Network Working Group D. Clark
Request for Comments: 1102 M.I.T. Laboratory for Computer Science
May 1989

Policy Routing in Internet Protocols

1. Status of this Memo

The purpose of this RFC is to focus discussion on particular problems in the Internet and possible methods of solution. No proposed solutions in this document are intended as standards for the Internet. Distribution of this memo is unlimited.

2. Introduction

An integral component of the Internet protocols is the routing function, which determines the series of networks and gateways a packet will traverse in passing from the source to the destination. Although there have been a number of routing protocols used in the Internet, they share the idea that one route should be selected out of all available routes based on minimizing some measure of the route, such as delay. Recently, it has become important to select routes in order to restrict the use of network resources to certain classes of customers. These considerations, which are usually described as resource policies, are poorly enforced by the existing technology in the Internet. This document proposes an approach to integrating policy controls into the Internet.

I assume that the resources of the Internet: networks, links, and gateways, are partitioned into Administrative Regions or ARs. Each AR is governed by a somewhat autonomous administration, with distinct goals as to the class of customers it intends to serve, the qualities of service it intends to deliver, and the means for recovering its cost. To construct a route across the Internet, a sequence of ARs must be selected that collectively supply a path from the source to the destination. This sequence of ARs will be called a Policy Route, or PR. Each AR through which a Policy Route passes will be concerned that the PR has been properly constructed. To this end, each AR may wish to insure that the user of the PR is authorized, the requested quality of service is supported, and that the cost of the service can be recovered.

In the abstract, a Policy Route is a series of ARs, which are assumed to be named with globally distinct identifiers. (The requirement for global names for ARs suggests that the name space of ARs is flat. That simplifying assumption is made in this RFC, but it should be possible to extend the scheme described here to permit nesting of ARs to reduce the amount of global information. The problem of adding structure to the space of ARs is an exercise for later study.) Before a PR can be used, however, it must be reduced to more concrete terms; a series of gateways which connect the sequence of ARs. These gateways will be called Policy Gateways.

Presently, the closest mechanism to policy routing in the Internet is EGP, the Exterior Gateway Protocol. EGP was constructed to permit regions of the Internet to communicate reachability information, even though they did not totally share trust. In this respect, the regions hooked together by EGP could each be viewed as Administrative Regions. However, the mechanisms of EGP imposed a topological restriction on the interconnection of the Administration Regions. In practice, this has proved unsatisfactory. Policy matters are driven by human concerns, and these have not turned out to be amenable to topological constraints, or indeed to constraints of almost any sort.

The proposals in this memo are designed to permit as wide a latitude as possible in the construction and enforcement of policies. In particular, no topological restrictions are assumed. In general, the approach taken in this memo is driven by the belief that since policies reflect human concerns, the system should primarily be concerned with enforcement of policy, rather than synthesis of policy. The proposal permits both end points and transit services to express and enforce local policy concerns.

3. Policy Routes

Almost all approaches to policy control share, to some degree, the idea of a Policy Route. The distinguishing component of a policy approach is the procedure by which the Policy Route is synthesized. One approach to synthesizing routes is to associate with each distinct policy a subset of all the gateways in the system, and then run a routing algorithm across the subset of the gateways. This approach has several drawbacks. It requires a distinct routing computation for every policy, which may be prohibitively expensive. It requires the global agreement on the nature and scope of each policy, which is at odds with the desire of Administrative Regions to establish their own independent policy assertions. Finally, it almost inevitably implies a topological restriction on the interconnection of regions.

Another synthesis approach is to have each Policy Gateway examine incoming packets and determine, based on local policy constraints, the most appropriate next AR. This approach might possibly work, but again has several drawbacks. First, it implies a substantial amount of computation at each Policy Gateway. More importantly, it removes the route selection from the location where it would most naturally be executed, the end-points of the connection.

It is useful to think of the interconnected ARs as a marketplace, in which various services are offered and users select among these services to obtain packet transport. By this analogy, it seems appropriate that the actual selection of the Policy Route should be made by the end ARs desiring to send the packets, rather than by the Policy Gateways. Looking to the phone system for comparison, it is the customer of the phone system who selects which of the long distance carriers to use, whether to purchase a fixed price service or pay incrementally for usage, and so on. In this proposal, therefore, Policy Routes are synthesized at the end point, where the packet originates, and are attached to packets in order to direct them through the appropriate series of ARs. In other words, Policy Routes are a form of source routing. The role of synthesizing a Policy Route is shared between the source AR and the particular source host.

In this architecture, therefore, the function of the Policy Gateway is not to synthesize the Policy Route, but to verify it. In the following sections, we will address the two questions of how a Policy Route is verified, and how a Policy Route is synthesized.

