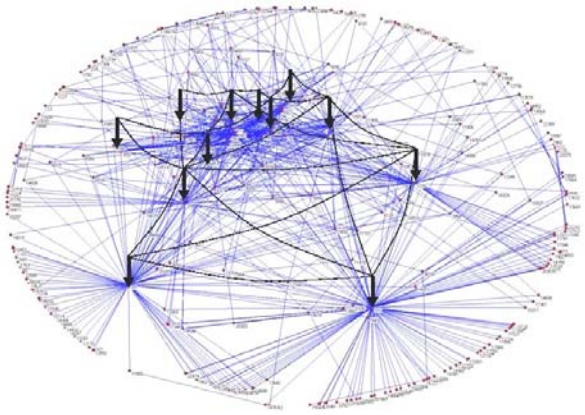

Figure 5 Filter net implementing route-based DPF in power-law network.

POWER-LAW NETWORKS AND FILTER PLACEMENT

Power-Law Connectivity

A power-law network is a graph whose node degree—the degree of a node is the number of edges incident on that node—has a power-law distribution:

$$\Pr\{\deg(u) = k\} \propto k^{-\beta}.$$

$\beta > 0$ is some exponent that depends on the application domain whence a graph comes from. For example, in Internet AS graphs and Web graphs, $2 < \beta < 3$.

As we have seen in Figure 2 (left), a power-law graph has high degree hubs that are joined by a backbone. From a static perspective, power-law graphs satisfy "a few are connected to many, many are connected to a few" (FMMF). An extreme form of this property is embodied by star graphs where nodes in the graph dangle from a single central node. In power-law graphs, the FMMF property is more relaxed: if we view nodes as molecules with different number of bonds, we can think of a "molecular stew" (of size n) where there are approximately (some constant times) $nk^{-\beta}$ molecules with k bonds sticking out, for k ranging from 1 to n. The different types of ingredients and their relative portion in a power-law molecular stew are depicted in Figure 6. When the stew is sufficiently stirred such that its ingredients bond to each other and anneal into a structure where no molecule is left with a dangling bond, we arrive at an instance of a power-law graph.

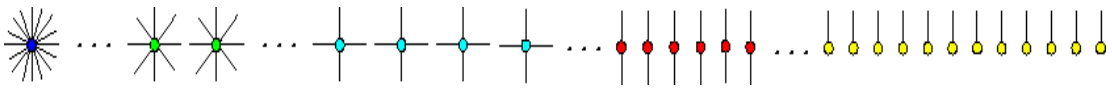

Figure 6 Ingredients of power-law molecular stew.

In a random graph of the same edge density (denote it p), the ingredients are essentially homogeneous: most molecules are of the same type having bond number np and deviation from this mean is exponentially rare. A regular graph is a graph where all nodes have the same degree. A random graph, therefore, is an almost-regular graph.

From a dynamic perspective, power-law graph invoke the proverb "the rich get richer and the poor get poorer" (RRPP). One way to evolve a power-law graph is to define a growth process where a new node, when it is added to the network, is preferentially attached to nodes with high degrees, in fact, proportionally so.

In the context of the Internet AS graph, a stub domain, when selecting a transit provider, has an incentive to choose a large transit AS. First, many other stubs are reachable with a single AS hop, and second, a large provider may project a sense of reliability—warranted or not—a consequence of the brand name effect.

Filter Placement Strategy: Vertex Cover

We use vertex cover as the filter placement strategy. Given $G = (V,E)$, a filter net $L \subseteq V$ is a vertex cover if for all edges in E at least one of the two end points is in L. There are two noteworthy algorithms for finding small VCs. The first is a procedure that is guaranteed to achieve constant factor 2 approximation: the size of the VC found is not more than twice the size of a minimal VC. The second is a greedy method that grows a VC by choosing a highest degree node, deletes the edges covered by the newly added node, then chooses the next node as a highest degree node based on the updated degrees.

For power-law graphs the greedy algorithm yields significantly smaller VCs than the constant factor approximation scheme. We use both, then choose the smallest. In practice, it is always the greedy algorithm that wins.

The vertex cover heuristic induces two properties that are relevant for scalable protective performance:

- PREFERENCE OF HIGH DEGREE NODES: Selecting a high-degree node allows simultaneous covering of multiple edges. In the context of transit ASes, the majority of neighbors correspond to stub domains—single-homed and multi-homed—which affords significant ingress filtering performance. The "catching of multiple birds with one stone" also affects small filter deployment.

- UNIFORM FILTER DENSITY ALONG ROUTES: For any route in the network connecting two nodes, VC implies that at least every other node along a route must be a filter node. Path-wise filter density assures effective coverage by the filter net despite its sparsity. This, in turn, is instrumental to affecting traceback with almost pin-point resolution.

The performance implications of the two properties in power-law networks are further discussed in later sections.

SCALABLE FILTER DEPLOYMENT

Partial Deployment: Power-Law Vs. Random Topology

The first performance result concerns the size of a small vertex cover—discovered by the greedy and factor-2 approximation schemes—as a function of graph size. Our principal benchmark graphs are AS topologies from Oregon Route-Views/NLANR which are obtained from BGP dumps. They form the most comprehensive measurement data to date. The benchmark suite is augmented in later

sections by measurement graphs from RIPE, CAIDA, USC/ISI, and UMich, and artificial power-law graphs obtained through Inet.

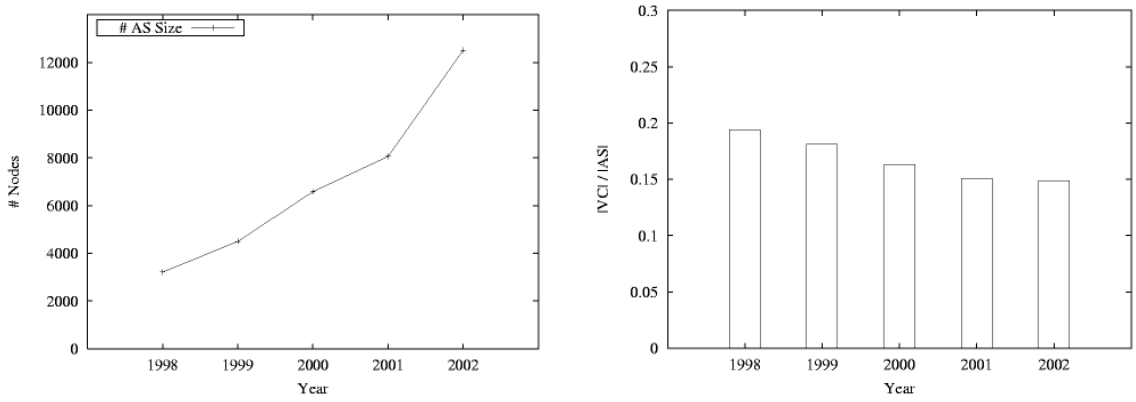

Figure 7 Left: Internet AS graph evolution 1997-2002. Right: Vertex cover size (%).

Figure 7 (left) shows the growth in Internet AS topology during 1998 (Jan.)-2002 (Jan.) where it has increased from about 3,000 domains to more than 12,000. The overall trend is superlinear. VC size, as shown in Figure 7 (right), has shrunk from about 18% to a little below 15% during the same period. For random graphs of the same size and edge density, Figure 8 shows that VC size increases to about 55%. It is this contrasting property that makes VC based filter deployment in power-law networks scalable. In later sections, we will consider further reduction in filter deployment that violate the VC property.

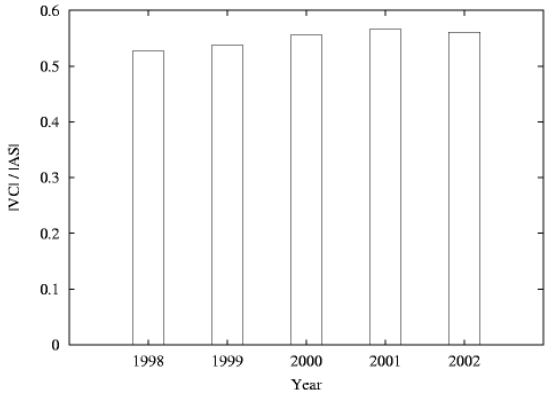

Figure 8 Vertex cover size (%) in random graphs of the same size and edge density.

Incremental Filter Deployment

An important property for incremental deployment is who in the VC membership is involved in the year-to-year change: either due to joining of new members of drop-out of old VC nodes. Figure 9 shows the percentage of ASes that are involved in the change as a function of node degree. We observe that during 1998-2002, the VC membership has an invariant core: nodes in the 1997 VC with degree greater than 27 do not drop out nor do new nodes get added to the VC that have degree higher than 27.

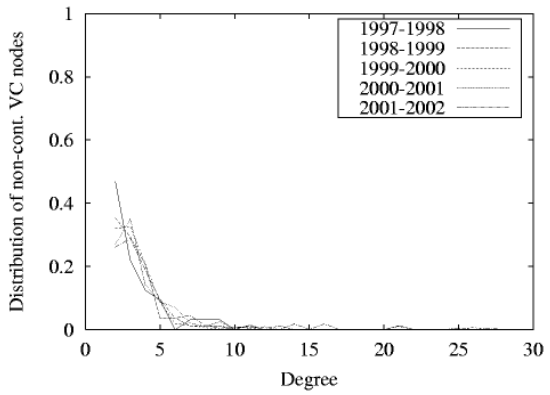

Figure 9 Year-to-year changes in VC membership as a function of degree.

SCALABLE PROACTIVE PROTECTION: CONTAINMENT

Containment: Power-Law Vs. Random Topology

Given 15% VC-based filter deployment in Internet AS graphs, how much proactive protection does it buy? Figure 10 (left) shows that the containment index reaches 99%. That is, with 15% deployment, 99% of all ASes can be made innocuous: no spoofed DDoS packet originating from these domains can reach their target domain wherever it may be. In the case of random graphs of the same size and edge density, Figure 10 (right) shows that containment falls below 65%. This is despite the significantly bigger VC size of 55% and correspondingly large filter deployment. Thus if Internet connectivity were random, route-based DPF would not be viable.

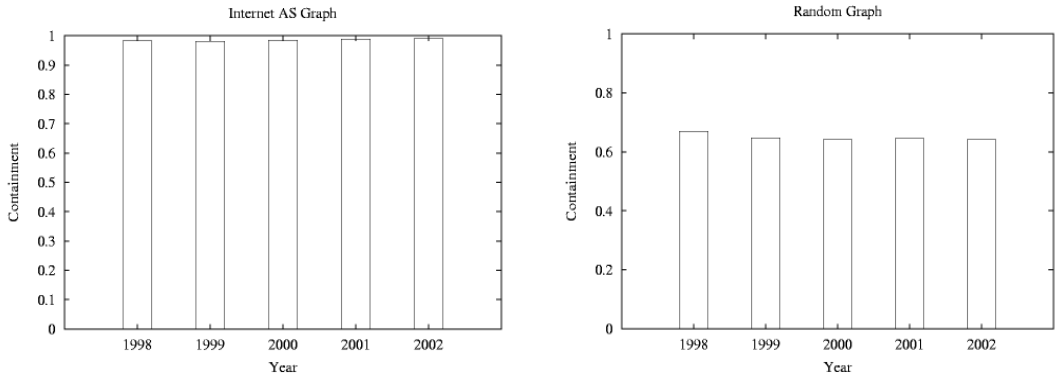

Figure 10 Containment index with VC-based filter deployment. Left: Internet AS topology. Right: Random graphs.

From an attacker's perspective, 99% containment performance in Internet domain topologies means that spoofed DDoS attacks can only be ventured from 1% of all Internet domains. Spoofed DDoS attacks, effectively, are shut down. Unspoofed DDoS attacks are still possible but because the source IP address reveals the physical origin of an attack packet, standard reactive measures employed in attribution may be instituted without delay. Presently, the same reactive measures take on the order of hours and days to take effect due to the time lag associated with traceback.

Maximal Filtering and Non-Shortest-Path Routing

The 99% containment result was obtained using semi-maximal route-based filters. For proactive protection, the more powerful maximal filter that additionally uses a packet's destination address to determine spoofing yields little additional gain: less than 0.3 improvement in the containment index as shown in Figure 11 (left). Since semi-maximal filtering already achieves 99% containment, maximal filtering requiring quadratic space is obviated.

Another network control factor that can influence protective performance is routing. The preceding results were obtained under shortest-path routing. In inter-domain routing where routing decisions are policy-based, the shortest-path assumption—although still an important policy criterion—can be violated. Figure 11 (right) shows containment protection under more relaxed routing algorithms that permit non-shortest-path routes.

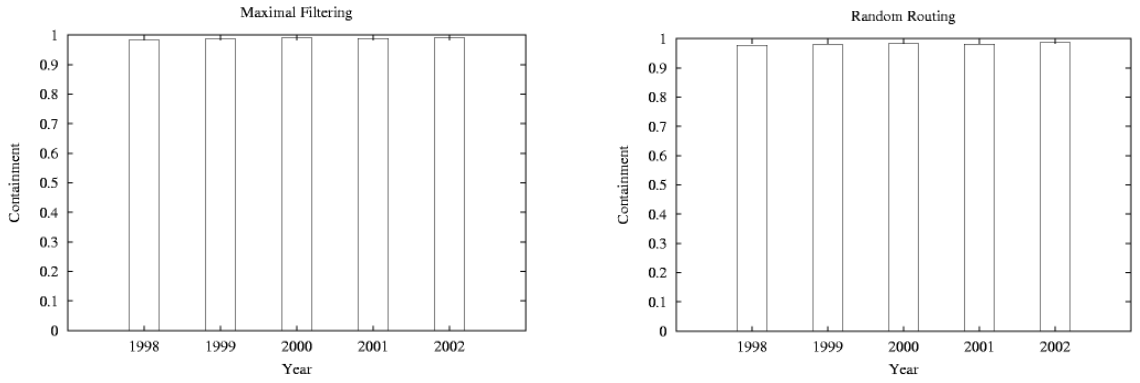

Figure 11 Containment protection. Left: Maximal filtering. Right: Random routing.

