# Security for Internet QoS

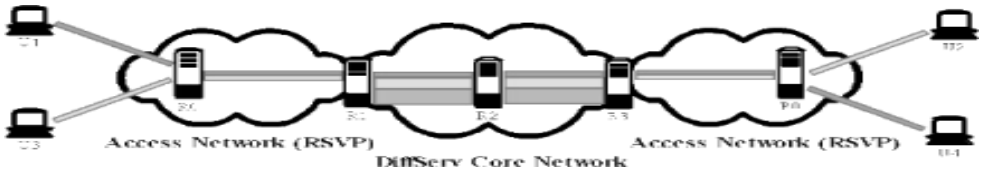
**Vincent Law 014-68-0083**
**Courtesy of UC Davis Computer Science Department ECS 289I Spring 2002**

**Abstracts**

**Internet QoS becomes a significant feature from Internet Service Provider because of limited resource availability. Without security, Internet QoS provision becomes a meaningless offer to the customers. This article outlines the Internet Security protocols which affects QoS. It divides the level of security into four areas and illustrates how each area works with QoS and describes several proposed or existing protocols that will enhance QoS security in that area. It first summarizes a proposal for secure routing protocols for end-to-end security, and how Quality of Protection (QoP) under the proposed ISCP can help end-to-end security with heterogeneous domains. Afterwards it describes various works on secure QoS, including two models for RSVP protection namely Resource Pricing and Selective Digital Signature with Conflict Detection (SDS/CD), secure QoS forwarding, and BGP/MPLS. It follows by introducing a secure network infrastructure protocol that enables parties having different security policies to coordinate and communicate with each other during security association negotiation. Then it gives an overview of policy specifications on security and management of general network components and their importance to the network in supporting secure Internet QoS. Finally it summarizes how these proposals comply with general common security policies. It intends to provide a report on various security researches in different areas to help make Internet QoS more effective.**

## Introduction

The addition of QoS on IP networks means the integration support of broad range of applications, such as voice and video, real-time distributed simulation and control, collections of data from sensors, etc. It also means the need of network support at both the packet level and connection level. Packet level involves scheduling, multiplexing, traffic shaping or smoothing, policing, packet dropping, and congestion control, while connection level includes signaling, admission control, routing, and resource reservation. A typical network has access networks, which have lower bandwidth and traffic, and core networks, which have higher bandwidth and heavier traffic. Integrated Service using RSVP is mostly used as the means for QoS in the access networks, while Differentiated Service is mostly used in core networks to reserve resources for aggregation of flows in order to guarantee QoS.




*Security needs for QoS*

Internet security awareness leads to the identification of four top-level security areas: *end-system security*, *end-to-end security*, *secure Quality of Service (QoS)*, and *secure network infrastructure*. *End-system security* is generally about firewall and other measures that will protect the end-systems or single host. *End-to-end security* is usually about the mechanisms used in the secure transmissions of data between two or more systems to maintain confidentiality, integrity, and authentication, such as using cryptography. *Secure QoS* involves the authentication and authorization of users requesting privileged network services in order to protect resources from theft or stolen traffic, which can lead to denial of service (DoS), caused by an unauthorized user. *Secure network infrastructure* is the prevention of the network infrastructure from being vulnerable so that attackers cannot take advantage of the flaws of the network. The last three areas are related to Internet QoS. This article serves as an amalgamation of security proposals and models currently applying on one of these three areas, and can serve as a reference in selections of security deployment for Internet QoS.

**End-to-End Security using Secure routing protocols**

People are not actively looking at the significance of secure routing until recently. Most routing protocols by design deal with single-network failures only, such as links down or nodes crashing etc. The vulnerabilities of tricked data traffic through routers have been overlooked. Threats can be classified as either external or internal, where external threats are from outside intruders who aim to disrupt the normal routing protocol operations and internal threats are from protocol participants whose purpose is to abuse the routing information in exposure.

*Routing protocol threats*

Main threats, which can come from either an outside intruder or a compromised intermediate system, to a routing domain include:

- *breaking neighbor relationship* by changing routing updates or intercepting traffic
- *replay attack* by retransmitting obsolete or duplicate messages to confuse the intermediate system which can result to making incorrect routing decision or denial of service
- *masquerading* by mimicking itself to a legal member or compromising or manipulating the authentication system
- *passive listening and traffic analysis* by monitoring possible confidential routing information or operation

*Sample routing attacks*

Known attacks on the routing protocols are either based on the RIP protocol or Open Shortest Path First (OSPF) routing protocol, which aims to be the future choice of intra-domain routing protocol and may replace RIP. Here are the brief descriptions on these sample attacks:

- *Black Hole Attack*: a compromised router broadcasting favorable link state and cost information as updates to the neighbors using distance vector routing protocol like RIP so that the neighbors will think that this router has the shortest path after recomputation
- *Table Overflow Attack*: a compromised autonomous system boundary router, based on OSPF, generating junk link state advertisements (LSAs) flooding to every router in the autonomous system without validation mechanism resulting in the failure of the routing protocol to successfully install new network entries and, worse, crashing the routers
- *Age Field Attack*: also known as *MaxAge attack*, LSA age fields are changed to MaxAge, the upper bound of LSA's age enabling the routing information database to purge that specific LSA, causing unnecessary flooding and refreshment which result in unnecessary bandwidth consumption, inconsistent routing information database, and incorrect routing
- *Sequence Number Attack*: an obsolete sequence number is generated, due mostly to implementation bugs such as improper sequence wrap-around process, for the LSA so that it is always greater, and newer by OSPF definition, to any future possible incoming LSA which will then be rejected

*Three-model secure routing protocol framework*

Dr. Felix Wu and his peers at North Carolina State University proposed and studied a framework for secure routing protocol containing three parts. *Topology model* defines topology relationship among various routing protocols. *Information model* defines the flow information inside an *intermediate system* participating in intra- and/or inter-domain routing. *Operation model* defines the general routing protocol operation procedures and reflects the information flow among intermediate systems. Security routing usually focuses on both the information model and operation model.