In determining that Policy Routes should be synthesized at the end point, it is important to distinguish between those aspects of routing that reflect legitimate policy concerns, and those aspects of routing which, in reality, relate to the detailed operation of the ARs. For example, if one were to represent Policy Routes using the existing Internet source route mechanism, which allows the end point to specify a series of gateways through which the packet should pass, the result would be that too much function has been transferred from the internals of the Internet to the end points. The end point would have to have knowledge of exactly which gateways are up and operational at a particular moment, and this degree of knowledge cannot be justified by policy concerns. Further, it would be necessary to run a systemwide gateway reachability protocol.

This proposal attempts to strike a balance between end point specification of those concerns legitimately related to policy, and local determination in the Policy Gateways of the more specific details necessary for reliable operation. This leads to a two-level routing model, in which the abstract Policy Route, a series of administrative regions, is specified by the end point as a form of source route, and each Policy Gateway selects the next actual Policy Gateway that is to be used to forward this packet. In other words, the abstract Policy Route is made concrete incrementally. This division of function does require that the source AR know if there are faults that have partitioned pairs of ARs that are normally connected together. This implies a global reachability protocol to be run for the purpose of providing information to the source AR, but it need only concern itself at the level of ARs, not at the level of gateways. In a later section on cost-recovery, the topic of gateway selection will be discussed in more detail.

An objection to a scheme such as source routing is that the potentially bulky source route must be in every packet, and must be evaluated for each packet. One solution to this performance problem is to employ a limited form of route setup, in which the actual Policy Route is carried only in the first packet of a sequence, and a short identifier or "handle" is included in subsequent packets of the sequence. Each Policy Gateway evaluates the PR on first encounter, and caches the result, which is then retrieved for later packets using the handle in the packet. The idea of a handle and caching, and the need for a form of route setup, is discussed later.

4. Verification of Policy Routes

As a packet arrives at a Policy Gateway, attempting to enter an AR, the Policy Gateway must decide whether it is legitimate to forward this packet, and if so, at what next Policy Gateway the packet should exit the AR (assuming that the final destination is not within the AR). The information available to the Policy Gateway to support its decision determines the range of policies that can be enforced. Determining what information is to be available is therefore a central feature of our proposal.

4.1. Identifying the User

Classic routing decisions, those minimizing some cost, are typically driven only by the destination of the packet. At a minimum, policy decisions must be based both on the source and the destination of the packet. In fact, source and destination addresses may not be sufficient to determine policy, for an AR may support different users with different rights, moreover a single user may wish to exercise different rights at different times. I suggest that to identify the user who is proposing to use this particular Policy Route, it is sufficient that the packets contain the source host and AR, the destination host and AR, and, optionally, a User Class Identifier, or UCI. In a later section, I discuss how to prevent misuse of the user class identifier.

In fact, the source and destination host address may not be needed to support the practical range of policy decisions required at intermediate ARs. Only the source and destination AR information may be necessary. If individual host addresses are to be used, that implies that intermediate ARs will want to keep track of the rights of individual hosts. It would be much simpler if the source AR could be trusted to permit only the proper hosts to use certain PRs. I will consider this further in a later section when I discuss the role of the Policy Controller.

4.2. Verifying the Route

The packet contains an abstract Policy Route: a series of AR identifiers. To validate this route, each Policy Gateway could store the complete selection of acceptable policy routes, and require that an incoming packet have a Policy Route that exactly matched one of the stored entries. This degree of constraint probably overspecifies the situation, and causes an information explosion. At the other end of the scale, Policy Gateways could simply be sensitive to the source AR and the destination AR. In some cases, particularly as regards to billing, this does not provide sufficient constraints. This proposal suggests that in deciding whether a given Policy Route is valid, a Policy Gateway should look at the source and destination ARs, and also the ARs immediately abutting the AR in question, called the entry and exit ARs.

One can think of the verification information in the Policy Gateway as a number of templates. Each template is associated with a valid set of users, as described by the source and destination host address and the optional User Class, and contains the four ARs described above, Source, Destination, Exit, and Entry. An incoming packet should be forwarded if, and only if, there is a template matching the information in the packet. These templates will be called Policy Terms.

4.3. Conditions

The Policy Terms, as described so far, do not permit the expression of a realistic range of policies. What is needed is the ability to attach to a Policy Term a number of conditions, which describe circumstances under which the term is valid. These might include what type of service (TOS) is available, what times of day the term is valid, what accounting options are valid, and so on. A time-of-day condition, for example, would permit networks, like time-sharing systems, to offer their off-peak capacity to a wider community.

In general, these conditions could be quite arbitrary. The important constraint on these conditions is that any condition imposed by the Policy Gateway must be understood by the end point, so that it can generate Policy Routes which will conform to the condition. If this is not so, and the Policy Gateway attaches capricious conditions to its policy terms, then the end points will construct Policy Routes in good faith which are rejected, leading to a failure to obtain service and serious dissatisfaction among users. For this reason, it is necessary that the nature of policy conditions be negotiated in advance.