Under semi-random routing, during next hop selection in BGP's destination-based routing, a candidate is chosen randomly but its parent in the routing tree is chosen by shortest-path. Under random routing, both the candidate and its next hop parent are chosen uniformly randomly independent of the length of AS-PATH advertisements received. Figure 11 (right) shows containment under random routing. The containment index remains close to 99%. Details in the routing policy have marginal influence on proactive protection, a consequence of the constraining influence of power-law connectivity on routing.

SCALABLE REACTIVE PROTECTION: TRACEBACK

Traceback: Power-Law Vs. Random Topology

The second of the two principal performance metrics is traceback. Recalling the definition of k-traceability, a node (viewed as a potential victim) is called k-traceable if the physical origin of any arriving packet—spoofed or otherwise—can be localized to within k sites. Figure 12 (left) shows the fraction of k-traceable nodes as a function of resolution parameter k. We observe that all nodes are 4-traceable, i.e., the network is 4-traceable, and more than 80% of the nodes are 2-traceable. Since the 2002 Internet AS topology has more than 12,000 nodes, a worst-case constant traceback resolution of 4 shows formidable traceback performance.

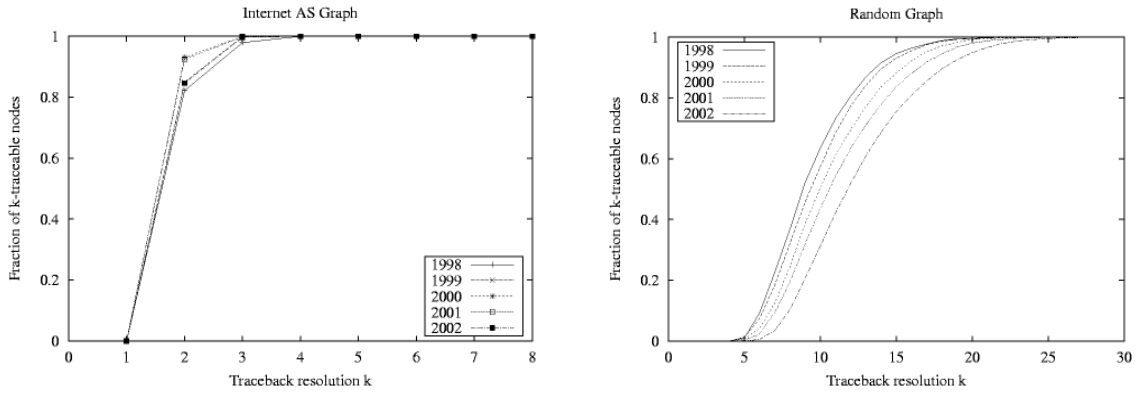

Figure 12 Traceback performance with VC-based filter deployment. Left: Internet AS topology. Right: Random graphs.

Figure 12 (right) shows that in random graphs of the same size and edge density, even with 55% filter deployment traceback performance is worse: the network is only 26-traceable. Degraded protection, despite higher deployment, mirrors the "bad-bad" performance characteristic observed in proactive performance. We emphasize that traceback in route-based filtering is immediate. This is in contrast to probabilistic packet marking (PPM), an elegant traceback technique that employs probabilistic path sampling using a constant header field. In PPM, significant damage has to be endured—a consequence of sampling—before reliable traceback can be reactively instituted. In route-based DPF traceback, a single packet suffices to determine attack location.

Maximal Filtering and Non-Shortest-Path Routing

For proactive protection, we showed that routing exerts marginal influence on containment performance. For traceback, the situation is a little different. Figure 13 (left) shows traceback performance under semi-random routing. We observe that traceback resolution has increased to 14.

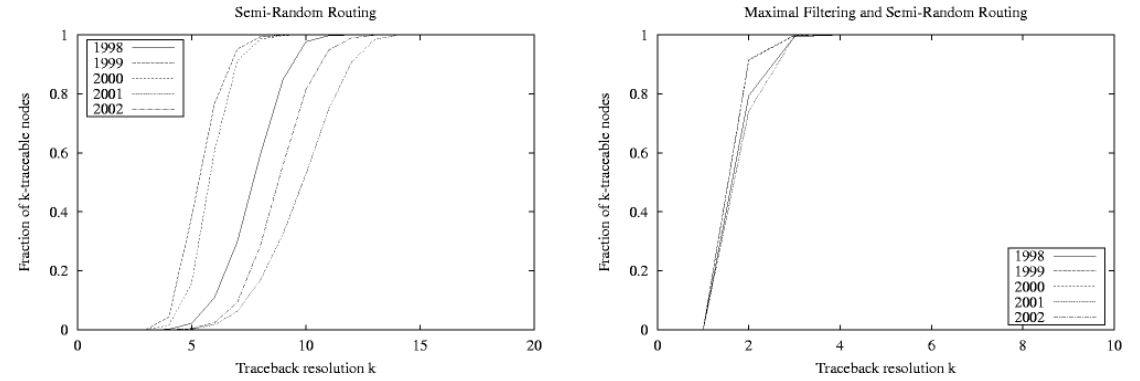

Figure 13 Traceback resolution under semi-random routing. Left: Semi-maximal filtering. Right: Maximal filtering.

Figure 13 (right) shows traceback performance under semi-random routing when maximal filtering is used. We observe that traceback resolution has returned to 4. Similar results hold for random routing under maximal filtering.

The power of maximal filtering vis-à-vis semi-maximal filtering can be discerned for traceback under non-shortest-path routing. Consider the network shown in Figure 14 (left) where node *e* receives a packet with source address *a* and destination *e*. We assume that shortest-path routing is in effect: for example, the paths from *a* to *e* and *a* to *f* are shown as dotted lines. Suppose *d* is the only filter node. Node e, upon receiving the aforementioned packet, can determine that *a*, *b*, or *c* must have sent the packet. Nodes *g*, *h*, and *i* disqualify since the filter at edge (*i*,*d*)—maximal or semi-maximal—will have discarded any spoofed packet sent from *g*, *h*, or *i*, destined for *e* with spoof address *a*.

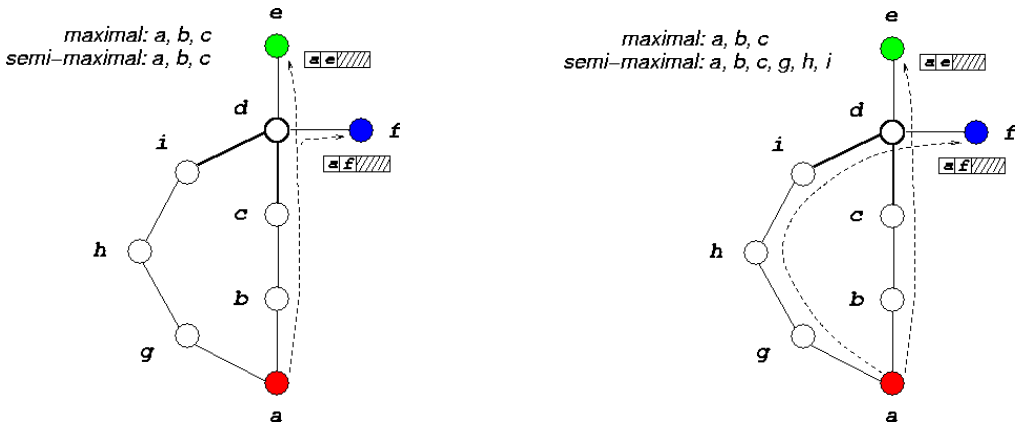

Figure 14 Traceback resolution under maximal vs. semi-maximal filtering. Left: Shortest-path routing. Right: Non-shortest-path routing.

Consider the same network configuration with the difference that the path from a to f is not shortest-path: instead of going through *b*, *c*, and *d*, it goes through *g*, *h*, *i*, and *d*. Under semi-maximal filtering, spoofed packets emanating from *g*, *h*, or *i*, destined for *e* with spoof address *a* cannot be discarded at edge (*i*,*d*) since a packet sent from node *a* destined to node *f* goes through (*i*,*d*). Thus traceback resolution at node e has increased from 3 to 6: nodes *a*, *b*, *c*, *g*, *h*, or *i* could have sent the packet. Under maximal filtering, at edge (*i*,*d*) on filter node *d*, the destination address *e* can be used to determine that a packet arriving with source address *a* and destination address *e* must be spoofed. Unless the scenario depicted in Figure 14 is prevalent, traceback resolution under semi-maximal filtering in power-law networks remains a small constant.

PERFORMANCE RESULTS: ROBUSTNESS

Other Internet Measurement Graphs: CAIDA, RIPE, USC/ISI, UMich

The main benchmark topologies were Internet AS measurement graphs from Oregon Route-Views/NLANR (1997-2002 period) which are based on BGP route table dumps. It is well-known that measurement topologies contain inaccuracies including missed peering relations and inconsistent route log information stemming from the long time duration associated with route table dumping. To show that route-based DPF performance is not sensitive to details in the underlying topology, we perform benchmark tests with Internet AS measurement graphs obtained from traceroute-based data (CAIDA and USC/ISI) and other BGP-based measurement data including RIPE (European BGP dump sites) and UMich (an integrated data set involving NLANR and RIPE dump data).

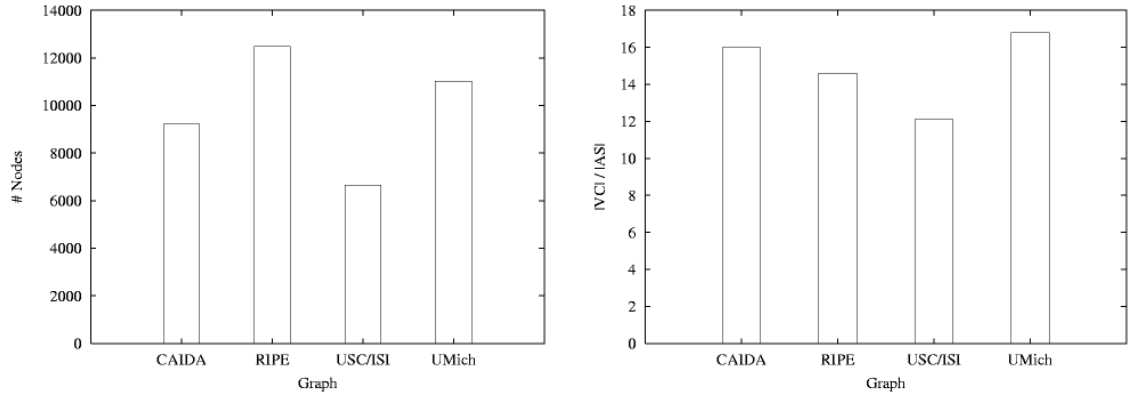

Figure 15 Expanded Internet AS benchmark suite. Left: Network size. Right: VC size.

Figure 15 (left) shows the network size of the four benchmark topologies: CAIDA, RIPE, USC/ISI, and UMich. Figure 15 (right) shows the corresponding VC sizes. We observe that the VC sizes range from 12% to a little below 17%. They are in the same ballpark range as Oregon Route-Views/NLANR graphs.

Figure 16 (left) shows proactive protection—containment—under the VC-based filter deployment in Figure 15 (right) for the expanded benchmark suite. We observe that the containment index lies in the 98-100% range, which is consistent with the containment result from Oregon Route-Views/NLANR data. Figure 16 (right) shows traceback resolution for all nodes under the same filter deployment. We observe that the network is 4-traceable which is the same traceback resolution as Oregon Route-Views/NLANR.

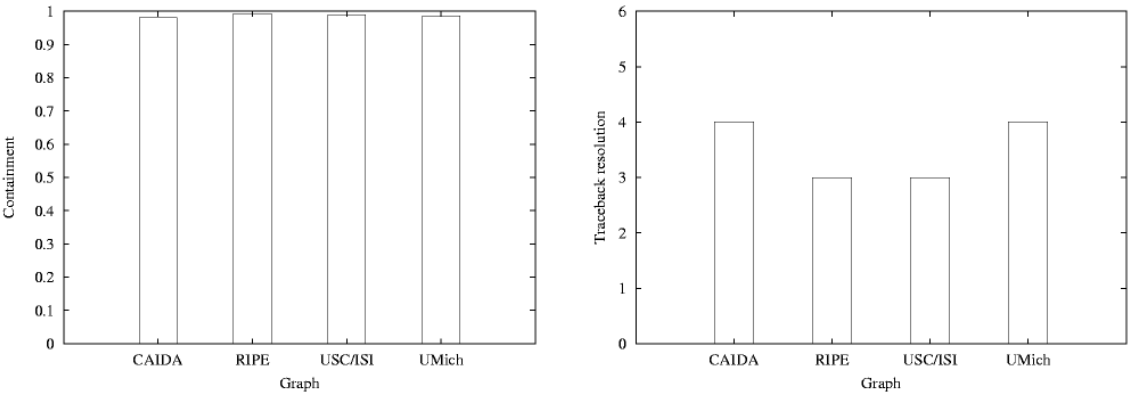

Figure 16 Expanded Internet AS benchmark suite. Left: Containment index. Right: Traceback resolution.

Artificial Internet Topology: Inet

Inet (Univ. of Michigan) is one of the better artificial topology generators aimed at mimicking the power-law connectivity observed in Internet measurement graphs. We augment the measurement-based suite with Inet generated artificial topologies. Figure 17 (left) shows VC size in Inet-2.2 power-law graphs configured to emulate the 1998-2002 Oregon Route-Views/NLANR measurement graphs. The VC sizes are above 30% which is approximately twice the size of VCs found in Internet AS graphs. Inet-2.2 focused at capturing the power-law degree distribution

without sufficient regard to other pertinent graph properties such as vertex cover which our work utilized for network security purposes. Using Inet-2.2 as a model of Internet AS topology would lead to inaccurate and possibly misleading results.

