

A NETWORK MANAGEMENT TOOL FOR INTER-DOMAIN POLICY ROUTING

Kraig R. Meyer¹

Dr. Deborah Estrin²

The Aerospace Corporation
PO Box 92957, MS M1-055
Los Angeles, CA 90009-2957
Phone +1 213-336-8114
Fax +1 213-336-5833
Electronic mail: kmeyer@aero.org

Computer Science Department MC0782
University of Southern California
Los Angeles, CA 90089-0782
Phone +1 213-740-4524
Fax +1 213-740-7285
Electronic mail: estrin@usc.edu

Abstract

In recent years, the number of organizations connected to the Internet has grown, resulting in an increasingly diverse set of connected users. As a result, network managers have growing concerns about the security and control of network resources. This concern has led protocol engineers to design routing protocols, such as Inter-Domain Policy Routing (IDPR), that include policy restrictions.

The introduction of these protocols present a number of new problems to network administrators as they may not be able to easily understand how the policies of different networks interact. A network manager would like to create policies that prevent unauthorized traffic, without impeding legitimate traffic flows. In the presence of charging mechanisms, a manager may wish to use policy terms to assist in minimizing costs by avoiding the use of commercial networks whenever possible. In this paper we describe a route synthesis simulation tool (RSST) which we have implemented to examine some of these network management issues. We show how it is used to explore some hypothetical situations that a network administrator would face in managing policy routing.

¹ Mr. Meyer's work was supported by a CERFnet Graduate Research Fellowship and by the National Science Foundation.

² Dr. Estrin's work was supported by the National Science Foundation.

5.3.1

Introduction and Motivation

- Internet is larger, more heterogeneous
- Routing is more difficult
- Potential misuse/abuse from wider variety of sources
- Policy Routing and the need for a routing management tool
- Managing Policy Routes in the context of Commercial Network charging

Introduction and Motivation

The topology of the Internet has grown in complexity in recent years, causing the problem of efficiently routing packets to their destinations to become increasingly difficult. At the same time, the variety of users and systems has increased. This has caused growing concern over the control of network resources—routers and links—and has resulted in a desire to introduce restrictions, or policies, into routers and/or routing protocols [BRA89], [ESTR89]. Protocols such as IDPR and BGP have been prototyped and/or implemented to address some of these concerns over policy and scale. As the Internet continues to grow larger, addressing policy and topology issues will become more problematic for network administrators.

Most models of policy routing conceptually divide the Internet into Autonomous Domains, or ADs, each of which is a group of commonly administered network resources. The restrictions on these resources are expressed as policies, which allow or disallow traffic based on source, destination, and other characteristics. As time progresses, most private ADs are likely to express more restrictive policies to limit the use of their resources. In some cases, this will cause policy interactions which prevent traffic flow between certain sources and destinations (either unintentionally or as an implicit goal of the policy). Network administrators will need to determine why certain destinations are unreachable. More generally, administrators need assistance selecting policy terms that are neither too restrictive nor too lax. Clearly a management tool is needed to help administrators resolve policy routing problems.

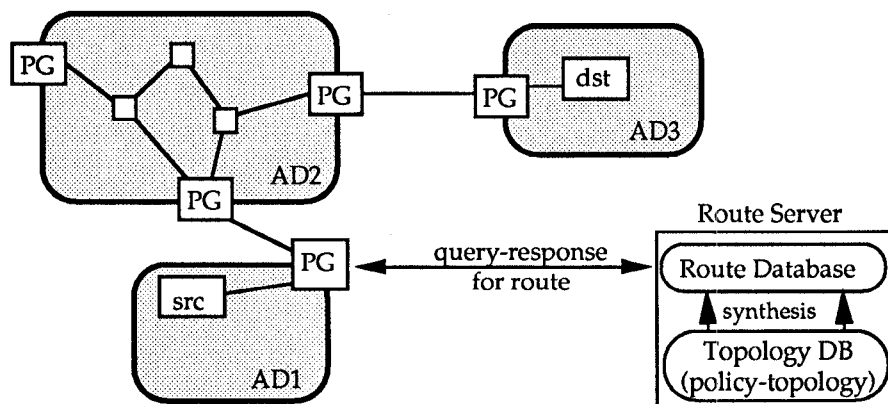
Commercial carriers, who are likely to provide a substantial portion of the Internet infrastructure in the future, may choose to charge on a packet-by-packet basis. From the commercial provider's point of view, this may simplify policy, since it doesn't matter how the network is used as long as the use is paid for. However, for a local network administrator, the presence of charging increases the need for other types of policy: whose traffic, and what kind of traffic, is permitted to traverse commercial networks when that use must be paid for? A management tool can help a network administrator create policies that minimize costs while maintaining reachability of all networks.