The most interesting and difficult conditions are those that relate to the dynamic state of the network. An excellent example is a bilateral mutual aid agreement between two transit ARs in which each agrees to carry the load of the other if the other should go down. To capture this agreement, each might wish to put in Policy Terms with the condition that they are valid only if some other AR is non-functional. In the earlier discussion of Policy Route synthesis, it was necessary for the ARs to run a global up-down protocol to describe the connectivity of ARs. This protocol is sufficient to allow the Policy Gateway to know that some other AR is non-functional, but care is required in the dynamics of this system to ensure that the end point in the PR have a consistent view of the up-down status of the world. Otherwise, there would be transient service outages, which again would lead to user dissatisfaction.

In general, this proposal asserts that policies should not be based on highly dynamic phenomenon. Administrative Regions should be thought of as stable entities which do not change state rapidly. Highly dynamic characteristics like queue length should be dealt with by proper engineering internal to the AR. Precisely because conditions must be propagated globally, attempting to base a condition on a highly dynamic parameter is liable to lead to system instability.

4.4. Ownership of Policy Gateways

In Section 1, all the resources of the network were described as being partitioned among the ARs. This statement does not extend to the Policy Gateways, which sit on the boundary between ARs. Either the Policy Gateway must be composed of two physical halves, connected by a wire, or there must be a joint agreement for the ownership and operation of the gateway. This is a matter for further study.

5. Examples of Policy Terms

This section presents examples of how policy terms would be used to express a range of practical policies. In order to give examples, it is necessary to define a notation for policy terms. The following is not necessarily the most compact form, but will be sufficient for some simple examples.

A Policy Term will be expressed as follows:

((Hs,ARs,ARent),(Hd,ARd,ARexit),UCI,Cg)

where:

Hs is the source host address,
ARs is the source AR,
ARent is the entry AR,

and these three values comprise the first "element" of the term, describing the permitted access looking toward the source. Similarly, for the destination, there is an element describing the host address, the adjacent AR, and the ultimate AR.

In addition to the two directional elements of the term, there is global information:

UCI is the User Class Id, and
Cg are any global conditions.

In many cases, an element will not want to constrain one of the values, and we will use the "*" symbol to indicate a "wild-card" match.

To construct some simple examples, here is a topology, where H elements are hosts, G elements are Policy Gateways, and Numbered elements are ARs.

```
      H1 ---  1 --- G1 -----  2 ------ G2 ----- 3 ----- H2
              |                                 |
              |                                 |
              |                                 |
              |---- G3 -----  4 ------ G4 ------|------ G5 --- 5
                              |                                |
                              |                                |
                              |                               H4
                              |
                              H3
```

In this picture, there are four hosts, five gateways, and five Administrative Regions.

First, consider AR two. It has no hosts attached, and models a transit service, such as the NSF network. It may have a very simple policy: it will carry any traffic between universities, without further constraint. If we let AR1 and AR3 be the regions of two particular universities, then its policy term could be written as:

```
AR2: ((*,1,*),(*,3,*),*,*).
```

This says that AR 2 agrees to carry traffic from AR 1 to AR 3, without concern as to the entry and exit AR, and for any hosts in these ARs.

This notation works, but is very bulky, as a new term is required for every pair of universities. There are several ways to compact the notation. First, we can use the * and a new symbol, "-", to broaden the terms a bit. For example:

```
AR2: ((*,1,*),(*,*,-),*,*)
```

would assert that AR 1 can use AR 2 to talk to any directly attached AR, where we use the "-" to mean that the exit AR must be the destination AR. In other words, the destination AR must be directly attached to AR2. If AR 2 only attaches to universities, then this would provide the proper constraint.

Another approach is to use the User Class ID:

```
AR2:((*,*,*),(*,*,*),University,*)
```

says that any traffic of any sort that has the User Class of University is acceptable.

Another, and perhaps most suitable notation, is to observe that the distinction between source and destination is actually artificial. While it helps in this memo to have names for the two ends, either end can be a source, depending on who sends the first packet. (A later section explores the bi-directional nature of PRs). A more general form of a PR is thus to permit any number of elements. That is, a Policy Term can have more than two elements, and the meaning of this is that a PR is valid if it uses any two of these.

For example, if university 5 wanted to use the AR2 service, AR2 might write a Policy term as follows:

```
AR2:((*,1,*),(*,3,*),(*,5,*),*,*)
```

which would permit a policy route between hosts in any two of the ARs 1, 3 and 5.

All the terms so far relate to the policies of AR2. If university 1 wanted to subscribe to this service, and use it to reach any other site, it would specify terms of its own. For example:

```
AR1: ((*,1, -),(*,*,2),*,*).
```

This term says that any host in AR 1 can use AR 2 as a path to any host in any AR. Again we use the "-" notation to indicate that the entry AR is the same as the source AR, in this case the AR writing the term.

