Internet Engineering Task Force
INTERNET DRAFT

Authors
R. Rajan
AT&T
S. Kamat
IBM

23/ May/ 1999

A Simple Framework and Architecture for Networking Policy
draft-rajan-policy-framework-00.txt

Status of Memo

Abstract

Many new protocols defined in the IETF have a regulatory component, i.e., network administrators have to decide what users, applications or hosts should have access to what resources/services under what conditions. Large-scale deployment of services using such protocols is critically

dependent on the presence of a network-wide policy infrastructure that allows Internet Service Providers (ISPs) and corporate administrators to regulate the network rather than configure individual devices. A key component of this effort is to evolve standards-based means of representing and storing policy information in a unified manner, with a focus on schema for storing policy information in directories. This document presents concepts, terminology, a policy framework and requirements for such a definition.

1. Introduction

Given the wide and varied usage of the term "policy", it is imperative that the word be clearly defined in the networking context. As a starting point, we shall use policy to denote the unified regulation of access to network resources based on administrative criteria. Policy is driven by the need to create multiple services over a shared IP infrastructure melding together different protocols. A complementary perspective on policy is to think of it in terms of (relatively) static performance goals or operational rules for a dynamic network environment supporting a variety of services. A policy infrastructure is the collection of information models, protocols and mechanisms through which the intentions of administrators can be represented, stored, communicated and translated into operations carried out at various network elements that encounter packet streams.

In order to motivate the domain of policy as defined in this document, consider the following examples:

(1) An ISP operator wishes to offer an ``extranet'' service to interconnect some company's campuses spread across the United States. Such a service includes assurances of traffic isolation, privacy, as well as bandwidth availability for traffic carried across the extranet. Such a service is pieced together from a variety of different protocols and components, e.g., network address translation to ensure connectivity, firewalls to ensure campus isolation, IPSec for encryption and authentication, MPLS for QoS provisioning across the backbone, etc. The regulatory components of such a service, to be provided by a policy infrastructure, include regulation of address translation to ensure connectivity, policies to prevent security violations at the firewall, and policies to automatically encrypt data from confidential servers across the extranet.

(2) In addition to the above ``extranet'' service, the operator wishes to offer 4 different QoS services, priced differently. The ISP allows customers to indicate their service preference on a packet-by-packet basis by setting DS codepoint in IP header appropriately. The added regulatory component of such a service is traffic volume verification through policing at network access points.

(3) A reputed telecommunications company wishes to offer unified voice and data services over cable modems. The billing and processing requirements for voice are different from data. The regulatory components of such a service include bandwidth management, prioritization of voice traffic, account verification and call routing.

(4) An administrator of an RSVP-capable intranet wishes to restrict individual Controlled Load reservations from engineering staff during the day (9am to 5pm) to a certain token rate and also limit the total bandwidth of such reserved flows.

(5) A network operator interfaces with another network operator at a peering point to support communication within their network. Network operator A supports 4 service levels within network A, while network operator B only supports two service levels. Network operator B maps the first two service levels of A to one type of its own service levels, and the remaining two into its second type of service level. The policy component of such a service is to configure and enforce the mapping between the service categories and enforce traffic contracts.

It is clear from the above examples that policy, i.e., the regulatory aspect of service provision, is very wide in scope. Obviously, the entire policy infrastructure cannot be unified and standardized all at once. Instead, standardization efforts must be aimed at protocols and representations that allow devices potentially belonging to multiple vendors to interoperate by distributing, sharing and interpreting policy information in a consistent manner. Further, while some mechanisms for regulation of services are best provided within protocols themselves, it is very unsatisfactory to completely isolate policy definition for different protocols from one another, for several reasons.

- Different services may benefit by using common policy transport mechanisms and semantics.

- Policies regarding different services are defined using common categories such as users, applications, hosts, etc. Administering these separately for each service could be time consuming and messy.

- Policies for different protocols may have interaction effects. For instance, network resources are wasted if a QoS policy reserves bandwidth for a flow that another security policy forbids.

- ISPs and corporate administrators would prefer to define new hybrid services, such as combining network access with other security and QoS features. These higher level services need to be administered in a uniform manner, which would be best done given a common policy infrastructure.

However, given the diverse needs of different environments and the heterogeneity of information availability and functionality in

network devices, the evolution of a single ubiquitous policy solution is very ambitious. Instead, we take a gradual approach which is aimed at well defined policy needs at first, working out to a more ambitious framework as newer policy requirements emerge. Our first aims are:

- 1. Derive a common terminology and conceptual framework for policy-based network control, with a few carefully selected services and protocols as the first targets.

- 2. Develop an appreciation for the policy needs of current services, with an emphasis on QoS and security services.

- 3. Distill from these a simple, common architecture that illuminates our workspace,

- 4. Present an overview of existing and proposed "pieces of the puzzle" that may be usefully employed to implement the architecture.