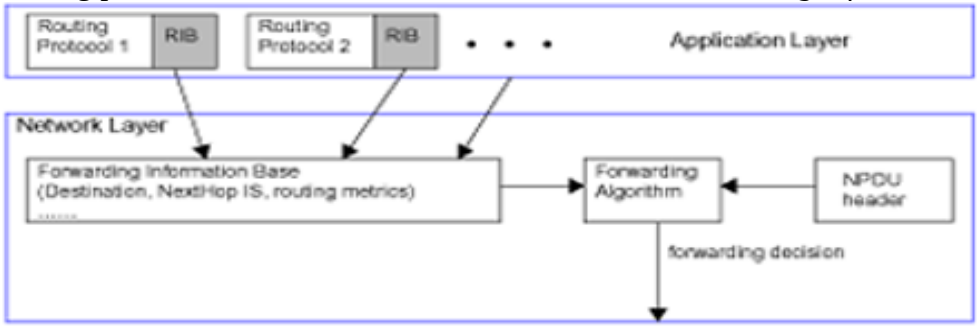
*Topology Model*

In order for routing protocols to deal with diverse network topologies, we commonly model the network topologies into two levels: *intra-domain* and *inter-domain*. Intra-domain routing handles routing procedures among single provider, who shares resources with other organizations, or single subscriber, who uses resources from other organizations. Inter-domain routing handles routing procedures spanning multiple providers and/or subscribers. An *administrative domain*, formed by a single provider or subscriber that spans a contiguous segment of an internet topology, provides inter-domain routing a convenient model to allow resource-contributing organizations to establish boundaries, such as firewalls, to protect and control access to their resources. A

connected intermediate systems set participating in single particular intra-domain routing protocol instance forms a *routing domain*. Since a single administration can have more than one intra-domain routing protocols, an administrative domain can have several different routing domains, and a routing domain can also be furthered divided into hierarchical routing areas. An *end system* is a host system usually not taking part in the routing and is usually connected to an intra-domain-routing-only intermediate system. A normal intermediate system can only talk with its intermediate system neighbor within the same domain, i.e. intra-domain only. The one performing both intra- and inter-domain routing is a *boundary intermediate system*. Without allowing overlapping administrative domains, an intermediate system can only belong to only one administrative domain.

*Information Model*

When an intermediate system receives a protocol data unit in a connectionless network, it makes its forwarding information based on the header in the protocol data unit and a forwarding table called the *forwarding information base (FIB)*, which stores the destination information and the routing metrics and characteristics along the route for its suitability evaluation. While the forwarding, and therefore FIB too, is handled in the network layer, the routing protocol governs the policy. The intermediate system constructs FIB using the information gathered from its participation in one or more routing protocols, each of which maintains an individual *routing information base (RIB)*.




*Operation Model*

A routing protocol is separated, based on the reflections of core procedures, into five components:

- *neighboring acquisition* defining how intermediate system or boundary intermediate system acquires neighbor information
- *neighbor reachability* defining neighbor relationship maintenance with previously acquired neighbors
- *routing information exchange* defining how and what to exchange routing information among intermediate systems based on these three pieces centered around RIB: *neighbor-in-RIB* storing information receiving from neighbor, *neighbor-out-RIB* storing information sending to neighbors, and *local RIB* storing necessary routing information excluding whole transit traffic, where the RIB needs either periodic update or event-driven polling or both to maintain its freshness dynamically
- *route generation and selection* determining what goes to FIB based on route selection algorithm in local RIB
- *neighbor relation termination* defining how to terminate a neighbor relationship

*Security-related characteristics of routing protocols*

Different routing protocols have different degree of immunity, with three favorable security characteristics: *self-stabilization* being able to return the disrupted network back to normal operation without human intervention within a reasonable time as long as the faulty hardware is either disconnected or repaired, *Byzantine robustness* in which the network can continue to operate properly even with the presence of *Byzantine failures* – failures not by cease operation but by performing arbitrarily – in some nodes, *fault detection* combining with proper security management and Byzantine robustness to enable the network to detect, identify and isolate faults without human intervention.

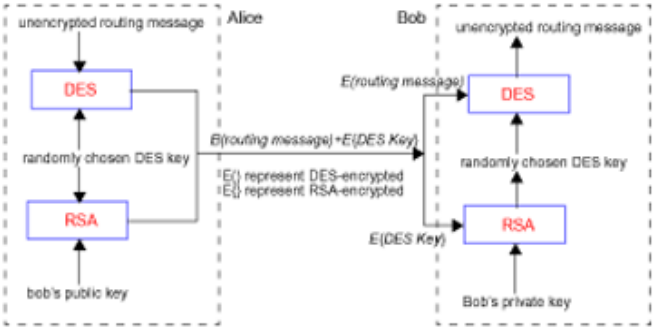
*Secure Routing Protocols Requirements*

The above attacks are all taking advantages of the lack of authenticity, integrity, and confidentiality. So these features as the contexts of secure services are the focus for the protocols in areas such as accessibility, intended usage, and network connectivity. The source and most, if not all, of the routing information should be authenticated. Vulnerabilities and possible threats should be identified. Security requirements should be determined based on both current and anticipated environments. Cryptographic techniques and authentication mechanisms should be incorporated to address both confidentiality and integrity. Current authentication techniques include *password*, *message digest signature*, and *public-key-based digital signature*. Password scheme simply requires the password provision to be carried in the packet header and checked by the neighbors, meaning the neighbors have to keep the passwords, but it does not have other authentication or integrity checking abilities. Message digest signature uses a one-way hash function, like MD5, to generate a digest as a signature using a secret key, and it allows hop-by-hop, but not end-to-end, authentication and integrity verifications against some attacks, such as spoofing especially when it is using BGP, etc. Public-key-based digital signature signs a message with the sender's private key and verified by the

sender's public key stored in the receiver(s). It is ideal enough because of its support in end-to-end authentication and integrity, but it suffers from algorithm performance and patent complications in exporting the key due to some key size limitations.

*Digital Envelope*

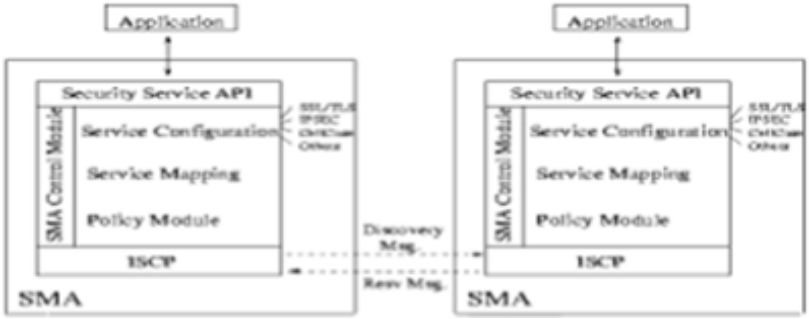
*Digital Envelope* is a new proposed technique making good use of the security features of public key model without having to suffer much in performance. It uses a key for a faster encryption scheme to encrypt the message, and the key itself will then be encrypted using public key scheme before both encrypted data will be combined to be sent to the destination. To recover the message, the destination firsts use public key scheme to decrypt and recover the secret key then perform the fast scheme to decrypt the message using the recovered key. Since the key size is most of the time much smaller than the message itself, using public key scheme on the secret key during the communication will be faster than sending the whole message using public key scheme. The use of keys in these schemes means that key management and distribution scheme, which can be either manual, automatic using some embedded mechanisms, or relying on existing schemes in the layers such as ISKAMP, has to be incorporated in each router too. Privileged information like shared keys should only be distributed to involved portions in the network.