5.3.2

IDPR: Inter-domain Policy Routing

- Millions of networks, hundreds of thousands of ADs
- Policy distributed as part of link state flooding
- Source routing at AD level is used

(Sample route from src to dst is AD1, AD2, AD3)



IDPR: Inter-domain Policy Routing

Inter-domain Policy Routing (IDPR) protocols have been designed and prototyped by the IDPR Working Group of the Internet Engineering Task force [STEE91]. A primary goal of IDPR is to develop a routing protocol that is able to route packets over a world-wide internet of millions of networks spanning hundreds of thousands of ADs [BRES90]. We developed and implemented a graphical Route Synthesis Simulation Tool (RSST) to evaluate the interaction of policy and charging in the context of the IDPR protocols.

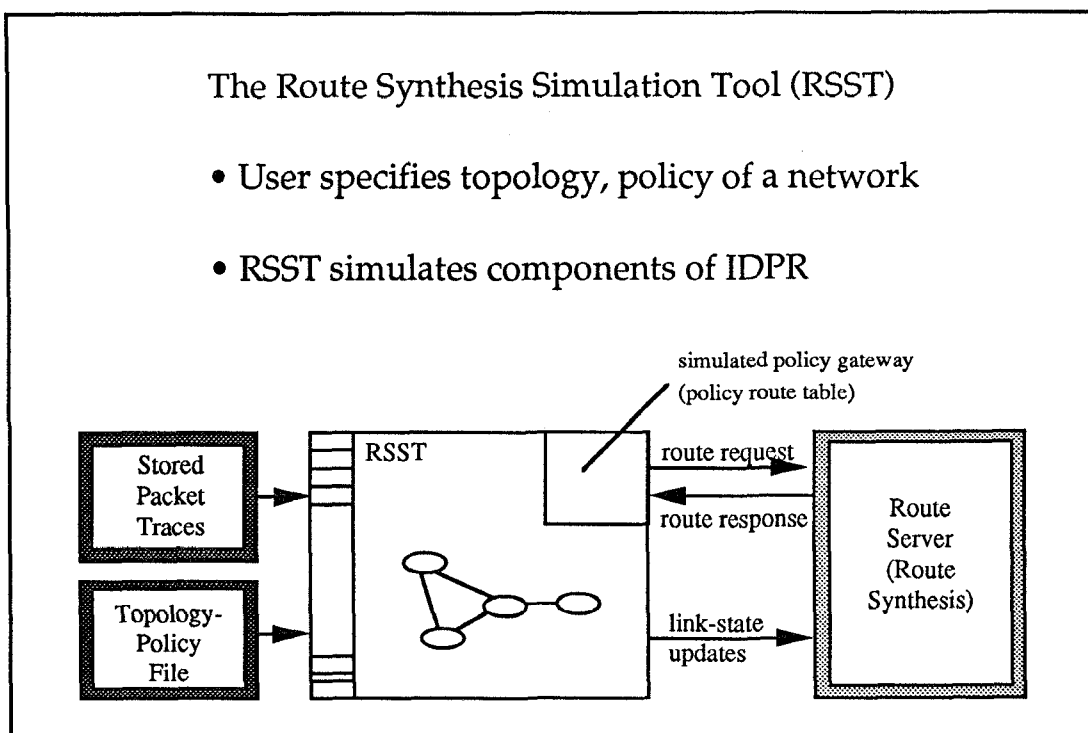
In IDPR, policy information is flooded with topology information in link-state style updates. From this, routes between a given source and destination can be synthesized given a specific set of criteria—typically type of service (path characteristics such as bandwidth and acceptable latency) and user class identifier (information specific to a given user, such as NSF or DOD researcher). IDPR routes are only concerned with which ADs a packet traverses, rather than the individual intra-domain gateways that are used to reach the destination.