- 5. Show that the architecture assumed does not restrict the solution space by outlining different feasible enhancements and extensions.

- 6. Obtain requirements for and constraints on a common policy representation that may be stored in a directory ("directory schema") with a focus on Quality of Service.

- 7. Define an associated policy information base (PIB) and management information base (MIB) that could be usefully employed in controlling and managing heterogenous devices.

A key motive for this effort is the drive to evolve standards-based means of representing information ("schema") in a unified manner, and storing such information in LDAP-accessible directories [6]. This document presents concepts, terminology, a policy framework and requirements for such a definition.

2. Requirements

2.1. Policy Requirements for QoS

In order to obtain some insight into the semantics of policy rules required for QoS and other services, we present an over-view of the policy needs of integrated and differentiated services.

Integrated services with RSVP signaling approach seeks to provide per-flow QoS assurances with dynamic resource reservation. A flow is defined by the 5-tuple (source address, destination address, protocol, source port, destination port). RSVP is a protocol for reserving network resources for unicast or multicast flows, with stringent requirements satisfied by a guaranteed service, and more flexible requirements by a relaxed controlled load service. In this context, there is need to provide policy control of individual flows, and regulate their ability to reserve network resources. RSVP is capable of opaquely transporting policy-objects which may be used by policy-aware nodes to derive more information about the user, application or end-points of communication. The role of policy

allows for highly granulated interactions with billing, accounting or accreditation systems resulting in informed control over the size and nature of the QoS reservations in the network. (See [8] for a discussion of policy based admission control framework and sample policies). Differentiated services (DiffServ), on the other hand, are aimed at traffic aggregates that may not correspond to fine-grained flows. The frequency of signaling is much coarser, and may occur through a variety of mechanisms (bandwidth brokers, off-line communication, service level agreements, etc.). The DiffServ approach relies more on administrative control of bandwidth, delay or dropping preferences, rather than per-flow signaling protocols to communicate service level information to network elements. For such services we wish to enable flexible definition of class-based packet handling behaviors and class-based policy control. See [7] for a discussion of DiffServ framework and sample behavior/service descriptions).

While these mechanisms address the issue of how to provide quality of service, the complementary issue of which packets are eligible for what services is a policy issue. For instance, an e-tailer may wish to provide preferential treatment to real-time transaction oriented web-traffic. Or an ISP may seek to ensure that voice-over-IP is assigned to a low-loss, low-delay class of service, while limiting the number of simultaneously supported voice calls. Or consider a network administrator of an RSVP capable intranet who wishes to restrict individual Controlled Load reservations from certain sources during the day to a certain token rate and also limit the total bandwidth of such reserved flows. In these and other examples, the utility of QoS services depends heavily on the presence of mechanisms for administering them. In fact, large-scale deployment of QoS services is critically dependent on the presence of a network-wide policy infrastructure that allows Internet Service Providers (ISPs) and corporate administrators to control the network rather than configure individual devices.

Speaking broadly, the policy needs of QoS administrators are of three kinds. First, administrators would like to be able to provision networks, i.e., earmark resources for use by certain types of traffic. Second, they would like to control the terms and conditions under which users are able to reserve bandwidth in the network.

Third, they would like to substitute one QoS service for another; for instance, require that integrated service categories be provided using comparable differentiated service categories. More elaborately the policy needs for QoS are as follows:

Integrated services using RSVP: RSVP has been devised to explicitly carry and process policy objects together with reservation requests and responses. In its simplest form, policy may be used to control the number, size and the nature of RSVP reservation requests depending on the information in the header of RSVP Path and Resv messages, or the TSpec and RSpec parameters. This use of policy in such an environment allows enterprises to be able to police QoS requests on a per- flow, per-user or per-application basis. However, a number of interesting and important forms of policy may only be expressed using policy objects carried with RSVP signaling messages These include objects that identify and authenticate users, applications or hosts, define the relative (reservation) priorities, carry accounting and charging information, etc.

Proxy RSVP: This refers to the use of policy to control the establishment of RSVP tunnels between routers or other intermediate devices within the network, and the use of these QoS tunnels by traffic flowing through these intermediate devices. There are three common proxy instances: RSVP tunnels for best effort traffic, RSVP aggregation, i.e., combining multiple RSVP tunnels into one, and Other QoS protocol to RSVP translation.

Differentiated services secured through provisioning: This includes the case of using policy to specify and control DiffServ within a domain, and, in an inter-domain scenario, control bilateral agreements across peer network boundaries. In such cases, policies are used to map across the two domain specific semantics, and enforce access control restrictions, such as ensuring that the amount of in-profile traffic is within the specified contractual limits.

Multiple QoS Protocols Tunneling through DiffServ: RSVP or other QoS protocols may be used within a domain, being mapped onto differentiated services across domains. In such cases, policies are needed at the domain boundary to translate between signaled and differentiated service semantics, to enforce traffic monitoring and to exercise access control over network resources.