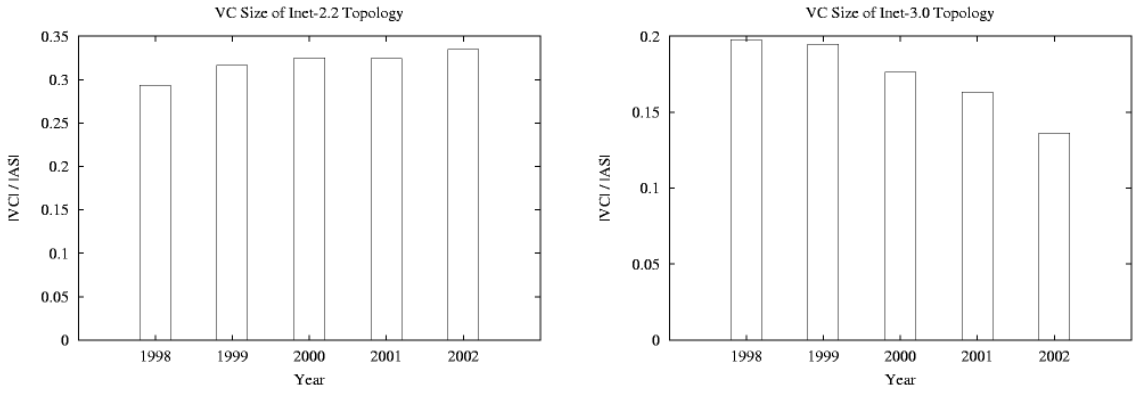

Figure 17 VC size of Inet generated power-law topologies. Left: Inet-2.2. Right: Inet-3.0.

The large discrepancy pointed out by our work, in part, prompted enhancements in Inet-2.2 aimed at capturing second-order properties in graph connectivity—the degree distribution may be viewed as a first-order property—leading to Inet-3.0. Figure 17 (right) shows the VC size results for Inet-3.0 which is markedly improved.

Figure 18 (left) shows the containment index of route-based DPF on Inet-3.0 generated topologies. For the artificial topology corresponding to the 2002 Internet AS graph, containment is above 90%. Overall Inet-3.0 is a little sluggish with respect to capturing proactive protection performance, part of which stems from treating all nodes in the graph as transit nodes. Figure 18 (right) shows traceback performance which is very close to that of Internet AS measurement graphs.

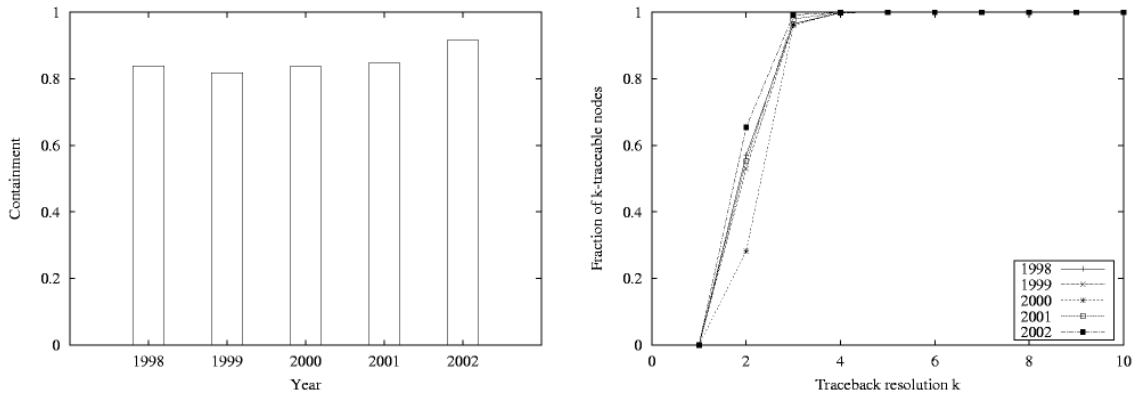

Figure 18 Route-based DPF protection in Inet-3.0 graphs. Left: Containment. Right: Traceback.

Artificial power-law topology generators, in the role of network benchmark tools, still have some ways to go. For example, Brite (Boston University), yet another power-law topology generator, shows significant inaccuracies when modeling Internet AS graphs with respect to vertex cover. A theory of power-law random graphs that generalizes classical random graphs pioneered by Erdös-Rényi is in its early stages of development. Our present understanding of power-law graphs and their

structural properties is incomplete, and performance extrapolations based on artificial power-law graphs need to be extra vigilant.

OPTIMIZATION: REDUCED FILTER DEPLOYMENT

The vertex cover filter placement strategy, by inducing preference of high degree nodes and path-wise uniform filter density, affects containment and traceback while achieving economy in filter deployment. The strict VC property—every edge must be incident on a vertex in the filter net—may be relaxed by considering almost-VC filter nets where some edges are not covered by nodes in the filter net. As long as the performance reduction stemming from reduced filter coverage is small, almost-VC filter nets provide an opportunity for optimization with respect to the trade-off in protective performance and deployment economy.

Figure 19 shows containment and traceback as filter deployment is varied from 18% down to 6%. At 15% we have a small VC, with 18% being a continuation of the greedy VC algorithm: edges whose two end points are not both covered are targeted in the continuation. The smaller filter densities 12%, 9%, and 6% are obtained from the small VC by pruning the smaller degree nodes. Figure 19 (left) shows containment as a function of filter deployment. We observe a linear decline in proactive protection starting from the 15% mark as filter density is decreased. At 6% deployment, we achieve close to 80% containment. Figure 19 (right) shows full traceback—i.e., all nodes satisfy the indicated *k*-traceability property—as a function of filter deployment. At 9% deployment, all nodes can localize the source of a packet to within 8 sites. At 6% deployment, traceback resolution degrades to 18 in the worst-case. Considering that there are a total of 12,000+ nodes, 18 is still a small constant.

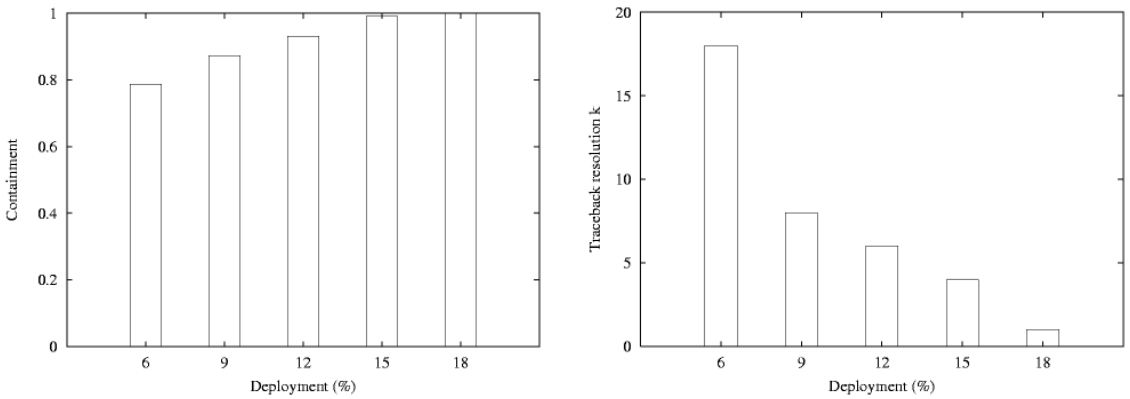

Figure 19 Route-based DPF performance under reduced filter deployment. Left: Containment. Right: Traceback.

The reduced, almost-VC deployment results show that by sacrificing some protective performance single digit deployment is achievable. A target operating point may be determined based on the trade-off relation along with other factors including policy considerations.

ROUTE-BASED DPF PROTOCOL

The preceding results showed route-based DPF performance under partial deployment in power-law networks assuming semi-maximal (or maximal) route-based filters. The issue of how to compute semi-maximal (or maximal) filters—i.e., protocol implementation—is context dependent and left unspecified. Our focus was in determining if route-based distributed packet filtering enabled

scalable DDoS protection: otherwise its raison d'être, along with protocol implementation, would be severely diminished. Here we address the protocol implementation issue.

Route-Based Filter Table Look-Up

Route-based DPF consists of two protocols that act at different time scales: route-based filter table look-up (at nanosecond time scale) and route-based filter table update (at tens of seconds time scale). Table look-up in semi-maximal filters with linear space complexity follows the same methodology used for fast IP table look-up: data structures and algorithms for tries and hashing. In power-law networks, the table size of semi-maximal filters, on average, is sublinear which renders table look-up faster than IP table look-up.

Figure 20 (left) shows the cumulative semi-maximal route-based filter table size distribution for the 2002 NLANR AS topology. We observe that 90% of the filters are less than 1,000 in size, and 80% have less than 100 entries. Figure 20 (right) shows the average filter size at ASes where the ASes are ranked by their degree: high to low. Domains that have an average filter table size exceeding 6,000 are of degree 2, those slightly exceeding 4,000 are of degree 3, and so forth. We observe that there is an approximately linear relationship between degree and average filter size.

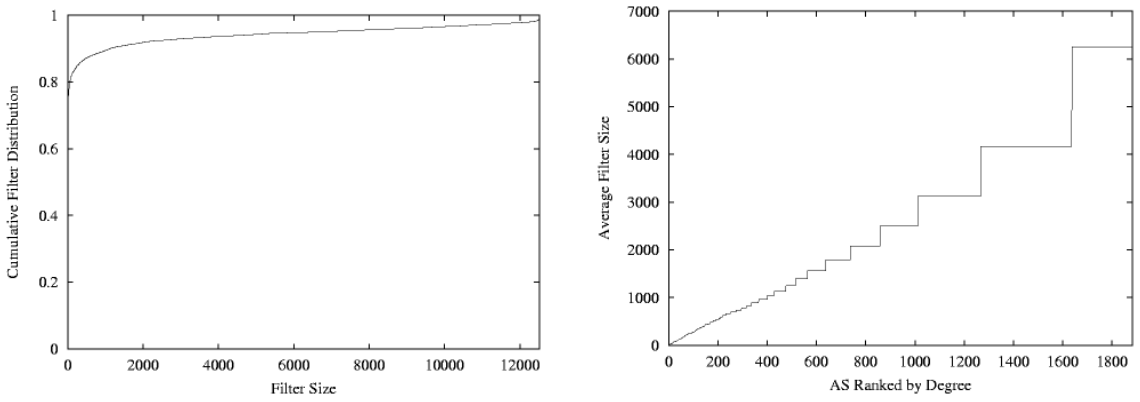

Figure 20 Filter table size distribution. Left: Cumulative route-based filter size distribution in 2002 NLANR topology. Right: Average filter size at AS ranked by AS degree.

One potential disadvantage of route-based filter table look-up vis-à-vis IP table look-up is that a filter table must exist for each input interface. First, we note that edges between ASes represent logical peering relations involving two or more border routers or exchanges: they do not represent physical interfaces at a single router. In today's backbone and access routers, interfaces number in the tens—for slow interface types the maximum number may reach above 100—which facilitates total memory usage that is on par with that of a single IP table.

Route-Based Filter Table Update: Issues

In general, computing semi- and maximal route-based filters is straightforward if routing information is known. In the static route-based DPF simulator, an optimized update protocol is employed that is both time- and space-efficient. This allowed performance benchmarking on large-scale network topologies involving 12,000+ nodes, each running a full TCP/IP stack. Power-law connectivity, by way of filter table sparsity, plays a crucial role in enabling efficiency.

Implementing route-based DPF in intra-domain routing governed by OSPF is simple: OSPF broadcasts global routing information which is used by every router in the system to compute its source-rooted routing tree based on Dijkstra's shortest-path algorithm. Route-based filter table update at routers in the filter net entails that OSPF's routing information be readable by route-based filtering. The filter update protocol computes the union of source-based routing trees and updates the filter table of every input interface with the allowed source (semi-maximal filter) or source-destination (maximal filter) addresses. The same is not true of RIP where global routing information is not locally discernible. An additional route discovery protocol is needed to perform route-based filter update.

In BGP, Internet's inter-domain routing protocol, the situation is mixed. BGP is a vector-path routing protocol that performs destination-rooted route tree expansion by sending AS-PATH update messages upstream from a destination AS. A border router of a transit domain, upon receiving multiple AS-PATH advertisements for the same target AS and IP prefix, must commit to a single path using its routing policy. The committed AS-PATH, prepended by the domain's AS number, is then advertised to select neighbors. A fundamental limitation of inferring global routing information through locally observable BGP state lies with destination-rooted route expansion and asymmetry of routes. Routes can be asymmetric under RIP and OSPF, however, under BGP's policy routing it is much more prevalent.

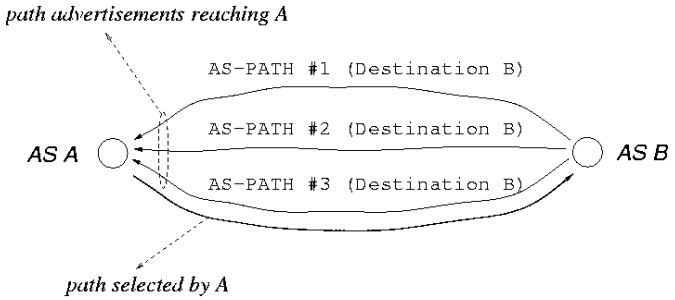

Figure 21 Limitation of route inference imposed by BGP's destination-rooted route expansion and route asymmetry.

Figure 21 depicts the inherent limitation imposed by BGP on route inference. AS *A* receives three AS-PATH advertisements originating at AS *B*. *A* selects AS-PATH #3 as its route to reach *B*. In general, due to a lack of reverse routing signaling, the path chosen by *A* cannot be known with certainty by other ASes in the system. *A* may choose to selectively advertise its own reachability through the neighbor that forwarded AS-PATH #1. This results in asymmetric routes between *A* and *B*.

Route-Based Filter Table Update: Protocols

We consider two approaches for dealing with the route discovery problem in BGP inter-domain routing aimed at computing route-based filter tables. The route discovery problem is not unique to route-based DPF. The methods described below are more generally applicable and of independent interest.

