

[Return to Table of Contents](#)

## Chapter 2 - Policy Routing Theory

- 2.1 What do you mean by "Policy"
- 2.2 Common IPv4 Routing Problems
- 2.3 Policy Routing Structure
- 2.4 Summary

Traditional IPv4 routing is summarized as "All routing is a destination-driven process." When a router looks at an IPv4 packet it cares only about the destination address in the header of the packet. It uses this destination address to make a decision on where to forward the packet. This scenario works fine for simple networks where all of the machines in the network only need to get out to some place.

Think of a standard driveway on a house. It starts at the house and goes to the road. When you get to the road you have two choices: turn right or turn left. The choice you make depends on where you want to go. Thus you make a destination-based routing choice. In the simplest case your driveway connects to a dead-end road and you will always turn the same way. This would be the single default route scenario.

But to somewhat extend the analogy, what if your driveway opened onto a 6-way intersection? And to complicate the picture further, one of the intersecting roads is a highway on-ramp that only permits sedans, one of the other roads is a gravel road only usable by tractors, and so on. In these cases you need to make a decision based on what you are driving as well as where you are going. These are the types of network setups where you route packets differently, depending not only on the destination addresses but also on other packet fields such as source address, IP protocol, transport protocol ports, or even packet payload. This type of routing is called *Policy Routing*.

### 2.1 Defining "Policy" in Policy Routing

The standard dictionary definition of policy is

1. A definite course of action adopted for sake of facilitation
2. A course of action pursued by a government or organization

Both of these definitions imply that a policy is a describing or proscribing set of rules and actions that encompass an ideal goal. And that implication fits in well with the scope of policy routing.

The policy in Policy Routing is to provide routing capability based on any or all facets of a packet. This includes not only the header information, but also the data contained within the packet itself. As you will learn in [Chapter 3, "Linux Policy Routing Structure,"](#) there are some concrete limitations in implementing the entire functionality of ideal policy routing. In [Chapters 5 through 8](#) you will implement and manipulate some very intricate networks. For now I will talk about the ideal nature of policy.

Even in the early days of networks, back in the early 1970s, there were discussions on what constituted the minimum set of information needed to route a packet. These discussions mostly covered the reality of the ARPAnet design, namely how to survive a nuclear holocaust and still provide network services. Within this scenario the destination-based routing made more sense because you we did not care how your packet was treated but were only concerned with getting your packet to the final destination. Many of the problems with the destination only treatment, such as long delays or retransmissions, were not as critical. The applications of that time, such as FP or Email, were much more forgiving. In the 1980s and through the explosion of the 1990s, the essential nature of the ARPAnet changed to become today's Internet. Today the need is to grease the wheels of business and not to care about the nuclear winter.

With the imperatives of business comes the need for security and flexibility. You need to protect the business assets while making it easy for customers and associates to access and use those assets. Now you do not have just a network, you have an intranet. An intranet connected to the Internet and to your supplier's, vendor's, partner's, and customer's intranets. The complexity of the network connectivity may be dealt with through standard dynamic routing, but what about the service and security?

Service is a key to success in any business environment. To provide a good impression of your business you coach and train your telephone and counter people to show competence and give good service. But how do you present a good image through your network? This question is becoming a prime concern for many companies as they struggle to differentiate themselves online. The answer lies in technology that was beginning to be discussed and considered back when the current Internet Protocol, IPv4, was making its debut.

As you probably have guessed from the terminology I have chosen, the not-so-secret technology is Quality of Service (QoS), also known as Differentiated Services (DiffServ) or Integrated Services (IntServ). While purists of the IP protocol will castigate me for doing so, as a network engineer I coalesce these services into a single thought. All of the various types of QoS exist to solve the same problem, namely how to provide different treatment for different data streams. And now the circle begins to close as the earliest implementations of Policy Routing were primarily implementations of various QoS concepts.

So when I speak about the "policy" in policy routing I refer to the set of describing and proscribing rules that implement the routing structure of a network or assortment of networks. This policy constitutes the ideal usages and services of the network. While the actual implementation of this policy may differ by device, by type, or by the nature of the data streams themselves, the overall effect implements the business imperative of the network. As you will see throughout this book, the policy can range from a simple concern for resources to a global scale definition of data stream priorities.