2.1.1. Policy Requirements for Security

IPSec [14] is a suite of protocols standardized for the purpose of ensuring secrecy and trust across the public internet. It is a complex protocol requiring participating hosts to negotiate a number of configuration parameters. The main issues that arise in IPSec regulation are outlined below:

1. End-to-End security configuration: In the interests of corporate policy, enhanced security requirements or ease of operation, administrators may wish to override protocol configuration defaults specified in IPSec documents, using alternative parameters or algorithms. There is a host of parameters including ISAKMP/Oakley exchange mode, authentication method, perfect forward secrecy requirement, hashing, authentication and encryption algorithms for the two phases of IPSec, timeouts, etc. Administrators must be able to configure the nature of security based on users, end-hosts, servers being accessed, time of day, path of communication, etc., as part of IPSec policy.

2. Gateway access: Often, multiple security gateways have to be traversed for two end hosts to communicate. Depending on the end host application, a gateway may either deny or permit the connection or require an IPSec tunnel from either the end host or another gateway acting as a IPSec proxy for the end host. Access through the use of gateways must be supported through policy.

3. Intranet access: Firewalls and security gateways are required to selectively admit inbound traffic based on the communication users, hosts, applications, the presence of authentication tickets (certificates), etc.

4. Proxy IPSec: In some cases, end-to-end IPSec may be deemed too burdensome, and administrators may configure firewalls to dispatch specified traffic within secure tunnels across the Internet. In this case, the policy language must be rich enough to allow for the classification of traffic based on end-users, hosts, communication path, time of day, etc.

In addition to policy requirements of IPSec, there are other protocols that have regulatory components. We wish to ensure that policy representation is compatible with the needs of:

- IP Filtering: Schema should be able to describe simple IP filters that may be used to configure access devices.

- IP Address Space Management: Address assignment and translation services provided by DHCP and NAT could also be regulated through policy.

- Network Access Control: Certain users and applications may not access intranets or certain servers/hosts. Their access must be proscribed through policy.

3. Policy : Terminology and Conceptual Framework

So far, we have seen the policy needs of QoS and security protocols. In this section, we develop some basic terms and concepts that will allow a shared understanding of the requirements for a policy infrastructure.

3.1. Conceptual Hierarchy

At the risk of simplifying several intricate features of policy, we present a conceptual model that describes different levels of abstraction at which regulation may be expressed and exercised.

Network Level View

This is a high-level perspective that incorporates topology, connectivity, end-to- end performance objectives, expressed in terms of static and dynamic resources in the network. The human administrator interacts with the network usually through a human friendly language or intuitive representation such as a GUI. Such a representation may also contain aspects of network monitoring, where the current state of the network is represented so that the administrator may ascertain the extent to which policy is enforced in the network. As an instance of a network level view, an administrator may require that all http traffic from server A to managers in campus B be transmitted through a gold- QoS, high-security tunnel with a 2 MBps reservation. The definition of "gold-QoS" and "high-security" may be deterministic or probabilistic, but we assume that these are fairly well defined in-context, and not vague and fuzzy. Note that the network level view is integrally tied to the services being administered. For instance, the network

view presented by a tool used for administering remote access to a corporate campus would be very different from another used to dynamically modify peering agreements between two large ISPs. Consequently, the network view cannot be standardized, and we must look towards a slightly lower level of the hierarchy where standardization efforts may be directed.

Role or Group Level View

A network level policy injunction requires different behaviors at various network nodes depending on their roles within the network. For instance, with respect to the above example, all firewalls and access routers in campus B have a similar role to play. They must be instructed to permit encrypted traffic from server A into campus B. Thus, the network view may be resolved into distinct role or policy group level views, which correspond to the policy objectives and requirements at like network nodes or interfaces. A particular network element may play several roles, and several such elements may play the same role. Each network element is controlled by the policies corresponding to all the roles that it plays. The role level policy view is resolvable into a collection of atomic injunctions called policy rules. Each policy rule corresponds to a statement of the form

If (PolicyRuleCondition) then (Action)

where (PolicyRuleCondition) is expressed in terms of administrative categories such as users, hosts, applications, etc., and (Action) identifies the associated privileges or deserved treatment. A simple policy rule for a firewall in campus B that would allow all IPSec packets from server A would have the condition "Host: Server A; Protocol: IPSec" associated with the action "ALLOW".

Device-specific or Configuration Level View

Each network node has vendor-specific resource allocation mechanisms and packet forwarding paths, and role level rules need to be ultimately translated into device-specific instructions. One aspect of a device specific view is the maintenance of caches, filters or classifiers to be placed in the datapath to identify the treatment associated with individual packets. A second aspect is the

reservation of resources and the delivery policy mandated services. For instance, to ensure that http traffic from server A is assured of bandwidth, it may be necessary to insert a classifier in an access router, and set the WFQ weights appropriately. It must be clear that device architectures, capabilities and configuration vary highly from vendor to vendor, and that there cannot be a common language for representing or unifying such configuration.