In the diagram above, the ADs are drawn as shaded ovals, and the policy gateways (labeled PG) are shown at the borders of these ADs. A route from source, src, to destination, dst, traverses three ADs and is specified (AD1, AD2, AD3). Note that within AD2 several gateways and links are traversed but these are not specified in the path. Routing in IDPR is source based, and the IDPR route is placed in the packet header when the first packet from src to dst leaves the source AD. Route synthesis is the responsibility of a policy route server.

The function of the policy gateway is twofold. First, it is responsible for routing packets to the next AD on their path—similar to any router in an IP network. However, when a policy gateway is the first (or local) policy gateway on a packet's route, it has the further responsibility of determining whether an appropriate path exists in the route table of the policy gateway. If necessary, it obtains a route from the route server, and sets up a route [ESTR91].

The Route Synthesis Simulation Tool (RSST)

- User specifies topology, policy of a network
- RSST simulates components of IDPR



The Route Synthesis Simulation Tool (RSST)

As mentioned previously, the purpose of the Route Synthesis Simulation Tool (RSST) is to assist in the evaluation of policy and charging issues in the context of the IDPR protocols. The RSST user begins by specifying the topology of a network that he or she is interested in studying, and the policies that are to govern the network. RSST then draws a map of the network in a window, and the user can run various types of simulations to see which destinations are reachable, via which route, from a given source. The user can ask "What if?" questions by changing the policy and topology in the simulation tool without actually changing the hardware or software configuration of a production network. Examples of this process will be presented in the slides that follow.

To support these activities, RSST simulates the activities of the primary components in the IDPR architecture. Link-state style updates are created from the user-specified topology-policy file, and are sent to a simulated route server. During simulation, routing demands are read from a stored trace file. These routing demands may be met either by a route stored in the routing table of the simulated policy gateway, or the route server may have to be called upon to synthesize an appropriate route.

Running a Simulation with RSST

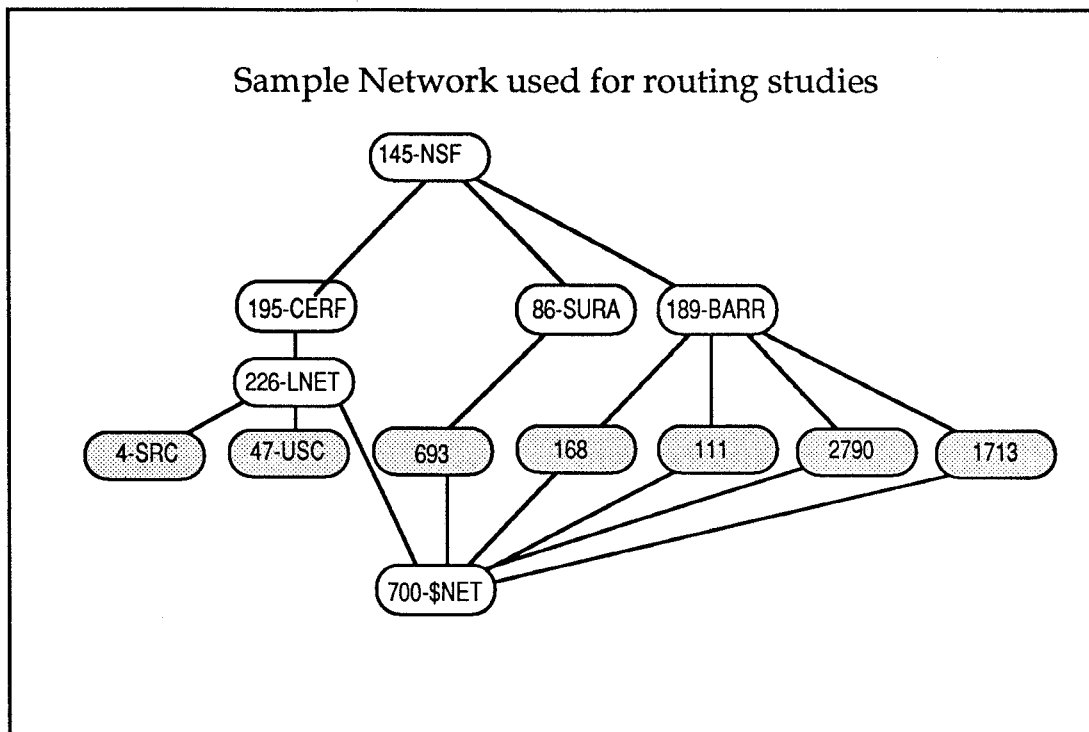
- Routing demands created from traffic trace or other list of destinations
- Policy Gateway requests routes from Route Server and builds routing table
- RSST keeps track of unreachable destinations, simulation statistics
- Routing table and simulation statistics are post-processed and displayed