The ARs numbered 3 and 5 are more interesting. While 3 is directly attached to 2, 5 is not. Instead, 5 has attached to 3. If 3 wants to use 2 for general transit service, it must provide a term similar to the one provided by 1:

```
AR3: ((*,3,-),(*,*,2),*,*).
```

If 5 wants to use 2, more terms are required. Since 2 is not directly attached, it cannot be named as the exit AR in a term written by 5. The directly attached AR, 3, is all that can be named:

```
AR5: ((*,5,-),(*,*,3),*,*).
```

Then AR3 must agree to carry the transit traffic for 5.

```
AR3: ((*,5,-),(*,*,2),*,*)
```

AR3 might not want to carry all forms of transit traffic for 5, but only of certain sorts or to certain locations. This could be expressed by restricting the previous term. For example,

```
AR3: ((*,5,-),(*,2,-),*,*)
```

would permit traffic from 5 to cross 3 to reach 2, but only to hosts directly in those ARs.

For some further examples, consider AR 4, which might represent the AR of a commercial user. It connects together the hosts of that user, for example, H3, and is connected to the other environment to permit cross-communication. Given the terms so far, no traffic will flow into this AR.

If AR 1 wants to permit communication with AR 4, it could add:

```
AR1: ((*,1,-),(*,4,-),*,*)
```

This would permit communication between hosts directly in each AR, but no transit traffic. In particular, H3 and H2 cannot talk. There are several different terms that would permit them to talk.

The direct path would be the following:

```
AR4: ((*,4,-),(H2,3,-),*,*)
AR3: ((*,3,-),(H3,4,-),*,*).
```

This would permit direct connection through G4. Note, for variety, that each term has been set up so that any host in the local AR can match, but only one host in the other AR. The combination happens to permit only H3 and H2 to communicate.

If G4 were not there, another path would be via AR 2, which could be permitted by suitable terms in ARs 1,2,3 and 4.

Even if G3 and G4 exist, no transit traffic will flow across AR 4 from 1 to 3. Even if 1 and 3 want it to:

```
AR1: ((*,1,-),(*,3,4),*,*) and
AR3: ((*,3,-),(*,1,4),*,*),
```

the lack of a term for AR4 will prevent a valid PR via that path. Only if AR 4 added:

```
AR4:((*,1,-),(*,3,-),*,*)
```

would AR 4 start serving AR a transit path from 1 to 3.

If AR4 added:

AR4: ((*,4,-),(*,*,*),*,*), any host in AR 4 could talk to any host anywhere else, but AR 4 would still not become a transit service.

These various examples demonstrate how individual ARs can offer Policy Terms that can be combined to form a route. The notation proposed here is probably not adequate to express the needed range of policies. For example, it may be desirable to have lists of ARs as part of a term, as well as single values and "*". Other notation might be proposed to permit exclusion of a limited set of ARs. It may also be appropriate to write elements that are directional, so that connections can be "opened" in one direction but not in others. This idea is vague in a connectionless architecture, but seems to relate to some real policy requirements.

In general, the problem of expressing policy terms in compact form is the same as the problem of constructing compact access control lists. There is still an ongoing argument whether access control lists should be ordered, and should permit exclusion, and so on. It would seem that the exact same issues arise here. Some experience attempting to express real policies may give guidance as to the expressive power needed.

6. Cost Recovery

Almost all of the existing Internet has been paid for as a capital purchase and provided to the users as a free good. There are limited examples of cost recovery, but these are based on an annual subscription fee rather than a charge related to the utilization. There is a growing body of opinion which says that accounting for usage, if not billing for it, is an important component of effective resource management. For this reason, tools for accounting and billing must be a central part of any policy mechanism. However, precisely because the administrative regions are autonomous, we cannot impose a uniform form of billing policy on all of the regions. Some of them may continue to provide service freely, or on the basis of an annual fee. Others may charge on the basis of resources consumed, but even here there may be variations in detail, as some may charge by the packet and others may charge by the byte. Again, in the telephone analogy, we see a variety of billing policies, with both local and long distance carriers selling service either on the basis of a monthly fee or on a fee-per-minute of usage, with time of day conditions attached. The billing problem is thus a very complicated one, for the user would presumably desire to minimize the cost, in the context of the various outstanding conditions.

If we are actually to pay for use of services, there is also the problem of collection. Using the current telephone system as an example, there are two strategies for collecting revenues. One is the pre-divestiture mode, in which the source AR (or the destination AR in the case of a collect call) serves as a single collection point for all of the ARs involved in the call. After divestiture, we see another paradigm, in which the transit AR separately bills the customer.

There are many reasons to support both collection formats. The primary reason for separate billing is that not all regions may wish to charge the user in the same units of currency. Some regions may wish to charge actual dollars, while others may wish to charge using some form of private allocation units. On the other hand, having a single point of collection is very convenient, because it eliminates a lot of duplicate effort in collection. It does, however, require a greater degree of trust and coordination among ARs.