BGP Extension

The first method entails a modification to BGP-4: whenever a new AS-PATH is adopted, send an AS-REFLECT-ADD message—a new message type—downstream toward the destination AS. AS-REFLECT-ADD is the same as AS-PATH prepended by the sender's AS number and IP prefix.

A downstream AS that receives an AS-REFLECT-ADD message forwards it to the next downstream AS. When AS-REFLECT-ADD reaches its destination, it is discarded. An AS-REFLECT-DELETE message is generated if an existing AS-PATH is overwritten in favor of a new AS-PATH. The deselected AS-PATH is prepended with the sender's AS number and forwarded downstream toward the destination as with AS-REFLECT-ADD.

Route-based filter table update, when configured at a BGP router, works as follows. The filter update module is assumed to have read access to AS-REFLECT-ADD and AS-REFLECT-DELETE messages. Upon seeing an AS-REFLECT-ADD message, the source AS address of the reflect message is added to the filter table (semi-maximal version) and a counter set to 1. If the source entry already exists, its counter is incremented. When receiving an AS-REFLECT-DELETE message, the source AS entry is accessed in the filter table and its counter decremented. If the counter hits zero, the entry is removed. In the maximal filter version, entries comprise of source-destination address pairs.

In the absence of link failures, the route-based filter update protocol can be shown to correctly compute semi- and maximal filters. In the presence of link failures, staleness can arise where an entry remains resident in filter tables when, in fact, it can be removed. Figure 22 (left) depicts a configuration where *A*, *B*, *C*, and *D*, reach *E* through the indicated AS-PATH. *B* fails—i.e., its links fail triggering BGP keep alive timeouts at affected BGP peers—prompting *A* to reroute to *E* using an alternate AS-PATH that, say, does not go through *D*. As a consequence, *D*, a filter node, is left with two stale entries: *A* and *B*.

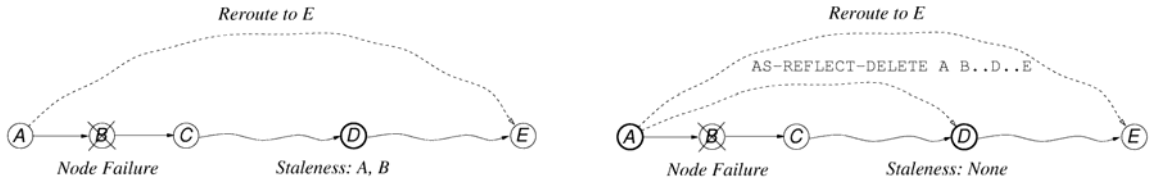

Figure 22 Left: Generation of staleness at filter AS D. Right: Removal of staleness through by-pass action at

Figure 22 (right) shows the same scenario with the difference that *A* is a filter node. *A*, detecting *B*'s failure, establishes a by-pass channel to its next downstream filter node *D* on the AS-PATH to *E*. ASes in the filter net form a coalition and know of the current membership. This prevents staleness by sending appropriate AS-REFLECT-DELETE messages: in this instance, one for itself and one for *B*.

The BGP Extension solution is simple and straightforward, but not preferred due to the requirement that all BGP routers must be updated with the extension. This goes against the premise of partial deployment.

Topological Filtering

Consider a VC-based filter deployment where instead of full-fledged route-based filtering ingress filtering is carried out. At transit nodes in the filter net, for all transit links to stub ASes (single-homed or multi-homed) ingress filtering is enforced. At transit links to other transit ASes, only egress filtering is performed. Figure 23 shows protective performance of VC-based partial ingress filtering. We observe that containment (97%) is significant but traceback resolution (260) is poor. The principal performance improvement of route-based filtering over ingress filtering under VC-based partial deployment lies in traceback resolution. The protocol implementation requirements of

partial ingress filtering are minimal, a matter of system configuration. However, its performance is limited with respect to traceback.

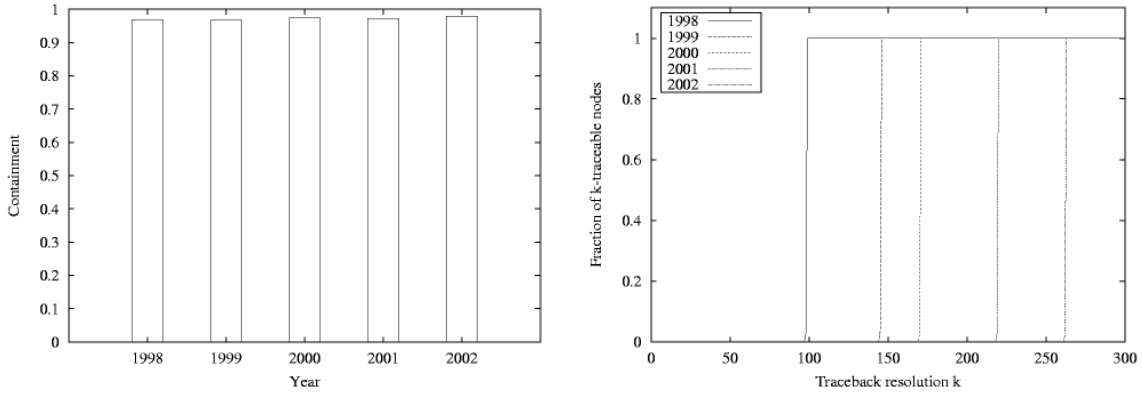

Figure 23 Performance of VC-based partial ingress filtering. Left: Containment. Right: Traceback.

In topological filtering, the goal is to harness the containment and traceback power of route-based filtering while overcoming the route uncertainty problem innate to BGP. At the heart of topological filtering are two complementary mechanisms: coarsification and isolation. Coarsification means that we group certain nodes into a super-node. Isolation means that super-nodes are separated from the rest of the network by surrounding filter nodes. Through their joint action, topological exploits power-law connectivity of Internet AS graphs to find filter placements that isolate non-filter domains into small super-nodes, called pockets, that are enclosed by filter nodes. A path to/from a pocket must go through a filter node.

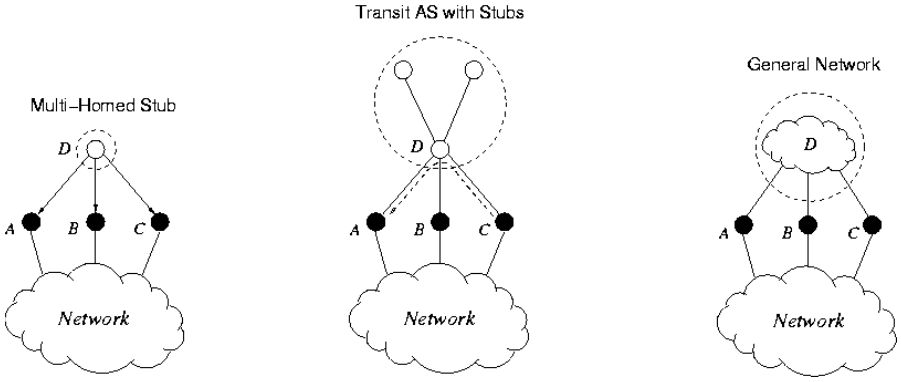

Figure 24 Filter-enclosed pockets: multi-homed stub, transit AS with two single-homed stubs, and general transit AS network.

The simplest pocket is a single-homed stub that is connected to the rest of the network by a provider who is a member of the filter net. A multi-homed stub, shown in the middle of Figure 24, is a pocket if all its providers are in the filter net. By definition, a multi-homed stub (except in cases of misconfiguration which sometimes arises in practice) does not provide transit service between its providers. The allowed address space originating from a multi-homed stub pocket is well-defined and visible to the members of the filter net via a dissemination protocol. The middle plot in Figure 24 shows a 3-AS pocket where one of the ASes in the pocket, *D*, is a transit AS, and the other two

are single-homed stubs. Filter nodes *A*, *B*, and *C* form a pocket corral. With respect to the allowed address space emanating from the 3-AS pocket, there are two possibilities:

- Non-Transit Pocket. Filter ASes *A*, *B*, and *C* do not utilize transit AS *D* in the pocket to route packets through each other. Any AS-PATH advertisement for reaching an AS outside the pocket that is forwarded by *D*—the AS-PATH necessarily contains *A*, *B*, or *C*—is rejected by the filter nodes. Effectively the pocket is turned into a multi-homed stub.

- Transit Pocket. Filter ASes, for some source and destination addresses not in the pocket, utilize *D* as a transit node. As long as the set of source addresses (for semi-maximal filtering) for which this is the case is small, the pocket is manageable in the manner of an multi-homed AS.

In the transit pocket case, even if the transit source address space is large, spoofing from within the pocket using outside source addresses can be detected by comparing ingress and egress transit traffic to/from the pocket visible at its perimeter. However, for DDoS attack this mechanism is not effective since it requires significant traffic exchange and coordination between filter nodes in the pocket corral. We show that route-based filtering affords a better way.

Figure 25 shows a routing tree rooted at AS *J*. *H* is a single-AS transit pocket, and *A*, *D*, *G*, and *I* are filters in the pocket corral. If filter placement is VC, the pocket's isolation property where non-filter transit ASes are surrounded by filter ASes is assured.

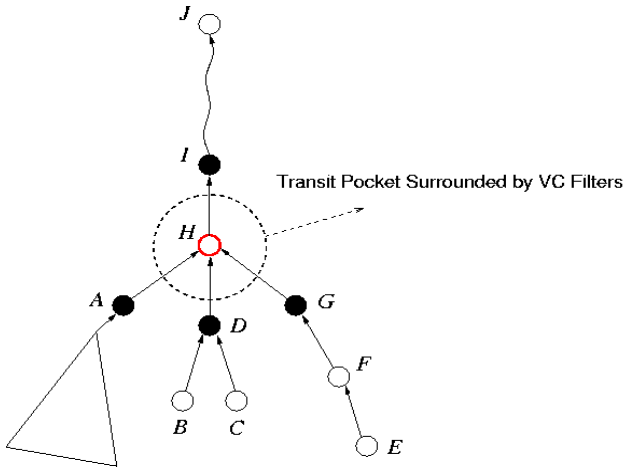

Figure 25 AS routing tree rooted at J. A, D, G, I are filter nodes, and H is a transit pocket.

Define S_B as the size of the maximum spoofable address space—inclusive the unspoofed address—a node in the subtree rooted at *B* can use to attack *J* without being discarded by a filter node. S_J denotes the traceback resolution of destination node *J*. *B* is a leaf node, hence a degenerate single-node subtree. Since its parent, i.e., provider, *D* is a filter node, $S_B = 1$. The same holds for *C*, and $S_D = 1$. (By default, we assume filter nodes perform egress filtering). We have $S_E = 2$, $S_F = 2$, and $S_G = 2$. In general, given a routing tree, filter net, and filter node *X*, the following recursive relation holds for S_X:

$S_X = \max\{S_{c(X)} : c(X) \text{ is a child node of } X\}$.

The max operator assures that traceback uncertainty does not additively amplify as long as there are filter nodes at junctions up the routing tree. A similar relation holds for a non-filter transit node Y when Y's neighbors are all filter nodes:

$S_Y = \max\{S_{c(Y)} : c(Y) \text{ is a child node of } Y\} + 1.$

Thus the traceback uncertainty contributed by a single-AS transit pocket is 1. Under VC-based filter placement, the condition that all neighbors of Y are filter nodes is satisfied. Moreover, in power-law networks, the diameter of the graph is small (e.g., the maximum path length in the 2002 NLANR/Oregon Route-Views AS topology is 11), hence by path-wise uniform filter density—one of the two key properties afforded by VC—a small traceback uncertainty is implied as long as tight traceback resolution at lower levels in the routing tree is achieved.

In the example shown in Figure 25, $S_H = \max\{S_A, S_D, S_G\} + 1 = \max\{S_A, 2\} + 1$. Tight traceback resolution at lower levels in a routing tree is enabled by VC's other key property: preference of high-degree nodes. In power-law networks, this yields filters placed at the center of locally star-like subgraphs whose traceback resolution is a constant 1. Since more than 80% of ASes are stubs—single-homed and multi-homed—a tight starting traceback resolution is assured and overall traceback performance rendered decisive. We note that ontainment performance is principally enabled by VC's preference of high-degree node property.

A protocol implementation of topological filtering is comprised of three parts:

- Phase I: Find a filter placement that achieves isolation with small pockets. Non-transit pockets need not be small to achieve effective protection. For example, it may suffice to contain and trace back spoofed DDoS packets at the granularity of nations for purposes of defending against nation-to-nation cyber warfare. For transit pockets, a small size is more critical. VC-based filter placement achieves both small non-transit and transit pockets. Customized, non-VC filter nets may be considered to achieve minimal filter deployment subject to a target protective operating point.

- Phase II: Discover address space of each pocket. For single- and multi-homed stubs, the address space is known to the providers who may be members of the filter net. For general pockets, as long as isolation holds (via Phase I), there is a simple protocol executed at filters in a pocket corral that discovers the pocket's address space. Figure 26 illustrates the method for a transit pocket where A and B are filters in the pocket corral and C is an internal AS. The key problem is to distinguish AS-PATH advertisements originating from the pocket (of unknown address space) from those originating outside the pocket but forwarded by the pocket. This is accomplished by checking if an AS-PATH advertisement arriving from the pocket contains a filter AS number belonging to the pocket corral.

- Phase III: Perform route-based filter update on the coarsified graph. By isolation, AS-PATH selection to destination ASes outside a pocket is revealed by IP traffic emanating from a pocket: once the next hop for a destination IP address (matching to an IP prefix) from the set of filter ASes in a pocket corral is revealed, the downstream AS path is determined and known to the filter nodes. Analogous to the BGP Extension but only deployed at BGP border routers at ASes in the filter net—true partial deployment—a DPF-ADD message belonging to the route-based

filter update protocol is sent to the next downstream filter node by-passing non-filter BGP nodes. At the downstream filter node, filter table update is performed and DPF-ADD further forwarded until it reaches the last filter node in the AS-PATH.

