Source Prefix Filtering in ROFL

Technical Report CUCS-033-09

Hang Zhao
zhao@cs.columbia.edu
Columbia University

Maritza Johnson
maritzaj@cs.columbia.edu
Columbia University

Chi-Kin Chau
chi-kin.chau@cl.cam.ac.uk
University of Cambridge

Steven M. Bellovin
smb@cs.columbia.edu
Columbia University

***Abstract*—Traditional firewalls have the ability to allow or block traffic based on source address as well as destination address and port number. Our original ROFL scheme implements firewalling by layering it on top of routing; however, the original proposal focused just on destination address and port number. Doing route selection based in part on source addresses is a form of policy routing, which has started to receive increased amounts of attention. In this paper, we extend the original ROFL (ROuting as the Firewall Layer) scheme by including source prefix constraints in route announcement. We present algorithms for route propagation and packet forwarding, and demonstrate the correctness of these algorithms using rigorous proofs. The new scheme not only accomplishes the complete set of filtering functionality provided by traditional firewalls, but also introduces a new direction for policy routing.**

I. Introduction

Firewalls have long been a mainstay of network security. While their utility has diminished in recent years [1], due to increasingly rich topologies, they are still valuable. In recent work [15], [16], we extended firewalls to work with MANETs (Mobile Ad Hoc Networks), using routing protocols to implement the firewall layer. In a MANET, a ROFL (Routing as the Firewall Layer) serves two important purposes: it not only helps implement a security policy, it causes unwanted packets to be dropped as early as possible, thus conserving battery power.

In common with most firewalls and routing protocols, ROFL makes its decisions based on destination addresses and port numbers. For wired networks, this is quite proper, since it is not possible to trust source addresses coming from beyond the firewall [5]. In MANETs, where connectivity patterns are constantly changing, the situation is subtly different. While it remains true that the behavior of "untrusted" nodes (i.e., those not protected by the firewall) cannot be relied upon for security purposes, adding source address constraints to MANET rules have a second, and equally important function: they define the boundaries of the *policy region*. That is, source address rules define the boundary between the "inside" — the portion of the network protected by the firewall — and the outisde. In a traditional wired network, the firewall itself is the boundary marker, with the network topology determining the inside and outside.

Consider the MANETs shown in Figure 1. In Figure 1(a), nodes I1 and I2 provide firewall functionality against outside node O1. In Figure 1(b), nodes I1 and O1 have moved, changing the connectivity patterns. I2 can no longer reach I1 except by going through O1; in addition, I4 now has a direct link to O1 and must activate firewall mechanisms against traffic originating from it. We must now rely on O1's source address to make such decisions, rather than on a fixed topology. (As is discussed later, we in fact rely on the cryptographically verified network identity of a network neighbor. By contrast, a fully distributed firewall, of the type described in [1], could have n^2 security associations, between every pair of nodes that are communicating. In fact, if we wish we can often have even fewer cryptographic associations: if a neighbor's purported identity is not named as "trusted" for any firewall rule, its true identity is irrelevant and can be ignored.)

In this work, we show how to extend ROFL to use source address rules to determine policy region boundaries. Since ROFL relies on routing, we are perforce describing a new form of policy routing (Section II). We provide proofs that our algorithms (Section IV) are correct (Section V), and show how to incorporate two criteria, cost and risk, into a routing metric framework (Section III). We use our new metric to map ROFL into the routing algebra framework (Section VII).

II. Policy Routing in ROFL

Our proposed ROFL [16] scheme allows early drop of unwanted traffic by intermingling filtering information with routing announcement. Traditional firewalls have the ability to allow or block traffic based on source address as well as destination address and port number. Doing route selection based on source addresses is a form of policy routing. The current version of ROFL performs traffic filtering focusing on destination IP address and port number only. Therefore, we extend the ROFL

Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

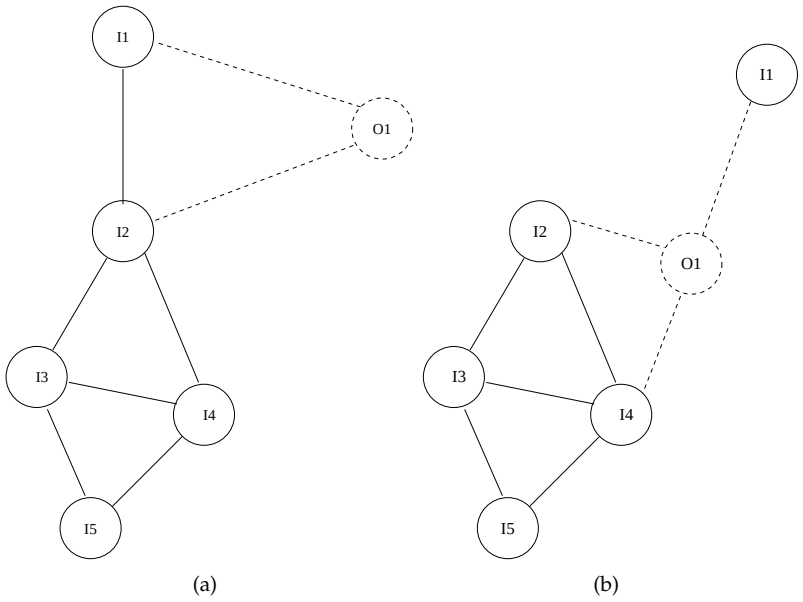

Fig. 1. Two different topologies in a MANET, after O_1 has moved.

scheme by including source prefix constraints in route announcement.

A. The Scheme

A routing announcement in ROFL is of the form

$$R = \{d : s/m, M\}$$

where d is a destination address prefix, s is the service port number, m is a prefix length, and M is a metric. We introduce a source prefix constraint S into the scheme,

$$R = \{d : s/m, S, M\}$$

where S indicates a set of source address prefixes such that data traffic coming from those address prefixes is allowed to access service s provided by destination address d. More precisely, we define $S = \{p_1, p_2, ..., p_n\}$; each p_i $(0 \leq i \leq n)$ is a source address prefix. Therefore no source prefix constraint is specified if $S = \phi$. (Alternately, we could simply set $S = \{0/0\}$, i.e., all addresses are accepted; for clarity in this paper, we prefer to distinguish between no source prefix constraints and one that happens to have no effect.) We remark that in some situations, it is possible to implement S as a Bloom filter [2] on the set of source addresses or networks of a given prefix length. Bloom filters are a space-efficient data structure that can compress the representation of a set of members in a compact manner, albeit with some chance of false positives. Bloom filters are particularly useful in MANETs, where there is little topological structure and each allowed node may be identified by a flat address.

Source prefix constraint in ROFL controls the propagation of routing advertisements as well as the packet forwarding procedure. A routing announcement will be passed to a node if and only if the node itself is allowed to access the advertised service. During packet forwarding phase, a packet will be dropped immediately if it is coming from a source whose address is not specified in the source prefix constraint of a matching route. Therefore, routing advertisements in this new scheme are handled similarly to any other routing announcements, except that another level of checking needs to be performed based on the packet's source address. The source node and any intermediate routers do a longest-prefix match on the advertisement. The packet is forwarded if and only if such a matching route is found and the source address in the packet header is contained in the source prefix constraints of that matching route; Otherwise, the packet is dropped.

The new scheme implement the complete set of filtering functionality provided by traditional packet filter firewalls by adding source prefix filtering into routing advertisement. For example, to allow data traffic coming from source address p_1 to reach destination host d on port number s, we simply announce a routing advertisement $R = \{d : s/48, \{p_1\}, M\}$. Blocking certain traffic is a bit trickier. If p_1 is the only source address that is not allowed to access service s on host d, we could announce $R = \{d : s/48, S, M\}$ with $p_1 \notin S$. Otherwise, if d wants to completely block traffic on port s only, it announces $R = \{d : s/48, \infty\}$ together with $R' = \{d/32, M\}$. Recall that traditional firewalls allow the wild card $*$ to appear in any field of source address as well as destination address and port number. In our new ROFL scheme, $*$ in destination port number is equivalent to $d/32$ and $*$ in source address can be described as $S = \phi$. If $*$ appears in the destination address field, we can easily adjust the

prefix length m to cover the corresponding subnet.

In the above scheme, route selection is based in part on the source address. Therefore the actual route announcement becomes a form of policy routing. There are two obvious approaches to inserting policy constraints into routing announcements. First, all relevant nodes along the route propagation paths could create or modify the policy statement, in accordance with some central policy. A better approach would be allowing only the node advertising the service — the route originator — to embed a source prefix constraint in routing announcement. Subsequent receivers of this route announcement should not alter the embedded policy statement. We suggest the second approach for a few reasons. First of all, it is the route originator that has the best knowledge of who is or is not allowed to access a certain service. Second, allowing intermediate routers to modify policy statements requires some kind of trust relationship established amongst them. More importantly, the first approach might work for static nodes; but would not be able to cope with dynamically changing topologies in some wireless environments, such as MANETs.