3.2. Policy Standardization

In order to evolve a schema, it is necessary to identify the level at which policy is to be standardized. The network level view

of policy is intimately tied with the to the remaining network management infrastructure, and it is unclear what would constitute an intuitive and yet flexible object model. Further, it involves an implicit knowledge of topology, and it is difficult to automate the translation of such a view to the device level. Consequently, the network level view is best left to different network management vendors to design and customize. At the other extreme, the device level view is too vendor-specific and scales poorly to large networks. In the rest of this document, we assume that role level views are sought to be standardized in schema, and that these are composed of "policy rules".

Given that standardization effort in policy should address policy definitions at the Role level, the next issue is to decide on a language framework to define policies. There are several design considerations and trade-offs to make in this respect.

1. On one hand, we would like a policy definition language to be reasonably human-friendly for ease of definitions and diagnostics. On the other hand, given the diversity of devices (in terms of their processing capabilities) which could act as policy decision points, we would like to keep the language somewhat machine-friendly, i.e., relatively simple to automate the parsing and processing in network elements.

2. An important decision to make is the semantic style of the language, e.g., procedural or declarative.

- The procedural approach would model network behavior that is to be regulated through policy in terms of states and pertinent events. In this model, policy directives are statements that control the state transitions and thereby regulate the network behavior. An example of state is installing or removal of packet classification filters and the appropriate configuration actions for traffic conditioning. Examples of events include device boot-up, packet arrival, etc.

- The declarative approach would simply describe the desired network behavior in terms of certain actions that should happen when specific conditions hold. For example, a policy directive that states that packets matching a specific traffic profile must be conditioned in a certain way is formulated in terms of conditions that describe the traffic profile and actions that describe the traffic conditioning behavior. A policy rule in this approach is written as if (policy condition) then <policy action>

The declarative approach has the benefit of simplicity, and facilitates hiding implementation differences, making it a suitable candidate for the policy definition language standard.

3. It is important to control the complexity of the language specification trading off richness in terms of features for ease of implementation. It is important to acknowledge the collective lack of experience in the field of networking policies and hence avoid the temptation of aiming for "completeness". We should strive to facilitate definition of the common policies that customers require today (e.g. VPN, QoS) and allow migration paths towards supporting complex policies as customer needs and our understanding of networking policies evolves with experience. Specifically, in the context of the declarative style language discussed above, it is important to avoid having full blown predicate calculus as the language as it would render many important problems such as consistency checking and policy decision point algorithms intractable. It is useful to consider a reasonably constrained language from these perspectives.

3.2.1. Policy Categories

Starting with the broad aim that administrators require mechanisms to control access to QoS resources, we are drawn to describe the specific bases or criteria for discrimination in QoS networks. These would include:

1. The end-points of communication: The source and destination IP addresses may be directly obtained from the IP packet, as long as no intermediate proxies that encapsulate packets are involved. Other information, such as source/destination MAC addresses or other layer 2 end-point information may be used to regulate communication.

2. The route or path of communication: The treatment of packets and the availability of reservable resources depend on the route along which the data packets flow. For instance, packets flowing over a dedicated line may require no particular reservation, while the same packets routed over a backup public network would need special treatment. Simplistic routing-based policies may be described based on the incoming or outgoing interfaces of devices at which the policy is enforced.

3. Communicating users or groups: The identity and organizational status of people involved in communication plays a large role in determining their access to network resources. Rich and versatile policy solutions are made possible by the availability of this

information is available within the network, for instance, using signaling protocols such as RSVP.

4. Application information: The characteristics of the particular application generating traffic, whether it has real-time requirements, for instance, will determine the quality and nature of resources allocated to the traffic. While some such information may be deduced from the port and protocol numbers in IP packets, a more comprehensive solution that does not involve laborious content inspection and state maintenance is possible only if the traffic generating host is involved in signaling application requirements.

5. Dynamic Network Characteristics: It is often useful to take the availability of resources in the network, or their scarcity,

into account while allowing or denying the use of resources by a particular flow or group of flows.

6. Accounting information: The ability to base resource access decisions on the availability of credits or tokens is seen as an important application of policy.