## 2.2 Common IPv4 Routing Problems and Solutions

As I implied earlier in the chapter, Policy Routing grew out of several different needs within IPv4 networks. One of the more recent and prominent discussions was the entire topic of QoS. But there have been many problems with the destination-based routing structures. Some of you may remember that it was only recently, in the early 1990s, that the Internet was allowed to carry any commercial traffic. Indeed there existed for some time parallel networks where one carried commercial traffic and the other carried Internet traffic. While this was a somewhat extreme and short-lived situation, there was call for and discussions about methods of differentially treating the routing based on the source of the packet as well as the destination.

Consider, for example, an educational institution that carried on correspondence with both a commercial entity and a governmental entity on the same project. Under the parallel networks the educational institution would have to send out two copies of all work and provide two different connections to the internal networks. They would need to use two completely different sets of internal routing tables to allow data from perhaps one experimental computer to be sent through a "commercial" connection and through a "research" connection. If the connections were indeed physically separate, then destination routing could still be used. But if the connections were intermingled (as was usually the case), at some point in time it would be necessary to artificially separate the data streams and route one of them through a defined path and the other through an alternate path. These decisions would be based on the source address as well as the destination address of the packet. This is Policy Routing.

Aside from this extreme example, you can see that even in the networks that were being developed internally by many corporations and other large entities there was a need to route packets based on the properties of the packet besides just the destination. An actual case I was involved in illustrates the vagaries of the routing needed under destination-based systems.

In this instance there was an accounting network that contained the user desktops for the auditing staff. All of the work these auditors did was on servers that resided on the main accounting network located in a different building on the corporate campus. Getting the traffic through to the accounting network from the auditor network required connecting to the main corporate campus backbone.

The problem resided in the fact that some other employees of the company had been recently reprimanded for attempting to access the accounting systems. These actions were caught by the auditing staff. After these incidents the accounting network itself was locked down physically and through other various network security measures. However, there remained the problem that any and all traffic from the auditing division could be seen and potentially used to access the accounting network. The corporate campus backbone was connected to many other buildings and in some cases all of the traffic on the backbone actually traversed several corporate campus buildings.

The company brought in my team to find a way to route the auditor traffic so that it passed through, or bypassed if necessary, the backbone in such a manner as to not allow any of the auditor network traffic to traverse these other buildings. As of 1998 this network configuration was still operational so I will not go into much detail as to the exact setup of the network involved. Generally speaking, the backbone network consisted of several multiply connected network segments bridged together. Each building had one or more network sections that were connected via routers into the local section of the core bridged network. In several cases there were redundant secondary connections between adjacent network sections both within and between buildings. The majority of the routing tables were setup to route all traffic through the backbone even if more than one connection existed. Due to the core network's bridging structure you could not access the backbone router connections directly. In order to isolate and verify the traffic flow from the audit network to the accounting network we ended up defining new IP networks on many of the sectional routers. We then had to set up a ring of static routes which would force our specific traffic from within those address scopes through different network paths.

Suffice it to say that what would have been a simple Policy Routing decision within the auditing and backbone routers ended up requiring us to set up multiple IP network scopes with different destination route tables. If we could have specified routing

based on IP source address we would not have had any problem setting up the route structure. We did try using strict source routing within the packet headers but that caused even greater problems with end station configurations. In the end the only method that worked was to use a completely different set of IP network addresses and set them up statically to force the route flow. Of course with a static forced route structure you end up with renumbering and routing problems when networks are added to or removed from the global structure.

This was one of the scenarios that started me down the path to Policy Routing. It was obvious even then that if we could have specified the source IP address or network as a routing factor we could have solved the problem quickly and easily. The need for redundancy, which drove the original destination-based routing, was rapidly coming into conflict with the security and structure of the network. Ideally, we should be able to have our redundancy and our security too.

### 2.2.1 The Quality of Service Explosion

Fast forward to the mid 1990s to the explosion of the other, and considered by many people the primary, driving force for Policy Routing structures, QoS. The commercialization of the Internet has driven the entire original assumption base of the IPv4 protocol family into intense scrutiny. The traditional packet delivery basis for IPv4 has been described as "best-effort." This summarizes the entire process quite aptly. The underlying reason for the split between UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) is related to the fact that IP (Internet Protocol) is a best-effort delivery service. Given this best-effort delivery, UDP packets are defined to get there if they can, when they can, and that is fine. TCP packets on the other hand are checked through all sorts of mechanisms to ensure that the packets do indeed get to the destination. In either case, the routing is left to the routers and is decided based on where the packet is going.