B. The Transit Node Problem

Consider the network topology shown in Figure 2. A node in Net A is advertising a service that a node in Net B wishes to access; however, the transit node T is not an authorized source for this service. That is, the source address policy advertised for this service does not include T; it does, however, include Nets B and C. What should happen?

One option would be to permit traffic from B to transit T; after all, Net B is an authorized source. This in turn would require that the border node in Net A advertise the service to T, which would presumably pass it on to Net B. This option is insecure: when packets for the service arrive at Net A, it is not possible to tell whether they originated from Net B or were forged by transit node T.

We thus adopt the following policies:

1) Routing advertisements are never propagated to a node not authorized for that service, according to the originator's policy.
2) Packets for a service are only accepted from nodes to which routing advertisements were sent. In a wired net, this is generally easy; in a MANET, it will likely require some form of cryptographic authentication of the neighboring node, since in a wireless environment it is generally very difficult to determine the precise source of a packet.

Note that these principles are not related to route aggregation. A shorter prefix may be transmitted to T if and only if such a prefix already existed and T was authorized for it.

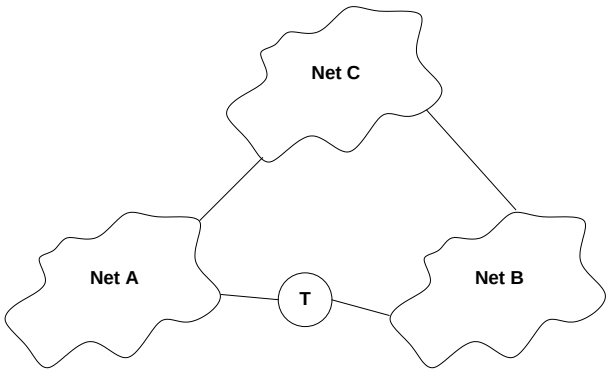

Fig. 2. Two subnets connected by both a transit node and a transit net.

III. Revisiting The Routing Metric

As always, our routing protocol requires a metric M that describes "cost" of a path. Metric construction varies. In BGP [11], the AS hop count is used. In OSPF [10], the network administrators specify arbitary metrics for each link.

Our needs are more complex. As described in [15], we require both a cost metric $\mathcal{C}$ and a risk metric $\mathcal{R}$. Intuitively, "cost" represents the total expense of using a given link. It may reflect bandwidth, power limitations and the difficulty of replacing or recharging batteries, etc. "Risk", by contrast, is related to the safety of a particular transmission. A node may have been captured or otherwise compromised (a significant issue for MANETs operating in hostile territory); if so, it should not be used for the path. The risk metric captures the exposure of a node to such events.

Our routing metric must handle both of these concepts. Furthermore, we wish for a scheme that can optimize both simultaneously, with the tradeoff between higher cost and lower risk made by local authorities. This tradeoff may vary over time; furthermore, it may be different for different services.

We define the cost to be the property of a link; risk is the property of a node. This is a reasonable approach on a MANET. As topologies change, the power consumed by a transmission will vary, thus affecting its cost. Similarly, in a battlefield environment the location of the front line may change, and with it the physical danger to any given node.

The tradeoff between cost and risk is represented by a value γ, $0 \leq \gamma \leq 1$. The actual metric used for route calculations is thus $M = \gamma\mathcal{C} + (1-\gamma)\mathcal{R}$.

While the precise sources of these values are not crucial, we do offer some plausible suggestions. We do suggest, however, that in general they should change no more frequently than the time required for the routing state to converge, or routing loops could occur.

Link costs are best determined by the transmitting node. If nothing else, it has the best knowledge of its

battery state and the power required to send a message. In MANETs, a meaningful approach may be to define link costs as proportional to the geographical distance between two nodes. In wireless communication, it is well-known that short-hop communications can improve the capacity of wireless, because short-range transmissions cause less interference than long-range transmissions. Hence, it is more preferable to choose a long path with many short-hop communications than a short path with a few of long-hop communications. This is best accomplished by using a sublinear relationship between distance and cost. Also, the link costs can be proportional to the number of neighbors of the transmitter; the more neighbors a node has, the more interference the transmitter will inflict on or receive from others.