3.3. Processing Policy Rules

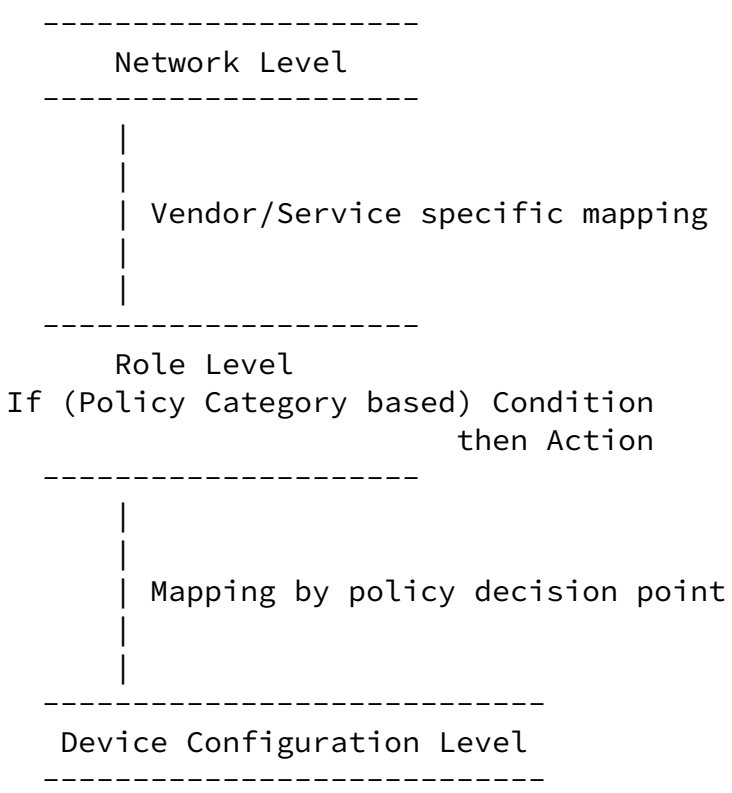


Figure 1: Hierarchy of Abstractions

If we conceptualize policy control in terms of an infrastructure designed to translate administrative intentions into transformations effected on packet streams, then its complexity arises from the need for mechanisms that gather spatially dispersed, dynamic network information and process them to make policy decisions. Consider a policy that allows traffic from certain users to be granted preferential QoS treatment. The complexity of implementing such

a policy depends on a variety of factors. On a policy capable end-host, it may be relatively easy to map the login id of the user to their name, and based on a table lookup, mark packets with the ToS pattern that connotes preferential service in the rest of the provisioned network. In the fortuitous case that users are tied to their workstations, network nodes may access a table that maps users to host addresses, and then use source or destination addresses in packet headers to grant preferential QoS treatment. Or a QoS protocol such as RSVP may be used to carry user information as policy objects, used to deny or grant bandwidth requests. Or, in the case of users dialing up from home, user information may be obtained from a protocol such as Radius, and then used by the access router to mark ToS bits of the packet stream. In any event, policies are expressed in terms of categories such as users, hosts, applications, account balances, etc. Network elements that handle packets need to (a) associate the information in the packet header with the policy category, (b) evaluate the policy conditions to see if they hold, and (c) carry out appropriate operations on the packet.

3.3.1. Associating policy categories with packet information

The mapping of policy categories to header information present in data packets may be static or dynamic; and may be made in a variety of different ways.

- If policy categories are identical to, or can be immediately deduced from data packets, the mapping of high level policy to enforcement is direct and static. An simple example of this is when policy is expressed in terms of source IP addresses. Direct mechanisms do not require state.

- Sometimes, is suffices to refer to lookup tables for resolving policy categories into packet header based conditions. The example is when users are tied to host addresses, but for administrative convenience policies are expressed in terms of users than host addresses. In this case the mapping is static, but indirect.

- Policy categories may be obtained in context, through state that has been established in the operational environment. This

is a very important case, as with user names or application information that can be obtained directly at an end host.

- The mapping between policy categories and header fields may be dynamic, in which case intermediary packets are used to communicate mapping information. For instance, policy category information may be transmitted by signaling protocols, and used to control the associated data stream. One example is the use of RSVP policy objects to ascertain the identity of a dial-up user in order to accept or deny a reservation. Note that this identity may now be used to put the data stream in a secure (proxy) IPSec tunnel. This latter use of information presented through one protocol to deliver a different service (cross-service policy function) considerably expands the utility of policy. Another instance of the use of intermediaries is when dynamic port number information (say for FTP data) is obtained by examining the content of signaling packets.

- In addition to obtaining information directly and through intermediary packets, a network device may contact a centralized network location to obtain information on the policy category a particular flow belongs to. An example of such indirect methods is when IP address to FQDN information is obtained from a DHCP server, triggered perhaps by an unfamiliar packet stream in an access router

3.3.2. Evaluating the Policy Condition

Once the policy categories associated with a packet or packet stream have been ascertained, any policy condition involving those categories must be evaluated. Typically, conditions involve checking set membership -- does "John Doe" belong to the group "Pseudonym-users", or is a certain account balance positive. Now, this stage of policy processing is relatively straightforward if the condition or relation being evaluated changes slowly or never. User and host groups are good examples of such stable policy conditions. However, certain other policy categories, such as account balances, are volatile and it is customary to use specialized servers and protocols to track their state. So the evaluation of the condition is outsourced or specialized information solicited from an external entity. An extreme example of spatially dispersed, volatile information is the evaluation of conditions based on network congestion. (Such a condition would, however, need precise semantics -- where is this state measured and how -- to be meaningfully implemented.) In this case, information regarding packet losses, queue lengths or delays must be solicited from multiple network elements and combined in order to evaluate the policy condition.