Single point collection also simplifies another sticky problem, lost packets. For most types of service, the user would presumably be offended if asked to pay for a significant number of packets undelivered because they have been lost before reaching the destination. If each region separately bills for its traffic, then to avoid billing for packets that are lost between that AR and the destination, it is necessary to have some form of lost packet reporting, which travels backward through system decrementing the counters of all the intervening ARs. If single point collection is performed, then the usage meters can be put in the destination AR, and periodically propagated to the billing AR, if that is a different region.

The discussion of lost packets makes clear an important relationship between billing and policy. If a Policy Route takes packets through a region of known unreliability, the regions preceding it on the path may be quite unwilling to forgive the charges for packets which have successfully crossed their region, only to be lost further down the route. A billing policy is a way of asserting that one region wishes to divorce itself from the reliability behavior of another region. The conditions in the policy terms, and corresponding policy routes, must therefore be able to capture two distinct conditions. The first is whether or not there exists a bilateral agreement between two ARs by which one agrees to be the collection agent for the other. The concatenation of a number of these agreements permits a single collection point to be used for the entire policy route. The other condition is whether or not the AR will accept packet and byte counts from the next AR downstream as the basis of billing, or whether the AR insists that the billing be based on the counts at the exit point of this AR. This condition allows an AR to build a wall between it and a subsequent unreliable AR. One can imagine certain regions agreeing to carry traffic into unreliable regions, but only grudgingly, knowing that the result is going to be user frustration which may be directed to all the ARs indiscriminately. The use of a specific policy condition can make clear to the end user which ARs do not view themselves as interworking harmoniously.

To enforce these mechanisms, the abstract PR which is included in the packet must be augmented with a number of conditions. First, for each AR there is a 3-way flag which describes whether the billing should be separately collected for the region, propagated back to the source (which corresponds to the normal telephone company paradigm), or propagated towards the destination (which corresponds to a collect call). Second, there is a flag which indicates whether the region is expected to accept from the next region downstream the packet and byte counts as the basis of billing. Third, there must be a charge code, a unique number somewhat resembling a credit card number to which bills may be sent. The Policy Terms in the Gateways must similarly be augmented to permit verification. The management of the charge code, insuring its uniqueness and preventing its abuse, is discussed later.

These conditions, which relate to agreements between two ARs, are somewhat different from the conditions previously discussed, such as time of day. Conditions relating to AR agreements will be called "bilateral conditions," while the others are called "global conditions." Note that even though bilateral conditions relate to the agreement between two ARs, they can have global effects.

7. Gateway Selection

In Section Two, this memo proposed that the end point should specify an abstract Policy Route, as a series of ARs, and the Policy Gateway at the entry to each AR should convert the next hop to a concrete route, selecting the Policy Gateway to exit from this region into the next. It turns out that this selection is not entirely devoid of policy concerns, and some additional conditions are required in the Policy Terms in order to make this operate properly.

In order that each Policy Gateway be able to select the next Policy Gateway on the route, it is necessary to have a table which lists all of the potential Policy Gateways that connect together adjacent regions. Presumably, this information is very slowly changing, and is not difficult to propagate. The more dynamic information that is needed is whether each of these gateways is up. It is therefore necessary that all of the Policy Gateways attached to a given AR must run a local up-down algorithm, one which hopefully can determine not only that each of the other gateways is up, but that its interfaces are up and that it is properly forwarding traffic. It is slightly complicated to design such a test. However, we do not have to design a strategy for propagating this information globally, because it is only needed by the other Policy Gateways attached to each region.

The policy matter related to concrete routes arises if there are several gateways connecting two administrative regions. As described so far, the exit Policy Gateway from any region (which is the entry Policy Gateway for the next region) is selected by the entry Policy Gateway for that region. In other words, each region may select its exit gateway, but has no control over its entry gateway. There are certain circumstances where a particular region might insist on being able to control the entry gateway used. Imagine two parallel transit regions, one which charges incrementally for service, the other of which provides its service as a free good. Obviously, from the point of view of the user, it is desirable to minimize the use of the charging AR, and maximize the use of the free AR. But this may lead to gross overloads in the free AR, and apparent discrimination against the charging AR. The owner of the free AR, therefore, might choose to impose a policy which says that it can be used only to reach certain points which are not directly connected to the AR which bills for its service, and the traffic must enter the free AR at the closest point to the destination. In other words, the free AR requires that it be allowed to choose its entry gateway so that it minimizes its costs (which are not, in fact, being billed), with the intent of shifting as much as possible of the cost onto the other network.

By adding more bilateral conditions to the Policy Terms and the Policy Route in the packet, it is possible to control the various options for Policy Gateway selection. At each boundary between ARs, there are only a limited number of ways to select the Policy Gateway. Either it is selected by the entry side, by the exit side, or by some collaborative algorithm specified through a bilateral agreement. (There might be several such algorithms, which requires the possibility of more complexity in the specification. In particular, if two adjacent ARs have agreed to use a common routing metric for some type of service, they may agree to make a common routing based on this metric.)