Risk is more problematic. In combat situations, an enemy may prefer to capture certain military ranks. In such situations, nodes with lower-ranking soldiers may be evaluated as a lower risk. Often, nodes are the best judge of their own risk levels; in other situations, they may be centrally determined. In either case, the actual value used must be known by, and hence transmitted to, all nodes that need to calculate M for a given hop. Generally, this will be the node's immediate neighbors, as everyone else will simply use the value of M broadcast by the routing protocol. That said, in environments where γ changes considerably more rapidly than $\mathcal{C}$ and $\mathcal{R}$, it may be worthwhile transmitting $\mathcal{C}$ and $\mathcal{R}$ rather than M.

γ has the most intriguing properties. It could be set centrally; alternatively, each node could use its own values. Again, what is important is that all nodes agree on M for each link; this can be done either by only broadcasting M, or by broadcasting $\langle \mathcal{C}, \mathcal{R}, \gamma \rangle$ in each advertisement.

Note that in the absence of aggregation points, metrics for a given service $d : s$ are never compared with those for another service: the longest-prefix match is always preferred, regardless of the difference in metric. Therefore, all of these values can be service-specific; in particular, γ could be advertised by the service originator $d : s$, and thus more properly be viewed as $\gamma_{d:s}$.

By folding cost and risk into a single value, the routing system automatically picks the optimum path between any two points. The total routing expense, then, is $\sum_{i,j} M_{d_i:s_j}$, the sum of the individual metrics. The total system expense, though, depends not just on the state of the routing system, but also on the traffic matrix. This is not known (or knowable) locally. Accordingly, we could in fact use $M = \Gamma\mathcal{C} + (1 - \Gamma)\mathcal{R}$, where Γ is a centrally-specified function of γ. By varying it, the overall system expense can be optimized.

IV. Algorithms

Routers are generally composed of two fundamental mechanisms, the control plane and the data plane. The control plane, sometimes known as route generation, produces a valid path from source to destination by exchanging routing information with other nodes. The data plane, or message forwarding, relays packets from node to node until they reach their final destination, following the selected route. In ROFL, we implement packet filtering by layering it on top of routing. ROFL is agnostic to the specific type of routing protocols used; only slight modifications are needed during the route propagation and the packet forwarding phases.

Because we do not change route calculations or prefix lookups, our new ROFL scheme can cope well with any distance vector or link state algorithms: route announcements in ROFL with source prefix filtering are handled the same way as conventional ones during this phase. There are some changes, however. Therefore, in this section we present our algorithms for route propagation and packet forwarding.

A. Route Propagation

Suppose a service provider with destination address d issues a routing advertisement $R = \{d : s/m, S, M\}$ with S being the source prefix constraint. Let N_b denote the set of neighbors for each node u. We use the following algorithm for route propagation:

RoutePROP(u, N_b)

```
1  if u is the service provider of s
2    R ← genRt(d, s, m, S, M);
3  else
4    R, S ← recvRt();
5    calculateFIB(R);
6  for each neighbor h ∈ N_b
7    if S == φ or h ∈ S
8      sendRt(R, h);
9    else
10     discardRt(R);
```

The above algorithm distinguishes different behaviors of a node u depending on whether u is the route originator (i.e. the service provider of s) or not (Line 1). Initially, the service provider generates a routing announcement R with appropriate routing information inclosed (Line 2), and then propagates R to all the neighboring nodes which are allowed to access service s (Line 6-10). Upon receipt of the routing announcement R, a node extracts the source prefix constraint information from R (Line 4), calculates its forwarding information base (FIB) to reflect the update from newly received route (Line 5), and then forwards R to its neighbors in a similar fashion (Line 6-10).

B. Packet Forwarding

When a routing path is established between a source and destination pair by underlying routing protocol, every node within a service's policy region will consult its local routing table T_R, as well as the source prefix

constraint to make the packet forwarding decision. More precisely, when a node u receives a packet K from its neighbor, it executes the following algorithm:

```
PKTFORWARD(u)
1  K ← recvPkt();
2  p_s, p_d ← procPkt(K);
3  if u is the destination
4    DONE
5  else
6    R ← LPrefixMatch(T_R, p_d);
7    if R ≠ φ
8      S ← extractRt(R);
9      if S == φ or p_s ∈ S
10       forwardPkt(K, R);
11   else
12     discardPkt(K);
```

Upon receiving data packet K (Line 1), node u retrieves source and destination addresses p_s, p_d respectively from the packet header (Line 2). If K arrives at its destination, we are done (Line 3-4); Otherwise packet K needs to be further forwarded based on u's local routing table. Node u performs longest prefix matching on destination address p_d (Line 6). If a matching route R is found, u extracts the source prefix constraint from R (Line 7-8). Then the packet is forwarded towards its destination if and only if no source constraint specified or it is originated from a legitimate source allowed to access this service (Line 8-10). Otherwise, the packet is discarded (Line 12).