3.3.3. Executing Policy Actions

There is tremendous diversity of capabilities and execution environments across network devices, and the ambitious goal of controlling them through a policy infrastructure has to strike the right tradeoff among design objectives. On one hand, it is important for policy actions to be clearly specified directives that different machines would interpret and execute in a comparable manner, i.e., policy must be implementable uniformly across the network. On the other hand, policy cannot be specified at the machine instruction or configuration levels, even though this would make for precise control. One reasonable compromise is to suggest that policy should control standard protocols and behaviors; and not be used to define new protocols. This does not preclude the use of policy as a glue to define new services by configuring and combining existing mechanisms, but it does move away from defining state machines of arbitrary complexity. In short, policy actions should parameterize existing protocols or behaviors.

4. A Simple Policy Architecture

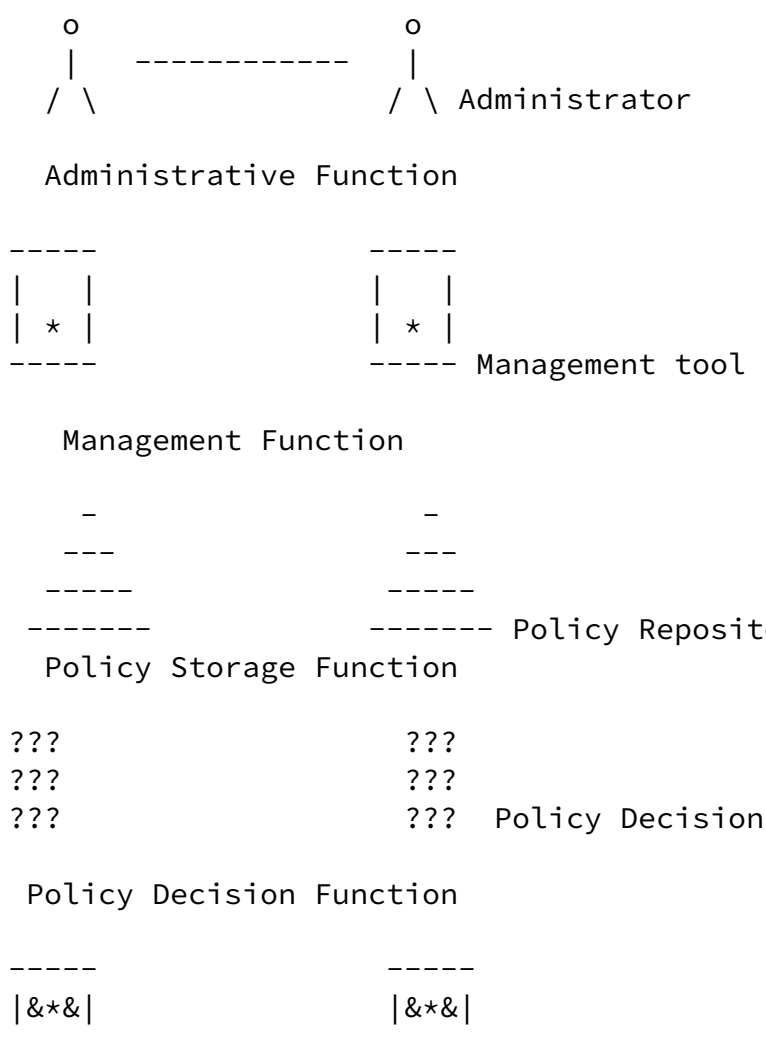
Considering the great heterogeneity of large networks -- varying device processing and storage capabilities, the differential availability of information regarding policy categories, as well as different policy requirements, control mechanisms and network protocols -- it is improbable that an entire policy infrastructure can be based on one single design. In keeping with the more restricted aim of this framework document of unifying policy representation, we first address architectural requirements for the creation, storage and communication of policy rules.

The administrator must be able to view and express requirements at the network level. At a minimum this requires a management tool that is capable of creating policy rules and storing them in the common format.

The policy repository is at the core of the architecture for unified policy representation. It acts as the memory of the network, and stores policy rules for easy retrieval. As policy rules are expressed in terms of relatively static categories, and the frequency of information retrieval is expected to far exceed update frequency, a directory is well suited to be a repository.

The policy enforcement function (PEF) is the functional entity in the data path that delivers network resources to packets. For example, in terms of QoS services, the enforcement function may perform policing, buffer management scheduling, etc. The PEP queries the

policy decision function (PDF) regarding specific actions that are to be applied in conditioning the packet stream. Thus, the PDP performs the key functions of mapping policy categories to information in data packet headers, and of determining which policy rules after evaluating the conditions. Note that the PDP is entity that brings together static rules and dynamic information required for their enforcement.