**Quality of Protection (QoP)**

Applying security in QoS routing means that there can be more than one security modules handling end-to-end security service. As a result, the security environment becomes heterogeneous, and sometimes even overlapping significantly. To conveniently manage and support all these modules, an "agent" is a nice approach in efficient distributed management for end-to-end security service. Dr. Wu and his peers further proposed an *Inter-Domain Security Management Agent Coordination Protocol (ISCP)* to provide good security capability and policy information communications, a.k.a. *Quality of Protection (QoP)*, among the security management agents in each policy domain, along with other classical software engineering features such as scalability, interoperability, and extensibility, to support Internet QoS.

*Security Management Agents (SMA)*



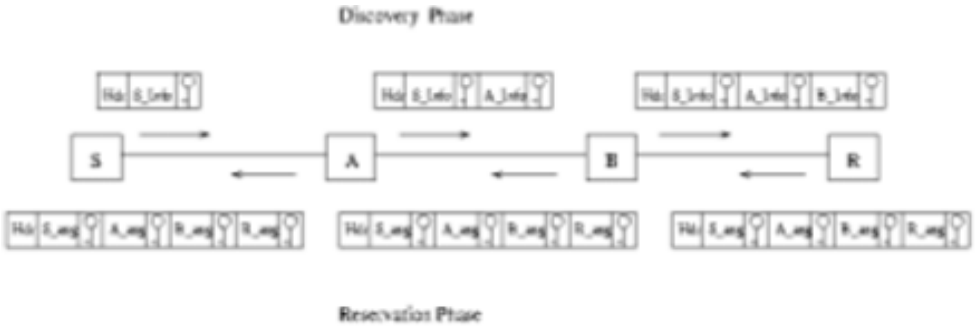

Two parties can achieve security communications upon the establishment of *security associations (SA)* after their mutual agreements during negotiations. However, if the parties do not have compatible security capabilities on either hardware or software, they will not be able to communicate and negotiate, and SA cannot be established. Even in the case when both parties have compatible security services, if the order is handled differently, this may result in the drops of all the data packets, i.e. DoS. For example, in tunnel mode if two firewalls have overlapped policies but with different SA order on the peering gateways, they could seriously block legitimate communication due to lack of mandatory SAs. A special case is when both hosts establish an SA using IPSec/ESP without awareness that a firewall on the path does not allow encrypted packets to pass. *Security Management Agent (SMA)* sits in management plane of any SMA-enabled node, which can be a switch, router, or gateway, resulting in DoS. SA is authorized to configure or re-configure various local security mechanisms at all protocol layers and is responsible for coordinating all network security-related activities.

*ISCP phases and objectives*

ISCP provides the transport function for security service negotiation and reservation in QoP implementations. The information in the messages ISCP transported during the negotiations includes service request, capabilities, SMAs' policies, request-related security configuration and assignment, and maintenance. Security context establishment is split into two phases: *discovery* and *reservation*. The scheme is adopting the soft-state approach, so it is very similar to RSVP, except that security context is the center of attention in ISCP. ISCP design has the following objectives:

- Efficiency during the two-phase process for end-to-end security context establishment
- Integration of QoP with QoS by adopting RSVP's scheme in ISCP
- Prevention of insider attacks, i.e. attacks from RSVP-enabled nodes
- Optimal operational efficiency and scalability

*ISCP messages*




During the phases, several types of ISCP messages are applied. They are as follows:

- In discovery phase, the sender first encrypts the *discovery message* using a locally generated secret key, followed by encrypting the key with the receiver's public key (to ensure only the receiver can receive) and the sender's private key (as a digital signature to prove that the message is from the sender) before sending the encrypted message and key downstream to the receiver in order to request requirements along the communication path, where any SMA node along the path having supported capability and service based on the security policy module will append additional security capability and policy, including appending encrypted verification using the node's keys, to the message, which will eventually reach the receiver who will decrypt the secret key and the message before analyzing all attached capabilities and policies in order to select an optimal set of SAs based on some mapping and configuration modules.
- The receiver now invokes the reservation phase by sending a *security reservation message* containing node-by-node assignment information along the reverse path of the discovery message, allowing each node to pick up its own assignment upon receiving the message and make corresponding security service setup or reservation before sending the confirmation message to the sender. Each assignment will be encrypted by the corresponding node's secret key (so that the corresponding node can decrypt its own assignment) and the secret key will then be encrypted using the sender's secret key (so that only the sender can receive all the secret key information for all the nodes).
- Those which participate in the SA establishment will send the sender the *Confirmation message* upon the completion of SA setup.
- Finally the sender notifies all nodes upon successful confirmations by sending the receiver the *ContextReady message* and then the receiver notifies the sender upon its being ready to start data flow under this security context by sending the sender the *ContextReadyAct* message.
- Any error during the transmission will be reported to the sender by issuing the *Error message*.
- Periodic updates by *refreshing discovery and security message* will be sent using the aforementioned distributed secret keys to keep the freshness and adapt dynamically to the route changes and possible intrusion events etc.
- All context and maintained state information will be deleted when a *teardown message* is sent from the sender to the receiver along the same security-context path at the end of data transmission. All contexts and state information maintained during the session will be deleted.

*ISCP Message Format Overview*

All the messages are having the following parts:

- Common header, which has field such as message type, source and destination address, security context handle, sequence number (to prevent replay attack), and checksum, which is used to prevent any section cutoff attack during the node information appending part of the message and is defined by Checksum(i) = MD5(Checksum(i – 1), Secret Key (i), non-mutable part of current ISCP message) where node i >= 0, Checksum(0) = 0 and the checksum is updated in every node
- Message body, which is different according to the message type.
    - Discovery message: security service request and capability/policy information of all SA nodes along the path, where capability information tells what security mechanisms supported and policy information tells the transfer policy and IPSec policy
    - Reservation message: each SMA node's security assignment
    - Soft-state and refreshing message: since it is only for refreshing, it has the same format as the discovery and reservation message
    - Error reporting message: error code and error value
    - Confirmation, ContextReady, ContextReadyAct or Teardown message: since it is just a notification, there is no need to have a message body as the header can already tell what kind of notification message it is

*ISCP compared with RSVP*

Since ISCP is working as end-to-end security mechanism, it only needs to be installed in the border network devices and security gateways or firewalls. The interior routers do not need to add its installation because all the routers do during the whole ISCP scheme is to cooperate and provide the corresponding resource information if they are ISCP-supported nodes or to just transparently forward the messages if they are not ISCP-supported. While RSVP has to maintain the per-flow state in each router, in ISCP the interior routers do not need to participate and have no obligation to process per-flow state since in ISCP state means domain-wise security context and the information has already been provided by the border devices or firewalls. Therefore, ISCP is more scalable than RSVP.