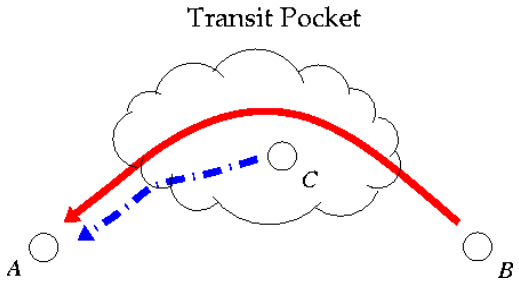

Figure 26 Pocket address identification. AS-PATH advertisement arriving at filter AS A in the pocket corral is void of filter AS number if it originates from internal AS C.

Phase I and II of the protocol change at large time scales—on the order of hours, days, or weeks—since the information concerns AS connectivity and IP prefix address space allocation that are independent of dynamic inter-domain route adaptation. Phase III transpires at the same time scale as BGP route table update.

INFRASTRUCTURE PROTECTION AND RESILIENCE

Infrastructure Attack

Increasingly DDoS and worm attacks are targeted at the network infrastructure, including routers and name servers, in addition to commercial, governmental, and public servers and individual users. Depletion of infrastructure resources, given its wide-spread repercussion, represents a critical vulnerability that must be protected against for the basic well-being of a network system. Hardware and software failures that are not necessarily of Byzantine origin are equally detrimental with respect to their impact on availability, robustness, and performance. In the case of route-based DPF, resilience against DDoS attacks that are aimed at disrupting route-based filter table update resulting in protective performance degradation, including malfunction, is a key concern.

We may distinguish two forms of attack: direct and indirect. In the direct case, flooding of TCP channels over which route-based DPF update are exchanged may be used to impede filter table update causing inaccurate filter table entries. As a consequence, packets with valid source addresses may be discarded—safety violation—a form of malfunction, and packets with spoofed source addresses that are otherwise so detectable by semi-maximal filters escape unnoticed—staleness—a form of inefficiency. If the signaling protocol is insufficiently secure with respect to authentication and integrity, bogus update messages may be infused into the filter net producing corrupt filter tables.

In the indirect case, the inter-domain routing protocol, BGP, on whose information route-based DPF depends, may be attacked thereby causing inconsistencies between BGP's route tables and route-based DPF's filter tables. BGP's well-known convergence problems may trigger prolonged transient periods during which inconsistency in table entries is exacerbated. BGP is vulnerable to targeted infusion of bogus AS-PATH updates that can propagate to other routing tables and cause significant table corruption as well as route instability.

Route-based DPF, once booted, protects all components of a network system including routers, name servers, and filters belonging to the infrastructure. Therefore an attacker aiming to bring down the network infrastructure is forced to target a local vulnerability—e.g., introducing corrupt route/filter table updates and generating link/site failures through intrusion at select systems—and hope that a weakening of the protective shield arises that can be exploited to further weaken protective performance. In the absence of self-stabilization, this may lead to a positive feedback loop that progressively degrades protective performance until eventual neutralization of the protective shield. We show that route-based DPF is resilient against localized attacks and failures, maintaining continued proactive and reactive protection without persistent safety violation and cumulating staleness.

BGP Convergence Dynamics

We consider a severe form of attack that brings down an entire AS. The backbone and border routers belonging to an AS are, in general, geographically dispersed. Hence the extent of damage stemming from an attack is realistically limited to AS-level link failures. Similarly for non-Byzantine faults such as severing of fiber lines and downtime due to power outages. We consider two types of AS failures: stub and transit AS. For the transit AS case, we select an AS that is heavily utilized by routes generated under BGP.

We show performance results for a 1997 (Nov.) NLANR/Oregon Route-Views topology containing 3,025 domains. We run BGP and route-based DPF—the BGP Extension protocol version—at time 0, and after stabilization, bring down an AS node at time 300 seconds. BGP stabilizes between 2,500-3,000 seconds, and our focus is to examine route-based DPF's protective performance during the transient period when the system is unsettled and in a state of flux.

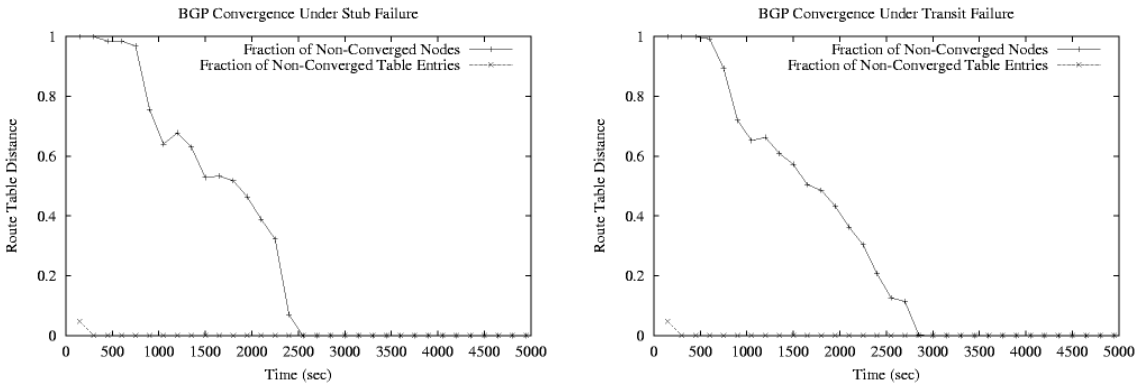

Figure 27 BGP convergence at 300-second AS failure. Left: Stub AS. Right: Transit AS.

Figure 27 shows BGP convergence when an AS is brought down. We measure route table convergence by defining a distance metric—route table distance—for two granularities: route table entry count and route table count. At a time instance, route table distance with respect to entry count denotes the fraction of route table entries across all route table entries in the system that have incorrect values relative to their values under converged route tables. Route table distance with respect to table count denotes the fraction of route tables that have one or more incorrect route table entries.

Figure 27 (left) shows route table distance as a function of time when a stub AS that is attached to a large (i.e., high-degree) transit AS is brought down. We observe that although the fraction of

non-converged route table entries is small, they are spread across many route tables and route table distance with respect to table count converges after 2,500 seconds. Figure 27 (right) shows BGP convergence under transit AS failure, where a medium-degree transit domain that is ranked 5th with respect to BGP routes that traverse through it is brought down. We observe a similar but slower convergence: the distance with respect to table count stabilizes short of 3,000 seconds.

Route-Based DPF Resilience: Safety

With the BGP convergence dynamics shown in Figure 27 as a reference point, we examine route-based DPF performance with respect to safety. By definition, semi- and maximal filters are safe: a packet whose address is not spoofed is never discarded. Under node failures leading to transient network dynamics, inconsistencies between route and filter tables may develop where packets with valid source addresses are discarded by filter tables containing incorrect entries.

We define safety violation for three granularities: filter table entry count, filter table count, and node count. At a time instance, safety violation with respect to filter table entry count denotes the fraction of filter table entries—across all filter table entries in the system—that have an entry value for discard when the corresponding entry in an ideal filter table dictates passage. The ideal filter table is defined with respect to the instantaneous—hence most up-to-date—snapshot of the route tables in the system. Thus the filter table entries are perfectly consistent with respect to the routes specified by the routing tables at that moment. Safety violation with respect to filter table count denotes the fraction of filter tables that contain one or more entries with a safety violation, and analogously for node table count.

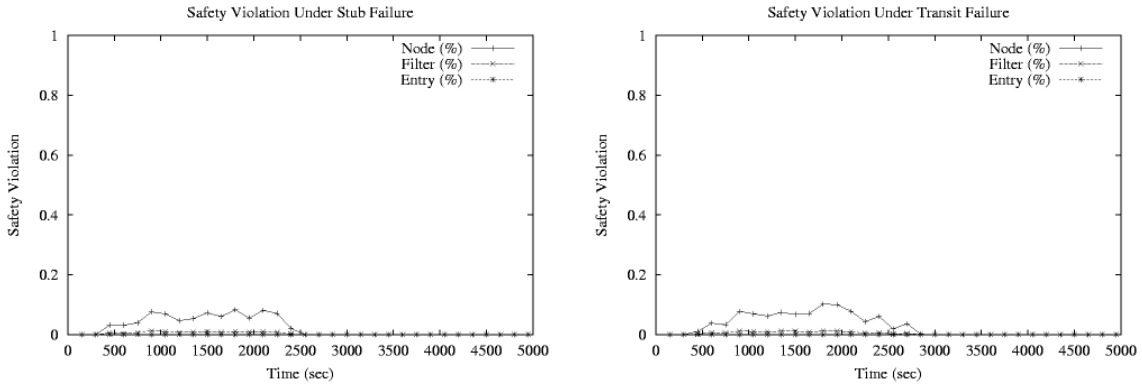

Figure 28 Safety violation during transient period due to AS failure. Left: Stub AS. Right: Transit AS.

Figure 28 (left) shows safety violation as a function of time for entry count, table count, and node count under stub AS failure. We observe that safety violation for both entry and table count is miniscule; node count safety violation stays below 10% during the transient period. Figure 28 (right) shows safety violation under transit AS failure. In both cases, the impact of safety violation is small and non-persistent: route-based DPF is able to operate without causing significant harm to unspoofed traffic.

Route-Based DPF Resilience: Staleness

Another important resilience property of route-based DPF is staleness: a filter table entry is stale if its presence, which allows passage to arriving packets with the designated source (and destination) address, is unwarranted in a perfect semi-maximal (and maximal) filter tables. Thus spoofed IP

packets which may be safely discarded and are so detectable in a non-transient environment are let through. Analogous to safety violation, we define staleness for filter table entry count, filter table count, and node count as the corresponding fraction of stale objects.

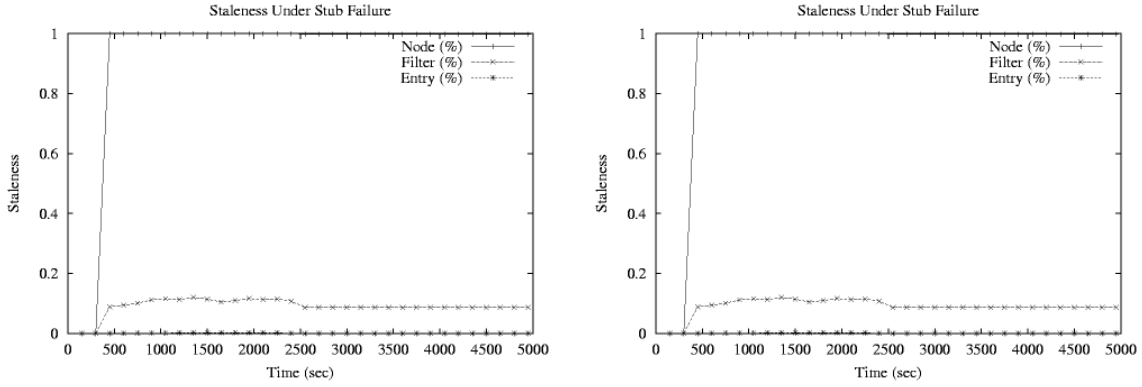

Figure 29 Staleness during transient period due to AS failure. Left: Stub AS. Right: Transit AS.

Figure 29 (left) shows staleness as a function of time for entry count, table count, and node count under stub AS failure. We observe that staleness with respect to entry count is miniscule, table count staleness around 10%, and node count staleness close to 100%. What this means is that a few stale entries are resident in 10% of the tables, and almost every node has a filter table with a stale filter table entry. In fact, examination of stale filter table entries reveals that all are due to the IP prefix of the downed AS which occurred in about 10% of the filter tables as a valid entry.

Route-based DPF Resilience: Containment and Traceback

The ultimate performance metric of route-based DPF performance under transient conditions remains containment and traceback. Containment and traceback at a time instance is defined with respect to the snapshot of filter tables in the system and their protective performance in the presence of staleness and safety violation.

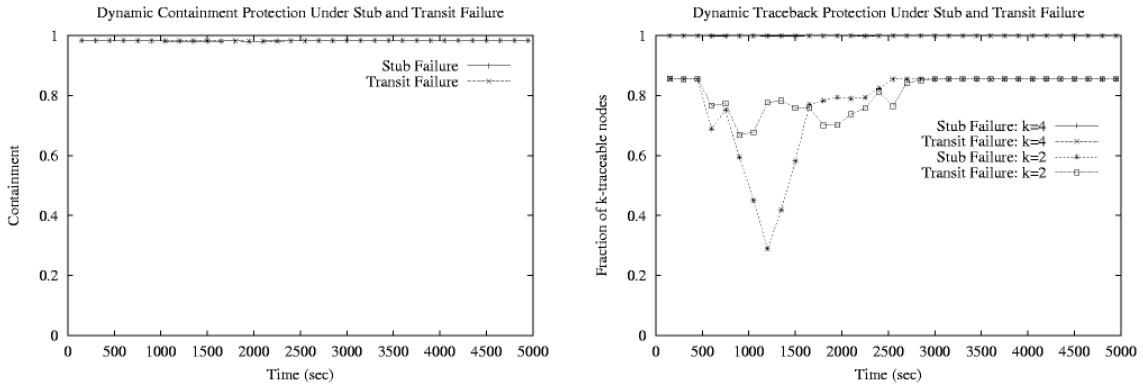

Figure 30 Containment and traceback performance during transient period due to AS failure. Left: Stub AS. Right: Transit AS.

Figure 30 (left) shows containment as a function of time under stub and transit AS failures. We observe that containment remains near 99%: proactive protection is preserved during transient

periods caused by node failures. Figure 30 (right) shows traceback performance as a function of time under stub and transit failures. For each case, we show the fraction of k-traceable ASes for $k = 2$ and 4. We observe that route-based DPF maintains perfect reactive protection with traceback resolution $k = 4$ during transient periods. That is, all received packets—spoofed or unspoofed—can be localized with respect to their physical origin within 4 sites. For $k = 2$, we find that the fraction of ASes that satisfy 2-traceability shows significant fluctuation during the transient period but stabilizes to above 80% after BGP and filter table convergence.