Now as a corporate or other entity, what if I am willing to pay for better, faster, or available packet delivery? Enter the entire rationale for QoS. The QoS basic premise is that I as a service provider of networking connectivity will guarantee you as the service consumer some guaranteed level of network service for which you pay a premium. Under IPv4 networks, this mechanism depends on the routing and queuing of network packets depending on the TOS (Type of Service) tag associated with the QoS service provided.

There are a few standards for TOS tags and many of the actual drawbacks of the system are related to the definition of a what is provided for a particular TOS tag. But that also is the bulk of the discussion and work relating to the various QoS systems today. For example, you can look at the various specifications and definitions for Differentiated Service (DiffServ) which specify the TOS values related to various types of queuing disciplines. The assumption of course is that the queuing structure, by providing mediated preferential access to the existing finite bandwidth connection, allows for better service of your packet stream. This begs the question of what additional structures are possible with QoS?

Routing structures that take advantage of the tagging of QoS are the natural extension. By providing a route that depends on the TOS tag in the packet, it is possible to now provide certain guarantees of packet delivery. Under simple queuing alone, about all you can guarantee is that the tagged data stream will get a defined percentage of the available bandwidth. With Policy Routing based on the TOS tag, you can add in methods of congestion avoidance and preferential packet routes.

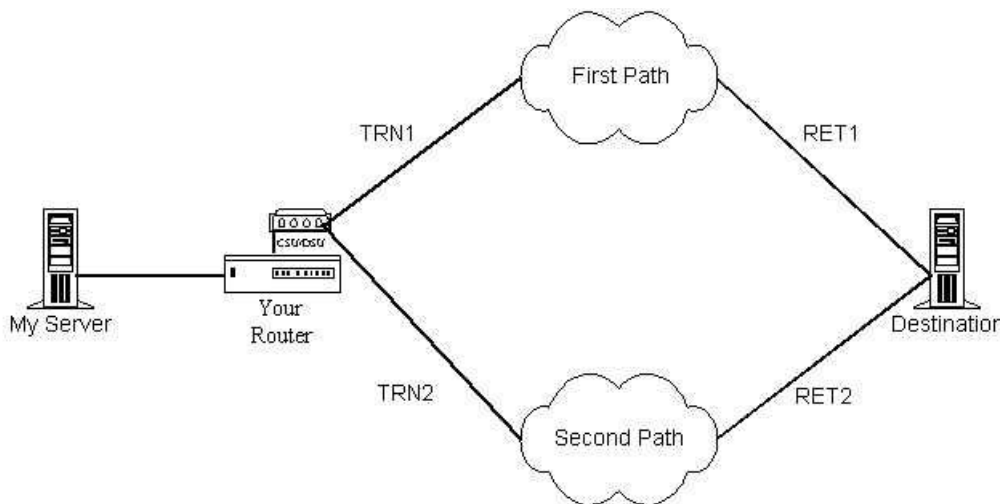


Figure 2.2.1 - Dual Path Cost Structure

Consider a setup as illustrated in Figure 2.2.1 where you have two potential data paths. Each data path is a private connection to a different network. Each of these networks is then connected to a final destination. As a peer to these external networks your

router is knowledgeable about the availability, load, and capacity of these networks. My agreement with you is to provide a certain bandwidth to my packets with a clause specifying an additional payment for low latency connections. You would like to maximize your income from my contract thus you will want to set up the policy in your routers to always send my packets by whatever path offers the lowest latency. Since your router knows the latency at points TRN1, TRN2, RET1, and RET2, you can calculate the latency of First Path as versus Second Path. Thus you can provide both a queuing guarantee to me and you can then route my packets for lowest current latency. This would work by having your router make a per packet or per queue decision based on the current calculations of Latency(TRN1 + RET1) versus Latency(TRN2 + RET2) and then routing the packet based on that decision. This packet selection can use the same TOS tag you use to queue on for the bandwidth guarantee.

## 2.3 Policy Routing Structure

This brings up the notion of the structure of a Policy Routed network. As you can surmise from this chapter, such networks comprise not only the actual routing structure itself but include the intelligence of the devices participating in the network. This is the main reason that Policy Routing was and often still is referred to as "intelligent routing." I prefer the term Policy Routing because "intelligent routing" could refer to anything better than simple static default routes.