*Session Control Table*

The sessions' states are created and maintained in *Session Control Table (SCT)*, which is maintained dynamically by *ISCP daemon*, has attributes such as

- Unique session ID
- Original service request
- Assignment and role of the node
- Pointers to control blocks of local capabilities servicing the request
- Last path time and last reservation initiation or refreshing time
- Previous hop address (for reservation message)

These attributes are stored in SMA control module's state controller. ISCP just simply hands the collected information to state controller for storing and maintenance purposes.

*ISCP's future*

Dr. Wu also mentioned that with the relatively more scalable feature of ISCP, more future researches will be focusing on a more scalable design and reducing the setup overhead. Possible approaches are taking advantage of existing SAs and aggregating refreshing messages etc. Another possible area of ISCP extension is to apply this model on multicast environment by merging the reservation messages.

**Secure QoS Forwarding**

Internet QoS packet flows can cause a new set of security problems. Therefore, it is necessary to authenticate and authorize users asking for those QoS values that are expensive in network resources, and it is also necessary to prevent unauthorized use of these resources and denial-of-service attacks by others. A two-stage security setup process, which can be either dynamic such as by an application or static such as by protocol or remote configuration, moves somewhat away from the pure datagram model, which requires checking and computations in all involved IP packets in the datagram and may be very demanding. In the first stage, the *setup* stage, routers and other network elements establish some state describing how to treat a subsequent packet stream. Most of the current QoS research, such as real-time service, has assumed an explicit setup stage and a classification stage. The setup stage is accomplished using protocols like RSVP, which also specify how to perform the subsequent classification. Then, in the second stage, the *classification* stage, the arriving packets are matched with the correct state information, known as *classes*, before being processed. Secure QoS forwarding involves setup stage security, and it is thus simply an extension to such protocols used in setup stage. To secure the setup process, a setup request needs to be accompanied by user credentials, known as the *high-level identification (HLID)*, that provide a trustworthy assurance that the requester is known and is authorized, like using password or other user-specific authorization, to make the request. While there could be any number of ways to organize the HLIDs, the objective of scaling suggests that a global user naming and authentication framework would be useful. Each packet to drive classification may also carry a *low-level ID (LLID)*, sometimes called a *cookie*.

*Cookie*

In current proposals for IP extensions for QoS, packets are classified based on existing packet fields such as source and destination addresses, ports, and protocol type. Cookie is distinct from the user address because the user privileges are not determined by the address in use. Change in the user's address does not modify the privileges. A packet's cookie acts as a form of tag used by some or all routers along a path to make QoS granting decisions to this packet. It might refer to a data stream between a single source-destination address pair, or more generally a range of data streams. For security

forwarding, IP datagram contains one cookie, which can be used at various network stages to map the packet to a class. The attributes of a cookie should be picked to match as broad a range of requirements as possible, and are summarized as follows:

- Its duration must match both the needs of the security protocol, balancing robustness and efficiency, and the needs of the application, which will have to deal with setup renewal when it expires. A useful end-node facility would be an automated setup renewal service. Besides, it has to deal with the durations for its mutable fields.
- The trust degree must be high enough to meet the most stringent but reasonable requirement.
- The granularity of the cookie structure must permit packet classification into classes fine-grained enough for any network resource selection. Therefore each separate packet stream from an application is expected to have a distinct cookie that there will be little opportunity for aggregating multiple streams under one cookie or one authenticator.

Cookie has to be authenticated in order to prevent theft-of-service or DoS. For performance, cookies have to be validated, but not so rigorously, on at least some selected packets at certain, even though not necessarily all, routers, which should also log the selected packets and the validation results as part of later audit activity. The basic authentication techniques, in terms of computational performance, bandwidth overhead, and effectiveness against various forms of attacks, are below:

- *Digital Signature*, which uses public key cryptography, uses a one-way hash function to compute a digest for a packet and encrypts the digest with the sender's private key associated with the cookie. The encryption part is also known as *signing*. The advantage is that any router can validate the data as long as it has the sender's public key as the authenticator, and it is secure enough because of the difficulty in guessing the private key and the hashed digest. The disadvantage is that the signature process and the validation process are not feasible because of overheads.
- *Sealing* does not use encryption, only uses a one-way hash function by generating a digest, known as a *seal*, for the packet, which is appended with some value making the value itself the secret "key" as the authenticator. All routers, which are trusted by the users, having the secret key can thus authenticate the packet by using the key to recompute the digest and comparing with the one being checked. This technique is faster than digital signature because of less computation overhead, but is considered less secure because any router having the secret key can forge a seal. A modified approach is to use shared secret between only an immediate pair of routers so that each secret does not need distribution and thus reduces the probability to be stolen by the attacker, as long as the downstream router trusts the upstream router as a representative for the cookie. However, there is still vulnerability in this approach because it cannot prevent replay attack.
- *Temporary passwords* attached a short-term secret quantity as a password as the authenticator to the packet header without any further protection. All the packets for a specified cookie will use the same password. The password is independent of the packet, which means it is not a function of the packet. Performance is better than the other two because it requires merely comparisons and the password does

not consume a lot of space in the packet header. However, since this password is visible to any involved router and any equipment along the path, this technique is much more vulnerable than the other two. One adaptation to make this technique safer is to use a sequence number in each of the packets so that intruders may fail to break in because of the difficulty in deleting legitimate packets or the detection of duplication of some packets of same sequence number.