Allowing the policy gateway to be selected by the AR which is on the far side of the gateway represents an interesting implementation problem. It would be possible to send some message in advance of the packet, which requests the next AR to select its entry gateway. To do this, it would figure out what its exit gateway would be, and then figure backwards to minimize its costs (for example) to select the potential entry gateway back into the immediate region. This is complicated to describe, and would probably be complicated to implement. One way to focus the problem is to observe that routes are bi-directional, because a packet flow is bi-directional, and it is very desirable that the packets from both directions follow the same route. Once a packet has come back along the reverse route, the gateway from which it emerges is precisely the gateway which should be used for future traffic in the other direction. But each gateway, in either the forward or reverse direction, must remember a decision made by another AR.

For this to work it is necessary that gateways not be stateless. If each Policy Gateway maintains a cache of recently computed Policy Routes, in particular remembering the result of computing the gateway for each abstract route, then by simply determining whether or not the forward direction or the reverse direction is allowed to constrain the gateway across this boundary, both policies can be enforced. But this requires building gateways with state, which has not been culturally acceptable in the Internet. I therefore consider as a separate topic the virtues of state in Policy Gateways. I believe that fairly simple algorithms exist to set up the required bindings in the Policy Gateways, but that problem is a matter for later study.

8. Flow States

The previous section suggested that the gateway needed to maintain state in order to tie together the forward and reverse halves of a flow. This solved the particular problem of tying together the routing decision which had been made in each direction, so that they could be used in the other. There are, in fact, a number of reasons why the two halves of the flow should be tied together.

- There is considerable overhead in accounting and collecting for the usage. It is clearly desirable to have both halves of the flow metered jointly.

- If the route is not bi-directional, then a failure in the node produces a uni-directional link. Uni-directional links are known to cause anomalous behavior in protocols.

- As part of resource management, it may be desirable for intermediate nodes to pass flow control information back to the source of the flow. If identifiable reverse-direction packets are passing through the gateway, then this information can be piggy-backed onto those packets.

An additional advantage of maintaining state in the gateway is that it will greatly reduce the overhead of dealing with incoming packets. There are a number of decisions which the Policy Gateway must make which are a part of forwarding a packet: it must validate the Policy Route against its terms, it must create or modify an accounting record, and it must select the next Policy Gateway. It is unreasonable to imagine performing these tasks from scratch for each incoming packet. Once these decisions have been made, the results should be cached, so that they can be used for subsequent packets.

The stateless gateway was proposed as part of the Internet design in order to insure a robust architecture. If the gateway has no state, then a crash of a gateway cannot endanger an on-going connection. If there is state in a gateway, and that state information is lost because of a crash, then it is possible that a flow would be disrupted.

In moving from a gateway with no state to a gateway which caches information, it is necessary to ensure that the cached information can be lost and reconstructed. The idea of keeping in gateways only that state which can be easily reconstructed I call "soft state."

9. Synthesis and Selection of Policy Routes

In this proposal, a packet contains a Policy Route, which is verified by each Policy Gateway along the way. This section discusses how the Policy Route is created in the first place.

PR creation cannot be done totally automatically by the system, but will in general require human judgment. Policies, after all, are matters of human concern. The approach to PR creation is thus a joint one, in which the system provides support to the persons setting policy.

Most commonly, the desired PR will be selected from among those available by first finding all valid PRs, and then picking one that meets the requirements of the user and has the lowest real cost. These two stages will be called synthesis and selection.

To synthesize a PR across a sequence of ARs, one must find a Policy Term in each AR that would permit such a PR. The Policy Terms in each adjacent AR must be compatible in their billing conditions and other particulars. One can imagine finding a sequence of Policy Terms that match, rather like dominoes, and reach from the source to the destination.

For a Policy Term at some AR to be acceptable as a part of a PR, the following must be true:

- The Source and Destination Host address and UCI must match the term,

- The Source and Destination AR must match the term,

- The Entry and Exit AR must match the adjacent AR in the route,

- The conditions in the term relating to the adjacent AR (e.g., billing) must match the conditions in the term from that region.

These conditions, of course, are exactly what the Policy Gateway would test in validating the PR when it is used.

As the route is synthesized from matching terms, the global conditions of each term are noted, and the combination of these become the condition under which the PR is valid. As a starting point of the synthesis the user may have indicated constraints on the acceptable conditions in order to limit the candidate terms in the synthesis.

The result of PR synthesis, which is somewhat similar to the computation in a link-state routing algorithm where each Policy Term represents an abstract link, is a potentially long list of possible PRs to each destination AR, each with attached conditions. The selection process must identify one of these which is actually to be used. The selection can be based on the conditions, and on the cost of each PR.

To determine the cost, it must be possible to ask each AR to identify the cost of using that Policy Term in the context of this particular set of Entry and Exit ARs. Either there must be an architected protocol for reporting these costs, or the task of cost determination must be left to humans to perform outside the system. The problem with architected cost reporting is that while some ARs may bill using real dollars, others may bill in terms of abstract usage authorizations which have no meaning outside that AR. Even so, I believe that we should attempt to define a representation for reporting the billing basis associated with each AR. This is a matter for later study.