The structure of a Policy Routing network encompasses participatory elements of the network. This ranges from the core routers through to the user end-stations. Any device that participates in the infrastructure of the network may also participate in the policy. Conversely, just because a device participates in the network infrastructure does not mean it participates in the network policy.

The primary source of the policy for the routing structure is usually found in the core routers. As the source for the majority of the routing decisions it is natural to expect them to provide the majority of the policy structure. As you will see later in this chapter, this is not always the case, especially in high security networks where the routing devices tend to be very limited in their capabilities.

A core router is usually designated as such because of the functions it provides to the network. Often this name is associated with the central router in a hub-and-spoke style routed network. However, the definition of a core router with regards to policy is the router or other security device that provides the main interconnectivity between the internal corporate network and external allied networks. The central router in a hub-and-spoke style system may not be the core router for the Policy Routing structure. It is a matter of perspective and usage. For this reason I will usually distinguish between central routers which are the main packet routing engines and core routers which are the main policy routing engines.

The recent trends within corporate networks have been to incorporate more advanced services for use. As services such as streaming audio and video are introduced to corporate networks there is a greater need to allocate and economize the resources within the network. Adding these services strains the central routers trying to handle all of the traffic being generated. The standard solution is to throw more money at the problem and purchase larger routers and higher-speed networks. But the reality in almost every case is that redesigning and optimizing the network traffic using Policy Routing returns a far higher cost savings by realizing greater efficiency. Of course such an optimization is not glamorous to senior management and not highly profitable for VARs (Value Added Resellers).

As in network security, policy and the resultant structures promulgated from implementing the policy must balance the need to provide a specific set of functions and the need to reduce the cost associated with providing such structures. These costs are not just material but also contain highly intangible human factor costs. In most cases if it is easier to bypass the policy when using the network people will do so. And when the policy in place renders usage of the network difficult, the policy hinders rather than helps. The analogy to network security policy structures is very precise and deeply intertwined.

A carefully crafted and implemented Policy Routing structure can assist on many fronts. As a concrete example, consider a policy routing structure that defines that all Internet traffic from the outbound call center be administratively denied. This serves to implement a security policy for Internet access, a network policy reducing extraneous traffic, and a business policy for divisional workflow. This multiplicity of usage points out the fundamental shift in today's corporate networks: The computing network defines and drives fundamental productivity in much the same way as the telephone network did before. And as business grew from single telephones to PBX systems and call centers, so too has computing grown to networks and NOCs (Network Operation Centers). As with the telephone usage policies earlier, network and security policies now concern themselves with the usage of the corporate network resources. And these policies are implemented through a Policy Routing structure.

### 2.3.1 Implementation Considerations for Policy Routing

Implementing a Policy Routing structure requires that all extant network usage policies be considered along with the actual logical/physical configuration of the network. In many cases, the logical and physical configuration of the network may change to facilitate the implementation. The best place to start when considering a Policy Routing structure is to map the logical

structure of the network. This logical map will show the network intermesh. The logical intermesh is important as most networks today still incorporate the single connection philosophy. The single connection philosophy defines that any two networks should only be connected at a single point.

In traditional routing, especially under RIP dynamics, you should not have two routes to any network. If you did, then only one of the routes would be used. This style of network design led to the two popular network topologies, the hub and spoke and the backbone. In a backbone system there is a central network with many routers attached to this backbone network. These attached routers connect the leaf or branch networks to the backbone network. Any conversations between leaf networks require traversing the backbone. A backbone system works well for distributed computing where the majority of each leaf network's traffic is within the leaf network. A hub-and-spoke system usually has a single large router that is connected to all of the leaf networks directly. Thus a hub and spoke is often referred to as a collapsed backbone system.

In either of these types of network, the implementation of a Policy Routing structure requires a careful analysis of the objectives and a clear understanding of the actual logical structure. Implementing Policy Routing on a leaf router when the traffic does not pass through that router not only wastes resources but can actively deteriorate network traffic flow. Worse still, implementing a Policy Routing structure without understanding the packet traversal paths and the oddities of the desktop operating systems connected to the networks can crash your network.