## RSVP protections for Secure QoS

*Resource Pricing*

One approach to make QoS secure is to extend security directly on QoS operation. A proposal suggests to price different users in resource allocations. Currently resource reservation does not require control, which can result in unauthorized users stealing extra resources and cause possible denial of service (DoS) due to lack of available resources for authorized users. In *Resource Pricing*, each user is allocated a budget b and a price p per unit for a certain kind of resource, which means that each user can have a maximum of b/p units of this resource. It is based on the notion of demand-based pricing in which it calculates resource demand based on price, which itself is calculated from demand. Therefore, it is a feedback system. It can reach an equilibrium point to satisfy resource utilization, i.e. total demands equal the supply. It determines fairness type under the situation how a user spends the allocated budget on several requested resources, and it suggests that a *fairness index* of 1.0 means completely fare and 0.99 is appropriate enough. It assumes a more general model of user behavior and traffic, and supports both reserved and dynamic resource pricing. One form of fairness is called *weighted max-min fairness*, in which each user is given a weight w. Each resource computes its own equilibrium price, and the user for a limited resource is allocated the amount for that resource in a total unit of the budget divided by the price for the resource. Then the price that user is charged is the price of the most expensive resource that user requires, and the allocated resource amount is proportional to the weight. Another form of fairness is called *weighted proportional fairness*, which is the form of fairness exhibited by TCP congestion control. Given the relative change for a user between one set of resource prices to another being the allocation amount change divided by the allocation amount, a price assignment is proportional fair if the sum of relative changes for all users between this price assignment and any other price assignment is at most zero. A user is allocated all resources by the amount of the user's budget divided by the price total of all resources requested by the user. A *utility curve* is relating resource allocation to the degree of utility or satisfaction of the resource by the user. An allocation is *equitable* if all users experience the same degree of utility under limited resources, while it becomes *utility-maximizing* if it results in the maximum of the network users' total utility. The speed of achieving equilibrium depends on the number of required iterations and the time needed to distribute prices and measure the demand change. A simulation based on 20 users competing for single resource resulted in only a few iterations required to reach equilibrium. Another simulation based on two groups of users, with one group having budgets twice as much as another, getting traffic of MPEG VBR video traces with price

distributed using ATM RM cells and updated every 10 ms, showed the result that bandwidth utilization was very high, with price distribution consuming only about 1% of data traffic bandwidth, i.e. low overhead, and allocation was always fair. The drawback of resource pricing is that even though utilization, fairness, and resource access are fair, prices, i.e. resource allocations, are not, so it is not suitable for applications which require stable resource allocations. To deal with fluctuated pricing, pricing must be computed to predict demand over long time interval, resulting in reserving resources for use, like the use of RSVP, once being acquired and possibly preventing newer users in accessing the network due to insufficient resources. As a compromise, two-price model adoption is an option, with one set of prices based on stable demands and another based on fluctuating demands, allowing users to request resources from their preferred set of prices. This is similar to the DiffServ model of premium and assured services, and the prices for reserved resources generally becomes higher than that for dynamically-priced resources, while keeping the high utilization and robust equilibrium achievement. To work with the current model of policy-based RSVP, the *signed policy object* is obtained from the authorization server using session initiation protocol as a proof of affordability of a certain price given for the resource and the signed policy object is then included in both the RSVP message and COPS (Common Open Policy Service) message, which uses a message integrity object using a 32-bit sequence number and authentication scheme to protect against replay attack and provide message integrity. The usual RSVP procedures are processed at the supported router that acts as *Policy Enforcement Point (PEP)*, based on the admission control decision in policy server, known as *Policy Domain Point (PDP)*. Upon successful resource reservation, the charge attached in the response message will be propagated back to the authorization server if desired. Since resource reservation depends on price, which depends on the policies applied on the user who requests the resource, it prevents unauthorized users from stealing resources.

IntServ involves resource reservation, like RSVP, and signaling on a per-flow basis. RSVP requires routers-awareness to maintain soft-state information. While it allows precise resource allocation, heterogeneous bandwidth support for multicast, and receiver-initiated reservation, its implementation has a lot of overhead and there are also scalability problems. RSVP/IntServ is also very weak securely and is especially DoS-attack-prone, because it involves trusted parties and it is inefficient in resource allocations and releases due to lack of central control. One main purpose of the attackers is to cause denial of service in network, a.k.a. Denial of Network Service (DoQoNS). DoQoNS can happen in either one of the two stages, before resource allocation and after. Before resource allocation, the attacks are on control flows, i.e. connection-level. These attacks can be in signaling, admission control, routing, and resource allocation. After resource allocation, the attacks are on data packets and data flow, i.e. packet-level. Examples of such attacks are scheduling, multiplexing, traffic shaping, packet dropping, and congestion control etc. Another likely attack is unnecessary or suboptimal resource reservation, causing the system to reserve excessive resources. Attackers can also degrade network utilization by interfering with reservation protocol such that the network can support only a small subset of its service capacity. In RSVP, the sender can send one of these messages to the receiver: *Find*, *TSpec*, *AdSpec*, and *Teardown*. Meanwhile, the receiver can send back to the receiver one of these messages: *RESV, TSpec, RSpec*, *Flowspec*, and *Teardown*. An attacker to RSVP can be an insider who controls RSVP-enabled router on the reservation path, an outsider on path who controls RSVP-disabled router on the reservation path, or an otherwise outsider who may control router or host not on the reservation path. In single-cast, attack scenarios can be a change in TSpec, AdSpec, or RESV that will result in either unnecessary reservation or utilization degradation. In multicast, there are attacks leading to incorrect reservation and drop in at least one of the connection links, and there are also attacks in which some member(s) of the multicast group attacks the other members of the same group. RSVP has difficulties in dealing with insider attacks because it is very difficult to tell whether an RSVP-enabled router is behaving correctly or not due to the existence of mutable objects such as AdSpec and RSpec. Therefore, hop-by-hop authentication becomes meaningless. To tackle the problem, *Selective Digital Signature and Conflict Detection (SDS/CD)* uses a detection algorithm with modified end-to-end authentication by separating target RSVP objects into constant and mutable. The constant objects (TSpecs) are digitally signed by the source and verified by the destination, while the mutable ones (RSpecs and AdSpecs) are digitally signed and sent with the RESV message by the destination as a commitment since once data reaches the destination it should not be changed any more. The signed "history" is then compared along the inverse route path with local observation by having the intermediate routers check the signed AdSpecs, and if there is any conflict due to the less amount of AdSpec signed than the downstream version of AdSpec, the protocol will react according to local *policy decision point (PDP)*. For multicast messages entering a merging point, the intermediate router will pick the RESV message with largest RSpec and forward it upstream. When the sender receives the RESV message, it verifies that was signed by a valid receiver. Sender digitally signs RSpec and piggybacks it with the next refreshing PATH message. Upon the reception of the refreshed PATH message, the intermediate router checks that the resource request in the sender-signed RSpec is at least equal to the resource request in the receiver-signed RSpec. However, it still cannot detect

or eliminate all possible types of tampering or malicious data traffic injections, as it can only prevent the attack from outsiders only, and it still cannot handle dropping attacks. Yet, SDS/CD provides better protection, and can be complemented with IDS techniques to at least be able to identify the dropping point. Dropping attacks can be resolved with *microeconomics concepts* by applying a pricing paradigm, which complements *COPS (Common Open Policy Service)* protocol with a Billing DB and a User DB in additions to COPS' own Policy DB, by statistical analysis of traffic patterns with profiling.