TECHNICAL CONTRIBUTION: WORM PROTECTION

CONTENT-BASED DISTRIBUTED PACKET FILTERING

Content-based distributed packet filtering is aimed at detecting and discarding IP packets carrying worm malware at transit points in the network system. The overall objective is the same as that of route-based DPF: achieve decisive protection with small deployment. The principal measure of protection is containment—conceptually similar but technically different from that of route-based DPF—where worms are prevented from spreading to other systems outside a small confinement, i.e., pocket.

Given a network topology $G = (V,E)$ and a set of filters $L \subseteq V$, let G' denote the subgraph induced by removing the vertices in L from V. G' consists of a set of maximal connected components—if the entire graph remains connected, then there is a single connected component—which we call pockets. Thus any two nodes in the same pocket are reachable by a path that does not contain a filter node. Even so, routing may still prevent such a route to be taken. However, we take a worst-case approach and consider achieving proactive protection against worm attacks under all possible routing.

Of particular interest is the giant component, i.e., a largest connected component under filter net L. An attacker's objective is to achieve maximal damage or infection using minimal effort. If the giant component is large, then a single worm planted at one of the nodes in the giant component through initial intrusion during attack preparation suffices to infect the whole pocket. If the giant component is small—for example, a constant—then the spread is locally contained. Since an attacker can afford to plant some number of worms during attack preparation, the distribution of pocket sizes is also of relevance.

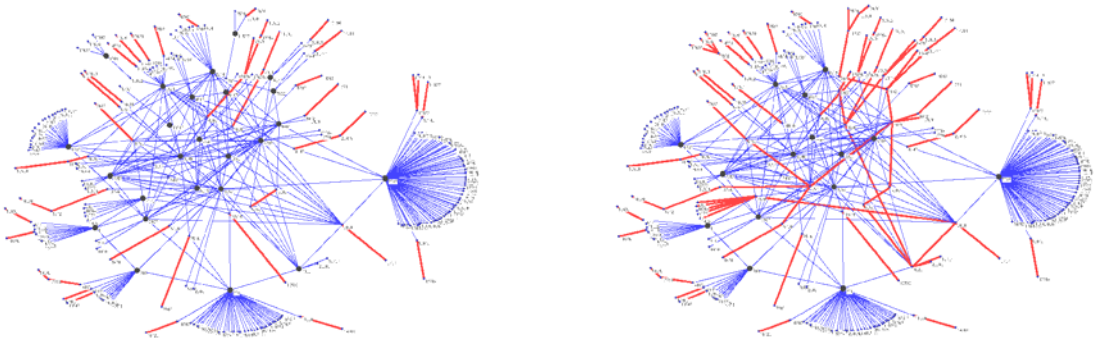

Figure 31 Pockets in 300-node Internet AS topology. Left: 25-node filter net. Right: 15-node filter net.

Figure 31 (left) shows a 300-node subgraph of an Internet AS graph where filters are placed at 25 sites. Filter nodes are shown as large filled vertices and connected components are highlighted by bold connected edges. The giant component has size 3, and worm infection is effectively contained. Figure 31 (right) shows the same system with a 15-node filter placement. There are too few filter nodes allowing large inter-connected components. The giant component has size 34.

SPARSE FILTER PLACEMENT

From an optimization perspective, we may define the following combinatorial optimization problem: given $G = (V,E)$ and integer $k > 0$, find a minimum filter net L such that the size of a giant component after removing L is less than k. This implies that all infections are contained within pockets of maximum size k. The optimization problem is NP-hard. A rigorous proof has not been advanced at the time of writing but it is expected to be routine.

Vertex cover maintains its utility as a placement strategy for worm filtering. This can be gleamed from the following simple fact: if the filter net is a VC, then all pockets are of size 1. Thus no infection is possible under VC. Since the condition of zero infection is unnecessarily strong—local containment with k a small constant is sufficient—this leaves room to reduce the size of the filter while achieving tight containment. The placement strategy is the same as that used in route-based DPF: VC-based pruning. The first property of VC—preference of high-degree nodes—assures that around 80% of the nodes (stub ASes) will be contained by picking large transit providers. The second property of VC—uniform filter density along paths—handles containment in the backbone.

Consider removing a filter node u from a filter net that is VC. It is easily seen that the maximum pocket size after removing u is bounded above by $\deg(u) + 1$. More accurately, if K_u denotes the number of neighbors of u that are not in the filter net, then the bound is exactly $K_u + 1$. A simple iterative procedure is to group the resulting connected non-filter nodes into a single super-node with a weight that indicates the size of the group. With respect to the coarsified—and now weighted—graph, the VC property is still preserved. The general update rule, after removing a filter node v, in a coarsified graph is

$$w(s(v)) = 1 + \sum_{x \in N(v)} w(x)$$

where $s(v)$ denotes the super-node that v is grouped into, $w(x)$ denotes the weight of node x, and $N(v)$ denotes the non-filter neighborhood of v. The object is to find relaxations whose maximal super-node weight does not exceed some target value k. Uniform filter density is a strong property such that relaxation, up to a point, continues to provide a corral of filter nodes surrounding the non-filter nodes. At a critical point, however, the corral breaks down, and along with it, local containment.

SCALABLE CONTAINMENT: CRITICAL FILTER DENSITY

Given $G = (V,E)$ and filter net $L \subseteq V$, two nodes u and v are reachable, denoted $u \leftrightarrow v$, if there is a path from u to v and vice versa in the network topology that does not traverse a node in L. The reachability relation generates a set of isolated equivalence classes. When reachability is defined with respect to route topology—a subset of all possible paths embedded in the network topology—the relation need not be symmetric. Route topology based reachability induces a partial order on the equivalence classes and further splinters their size.

Figure 32 shows the size of a giant component as a function of filter density for a 12,517-node 2002 (Jan.) NLANR/Oregon Route-Views Internet AS topology and a corresponding random graph with the same number of nodes and edge density. In both cases, we observe a threshold phenomenon or phase transition at a critical filter density. In the case of the Internet AS topology, it is near 3%; in the random topology, it is near 29%. Figure 32 depicts the critical role played by power-law connectivity: if the Internet were connected like a random graph, about 30% filter

deployment would be needed to affect proactive protection. Due to its power-law connectivity, however, the required deployment for achieving near-perfect containment is below 5%.

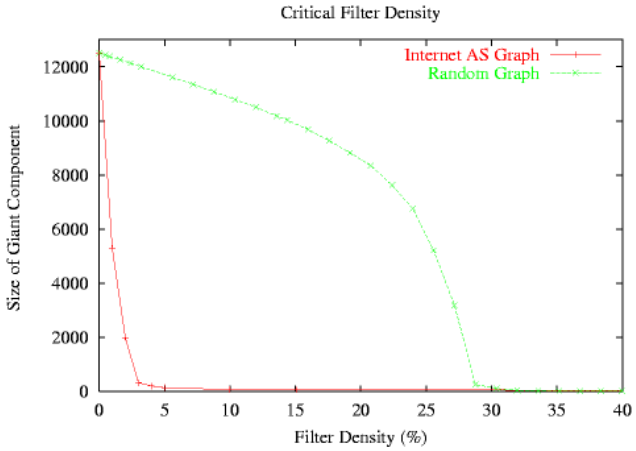

Figure 32 Critical filter density. Size of giant component as a function of filter density for 12,517-node Internet AS topology and corresponding random topology.

From an engineering perspective, presence of a phase transition around a critical filter density implies that deployment below the critical density is ineffective, and deployment above is superfluous. Without knowledge of the existence of a phase transition and its location, sound engineering would not be feasible. For example, heuristics such as "let's deploy at the top 100 ASes" are bound to yield results that may lead to misleading conclusions.

Since a worm attack may be staged from several initially compromised sites, from an attacker's perspective, it is useful to know what the distribution of pocket sizes for a given filter net is such that maximum damage with minimum effort can be achieved.

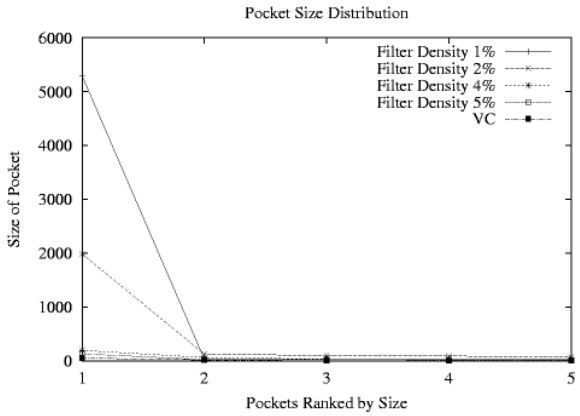

Figure 33 Pocket size distribution under varying filter density for 12,517-node Internet AS topology ranked by size.

Figure 33 shows the distribution of pocket sizes under varying filter density for the 12,517-node Internet AS topology. Pockets are ranked by their size. For filter densities above 4%, the ranked filter sizes are nearly flat as expected from the giant component size in Figure 32. For 2% and 3%

filter densities, we observe that there is a unique giant component of size 1,984 and 5,293 nodes, respectively. The second biggest pocket is of size 127 and 29. Thus, not only is it ineffective to deploy filters below the critical filter density, under such deployment, a single worm placed anywhere in the giant component pocket suffices to infect a significant portion of the system.

PROACTIVE PROTECTION: ROBUSTNESS

The existence of a critical filter density and its approximate location in the 4% vicinity is further confirmed by computing the giant component size for the extended Internet AS measurement benchmark set: CAIDA, RIPE, USC/ISI, and UMich. This is shown in Figure 34 (left). We observe that a sharp transition occurs left of the 5% mark. Figure 34 (right) shows the giant component size as a function of filter density for the artificial topology generator Inet-3.0. We observe a matching threshold behavior near 4%.

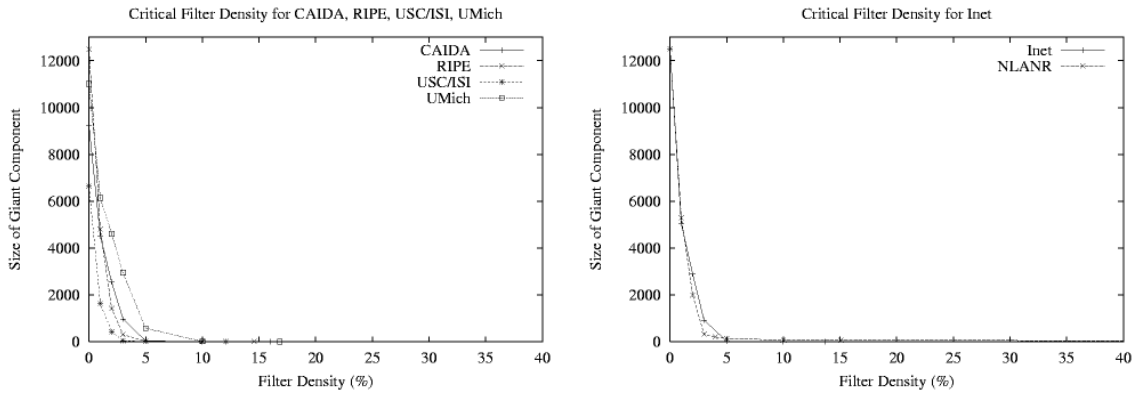

Figure 34 Critical filter density. Left: Size of giant component as a function of filter density for CAIDA, RIPE, USC/ISI, UMich AS topologies. Right: Corresponding results for Inet-3.0.

Threshold phenomena represent a manifestation of the 0/1 law in probability theory. We know from random graph theory that monotone graph properties obey sharp threshold transitions. For power-law graphs, we establish a phase transition with respect to filter density where filter sites are chosen according to VC-based pruning. Our results indicate that the critical filter density is significantly smaller than that of random graphs. Most importantly for network security, we demonstrate an engineering application of phase transition for worm attack prevention where knowledge of its existence and location is crucial for effective defense planning and engineering.

REACTIVE PROTECTION: TRACEBACK

Traceback under content-based DPF for worm attack prevention is straightforward. When filter deployment exceeds the critical filter density, worm spreading is contained within small pockets. By utilizing content-based filters as sensors that emit an alarm when a worm signature is detected, source localization can be achieved with traceback resolution the same as the pocket size wherein detection occurs. If k is the giant component size under a given filter net, then all worm attacks are k-traceable. Thus worst-case traceback resolution is the same as shown in Figure 32 which depicts giant component size as a function of filter density.

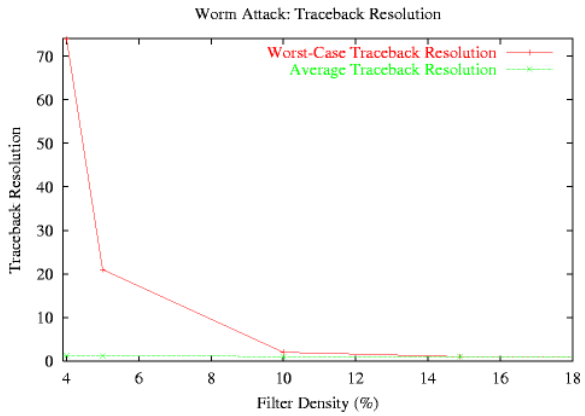

Figure 35 Average and worst-case traceback resolution as a function of filter density.

Figure 35 shows a more detailed picture of traceback resolution as a function of filter density with respect to worst-case (i.e., giant component size) and average traceback resolution. At 4% filter density, worst-case traceback resolution is 74, at 5% it is 21, at 10% it is 2, and at 15% (i.e., VC) k = 1. Average traceback resolution is 1.27 at 4% filter deployment and converges to 1 at 15% deployment. For practical on-line traceback, the identity, size, and membership of pockets must be maintained so that a worm signature detected at a pocket incident on a filter node can be looked up with the appropriate key to determine the identity of its members.