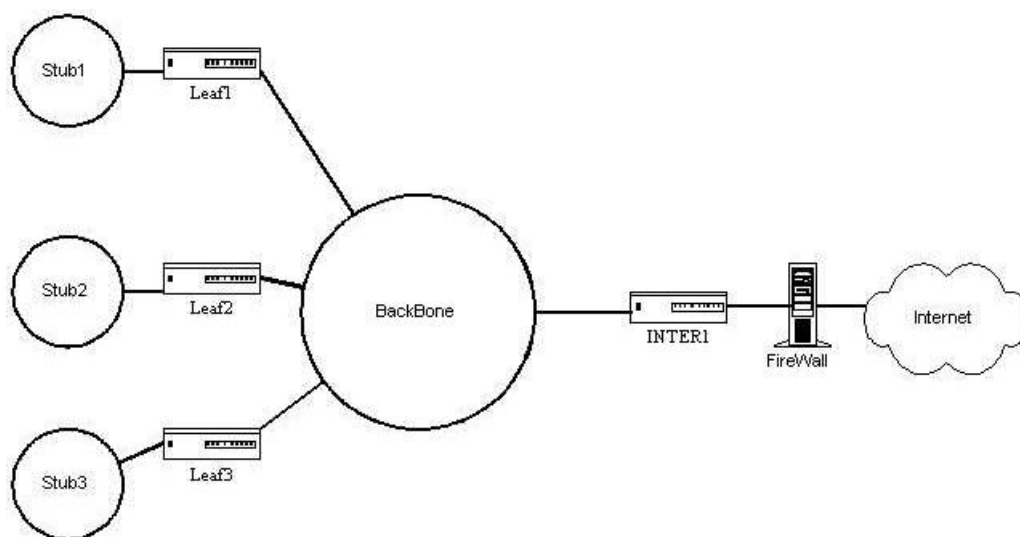


Figure 2.3.1 - MultiNetwork Policy Structures

To illustrate, consider a backbone network where several of the stub networks contain traffic destined for the Internet as illustrated in Figure 2.3.1. The connection to the Internet is through a router, Inter1, connected to the backbone and to an external network. Originally all the stub networks sent their traffic to Inter1 as the default gateway. Then Inter1 would send on all the traffic to the Internet. A firewall placed between Inter1 and the Internet connection was responsible for blocking the traffic from those stub networks not allowed to the Internet.

As the company grew the backbone network was upgraded twice to faster speeds due to the increasing level of traffic. There were also attempts to use static routes to contain internal traffic. When the backbone network reached saturation again, the decision was made to try and tune the network. As a first attempt traffic control was implemented using a policy on the Inter1 router. This policy mediated Internet access based on source IP address. As a result the backbone network collapsed on the first full business day after the implementation. Why?

A packet dump of the backbone showed that the packets destined for the Internet from the stub networks arrived at Inter1 and that Inter1 then sent back an administratively denied response packet. The desktop OS in use ignored these packets and resent the request. This caused the traffic on the backbone from these requests to triple in volume which overwhelmed the traffic capacity of the backbone. Under the old method the firewall between Inter1 and the Internet had simply dropped the request packets so there was no further response and the original request timed out.

The successful second attempt implemented the policy on the leaf routers as well as Inter1. The leaf routers were instructed to simply drop, or "blackhole," denied packets based on the source address incoming from the stub network. Packets from addresses that were allowed out onto the Internet were additionally tagged with a TOS field id that defined which stub network the packet came from and the protocol used such as http. Router Inter1 was configured to queue these packets according to the

network policy allowed rates for those services. Later on the leaf routers were configured to TOS tag and differentially route various internal network data flows, thus maximizing the available backbone bandwidth and making the network seem much faster than ever before to the end users.

## 2.4 Summary

Defining and deciding what constitutes a Policy Routing structure is harder than implementing one. Most importantly you should understand that routing policy is NOT policy routing. Policy Routing refers to the network-centric structure that performs the routing. Routing policy is an administrative or otherwise externally imposed requirement on the network. While they are very closely intertwined, they need to be understood separately. That is why in this chapter I spoke of using Policy Routing structures to implement security policies, network policies, and routing policies.

When considering the implementation of a Policy Routing structure you must understand your entire network and the scope of the network operations. Understanding both the uses of your network and the operations of the protocols traversing your network is critical to designing a good Policy Routing structure. In short, if you do not know how a simple TCP transaction differs from a UDP transaction at the packet level, you will burn yourself on the flame of Policy Routing. Like any good tool it requires skill, knowledge, and a good idea of what you want before you will get usable results.

[Return to Table of Contents](#)