Running a Simulation with RSST

During a simulation, the simulated policy gateway attempts to satisfy routing demands from a specific source AD to a list of destinations supplied by the user. These routing demands may be based on the destination fields of a previously recorded packet trace, or it may be a list of individual destinations selected by the user.

To meet these simulated routing demands, the gateway maintains a policy route table of all computed routes that have been used during some recent window of time. Routes that have been unused for longer than some cache expiration time are invalidated. If an appropriate route can be found in the gateway's policy route table that meets the current destination demand, no further action is necessary. If the policy gateway does not have a route, it generates a simulated route request and sends it to the simulated route server. If the route server is able to calculate a route, the route is returned and added to the policy gateway's routing table. In some cases, there may be no valid route to a given destination because of policy and topology restrictions.

In addition to storing policy routes, RSST also stores the number of times each route is used, as well as a list of all unreachable destinations and the number of packets intended for those destinations. When the simulation is complete, this information is displayed along with the cost of providing each route, based on the ADs traversed in the route and the number of packets that used the path. If a traffic trace is used, information about the trace is also provided, such as the number of packets processed and the trace duration.



Sample Network used for routing studies

To illustrate how this tool can be used by a network administrator to solve policy routing problems, we ran simulations using the network above. We created this network by starting with a subset of the present-day Internet and then adding to it in a way that we believe to be an accurate representation of some future growth. At the top is the NSFNET backbone, which we call AD 145. It is directly attached to ADs 195, 86, and 189, which are NSF regional networks. The row of shaded ovals represents stub university and corporate networks, which are the sources and destinations of all the traffic in our simulations. In addition, we have added a few connections to a commercial network, which we have labeled AD 700 in our picture and called \$NET (DOLLAR-NET).

The next four slides will show how a network administrator uses RSST to deal with several hypothetical situations which arise as part of managing policy routing.

In the first situation we examine, no restrictive policies have been introduced into the network. Every autonomous domain will carry traffic from any other autonomous domain, and all traffic takes the shortest path between the source and destination.

Given this scenario, the administrator at AD 4 has determined that s/he must reduce the amount of money paid to AD 700, which is a commercial network that charges on a per-packet basis. To determine why the bill is so high and evaluate methods of cutting costs, the administrator records a 30 minute traffic trace. He uses this traffic trace to drive a simulation in hopes of seeing what traffic is going through AD700.

To simulate this process, we took a 30 minute traffic trace and mapped the local network number to AD 4. All other network numbers in the trace are mapped to one of the stub AD numbers, or are ignored (the trace had too many destination networks to use in a simple example). Thus, all traffic in the simulation is between AD 4 and one of the other stub (shaded) ADs. We are only interested in traffic that originates at AD 4 since traffic destined for AD 4 does not represent a routing demand on the local policy gateway.

5.3.6

Evaluating Charging Mechanisms

- How can policies be introduced to save money?