PROTECTION UNDER ROUTING REACHABILITY

Topological reachability $u \leftrightarrow v$ does not imply that routing actually permits a path that goes from u to v, or vice versa, without traversing a filter node. Routing can exert an additional constraining influence on reachability that aids sparse filter deployment.

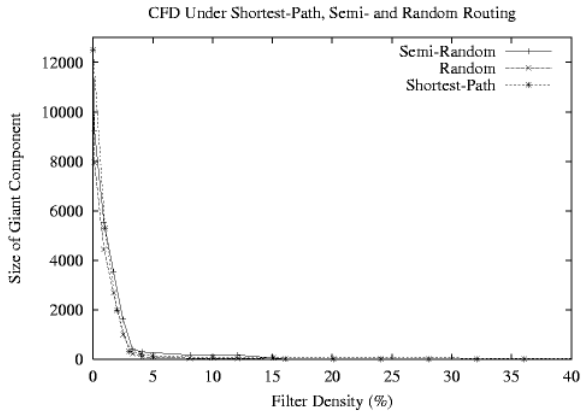

Figure 36 Critical filter density under shortest-path, semi-random, and random routing.

Figure 36 shows giant component size as a function of filter density under shortest-path, semi-random, and random routing for the 2002 (Jan.) NLANR/Oregon Route-Views AS topology. Performance difference across the three extreme routing schemes is negligible. In other words, power-law connectivity exerts a strong influence on topological reachability so that routing is unable to find paths that by-pass strategically placed filters in the filter net.

FINITE TIME HORIZON DYNAMICS

Worm Propagation Dynamics

The preceding performance results were structural meaning that time dynamics involved in worm spreading were not considered. Structural results correspond to worst-case performance since worm propagation—across the entire network system—transpires instantaneously. From a dynamical system perspective, structural results may be viewed as asymptotic damage that is incurred by the network system when an attacker is given all the time in world with respect to a given protective system. In practice, we are interested in the extent of infection at finite time scales ranging from a few days down to a few minutes. As Keynes has remarked, "In the long run, we are all dead." The object of worm attack protection is to prevent wide-spread worm infection, not attempt recovery after significant damage has already been done.

We used the DaSSF-Turbo simulation environment to evaluate worm propagation in the 12,512-node 2002 (Jan.) NLANR/Oregon Route-Views AS topology with full TCP/IP stack running at each node. Each node, i.e., AS, is treated as a single router/host. When a node is infected by a transmitted worm, we treat the entire AS as being infected, a worst-case assumption. The worm attack is instigated from 10 initially compromised sites. When content-based filtering is active, one compromised attack node is selected from each of the top 10 pockets. Thus this includes a compromised node in the giant component. We assume a sophisticated (or lucky) attacker who has access to all the information that the defender has, including the location of filter sites.

The worm model has three key parameters: the scan rate which determines the frequency by which a target is selected (by default, 2 scans/sec), infection time which determines how long it takes for a newly infected node to become an active attacker (by default, 1 second), and scanning strategy which determines how a target is selected. Under random (or global) scanning, a target is selected uniformly randomly from the total address space. Under local scanning, a node selects a target from its immediate neighborhood uniformly randomly.

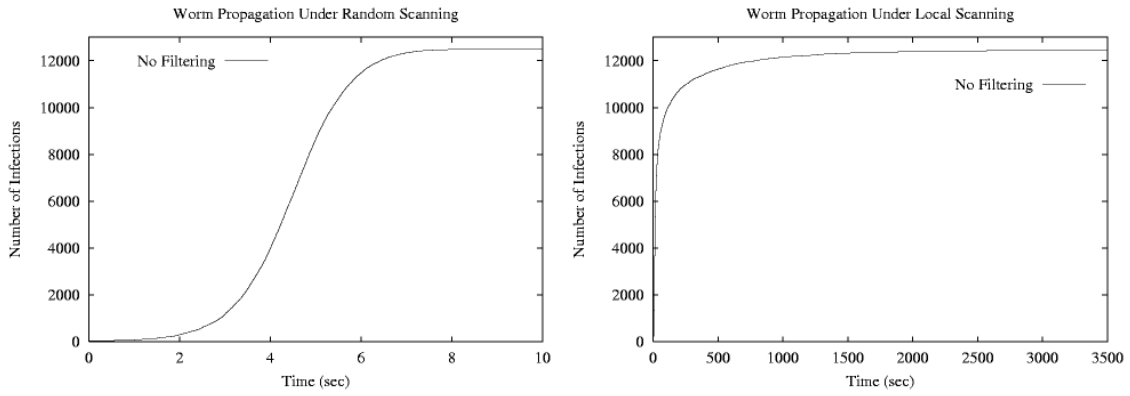

Figure 37 Worm propagation dynamics without content-based filtering. Left: random scanning. Right: Local scanning.

Figure 37 (left) shows the spread of infection as a function time under random scanning when no filtering is performed. We observe a characteristic *S*-curve infection dynamics predicted by epidemiology. Under random scanning without filtering, infection spreads rapidly taking less than 10 seconds to infect the entire system. Once a single host within an AS is infected, it is just a matter of

time before other vulnerable hosts in the same AS get infected. Figure 37 (right) shows infection dynamics under local scanning. At 10 seconds, 2,400 nodes are infected, but to reach full infection it takes significantly more time.

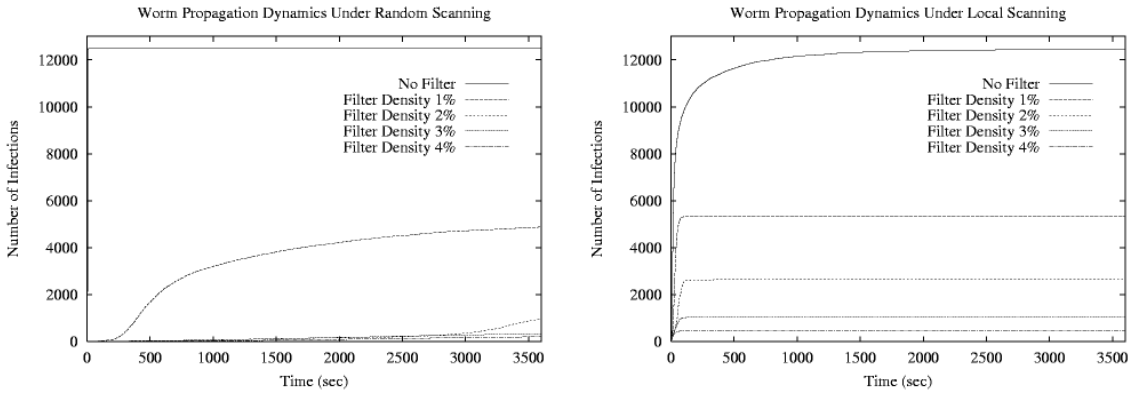

Figure 38 Worm propagation dynamics with content-based filtering. Left: random scanning. Right: Local scanning.

Figure 38 (left) shows worm propagation dynamics under random scanning when filtering with 1%, 2%, 3%, and 4% deployment is invoked. The case of no filtering is shown as a reference point. We observe that 2% and 3% filter deployment can significantly slow down the infection rate. For example, at 2% deployment eventual infection is 2,658 nodes. However, at 1,800 seconds infection is at 141. Figure 38 (right) shows infection dynamics under local scanning. We observe a rapid rise followed by convergence to steady-state infection. Unlike in the case of no filtering, convergence to maximal infection occurs quickly. For example, with 2% filter deployment, convergence to eventual infection 2,658 occurs at time 360 (sec). Thus local and random scanning exert a significant impact on the time dynamics of worm propagation, with and without filtering.

Finite Time Critical Filter Density

Figure 39 (left) shows infection as a function of filter density under random scanning at finite time horizons: 5, 10, 30, and 60 minutes. Phase transition at small time scale occurs later (i.e., smaller filter density) than at large time scale. This implies that a time window—albeit small—is available during which an even smaller filter deployment suffices to hold a worm attack in check. This is relevant if filters in content-based DPF are selectively activated on-demand, akin to a multi-layered defense. During normal operation only a core immune response is active, with full-fledged activation of the filter net triggered when it is deemed necessary to do so. Content-based filtering is a technology that is still at a primitive stage, both with respect to worm signature computation and efficient signature matching. Imperfect signature computation implies that false positives may arise that can adversely affect normal IP traffic. To reduce the impact of safety violation, the immune response may be operated at low intensity—principally in detection mode—until distributed detection by the filter net yields multiple alarms that can be used as a high order signature and invoke a heightened response.

Today's content-based worm signatures number in the thousands. With respect to efficient signature matching, performing per packet signature matching with thousands of signatures puts a heavy burden on processing overhead. This can impede a router or network processor's ability to perform filtering at line speed. A multi-layered defense that narrows an attack's signature set to a manageable subset may be critical in facilitating line speed filtering, a key requirement of network

oriented filtering. Imperfections may also lead to staleness where worm packets go undetected and are let through. The Achilles' heel of current content-based filter technology lies in effective signature computation for new forms of worms, including mutations (i.e., polymorphic and metamorphic worm variants). This is one of the challenges under investigation.

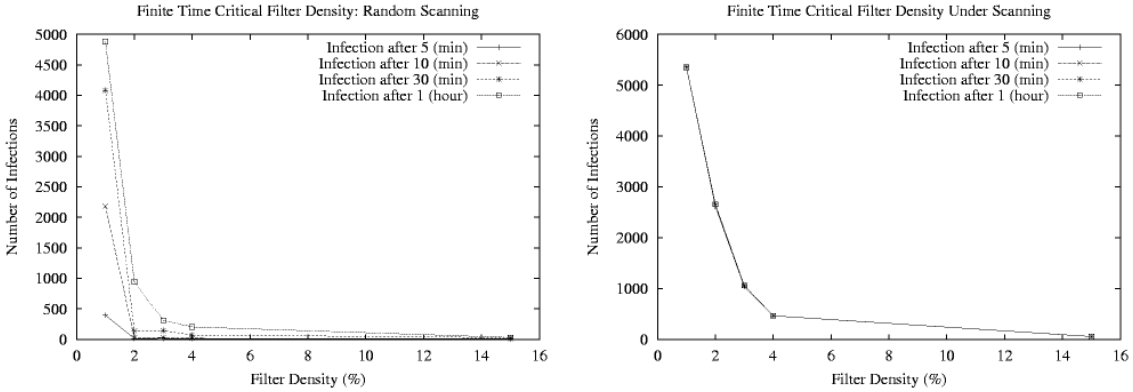

Figure 39 Finite time critical filter density. Left: Random scanning. Right: Local scanning.

Figure 39 (right) shows finitary critical filter density under local scanning. Unlike the case of random (or global) scanning, convergence of steady-state infection is rapid and negligible breathing room is afforded by sub-critical filter density deployment. The reason for this is that under intelligent initial attack configuration, such as top ten pocket infiltration, damage is dominated by the largest pocket—giant connected component—which renders contamination inherently local (in the distance metric of topological reachability). The diameter of the global AS-level Internet is small—11 in the case of the 2002 (Jan.) NLANR AS topology—which allows rapid propagation through locally expanding contamination, i.e., percolation.

REFERENCES AND RELATED WORK

The following sections contain select references and related work, including reports and publications by members of the Network Systems Lab.

REPORTS AND PUBLICATIONS FROM THE PROJECT

Published Papers and Reports

The foundational papers for the project are:

- K. Park and H. Lee. On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets. In *Proc. ACM SIGCOMM '01*, pp. 15-26, 2001.

- K. Park and H. Lee. On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack. In *Proc. IEEE INFOCOM '01*, pp. 338-347, 2001.

The first advanced the notion of route-based distributed packet filtering and its effectiveness in power-law networks. It is one of the first papers making a link between power-law connectivity and network security. The second paper gave a performance evaluation of probabilistic packet marking. Its performance in power-law networks under partial deployment for IP traceback and networking monitoring is under exploration.

The next report provides a comprehensive description of the Static DPF Simulator (version 2), which was released in May 2002 to the DARPA community.

- A. Selcuk, K. Park, and H. Lee. The Static DPF Simulator (v.2). Technical Report CSD-TR-008, Department of Computer Sciences, Purdue University, September, 2002. (http://www.cs.purdue.edu/nsl/dpf-v2.pdf).

A description and performance evaluation of the DaSSF-Turbo simulation environment is provided in Hyojeong Kim's M.S. Thesis:

- Hyojeong Kim. *Performance Evaluation of Route-Based Distributed Packet Filtering for DDoS Prevention in Large-Scale Networks*. Master's Thesis, Purdue University, December, 2003 (http://www.cs.purdue.edu/nsl/hjkim-ms-thesis.pdf).

A comprehensive document describing DaSSF-Turbo is in preparation for release with the DaSSF-Turbo software package. The thesis describes the BGP Extension based route-based DPF protocol specification, and provides performance evaluation of route-based DPF under infrastructure attack.

Papers in Preparation

The focus of our effort in the past two years has been on generating new research results, technologies and tools. Key papers describing the results are under preparation. They are:

- K. Park, A. Selcuk, and H. Lee. Scalable DDoS prevention in power-law network using route-based distributed packet filtering. In *preparation*, 2003.

The full paper (of ACM SIGCOMM '01) that introduces and provides performance evaluation of route-based DPF in power-law networks.

- H. Kim and K. Park. Toward large-scale network simulation in PC cluster environments. In *preparation*, 2003.

  The technical paper describing DaSSF-Turbo and its use in facilitating large-scale network simulation. Benchmark records are set for BGP using DaSSF-Turbo's measurement subsystem that facilitates memory, CPU, and communication cost management.

- B. Bethala and K. Park. Scalable protection against worm attacks in power-law networks. *In preparation*, 2003.

  This paper shows the effectiveness of content-based DPF in power-law networks under known worm attacks. Existence and location of the critical filter density are identified, and dynamic performance evaluation carried out.

- H. Kim, K. Park, and A. Selcuk. Scalable protection against DDoS infrastructure attack in large-scale dynamic networks. In *preparation*, 2003.