C. Discussion

Our *route propagation* and *packet forwarding* algorithms described above are similar to those dealing with conventional routing advertisements. With our new ROFL scheme, though, there is an additional check against the source prefix constraint which constraints route propagation and packet forwarding decision at each node.

A crucial question in any firewall design is defining which nodes should be allowed access to the protected service. In ROFL, this translates to defining for which routing announcements are forwarded to which nodes, i.e. the policy region. With source constraints added into routing announcement, the service provider has full control of the propagation of its service announcement. On one hand, routing announcements won't be seen by nodes that are not allowed to access this service; On the other hand, data packets originating from illegitimate sources will be dropped far earlier, since each router along the path is now acting as a firewall to perform packet filtering based on the destination address and port number as well as source address.

V. CORRECTNESS

We claim that our new ROFL scheme with source prefix filtering will not cause any routing mistakes. More precisely, under the assumption that the underlying routing protocols are correct, we claim that our algorithm (a) will produce equivalent results for packets not blocked by policy constraints, and (b) will properly drop unwanted packets.

THEOREM 1 ***Correctness of the Route Propagation Algorithm***
The route propagation algorithm ensures that route advertisements for certain service only propagates to legitimate nodes that are allowed to access this service.

Proof Suppose a service provider T generates a routing advertisement $R = \{d : s/m, S, M\}$ to announce its service s with source prefix constraint S. Without loss of generality, our discussion focuses on scenarios with the presence of a valid source prefix constraint (i.e. $S \neq \phi$). Let p_u represent the address of a node u. When R is propagated into the network before it reaches any aggregation point, to demonstrate the correctness of this algorithm, we have two separate scenarios:

1) $p_u \in S$ If the current node is allowed to access service s provided by T, then u should be able to see the route advertisement R originated from T. According to line 6-8 in our route propagation algorithm ROUTEPROP(u, N_b), if u is a direct neighbor of T, it should receive R from T. If u is multiple hops away from T, it can receive R from its neighbors; otherwise, it implies that none of its neighbors are allowed to access service s. Since transit nodes are not permitted to carry such traffic in our scheme, u cannot establish a valid path to T by any means.
2) $p_u \notin S$ If the current node is prohibited from accessing service s provided by T, then u should not see route advertisement R originated from T. Lines 9-10 in our route propagation algorithm ROUTEPROP(u, N_b) guarantee that none of its neighbors will forward the routing advertisement to u since $p_u \notin S$.

Therefore, the route propagation algorithm guarantees that illegitimate nodes won't see routing advertisements originated from the service provider.

THEOREM 2 ***Correctness of the Packet Forwarding Algorithm***
The packet forwarding algorithm ensures that permitted packets will arrive at the destination following the selected path; whereas non-permitted packets will be dropped early by intermediate routers along the path acting as firewalls.

Proof The correctness of packet forwarding algorithm relies on the assumption that the underlying routing protocol will generate a valid routing path between each pair of source and destination nodes. Suppose a packet K is originated from a source node with address p_s to

access service s provided by destination node T. With the new ROFL scheme, each router along the selected path acts as a firewall. To demonstrate the correctness of our packet forwarding algorithm, we have the following two separate cases:

1) $p_s \in S$ Packet K is coming from a legitimate source allowed to access service s. Each router u along the path performs longest prefix matching by consulting its local routing table. Once a matching route R is found, u makes a forwarding decision through a second level of checking against source prefix constraint S embedded in route R. Since $p_s \in S$, u decides to forward this packet according to line 6-10 of the packet forwarding algorithm PKTFORWARD(u). The same process repeats until K reaches its final destination.
2) $p_s \notin S$ Packet K is coming from an illegitimate source prohibited from accessing service s. Each router u along the path performs longest prefix matching by consulting its local routing table. If a matching route R is found, u makes a forwarding decision through a second level of checking against source prefix constraint S embedded in route R. Since $p_s \notin S$, the packet is dropped immediately according to line 11-12 in PKTFORWARD(u), that performs packet filtering; Otherwise, K is still dropped according to the underlying routing protocol.