While PR synthesis may be an automated process, selection probably is not. While cost minimization will help prune the list, and some routes may be rejected automatically on the basis of conditions, part of the selection will in general require human judgment. This observation, together with the observation that PR synthesis may be costly, suggests first that synthesis and selection cannot be done for each packet or indeed each time a transport connection is established, and second that it should not be done separately for each host in the AR.

Instead, each AR should have one (or more) Policy Servers, servers inside the AR which support the management of PRs. The Policy Server would perform a number of functions.

- It would store the Policy Terms for the AR, and make them available to the Policy Gateways and the Servers of other ARs as appropriate.

- It would synthesize potential PRs to reach other ARs, and remember which of these have been selected for use.

- It will respond to requests from hosts in the AR for PRs, and return them so that they can be included in outgoing packets.

- It will participate on behalf of the AR in AR up-down protocols, and other inter-AR routing algorithms.

- It will remember the location of all Policy Gateways attached to this AR.

- It will provide the management interface for those persons who must establish AR policy: setting of local Policy Terms, selection of Policy Routes, and so on.

A host wishing to send packets outside the local AR must first obtain a PR to put into the packets. In the normal case, it would do so by directing a request to the local Policy Server, supplying the desired destination and other negotiable conditions. (For example, the TOS is negotiable, the current time is not.) The Server, based on this input, must select the most appropriate PR and return it.

At this point in the process, human intervention is not reasonable, as it would take much too long. By now, sufficient selection must have been done so that automated PR selection is possible. The most direct implementation is that the manual selection process should yield an ordered (or partially ordered) list of potential PRs, and the list is searched in order until a PR is found that matches the destination and conditions. That PR is then returned.

10. Security

There are a number of aspects of this scheme which present opportunities for abuse. In essentially all cases, the possible abuse is theft of network resources or improper charging. They thus have a somewhat different nature than problems related to corruption or disclosure of data. Mechanism to insure proper use and charging of resources often tolerate minor abuse in exchange for ease of operation. Also, control is often based on detection and recovery rather than prevention. Assumptions of this sort are probably acceptable here as well. An isolated packet, which is not a part of any sequence of packets, may be too small an item to account for or control. But if a significant stream of packets goes unaccounted, this is less acceptable.

There are three general options for abuse. One is to falsify the user identification information in the PR, the source and destination host, the User Class Id and the charge code. Another is to take a valid PR and misuse it intact. And the third is to read out a valid charge code from a PR and then make additional charges against it.

To protect against putting false user identification information into a PR, the PRs should be sealed or signed, using a crypto sealing technique. Since Policy Servers are the source of PRs, the sealing can be done by the Server. This would require that the seal or digital signature of each Server be known, but avoids the need to have each host known. The Server would be trusted to seal only valid PRs. It must only put User Class Ids and charge codes into PRs from a source permitted to use them, for example.

Assuming a public key system, each Policy Server could have a separate key pair, the public half of which was advertised in some way. It is a matter for further study exactly what parts of the PR need be sealed.

If the Policy Server violates this trust, and uses a UCI or charge code with an unauthorized host, there are two sub-cases: the false source host is in the same AS, or is outside it. If it is outside, this can be detected by inspection of the PR, since the relation between AR and network number is (almost) static. One approach is to make an AR identifier part of the charge code, so that use of the code can be rejected unless that AR is the source AR for the packet. This works, but prevents using charge codes from a foreign location. Other more general techniques could probably be proposed.

If the false source host is inside the AR, then further steps are required to prevent the problem. One general solution is to note that a PR is valid only if sealed by a Policy Server. Any AR attempting to collect for usage should be required to keep a copy of the PR as proof that the route was used. If there seems to be unauthorized use of a charge code, the owner can ask to see the PR which generated the charge, which will show the Policy Server which constructed the route. If this is an unauthorized use, action can be taken against the AR owning that Server, with the sealed PR as evidence. In other words, detection and redress may be more effective than prevention.

If we can assume that the Policy Server for a particular region is as trustworthy as that AR requires, there is still the problem of a Server of one region trying to steal from another AR. This could be done, for example, by taking a valid PR, and sending data forward along it from the "middle" of the route, so that what appears to be coming from one source is actually coming from another in a different AR.

This would require that packets coming back along the route towards the original source be rerouted to the false source, which would require that the whole routing function within the AR be corrupted. It is unlikely that this would go long undetected, but if direct control of this class of fraud is needed, it could be achieved by requiring any AR intending to charge against a particular PR to obtain from time to time a confirmation, sealed by the Server of the source AR, that its policy gateway has in fact forwarded some number of packets using this PR. This sort of function is probably overkill, but this class of fraud needs to be considered.