**Secure QoS using BGP/MPLS**

MPLS and BGP are both routing protocols that most Internet QoS providers are adopting. Besides routing purposes, they also provide some sort of security.

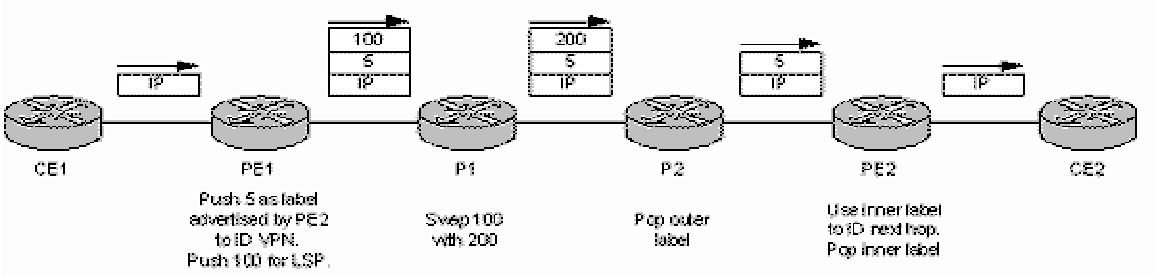
*MPLS*

*Multiprotocol Labeled Switching (MPLS)* is one of the easiest and most important network models to offer Internet QoS guarantees because of its router/switch-based nature. When an IP packet enters an MPLS ingress router, the router, which is a *Label Switch Router (LSR)*, assigns the packet to a *Forwarding Equivalency Class (FEC)*, which is based on various *Access Control List (ACL)* matches such as source and destination addresses, next hop, application type, and *Differentiated Service (DS)* flag, followed by assignment to a *Label Switched Path (LSP)* by adding a 32-bit *MPLS header*, to the packet header before finally sending it to the next router or destination. This forwarding is policy-based. This ensures the label of only local significance because the label is only relevant between the two neighbors in the route. The MPLS header has 4 fields, namely a 20-bit *label* field, 3-bit *class of service* field, a 1-bit *stack* field that supports hierarchical label stack, and an 8-bit *time to live* field. LSP routing can be defined using such constraint based routing protocols as *Constrained Shortest Path First (CSPF)*, and LSP signaling can be based on either *Label Distribution Protocol (LDP)*, *RSVP*, or *Constraint-based Label Distribution Protocol (CL-LDP)*.

*BGP*

*Border Gateway Protocol (BGP)* is a classless inter-domain, here also known as inter-autonomous-system (inter-AS), TCP routing protocol, where an autonomous system is a set of routers under the same technical administration which uses interior gateway protocol and common metrics for routing within the domain and exterior gateway protocol for routing to other domain(s). The fourth version of BGP, BGP-4, also introduces mechanisms which allow route aggregation, including AS path aggregation. A BGP speaker advertises to its neighboring ASs' speakers only those routes that it itself uses. Therefore, it supports the "hop-by-hop" routing paradigm generally used throughout the Internet. For those policies unable to be supported by the "hop-by-hop" routing paradigm, they need to enforce such techniques as source routing to enforce. Interior routing protocol provides an AS a consistent view of its interior routes, while having all BGP speakers within the AS maintain direct BGP connections with each other provides

the AS a consistent view of the exterior routes. Using a common set of policies, BGP speakers reach an agreement as to which border routers will serve as exit/entry points for particular destinations outside the AS. Connections between different ASs' BGP speakers are referred to as *external links*, where connections between same AS' BGP speakers are referred to as *internal links*. Routes are stored in the Routing Information Bases (RIBs): namely, the *Adj-RIBs-In*, the *Loc-RIB*, and the *Adj-RIBs-Out*. Advertised routes must be present in the Adj-RIB-Out. Local routes within the local BGP speaker must be present in the Loc-RIB, and the next hop for each of these routes must be present in the local BGP speaker's FIB. Routes received from other BGP speakers are present in the Adj-RIBs-In. A BGP speaker sends message to its peers regarding the routes in one of the four message types: OPEN, UPDATE, KEEPALIVE, and NOTIFICATION. OPEN message is for establishing a connection. UPDATE message is for updating route information including any possible route disconnection. KEEPALIVE message is for responding the successful OPEN message and keeping fresh of the route. NOTIFICATION is for sending messages upon detecting error before tearing down the BGP connection which experiences problem. BGP operation is in terms of finite state machine and has one of the six states: *idle*, *connect*, *active*, *opensent*, *openconfirm*, *establish*. *Idle* state is the initial state which has no BGP connection and refuses any incoming BGP connection. When a connection is requested from outside peer, which is in *active* state, it changes the state to *connect* and waits for the completion of the transport connection. BGP resource will be allocated in the respective ends. Upon the completion, an OPEN message is sent and it changes the state to *opensent*. Upon the successful receiving and checking of the OPEN message, a KEEPALIVE will be sent and the state now becomes *openconfirm*. Upon the receiving of the KEEPALIVE, the state finally becomes *established* and the connection can freely exchange UPDATE, KEEPALIVE and NOTIFIACTION messages. Upon any receiving of the NOTIFICATION, the connection will be torn down and the BGP resources between the pair will be released.




*Security requirements of BGP/MPLS*

There is increasing concern of MPLS architecture security. MPLS requires unique destination given an address. From a security perspective, the basic requirement is to avoid the situation in which packets destined to a host within a given VPN reach a host with the same address in another VPN or the MPLS core. The internal structure of the MPLS core network like provider edge (PE) and provider (P) elements should be invisible to outside networks, including the Internet or any connected VPN. It is advantageous if the internal addressing and network structure remains hidden to the outside world such that with this limited visibility, attacks like DoS become more

difficult. The following table shows two kinds of network attacks by outsiders, *DoS* and *Intrusion*, and their relationships with the network accessibility.

| | **Has Access** | **Has No Access** |
|---|---|---|
| **Authorized User** | Normal | Denial of service |
| **Unauthorized User** | Intrusion | Normal |

Therefore, to avoid DoS machines should not reachable to outsiders by packet filtering and address hiding, and to avoid intrusion easily-abused protocols have to be hardened and the network has to be made as inaccessible as possible, which could be achieved by a combination of packet filtering or use of firewalls and address hiding. A key issue in a pure IP network is easy address spoofing. In MPLS case, since it works internally with labels instead of IP addresses, MPLS has to make sure these labels cannot be spoofed by outsider as easily as IP addresses and packets will not be maliciously inserted through, for example, another *customer edge (CE)* of an MPLS VPN or an MPLS core, by a potential attacker with a label that he/she does not own.