Therefore, the packet forwarding algorithm guarantees that permitted packets will arrive at destination following the path generated by underlying routing protocol; non-permitted packets will be dropped by intermediate routers along the path that also perform packet filtering functions based on destination address and port number as well as source address.

VI. Aggregation Points

Although ROFL as defined will work without further enhancement, installing it as part of a larger network could be seen as unfriendly: blasting that many extra routes into, say, the Internet is considered improper. Indeed, the entire rationale for CIDR [6] is that distant networks need only see a single short prefix that covers many networks; ROFL should not frustrate that scheme. We propose two basic approaches to resolving this: external and internal aggregation.

External aggregation is the approach taken by today's ISPs to deal with overly-long prefixes announced via BGP. In one variant, known as proxy aggregation, an ISP will generate a single short prefix that covers multiple longer prefixes, and re-announce the shorter one rather than the long ones. In practice, this is very rarely done. The second variant, occasionally known as "satanic philtres" [12], is unilateral: an ISP will drop announcements for too-long prefixes. It assumes that the covered networks will be reachable via some shorter prefix; if they are not, their administrators should arrange for proper CIDR-based addresses.

This latter issue (as well as politeness) impels us to incorporate an aggregation scheme into ROFL. (Other justifications for ROFL aggregation are given in [16].) We define an *aggregation point* as a node that receives full ROFL announcements but generates fewer re-announcements of shorter prefixes. Typically, this is done at the boundaries of a MANET, though it can be done at other points, both within the MANET and outside it. There are three basic issues: how aggregation points are defined, what prefix should be announced, and what metric should be used.

For the first, we use notation similar to source prefix constraints: we define the ROFL region as a set of prefixes. Any neighbor whose address is not in this set receives only the shorter, covering announcement. Note, though, that the aggregation policy prefix is not the same as the firewall source prefixes; the two are used independently. That is, packets from the wide-area net are dropped if they do not meet any source address constraints for firewall rules; furthermore, firewall rules will frequently exist within the ROFL region.

The aggregation policy is generally a matter of static configuration, and hence is not passed along via a routing protocol. However, since in a MANET any node can be a boundary node, all nodes must have this configuration information. Exactly how this is provided is beyond the scope of this work; we assume that it will be done as part of general node provisioning.

We treat the actual prefix to be re-announced similarly: it is a static policy decision, installed on all nodes.

Metrics are a more interesting problem, since we may wish to preserve some notion of cost even in the wide-area network. Consider: a MANET may consist of several subnets, with multiple attachment points to the wired network. If a given subnet is much closer to one of them, this information should be preserved.

It is not obvious how to do this. Even if everyone is using the same values for $\mathcal{C}_i$, $\mathcal{R}_i$, and γ, aggregation points will see many values. Which should be used? The minimum? The maximum? The mean? The median? We suspect that median is correct, but that mean is easier to calculate.

One more optimization can be introduced. As explained in [15], local optimizations can be introduced. The routing algebra is used to merge redundant announcements, though in this scheme S must match for two routing advertisements to be merged. As above, treatment of the metric can be a complex process.

VII. Modeling ROFL Using Routing Algebra

We now discuss some practical issues when modeling the new ROFL scheme with source prefix filtering using the routing algebra [8], [13].

A. Metarouting

Routing algebra is motivated by the recognition that a path may not be only associated by a metric cost, but also can be abstractly associated by a signature. And the operation that translates a signature on an in-coming link to an out-going link can be heterogeneous on different nodes of the network.

In the following, we briefly describe routing algebra and discuss the relevance to ROFL. A routing algebra is a tuple:

$$A = (\Sigma, \preceq, L, \oplus, O)$$

where Σ is a set of signatures for describing paths, and $\preceq$ is a preference relation over signatures, L is a set of labels assigned by routers on the ongoing links, and $\oplus : L \times \Sigma \rightarrow \Sigma$ is an operator that produces a signature when a path with a certain signature is extended by a link with a label.

A sufficient condition in routing algebra that guarantees the existence of network-wide consistent condition is monotonicity [13]:

$$\lambda \otimes \sigma \succeq \sigma$$

for any $\sigma \in \Sigma$ and $\lambda \in L$. The basic idea of monotonicity is to make all extended paths carrying a less preferable signature, and hence, creates loop-free preference. Monotonicity is shown to imply a well-known condition for the convergence in BGP with customer-provider-peer relations [7].