Obviously, a more detailed study will be required of the problem of resource theft, but I believe that a mechanism can be made to work based on:

- Local trust of the Policy Server within each AR.

- Sealing of the PR by the Server.

- Selective validation of the seal at the Policy Gateway.

- Selective consistency checking of the PR at the Policy Gateway.

- Use of seal on PR as evidence of the source of the PR.

11. An Experimental Program -- Migration to Policy Routing

The proposal above calls for several Internet components not present today: the Policy Route IP option, Policy Gateways, Policy Servers, and support protocols such as the global AS up-down protocol and the local (to the AS) Policy Gateway up-down protocol. Any plan for introduction of policy routing must provide a method to experiment with the concept without changing all the hosts and the gateways now in place.

Since the Policy Server is a new component which can be added to the Internet without changing any existing components, it is easy to put that facility in place. This, then, becomes the central part of an experimental plan. Later, it is possible to imagine adding the policy controls to some of the gateways. Most difficult will be modifying all the hosts to use the PR IP option. Based on our experience with adding minor features such as IP subnetworks, it will never be possible to get the PR option into all the hosts, and policy routing must be made to work anyway.

Taking into account these difficulties, here is a concrete experimental plan, in three phases.

In Phase I, software for a Policy Server is created, and made available to all potential ARs. As a part of its function, it has two "temporary" feature, to mimic the function of the missing host and gateway support.

To mimic the function of the policy gateway, two policy Servers are placed "near" a current function gateway which happens to connect the two ARs, one on each side of the current gateway, and representing their respective ARs. These two Servers then proceed to fool the current gateway as follows.

- The current gateway is given the two Servers as neighbors in its routing exchanges. In this way, the Servers can control which network numbers are advertised. This is similar to the way "gated" is used today to control routes.

- A packet entering the AR is directed to the "near" Server inside the AR, which performs the functions of the Policy Gateway and then resends the packet. This may require the use of a regular source route in some cases, but can probably just be done by rewriting the destination IP address in the packet. (Note that the IP PR option proposed in the Appendix has fields for the original IP source and destination, so that these fields can be reused in forwarding the packet from gateway to gateway.)

To deal with the lack of host support for the PR option, we again make use of the Server. Since the Server is the recipient of all routing information coming into the AR (since it has been set up as the neighbor of the current gateway at the actual AR boundary) it alone knows the proper routes out. Internally, it advertises itself as the default gateway to all networks outside the AR, so that it receives all the packets intending to leave the region. It, rather than the host, adds the PR option and then sends the packet on the Policy Gateway (or the matching Server in the next AR playing its part) for relaying.

By controlling how routes are propagated by the regular gateways, it is possible to prevent hosts from manually setting up routes to bypass the Servers. In any event, enforcement is not the primary concern in Phase I of the experiment.

In Phase II, certain of the current gateways are augmented with the Policy Gateway functions. This will make enforcement easier, and eliminate the extra hop which the packet had make in Phase I, as it passed from one Server to another through the current gateway. At the same time, some of the hosts are modified to insert the IP PR option into the packet at the source. This will explore the problems of PR selection.

In Phase III, the PR design is proposed for general implementation.

12. Policy Route Setup

One objection to this scheme is the large size of the IP PR option. With all the information proposed in this memo, it is larger than the IP header itself. However, this problem can easily be avoided; the PR option seldom need be sent.

Since the Policy Gateways are going to cache the result of processing the PR, the cache holds the equivalent of the PR. All that is required is a very short option in the packet which is a handle that permits the gateway to find the correct cache entry. This handle would be included in the original IP PR option, and then repeated in every packet. The Policy Server which generated the PR could select the handle, so it would be unique for each AR. Perhaps the AR id and a 16 bit UID would be sufficient.

The full PR option needs to be in the packet only if the cached Information in the gateway is lost. If a gateway crashes or the route changes, the end point must reconstruct the caches in the series of gateways that form the route. The end point could determine that this was necessary either when a gateway reports explicitly that it does not have an entry corresponding to a handle, or when the host determines that it is not getting the desired service.

This sort of action can be thought of as an extension to the idea of retransmitting. In transport protocols such as TCP, the host keeps track of the behavior of the network, and if it believes that something is wrong (e.g., there is a lack of an acknowledgment), it takes action to restore the desired service. Other examples include switching to another gateway if the currently active adjacent gateway seems to be down. Sending the full PR option in the packet is just another example of allowing the end node to restore the state of the connection if it seems to be broken.

Using this model, most packets would have only a short option (perhaps 12 bytes).

This idea of restoring the state in the gateway as needed achieves the idea of "soft state" mentioned earlier, and allows gateways with state to achieve the same robustness associated with datagram networks.

Author's Address

David D. Clark
Massachusetts Institute of Technology
Laboratory for Computer Science
545 Main Street
Cambridge, MA 02139

Phone: (617) 253-6003

Email: ddc@LCS.MIT.EDU