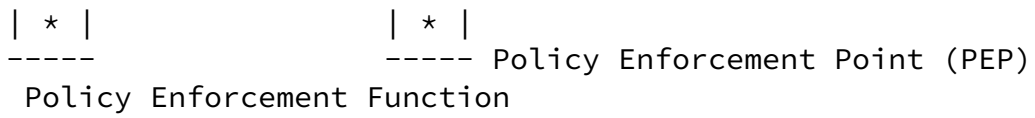


Figure 2
Policy Functional Model

4.1. The Role of Existing or Proposed Standard Protocols

As shown in the above figure, each of the functions described above may be implemented in a distributed or even replicated manner.

The management function may be spread across multiple management tools, or several tools may simultaneously be used to populate the repository. Likewise, the policy repository may be replicated or distributed across several directories. In many cases, the policy decision function may be substantially located at a specialized policy server that communicates with multiple network devices at which policy enforcement occurs. The policy decision function may also be split across several policy decision points which collaborate in making decisions. If multiple functions are not co-located, or if the same function is distributed across multiple devices, it is necessary to define standardized protocols for communication between different boxes. There are a number of existing and proposed protocols and mechanisms that may be usefully employed to realize the above architecture, which we discuss in this section. In the next section we discuss enhancements to the basic model to provide enhanced functionality.

Directories make good policy repositories, and are commonly accessed using LDAP, a lightweight protocol used extensively to communicate information about users, hosts and applications. Traditional databases, using a language such as SQL provide greater functionality in the repository, at the cost of being more heavyweight. While there are many choices of protocols for directory/database access from the management tool and PDF, LDAP appears to be favored by a number of vendors and users for its ubiquity, versatility and flexibility. LDAP schemas are versatile and allow considerable flexibility in choice of back-end directory management. Further, the LDAP client-server protocol is widely implemented and used for supporting a wide range of directory enabled applications. However,

there are a number of shortcomings of LDAP that must be clearly understood by implementers. (Para on shortcomings of LDAP to be added -- asynchronous notification, replication support, security, referential integrity, support for "templates", limitations of query language.)

The RAP working group has recently moved to standardize the COPS [5] protocol for outsourcing RSVP decision making from a PEF to PDF. This protocol is modular and may be extended for other policy outsourcing requirements. More recently, the Diameter protocol has been proposed for outsourcing authentication and security decision making. Extensions have been proposed to enhance Diameter to handle QoS outsourcing as well. Another outsourcing protocol that is currently proposed is the Security Policy Protocol in the IPsec WG.

SNMP is likely to continue as an integral component of the network management infrastructure. It can play many roles in implementing the above policy architecture. SNMP can be used at the PDF and PEF to record errors, policy usage and notify administrators of unusual

occurrences within the network. It may be used by the management tool to notify the PDF of a policy update (as such an asynchronous notification feature is not yet available with LDAP). A number of other protocols such as telnet or SSH with CLI (Command Line Interface) and TFTP (could be combined with IPsec) are complementary to SNMP or COPS.

4.2. Enhanced Functionality

The basic functions described above may be extended by vendors seeking to enhance flexibility, ease of use, scalability or security. In this regard, there are multiple logical functions at which the same enhancement may be usefully provided. For instance, it is conceivable that some form of "sanity checking" for policy rules is part of each of the above pieces. The "correct" placement of extended functionality depends on access to information. For instance, if there are multiple administrators in a network, but one physical directory server, then the repository function is perhaps best enhanced to handle network level sanity checking. Below, we describe some obvious enhancements to each of the functions, with the caveat that the list is neither necessary nor exhaustive.

Apart from being a simple input device for policy definition, management tools can use intuitive user interfaces to bring together network topology, connectivity and performance, with a language for expressing policy categories and goals. In this case, management tools function as translators of the network level view into a role level view. Such translation ensures that policy is uniform across the network, and reduces the need for expensive uniformity checks. (As an example for the need for uniformity, consider two firewalls in peer campuses that cannot communicate, being configured to use different IPSec encryption methods for the same traffic.) Further, management tools can perform several sanity checks on rules. They ensure that policy objects are syntactically correct, that objects referred to in policy definition exist, that policies are uniform across the network, and that multiple rules defined for each role are consistent. To the extent that the management tool has access to network device functionality and resource availability information, these may be included in consistency checking.

Policy repositories vary in their ability to be reliable, secure and distributed. They support query languages of differing complexity. Repositories enhanced to provide native support for policy categories would be natural locations to optimize a variety of policy consistency and sanity checks.

The policy decision function is the merging and processing point for static and dynamic information. A variety of protocols different information exchange protocols may be supported here, especially those for gathering volatile data required for decision making. PDPs may talk to different servers and devices in the network, or even to other PDPs, in order to collect billing and accounting information, group membership, addressing and routing information, as well as resource availability and usage. In addition, a PDF may be aware of the resources and capabilities of the PEPs it is connected to, and hence able to flag policy rules that cannot be applied because of device conditions. Some of the shortcomings of existing mechanisms, such as the lack of asynchronous notification in LDAP, may be addressed by defining specialized protocols between functional entities.