Monotonicity also applies to distributed or hybrid firewalling, such that the existence of robust firewall configuration can be attained in a machine-checkable manner. This lends the usefulness of routing algebra to the context of security systems.

The more general setting of routing algebra allows us to model sophisticated operations of ROFL, which otherwise can not be regarded as shortest-path routing in the Internet. We aim to prove that a sufficient condition that is similar to monotonicity in policy-based routing will be also useful to establish robust deployment of firewall by ROFL. We will use our policy algebra — a formal model of outsourced firewall policies — to map ROFL into the routing algebra. Our goal is to show how the deployment of firewalls can be checked in an automatic manner to guarantee that ROFL can converge to a network-wide firewall enforcement in a deterministic manner, even in the presence of asynchronous communications among routers.

B. Policy Regions

A crucial question in any firewall is defining the policy region associated with a particular service. As discussed previously, our approach is to embed source prefix constraints in the routing announcement originating from the service provider. Therefore, we use one basic routing algebra TAGS(String) to define the set of routers allowed to access the protected service. This information is embedded in the initial routing announcement originated from the service provider. We use another basic routing algebra component PROGS(A) to extend labels to programmatic labels. We define that

$$\lambda \oplus \sigma = \begin{cases} \lambda_1 \oplus \sigma = \sigma' & \text{if } u \in \text{TAGS(String)}; \\ \infty \oplus \sigma = \phi & \text{otherwise.} \end{cases}$$

Since $\lambda \oplus \phi = \phi$ for all $\lambda \in L$, the routing announcement for that protected service will never propagate out of the defined policy region.

C. Cost and Risk Metrics

In ROFL, each routing announcement is associated with a cost metric. Cost metric in the routing problem is different from the one in the optimization problem. In that problem, we are trying to find optimal policy enforcement points in order to minimize the total cost. In the routing problem, cost is incurred at each node along the routing path. Cost determines route preference. It is modeled as a label of a specific link.

To construct a routing protocol that satisfies M/SM, we do not need a cost metric what increases monotonically as the routing announcement propagates. What we need is only

$$\begin{aligned} \text{M:} \quad & \sigma \preceq \lambda \oplus \sigma \\ \text{SM:} \quad & \sigma \prec \lambda \oplus \sigma \end{aligned}$$

This is true as long as the $\oplus$ operation of constructing cost signature can be modeled as $ADD(1, n)$ or something similar.

There are a number of approaches we can take. The metric defined in Section III clearly meets these constraints, as long as either every link has a positive cost or (for $\gamma = 0$) every node has a non-zero risk.

Other possibilities were discussed in [15]. The cost metric in ROFL, for example, may reflect the size of the routing table and the battery power of each node (which is the major concern in MANETs). Thus we have cost metric $C \propto \frac{\#routingtable\_entries}{battery\_power}$.

We can define the risk metric simply as the number of hops from the enforcement point (the node which implements the firewall rule as routing entry) to the protected node (the node which provides that service). Therefore, we can define the algebra associated with risk metric as $Risk = SEQ(0, n) = \langle \Sigma_R, \preceq_R, L_R, \otimes_R, \mathcal{O}_R \rangle$, where the preference relation is defined as $\sigma_1 \preceq \sigma_2 = |\sigma_1| \leq |\sigma_2|$, the $\oplus$ operation is defined as $i \oplus \sigma = i :: \sigma$.

Now we have two sub-algebras that capture the cost and risk metric associated with path signatures. Since now we have multiple route metrics when dealing with route selection, we can apply the lexicographic comparison in [9]: the most important attribute of each route is considered first, and if this does not give enough information to decide which route is better, the next attribute is considered, etc.

D. Route Aggregation

To reduce the growth of routing table size, we introduce the notion of aggregation points in ROFL. With routing algebra, we can model this using scoped product $A \odot B$, where algebra A is used between administrative entities, and algebra B is used inside for each administrative entities. Thus the path signature for algebra B is updated each time within administrative entities; once across the boundary, the path signature for algebra A is updated with path signature for algebra B re-initialized.

VIII. Conclusions

Apart from adding source prefix filtering to ROFL, we have accomplished three other things: we have shown how to incorporate flexible tradeoffs between cost and risk; we have devised a new approach to policy routing; and we have shown how our scheme can be modeled by metarouting.