  This paper describes the BGP Extension based route-based DPF protocol and its resilience against failures and infrastructure attack in dynamic network environments.

- H. Kim and K. Park. Partitioning and load balancing in power-law networks. In *preparation*, 2003.

  This paper provides an in-depth study of the partitioning problem for power-law networks where the effect of power-law connectivity on load balancing and parallel speed-up is evaluated. A new algorithm is advanced.

- C. Smith, B. Armbruster, and K. Park. A filter placement problem with application to computer network security. In *preparation*, 2003.

  This paper shows that perfect proactive protection (i.e., containment index = 1) is *NP*-hard, and shows special cases where a polynomial algorithm exists for optimal filter placement.

- Kihong Park. On the effectiveness of probabilistic packet marking for IP traceback under partial deployment in power-law networks. In *preparation*, 2003.

  This paper demonstrates that probabilistic packet marking can be used for IP traceback in power-law networks under sparse deployment.

POWER-LAW NETWORKS

Power-Law Network Modeling and Analysis

Several papers and data sources are relevant for power-law network performance evaluation including network security. First, we mention relevant modeling and analysis papers aimed at describing power-law networks and their properties.

- M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *Proc. ACM SIGCOMM '99*, 1999.

  This paper performs a comprehensive analysis of NLANR AS topology data and shows that Internet AS topologies exhibit power-law connectivity with respect to their degree distribution.

- W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *Proc. ACM STOC '00*, pages 71–180, 2000.

  This paper provides and early model for power-law random graphs. A more cleaner and improved random graph model based on average degree sequences that generalizes classical random graph theory is given by the next paper.

- F. Chung and L. Lu. The average distance in a random graph with given expected degrees, *Proceedings of National Academy of Sciences*, to appear, 2003.

  Bollobas' textbook, *Random Graphs* (2nd ed., 2001), includes a brief treatment of small world phenomena, a property of power-law graphs (cf. Chapter 10).

- R. Albert, H. Jeong, and A. Barabasi. Diameter of the World Wide Web, *Nature* **401**: 130-131, 1999.

  This paper shows that the connectivity structure of the World Wide Web has a power-law degree distribution. Power-law connectivity of WWW graphs has been independently discovered by several others, including Kleinberg *et al.* (1999).

Power-Law Measurement Topology

The main benchmark suite for our study are Internet AS measurement graphs obtained from BGP dumps as part of the Oregon Route-Views project. The two sites are:

- NLANR data: http://moat.nlanr.net/Routing/rawdata

- Oregon Route-Views: http://archive.routeviews.org

The RIPE AS topology data, also based on BGP dumps, are available from http://www.ripe.net. AS topology data from CAIDA which are based on traceroute data are available from http://www.caida.org. Traceroute based AS topology data from USC/ISI (Mercator project) were made available by Ramesh Govindan. The UMich data, which is a combination of NLANR and RIPE data with additional processing, were made available by Sugih Jamin. Inet is available at http://topology.eecs.umich.edu/inet.

DDOS REFERENCES

The overall state of DDoS research and associated technology is as yet immature and much remains to be done. Presently the tried-and-tested, directly usable method is redundancy-based service distribution, an application/network layer technique that service providers can use to enhance availability when subject to DDoS attack. The simplest way is to replicate services across several proxies so that a single-point-of-failure is eliminated. Since this is already done for load balancing and fault-tolerance purposes, multiple birds, including DDoS, are caught with one stone. A

limitation of this approach is that it is per-provider driven (i.e., provides local protection in the sense of a firewall). Also, these systems are vulnerable to infrastructure attacks given their critical dependence on name servers.

In general, the *k*-out-of-*n* property may be employed to encode an object of size *k* into a larger object of size $n = k + h$, and distribute the *n* pieces across multiple (ideally *n* independent) sites. Here *h* represents the amount of redundancy. To retrieve the original object, a fetch is performed and as long as *k* pieces arrive in a timely manner and unharmed—it does not matter which *k* pieces—recovery is assured. This method has its roots in forward error-correction (i.e., channel coding), with a space-optimal code provided by Rabin:

- Michael Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM* **36**(2): 335-348, 1989.

The *k*-out-of-*n* property has been applied in a number of distributed system contexts including real-time distributed database applications:

- Azer Bestavros. An Adaptive Information Dispersal Algorithm for Time-critical Reliable Communication. In I. Frisch, M. Malek, and S. Panwar, editors, *Network Management and Control, Volume II*, chapter 6, pp. 423-438. Plenum Publishing Corporation, New York, 1994.

Another interesting development is probabilistic packet marking (PPM) which seeks to identify the path undertaken by a DoS attack by stamping packets with link/hop information at transit points in a network. As with source routing—but in reverse form—the method is trivial if hop information is appended to the packet header. A growing header, however, is undesirable. The innovation achieved by PPM is the use of constant space traceback. The idea is for a router to stamp a traversing packet with local hop information probabilistically. A downstream router may overwrite the mark placed by an upstream router, but if sufficiently many packet traverse the same route—an assumption that holds for DoS—each transit point of a path gets sampled with high probability and the path taken by a flow may be reconstructed at the receiver.

The idea behind PPM was first described by Burch and Cheswick (1999), however, Savage *et al.* deserve the credit for recognizing its utility for IP traceback under spoofed DoS attack and studying coding issues:

- S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *Proc. ACM SIGCOMM '00*, pp. 295-306, 2000.

On the coding side, Adler provides a 1-bit method for IP traceback—a generalized form—and analyzes its trade-offs, including sampling:

- Micah Adler. Tradeoffs in probabilistic packet marking for IP traceback. In *Proc. ACM STOC*, pp. 407-418, 2002.

Park and Lee study the performance properties of PPM, including its vulnerability to spoofing of the marking field. They also show that PPM is limited in its effectiveness when faced with DDoS attacks where traffic volume is distributed over many paths. This enables a DDoS flow to "fly under the radar" undetected due to PPM's sampling limitation.

- K. Park and H. Lee. On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack. In *Proc. IEEE INFOCOM '01*, pp. 338-347, 2001.

iTRACE is a messaging (ICMP) based IP traceback variant. Hash-based traceback is a storage based scheme that may be used in audit trail analysis. It is principally of interest due to its coding prowess, an application of Bloom filters.

Moore *et al.* present an elegant method for detecting spoofed DDoS attacks based on backscatter analysis which samples reflected packets assuming random source address spoofing.

- D. Moore, G. Voelker, and S. Savage. Inferring Internet denial-of-service activity. In Proc. *USENIX Security Symposium*, pp. 9--22, 2001.

Ingress and egress filtering are needed, but by themselves, are limited with respect to protective performance. As discussed in the report, they may be viewed as special cases of route-based filtering. Other methodologies for DDoS attack protection such as pushback (e.g., Floyd *et al.* and *Yau et al.*) are neither practical nor intellectually interesting. Chang provides an overview of DDoS work.

- R. Chang. Defending against flooding-based distributed denial-of-service attacks: A tutorial. *IEEE Communications Magazine*, pp. 42-51, 2002.

WORM REFERENCES

Worms, which are network born viruses, present a pressing cyber threat to the well-being of the Internet, even more so than DDoS. Denial of service is a resource depletion attack whose consequences for a networking and service provisioning system are severe, but stop short of corrupting or destroying information by overtaking a system.

One of the foundations of worm dynamics is epidemiology, which studies how infection spreads in a population through contact various processes. In the early 1990s, Kephart and White wrote a sequence of papers that emphasized the connection between epidemiology and computer viruses which were garnering attention, including the popular media (e.g., Michelangelo virus) In classical epidemiology, as a function of birth (infection) and death (clean-up) rate, there is a threshold below which wide-spread infection cannot take hold, and vice versa, when the threshold is exceeded. The *S*-curves shown in the report are typical instances of infection propagation dynamics. A textbook reference on mathematical epidemiology and an excellent survey are provided by

- Normal Bailey. *The Mathematical Theory of Infectious Diseases*. Oxford University Press, New York, 1987.

- Herbert Hethcote. The mathematics of infectious diseases. *SIAM Review* **42**(4):599-653, 2000.

The key difference between classical epidemiology and worm propagation is that it takes but a few minutes for a fast scanning worm to spread to the bulk of unpatched systems in a large-scale network. For all practical purposes, there is no operational end system-based recovery rate (i.e., death rate) active at small time scales relevant for reactive protection. If an end system is already patched against a known worm, then it is safe. If not, with high likelihood it will get infected. An epidemiological study of worm propagation, for research purposes, is not very meaningful. Recent efforts in this direction include Zou *et al.* (2002) and Chen *et al.* (2003).

End system patching is necessary but not sufficient to achieve effective protection. Content-based DPF at transit points via a strategically placed filter net is required to prevent massive infection and damage in future worm attacks. This leads to a key challenge: how to defend against new worms including mutations. Of these, reliable worm signature computation and matching are at the heart of the problem. Most of the work on signature computation and malware detection has focused on viruses resident on end systems that are embedded in other programs. Examples include:

- Brian Chess. Improving computer security using extended static checking. In *Proc. IEEE Symposium on Security and Privacy* '02, pp. 160-173, 2002.

- M. Christodorescu and S. Jha. Static analysis of executables to detect malicious patterns. In *Proc. USENIX Security Symposium,* pp. 169-186, 2003.

- J. Kephart and W. Arnold. Automatic extraction of computer virus signatures. In *Proc. Virus Bulletin Conference*, 1994.

- R. Lo, K. Levitt, and R. Olsson. *MCF*: a malicious code filter. *Computers & Security,* **14**(6):541-566, 1995.

- M. Schultz, E. Eskin, E. Zadok, and S. Stolfo. Data mining methods for detection of new malicious executables. In *Proc. IEEE Symposium on Security and Privacy,* pp. 178-184, 2001.

Formal methods are extremely limited in dealing with non-toy real-world environments—the only success story is that of VLSI testing where layout regularity renders the problem tractable. Some virus specific observations can be found in

- D. Chess and S. White. An undetectable computer virus. In *Proc. of Virus Bulletin Conference*, 2000.

- Fred Cohen. Computer viruses—theory and experiments. *Computers and Security*, vol. 6, pp. 22--35, 1987.

The main hope with worms is that they are transmitted by packets and target a small set of end system vulnerabilities, buffer overflow being the dominant flaw targeted by the bulk of recent and past worms. End system vulnerabilities that can be triggered remotely are restrictive and admit the possibility of network based content-based signature detection that are not available to viruses resident on end systems.

For example, in the case of the Sapphire worm, it targeted a stack buffer overflow vulnerability of Microsoft SQL Server Resolution Service (SSRS) that can be triggered by a single UDP packet. SSRS provides a referral service—returns an IP address and port number of a SQL server—by listening on UDP port 1434. A content-based filter only needs to be sensitized to UDP packets with port number 1434 and perform a length check on an argument inscribed in the payload. A similar observation holds for the Blaster worm which targeted a buffer overflow vulnerability in Microsoft DCOM RPC. By inspecting TCP packets with port number 135 (and a few others), content-based filtering can be effected. In fact, for DCOM RPC which is intended to be used in intranets, simple discarding of all TCP port 135 packets (which most firewalls do) would suffice.

Worm malware may be distributed over several packets, however, the constrained nature of the remote services they target makes it significantly harder to obfuscate their malicious nature when

compared to virus that operate in a less constrained environment. Our current work is directed at solving the content-based filtering problem for worms for use in content-based DPF.

SCALABLE SIMULATION REFERENCES

ns-2 and OPNET are popular network simulators used in academia and industry. The LBNL network simulator (*ns*) is an event-driven simulator that was derived from Keshav's REAL simulator which, in turn, was based on NEST. Although *ns* has done much to bring network simulation to the masses, it has limitations on two grounds: one, it is a serial program and simulation of moderate size networks with nontrivial traffic rates exacts a heavy burden on simulation time, and second, its event-driven nature makes it difficult to extend it to a more flexible process-oriented environment (e.g., dynamic routing is awkward to implement).

SSF adopts a process-centric view, similar to that of the *x*-kernel, that facilitates a flexible and scalable—with respect to process and event coordination—simulation in large-scale networks. DaSSF is a C++ based realization of the SSF framework for parallel and workstation cluster environments. DaSSF provides advanced synchronization, and comes with a TCP/IP stack and DML network configuration support. Further information on SSF, *x*-kernel, and DaSSF are available at

- SSF: http://www.ssfnet.org/homePage.html
- *x*-kernel: http://www.cs.arizona.edu/xkernel
- DaSSF: http://www.cs.dartmouth.edu/jasonliu/projects/ssf/index.html

PDNS is a simple-to-use parallel extension of *ns* running over DSM multi-processor environments connected by Myrinet/Ethernet. Although it is a reasonable parallel extension of *ns*, it still structurally suffers under the limiting event-driven abstraction exported by *ns*. To get the best of both worlds, albeit sacrificing with respect to the deployment/user base of *ns* for the long term, we have chosen DaSSF as our simulation kernel for its support of workstation clusters. Following is a link to PDNS:

- PDNS: http://www.cc.gatech.edu/computing/compass/pdns/index.html

Yet another attempt at facilitating large-scale simulation is based on fluid simulation wherein individual packet information—when deemed allowable—is dispensed to affect more simple book keeping and scalable state management. The problem with fluid based simulation is two-fold: one, the situations in networking where this is useful may be severely limited (a simple form of congestion control may be one of the few applicable instances), and two, once traffic is multiplexed and starts mixing, the aggregated traffic takes on increasingly complex shapes whose book keeping may not be significantly simpler than that of event-based tracking. Liu *et al.* (1999) and Kiddle *et al.* (2003) represent attempts in this direction.

Our own work, building on top of DaSSF, focuses on facilitating a simulation environment for large-scale network topologies where automated network configuration, network and system measurement/monitoring, and partitioning for memory, CPU, and communication load balancing comprise critical components to enabling useful simulation. The architecture of DaSSF-Turbo is described in Hyojeong Kim's M.S. Thesis. A separate document describing DaSSF-Turbo will accompany the public release of DaSSF-Turbo.