Simulation Statistics:

Time: 30 minutes
Packets: 1008 total
47 outbound (processed)
Policy Routes: 6 synthesized from source AD 4 (0 not installed):

DestAD	hops/\$	usage/\$	tos	uci	Computed Route -- AD:VG:PRid (charging policy)			
693	4/3	7/21	0	0	4:0:1(0)	226:0:1(0)	700:0:1(3)	693:0:1(0)
168	4/3	7/21	0	0	4:0:1(0)	226:0:1(0)	700:0:1(3)	168:0:1(0)
111	4/3	1/3	0	0	4:0:1(0)	226:0:1(0)	700:0:1(3)	111:0:1(0)
47	3/0	24/0	0	0	4:0:1(0)	226:0:1(0)	47:0:1(0)	
1713	4/3	1/3	0	0	4:0:1(0)	226:0:1(0)	700:0:1(3)	1713:0:1(0)
2790	4/3	7/21	0	0	4:0:1(0)	226:0:1(0)	700:0:1(3)	2790:0:1(0)

Total cost (this simulation) = 69
Average route length (hops) = 3.833333
All requested ADs were reachable -- 0 dropped packets.
TOTAL 0 packets dropped

Evaluating Charging Mechanisms

When we run this simulation, we find that AD 4 communicated with six different ADs during this 30 minute window. For each destination AD there is an entry in the routing table that shows the following information: destination AD number, number of hops in the path and the total cost of the path; the number of packets that used the path and the total cost of providing the path; the type of service (TOS) and user class identifier (UCI) of the path, and the actual AD level path.

For example, in the first entry we see that AD 693 has a 4 hop path that costs 3 units to traverse; the path was used by 7 packets and cost a total of 21 monetary units to provide communication during this 30 minute period. The TOS and UCI are both zero, and the actual ADs traversed in the path (4, 226, 700, and 693) are listed.

By examining the routing table, we see that 5 out of the 6 paths listed in the routing table traverse AD 700, the commercial AD that charges on a per-packet basis. These paths incurred an expense of 69 units during this 30 minute time period. If we examine the topology illustrated on the previous slide, we see that all of these destinations are also reachable via the NSFNET backbone (which does not charge), but this path is not presently used because it is longer than the path through the commercial AD. The administrator at AD4 notes that by writing a policy that prohibits the use of AD 700, s/he can completely eliminate network charging expenses without affecting AD reachability at all.

Eliminated Charges, Longer Paths

Simulation Statistics:

Time: 30 minutes
 Packets: 1008 total
 47 outbound(processed)
 Policy Routes: 6 synthesized from source AD 4 (0 not installed):

DestAD	hops/\$	usage/\$	tos	ucl	Computed Route	-- AD:VG:PRId (charging policy)				
693	6/0	7/0	0	0	4:0:1(0)	226:0:1(0)	195:0:1(0)	145:0:1(0)	86:0:1(0)	693:0:1(0)
168	6/0	7/0	0	0	4:0:1(0)	226:0:1(0)	195:0:1(0)	145:0:1(0)	189:0:1(0)	168:0:1(0)
111	6/0	1/0	0	0	4:0:1(0)	226:0:1(0)	195:0:1(0)	145:0:1(0)	189:0:1(0)	111:0:1(0)
47	3/0	24/0	0	0	4:0:1(0)	226:0:1(0)	47:0:1(0)			
1713	6/0	1/0	0	0	4:0:1(0)	226:0:1(0)	195:0:1(0)	145:0:1(0)	189:0:1(0)	1713:0:1(0)
2790	6/0	7/0	0	0	4:0:1(0)	226:0:1(0)	195:0:1(0)	145:0:1(0)	189:0:1(0)	2790:0:1(0)

Total cost (this simulation) - 0
 Average route length (hops) - 5.500000
 All requested ADs were reachable -- 0 dropped packets.
 TOTAL 0 packets dropped.