It is well-known that trying to optimize according to two different criteria is difficult, since maximizing one criterion will sometimes cause an opposite effect on another. Our equation permits each network or even each node to balance two different criteria, according to local needs.

This flexibility comes at a price, though: administrative complexity. Not only must firewall rules be configured for each node and service, itself an error-prone procedure [14], the appropriate values for cost, risk, and the tradeoff (γ) must be set as well. This complexity can be quite serious, especially in tactical MANETs, where some changes will need to be made while under enemy fire. The human interface issue will be explored in a forthcoming work. That said, ROFL offers a major complexity advantage over conventional firewalls: rules are set locally and do not interact in the way that conventional firewall rules can [3], [5]. This permits easy introduction of new nodes and services, with much less chance of a configuration error opening other services to attack.

We would also like to extend our work to handle inter-domain MANET routing [4]. At a minimum, cost, risk, and γ will need to be per-domain values. That, however, may result in routing oscillations. We conjecture that we can use the routing algebra to establish appropriate relationships between the cost and risk values for different domains, and thus avoid the problem.

Adding source prefix constraints to a routing protocol has always been problematic: it is often unclear how to balance a packet that matches in one field (i.e., destination address) but does not match in the other: which should dominate? By marrying routing to packet filtering, we resolve that issue: a routing advertisement for a given destination is forwarded to a neighbor if and only if that neighbor is allowed to send packets to that destination. There is thus no ambiguity; a forwarding node's routing tables can never contain such a conflict for any packet it can legitimately pass along. We suggest that this approach to policy routing may be useful in other contexts.

Acknowledgements

We would like to thank Tim Griffin for his advice on policy routing and the routing algebra.

References

[1] S. M. Bellovin, "Distributed firewalls," *;login:*, pp. 39–47, November 1999.

[2] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of ACM*, vol. 13, no. 7, pp. 422–426, July 1970.

[3] D. B. Chapman, "Network (in)security through IP packet filtering," in *Proceedings of the Third Usenix Unix Security Symposium*, Baltimore, MD, September 1992, pp. 63–76. [Online]. Available: http://www.greatcircle.com/pkt_filtering.html

[4] C.-K. Chau, J. Crowcroft, K.-W. Lee, and S. H. Wong, "Inter-domain routing for mobile ad hoc networks," in *MobiArch '08: Proceedings of the 3rd International Workshop on Mobility in the Evolving Internet Architecture*. ACM, 2008, pp. 61–66.

[5] W. R. Cheswick, S. M. Bellovin, and A. D. Rubin, *Firewalls and Internet Security; Repelling the Wily Hacker*, 2nd ed. Reading, MA: Addison-Wesley, 2003. [Online]. Available: http://www.wilyhacker.com/

[6] V. Fuller and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan," Internet Engineering Task Force, RFC 4632, Aug. 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4632.txt

[7] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Trans. Networking*, vol. 10, no. 2, pp. 232–243, April 2002.

[8] T. G. Griffin and J. L. Sobrinho, "Metarouting," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 1–12, September 2005.

[9] A. Gurney and T. G. Griffin, "Lexicographic products in metarouting," in *Proc. ICNP 2007*, October 2007.

[10] J. Moy, "OSPF Version 2," Internet Engineering Task Force, RFC 2328, Apr. 1998. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2328.txt

[11] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," Internet Engineering Task Force, RFC 4271, Jan. 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4271.txt

[12] J. Rexford, S. M. Bellovin, and R. Bush, "Some initial measurements of prefix length philtres," NANOG talk, May 2001, http://ran.psg.com/~randy/010521.nanog/index.htm.

[13] J. Sobrinho, "An algebraic theory of dynamic network routing," *IEEE/ACM Trans. Networking*, vol. 13, no. 5, pp. 1160–1173, October 2005.

[14] A. Wool, "A quantitative study of firewall configuration errors," *IEEE Computer*, vol. 37, no. 6, pp. 62–67, 2004.

[15] H. Zhao and S. M. Bellovin, "Policy algebras for hybrid firewalls," Department of Computer Science, Columbia University, Tech. Rep. CUCS-017-07, March 2007, also presented at the Annual Conference of the ITA, 2007. [Online]. Available: http://mice.cs.columbia.edu/getTechreport.php?techreportID=453

[16] H. Zhao, C.-K. Chau, and S. M. Bellovin, "ROFL: Routing as the firewall layer," in *Proc. New Security Paradigms Workshop (NSPW '08)*, September 2008.