*Security advantages of BGP/MPLS*

MPLS allows distinct VPNs to use the same address space, and so it adds a 64-bit *route distinguisher (RD)* on top of each 32-bit *IPv4 address* to make VPN-unique addresses also unique in the MPLS core. This "extended" address is also called a "VPN-IPv4 address" allowing the MPLS service customers not to change their current network addressing. Routing separation between the VPNs can also be achieved by having every *Provider Edge (PE)* router maintains a separate *Virtual Routing and Forwarding (VRF)* instance for each connected VPN. Each VRF on the PE router is populated with routes from one VPN, through statically configured routes or through routing protocols that run between the PE and the CE router. Since every VPN results in a separate VRF, there will be no interferences between the VPNs on the PE router. Multiprotocol BGP (MP-BGP) adds and exclusively exchanges VPN identifiers, such as route distinguisher, across the MPLS core to maintain the separation, and the core network does not redistribute the BGP information but instead maintains insider VRFs, allowing separate routing across the MPLS network. Given the addressing and routing separation across an MPLS core network, we can assume that MPLS offers, in this respect, the same security as comparable Layer 2 VPNs such as ATM or Frame Relay. It is not possible to intrude into other VPNs through the MPLS cloud, unless this has been configured specifically. BGP/MPLS offers security in a way that routes are kept separate in BGP routing updates through the use of unique identifiers in BGP route target extended attributes. These mechanisms are internal to the service provider and so the structure is invisible to the customers. Besides, the only relevant information that the router knows is the address of the next hop, thus it protects against the attackers who are trying to steal information about the core for attacks such as DoS, spoofing, or session hijack. As long as it is

properly configured for address space and routing separation, the router is almost impossible to be attacked. The only way for an attacker to attack is to spoof the customer address space or exist physically at a targeted router location to spoof the MPLS label, but this is the business of how the customer takes security precautions. Therefore, this implies same level of security as IPSec. Another big advantage of BGP/MPLS is its scalability allowing it to provide multicast service. Its ease of troubleshooting inter-domain routing and ease of deployment of latency sensitive applications makes it the popular choice for multicast purposes such as videoconferencing. Therefore, BGP/MPLS security works very well with the advanced QoS features of MPLS.

*BGP challenges*

Although the use of BGP/MPLS is generally secure, the TCP connection during BGP establishment may reveal some vulnerability in which spoofed TCP segments can be introduced into the connection stream, and as a result a false connection can be established to favor the malicious user. One proposal to protect BGP sessions is to use MD5 signature digest on TCP header and segment data. Although this proposal suggests MD5, it is open to any other hashing algorithm such as SHA-1.

**Security Infrastructure under Secure Quality of Service Handling (SQoSH)**

In recent years there are proposals for programmable network infrastructure allowing programmer access to network resources and data structures, and they aim to introduce new services. These programmable networks, also known as *active networks*, are for packet-switched networks, either on a per-user or per-packet basis. However, this also introduces more security risks and vulnerabilities. Security, as a result, becomes a significant issue in active network. That introduces *SANE*, the *Security Active Network Environment*, as a security infrastructure for active networks to guarantee Quality of Service (QoS). Dr. D. Scott Alexander at Lucent, and Dr. Jonathan M. Smith and his peers at the University of Pennsylvania proposed an architecture called *Secure Quality of Service Handling (SQoSH)*, as a means of *Active QoS*.

*Active networks and their challenges in security*

An active network infrastructure is very different from current Internet infrastructure. In Internet, buffer memory and CPU cycle used to locate the correct route are the only resources consumed by a packet. Hence, the overhead is not that much but that also easily enables DoS attack due to its simplicity. Furthermore, current Internet in general is difficult to provide enforceable QoS guarantee. In terms of security, current Internet infrastructure considers the network secure as long as it can protect against *admission failure* and *policing failure*. Admission failure is the result of unauthorized access of resources like unauthorized RSVP reservation to a specific node, where policing failure is the consequence of vulnerable security policies such as overuse of certain ports resulting in the inability to meet QoS requirements upon any QoS traffic to those ports. The flexibility of an active networking infrastructure causes huge potentials in exposing and

expanding its threat model for attacks towards itself. One typical example is the ability of DoS attack to abuse a variety of infrastructure resources such as CPU cycles, output link bandwidth, and storage etc., because of its taking advantage of their exposures due to loaded programs.

*SANE*

*Secure Active Network Environment (SANE)* provides the following security services to an active network:

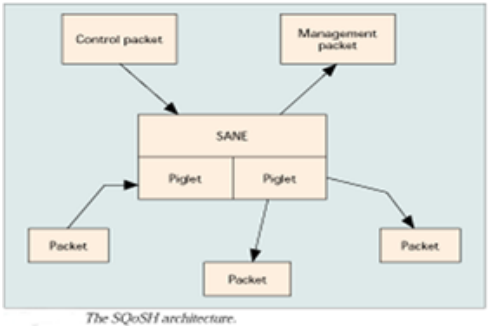
- Secure bootstrapping and component recovery
- Cryptographic primitives
- Packet encryption and authentication
- Secret key creation and exchange using *key establishment protocol (KEP)*, which supports secure bootstrapping, session-key establishment, principal authentication and authorization, secure neighbor node discovery so that the new node can establish trust relationships with neighbors to secure infrastructure information sharing
- Administrative domains to enforce security restrictions and allow border elements to act like firewalls
- Naming service for secure module identification

A system combines the SANE elements based on one of these design principles:

- Dynamic checks, while the active node is operating, should be fast since that happen frequently
- Static checks, usually performed before the active node enters operating state, can be expensive due to their relatively infrequent happenings
- If possible, improve the system performance by using the static checks during compilation to eliminate the need of dynamic checks during operations

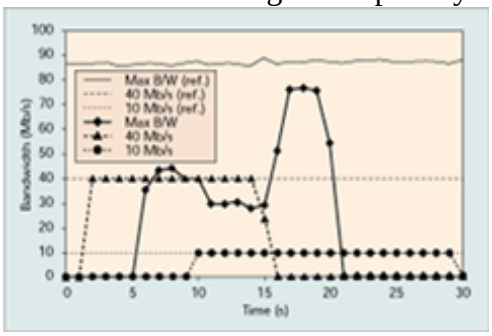
SANE protects resources such as access to standard and loaded modules, CPU cycles, allocated memory, number of packets, latency and bandwidth requirements. It associates cryptographic credentials with modules to achieve secure manageable and controllable module loading by requesting a certificate for a particular module or just simply allowing universal loading of such module. Most low-cost modules, like *ping*, are universally allowed under SANE. Two kinds of certificates in the packets are recognized by SANE: *administrative* and *regular*. Administrative certificates allow any or all loadings into the system, i.e. they have higher privileges given by the system administrator. Regular certificates permit selected module loadings under specified usage patterns.