- Routes between AD 4 and AD 693:
 old 4 hop route: AD4, AD226, AD700, AD693
 new 6 hop route: AD4, AD226, AD195, AD145, AD86, AD693

Eliminated Charges, Longer Paths

The new policy prohibits any AD4 traffic from using AD 700 as a transit. The new simulation shows that all destinations are still reachable, but the average route length in hops has increased from 3.8 up to 5.5. If we compare the old route from AD 4 to AD 693 to the new route, we can see an example of how the route length has increased. (Refer to the slide with the map). The old route took 4 hops and traversed AD4, AD226, AD700, and AD693. The new route takes 6 hops and traverses AD4, AD226, AD195, AD145, AD86, and AD693.

We see that there is no cost to support this combination of traffic and policy. This is because AD 700 is not used at all.

For our next example, suppose AD 2790 has been informed that it should not use the NSFNET backbone any more because it is a commercial organization. The NSFNET backbone could implement a policy that prevents this traffic from flowing, or AD 2790 could be a good citizen and write a local policy that prevents the use of this AD in any of its routes. Regardless of which policy is introduced, we find that routing between AD 2790 and AD 4 breaks. Even though there are two physical paths between the two ADs, neither one of them can be used because of the interaction of policy terms.

Resolving Routing Problems between AD 2790 and AD 4

```

-----
Policy Routes: 5 synthesized from source AD 4 (0 not installed):
-----
DestAD  hops/$  usage/$  tos  uci  Computed Route  -- AD:VG:PRid (charging policy)
-----
693      6/0      7/0      0    0    4:0:1(0)  226:0:1(0)  195:0:1(0)  145:0:1(0)  86:0:1(0)  693:0:1(0)
168      6/0      7/0      0    0    4:0:1(0)  226:0:1(0)  195:0:1(0)  145:0:1(0)  189:0:1(0)  168:0:1(0)
111      6/0      1/0      0    0    4:0:1(0)  226:0:1(0)  195:0:1(0)  145:0:1(0)  189:0:1(0)  111:0:1(0)
47       3/0      24/0     0    0    4:0:1(0)  226:0:1(0)  47:0:1(0)
1713     6/0      1/0      0    0    4:0:1(0)  226:0:1(0)  195:0:1(0)  145:0:1(0)  189:0:1(0)  1713:0:1(0)
-----
Total cost (this simulation) = 0
Average route length (hops) = 5.400000

Unreachable  Packets Dropped
-----
AD 2790      7 packets
TOTAL        7 packets dropped.
-----

```

After policy fix:

```

-----
Policy Routes: 6 synthesized from source AD 4 (0 not installed):
-----
DestAD  hops/$  usage/$  tos  uci  Computed Route  -- AD:VG:PRid (charging policy)
-----
693      6/0      7/0      0    0    4:0:1(0)  226:0:1(0)  195:0:1(0)  145:0:1(0)  86:0:1(0)  693:0:1(0)
168      6/0      7/0      0    0    4:0:1(0)  226:0:1(0)  195:0:1(0)  145:0:1(0)  189:0:1(0)  168:0:1(0)
111      6/0      1/0      0    0    4:0:1(0)  226:0:1(0)  195:0:1(0)  145:0:1(0)  189:0:1(0)  111:0:1(0)
47       3/0      24/0     0    0    4:0:1(0)  226:0:1(0)  47:0:1(0)
1713     6/0      1/0      0    0    4:0:1(0)  226:0:1(0)  195:0:1(0)  145:0:1(0)  189:0:1(0)  1713:0:1(0)
2790     4/3      7/21     0    0    4:0:1(0)  226:0:1(0)  700:0:1(3)  2790:0:1(0)
-----
Total cost (this simulation) = 21
Average route length (hops) = 5.166667
All requested ADs were reachable -- 0 dropped packets.
TOTAL        0 packets dropped.
-----

```

Resolving Routing Problems between AD 2790 and AD 4

When routing to AD 2790 breaks, the administrator at AD 4 runs several simulations to determine the cause. The first simulation shows the situation that the administrator is already aware of: namely, that routing to AD 2790 is broken. It also shows paths to ADs adjacent to AD 2790, which are functioning as before. If the administrator relaxes the local policy terms and re-runs the first simulation again, s/he will realize that AD 2790 is reachable via AD 700 but not via AD 145 (This simulation is not illustrated in this slide).