5. Policy Representation Requirements

The directory provides a convenient repository of the resource regulation policies, which may be accessed by a number of different policy decision points. The following considerations must guide the evolution of standardized means of representing policy (see [1] for instance,) also known as policy schema:

- The schema must be defined at the role or policy group level view, and not at the network level nor at the level of device configuration.

- It is assumed that the policy repository will not store volatile information required for policy decision making. Hence, the schema must be devised to express administrative intent in terms of relatively static categories. The policy decision point will map policy categories to identify packet streams that need to be regulated.

- The schema definition should be generic enough to support a wide range of resource control environments.

- Policy schema must be used to configure and control standardized protocols and services, and not as a low-level programming language for networking devices. These schema must support the needs of QoS and security as discussed earlier in this document.

- The schema shall be designed to be extensible to the needs of other network services.

- From this perspective, it is desirable that the schema facilitates definition of a wide range of policies varying in

their complexity. Simple policies (the common case) should be easy to specify, and there should be sufficient hooks to define sophisticated policies within the schema. Using the language analogy, an administrator's ability to define complex resource regulation policies should not be limited by the structure of the schema, although it may be limited by the available implementation of the policy enforcement environment.

- The schema should facilitate simple addition and deletion of new rules, automated checks for rule ambiguities, and allow for diverse methods (varying in efficiency and ease of implementation) to be employed in the policy decision entity to search through rules.

- While compactness of representation is of concern, it is subordinate to the needs of expressiveness and extensibility listed above.

6. Security Considerations

There are two potential security considerations, both of which may be addressed through standards compliant mechanisms. The first is the unauthorized access to read or change policy rules and related objects in the directory repository. The schema in this document SHOULD be used in conjunction with an LDAP access control mechanisms, see for instance [12]. The second exposure for violation of security lies in the communication between policy decision point and the directory repository. Such communication SHOULD be secured, with both ends mutually authenticated using SSL/TLS or IPSec.

Acknowledgments

Thanks to Partha Bhattacharya and Skip Booth for useful discussion and suggestions in this problem space. We also thank many others who have read and commented on this draft in various forms.

References

[1] J. Strassner and E. Ellesson, Policy Framework Core Information Model", draft-ietf-policy-core-schema-01.txt, February 1999.

[2] D. Piper, ``The Internet IP Security Domain Of Interpretation for ISAKMP'', draft-ietf-ipsec-doi-07

[3] Bhattacharya, P., and R. Adams, W. Dixon, R. Pereira, R. Rajan,

"An LDAP Schema for Configuration and Administration of IPSec based Virtual Private Networks (VPNs)", Internet-Draft work in progress, October 1998

[4] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification. RFC2205, Sept. 1997.

[5] S. Herzog, A. Sastry, R. Rajan, R. Cohen, J. Boyle, and D. Durham, The COPS (Common Open Policy Service) Protocol Internet-Draft, draft-ietf-rap-cops-00.txt, Jan. 1998.

[6] W. Yeong, T. Howes and S. Kille, Lightweight Directory Access Protocol, RFC1777, Mar. 1995.

[7] K. Nichols and S. Blake, Differentiated Services Operational Model and Definitions, Internet-Draft, draft-nichols-dsopdef-00.txt, Feb. 1998.

[8] R. Yavatkar, R. Guerin and D. Pendarakis, A Framework for Policy-based Admission Control Internet Draft, draft-ietf-rap-framework-00.txt, Nov. 1997.

[9] S. Judd and J. Strassner, Directory Enabled Networks - Information Model and Base Schema - Draft v3.0c5 DEN Specifications, Sep. 1998.

[10] P. Bhattacharya et. al., An LDAP Schema for Configuration and Administration of IPSec-based Virtual Private Networks, Internet Draft, draft-ietf-ipsec-policy-ldapschema-00.txt, Oct. 1998.

[11] Desktop Management Task Force, Common Information Model (CIM) Specification, Version 2.0, Mar. 1998.

[12] E. Stokes, D. Byrne, B. Blakeley and P. Behera, Access Control Requirements for LDAP, Internet Draft, Sep. 1998.

[13] J. Strassner and E. Ellesson, Terminology for describing network policy and services Internet draft, draft-strassner-policy-terms-00.txt Aug. 1998.

[14] S. Kent and R. Atkinson Security Architecture for the Internet Protocol RFC 2401 Nov. 1998.

AUTHOR'S ADDRESS

Raju Rajan AT&T Labs Research 180 Park Avenue, PO Box 971 Florham Park, NJ 07932 email: rajan@research.att.com

Sanjay Kamat IBM T. J. Watson Research Center 30 Saw Mill River Road Hawthorne, NY 10532 email: sanjay@watson.ibm.com

Full Copyright Statement