Since no other destinations are affected (that is, all other destinations are still reachable via AD 145), there is no reason to completely remove the restriction on traffic via AD 700. The desirable policy is to allow traffic between AD 2790 and AD 4 to use AD 700, but to prohibit the use of AD 700 for all other traffic.

After the appropriate changes have been made to the policy, a final simulation is run. We see that this fix results in the desired behavior, namely that traffic to AD 2790 is permitted to travel via AD 700 (DOLLAR NET), but traffic to all other ADs still travels via AD 145 (NSFNET backbone). Thus, AD 4 maintains reachability to all destinations while minimizing its use of the commercial network. Also, AD 2790 complies with the NSFNET guidelines and eliminates its use of the NSFNET backbone for commercial purposes.

The penalty for meeting these policy demands is slightly longer routes—the average route length has increased from 3.8 hops in the first simulation to 5.17 hops in this final simulation.

Conclusions from this study

- Policy is difficult to construct
- Policies will interact in nonobvious ways
- RSST or similar tool useful in future planning
- RSST or similar tool useful in problem solving

Conclusions from this study

This study demonstrated several major points. The first is that large internets comprised of many different organizations will have heterogeneous policies that reflect different organizational requirements and philosophies. The next point is that policy is difficult to construct in even moderately large internets. For example, when we were trying to identify the policy terms that would fix the connectivity problem between the two ADs, it took four attempts to get the set of policies to work in the desired manner. Doing this with RSST took about 5 minutes. However, doing this in a production network would have required restarting a policy gateway and/or a route server several times. Working this same problem out on paper would likely be even more challenging, as each change in policy requires performing a shortest path algorithm over the entire graph—and each node in the graph has multiple policies that affect the algorithm in different ways.

The connectivity problem examined in the previous slides illustrates that any time even moderately restrictive policies are used, there can be nonobvious interactions of policies that affect connectivity. A simulation or management tool aids in solving these types of problems rapidly.

RSST allows future policy changes to be evaluated before they go into effect. For example, if the NSFNET backbone policy was going to change in a week, that policy could be entered into the simulation tool and potential conflicts could be identified and resolved in advance. As in one of our examples, that may require changing local policy. In the case that changes are not announced in advance, the tool also assists in the problem solving process.

Finally, a simulation tool has a distinct advantage over a tool that manages a production network. In a production network, an administrator can only change the policies in the parts of the network that s/he controls. In contrast, any AD's policy can be changed in a simulation. Once an administrator has shown in simulation that a certain combination of policies works correctly, s/he may be able to convince a colleague in another AD to change those policies.

5.3.10

References and Acknowledgements

References

[BRA89] Hans-Werner Braun, "Models of Policy Based Routing," RFC 1104, Merit/NSFNET Project, Merit Computer Network, June 1989.

[BRES90] Lee Breslau and Deborah Estrin, "Design of Inter-Administrative Domain Routing Protocols," Proceedings of the 1990 ACM Sigcomm.

[CáCE91] R. Cáceres, P. Danzig, S. Jamin, D. Mitzel, "Characteristics of Individual Application Conversations in TCP/IP Wide-Area Internetworks, Proceedings of the 1991 ACM Sigcomm.

[ESTR89] D. Estrin, "Policy Requirements for Inter-Administrative Domain Routing," RFC 1125, University of Southern California, November 1989.

[ESTR91] D. Estrin and M. Steenstrup, "Inter-Domain Policy Routing: Overview of Architecture and Protocols," ACM Computer Communications Review, 1991.

[STEE91] "Inter-Domain Policy Routing Protocol Specification: Version 1," Internet Draft edited by Martha Steenstrup, BBN Communications, February 1991.

Acknowledgements

The simulated policy gateway code used in RSST was written largely by Gene Tsudik, and the simulated route synthesis code by Lee Breslau. Danny Mitzel, Sugih Jamin, and Peter Danzig provided various traffic traces that were used in our simulations. The authors would also like to thank Steve Hotz and other members of the IDPR working group for comments regarding useful features in RSST, and to thank Jeff Thomas, Mark Gabriele, and Dennis Persinger for assisting in the document review process.