*SQoSH architecture*




*The SQoSH architecture.*

SQoSH's main goal is to protect against admission failure and policing failure by balancing all the factors of performance, usability, flexibility, and security. This suggests that the architecture has to be *front-loaded* in order to reduce subsequent decision cost needed for every single packet. *SANE* performs front-end cryptographic operations required for access granting to assure authentication, allowing the OS to focus on whether the requested resource can be allocated or not. The OS provides basic packet delivery operations for SANE using controlled multiplexing of the shared network resource, and demultiplexes all packets destined for SANE. A full scale SQoSH system would consist of a multiprocessor with OS instance on each device-managing processor so that the processor's OS manages all I/O devices and performs resource scheduling by responding to device interrupt. The OS in the SQoSH system thus becomes the manager of interrupts, buffering, and status polling etc. The host OS will then be protected from device-initiated attacks. Simulation using three users, one being malicious trying to steal resources, another one requesting about 40 Mb/s and the last one requesting about 10 Mb/s, under the environment of available bandwidth of approximately 85 Mb/s, also echoed the statements. Hence, SQoSH offers controlled access to local or remote system resource allocations, thus providing efficient-enough security in *Integrated Services (IS)* resource allocations and during which privacy and integrity of media streams have to be preserved.




*SQoSH applications*

SQoSH architecture is proposed as a powerful resource management tool in a network. It is expected that Active QoS can adapt to economic environments such as capturing complex auction decisions easily by active packets in a programmable infrastructure. It can also work well in military environments in which hierarchical command responsibility maps to multiple service and security classes. SQoSH ensures authentication in a control request and preserves corresponding resources for that class so that whenever there is any delivery there will not be any delay since the delivery may be critical to the whole military action. It therefore provides the integrated admission control

and policing that conventional QoS, especially RSVP and ATM signaling protocols which presume the trust of the administrative entities, lacks.

## Security Management on programmable QoS network components

Programmable network components are growingly popular in modern networking because they support adaptive QoS required mostly by multimedia applications and mobile computing users. To provide a seamless ubiquitous computing environment required for fast service creation and resource management through a combination of network-aware applications and application-aware networks, portable intelligent communicators will need to make use of local network services. Adaptive networks must support rapid customized service deployments for potentially mobile corporate and individual users. Many mechanisms are being promoted for programmable network components, including code-carrying IP packets executed by the routers traversed by the packets, scripts or interpreted codes loaded via management interface like Java applets, and mobile agents carrying both codes and data to autonomously migrate around the network. However, there is a potential that these codes may contain serious malicious or inadvertent bugs. Therefore, instead of freely allow total adaptive component behavior, policies are imposed to present suitable restrictions.

### *Network Policy Model*

It is difficult to specify and analyze programmable network security for resource access and update because of the presence of many different organizations as users and the combinations of many heterogeneous components such as databases and firewalls controlled by different organizations. Security management also needs adaptivity to specify actions in response to security violations and network-based attacks. Policies are persistent rules governing system behavior choices derived from business goals, service level agreements, or trust relationships within or among enterprises, and have two kinds: *obligation policies* and *authorization policies*. Obligation policies are event-triggered condition-action rules defining the network conditions usually for resource reservation, queuing strategies, router code-loading or reconfiguration etc. They may be user-specific or application-specific and mostly are not involving error correction. Authorization policies define accessibilities to service and resource. It is more desirable to have policies dynamically update themselves based on the distributed entities, and it is more practical to specify group-related policies instead of individual-based policies due to many millions of individual users and resources. While an obligation policy requires a relevant authorization policy for defined permissions, it does not imply an authorization policy.

### *Security Policy Model*

A common security policy model for specification and enforcement of organizational access control is *Role-Based Access Control (RBAC)*. RBAC is role-based instead of user-based, mapping users' role assignments to access permissions. Multiple users can be assigned to the same role and multiple roles can be assigned to the same user, and

constraints may be applied between users and roles, between roles and permissions, or between roles themselves. Its goal is to simplify permission management in large organizations using structural, hierarchical, reusable, and inheritable approaches, similar to object-oriented approach in programming. In situation when a senior group may also inherit exceptions among the inherited roles from any junior role, a *private role* is created for the group instead to group those not inherited upward in the hierarchy. RBAC is preferred to be *capability-based*, in which responsibilities for inherited permission collections are delegated to the end-user's system prior to access control check to remote systems during remote invocation, because this reduces the network overhead due to the complexity of possible multiple remote invocations required by role checking. The introduction of RBAC models revolutionized access control in which access control policies can be implemented on the basis of clearly specified organizational policies instead of embedded implementation.

*Trust Policy Model*

*Trust Management* is a framework that supports sophisticated authorization policy specification and implementation using public key certificates as credentials to authenticate identities or group memberships. *Trust Policy* assigns client to a group similar to a role, where the group will then be assigned authorization rules, in the form of X509 certificates, on resource access. Therefore, access control policies and role policies can be assigned by different authorities. The fields in the certificate define group membership criteria, and can contain related links to other certificates. An XML-like script language, *Trust Policy Language* from IBM, is a popular choice to define trust policies, especially in e-commerce and Internet applications because they need flexible authorization policies. However, since XML syntax does not support inheritance, it is not suitable for specifying security management policy.

*Management Policy Specification*

Service level goals can be integrated with policies to support multiple-level adaptability in a network both at hardware and within network-aware applications and application-aware networks like active networks, and management and security are closely linked. Policy management offers the interoperability between QoS and security to protect QoS from any malicious abuse. In many cases, management policies are specified in a script language like those as follows:

- Lucent's *Policy Definition Language* is an event-condition-based language based on obligation policies using has two simple main constructs, policy rule corresponding to the obligation policy and event-triggering rule, and is widely used in Lucent's switching products. Like other scripting languages, it does not support object-oriented approach of reusability in its specifications.
- DMTF and IETF defined a policy information model as an extension to the Common Information Model (CIM) called *Policy CIM (PCIM)*. This model is based on entity abstraction and representation in a managed environment in terms of properties, operations, and relationships, and is independent of any specific repository, application, protocol, or platform. The model is a mapping of PCIM to

a directory schema so that a *Lightweight Directory Access Protocol (LDAP)* directory can be used as a repository, making policies as objects stored in an LDAP directory service to be retrieved by policy consumer, or *policy decision point (PDP)*, later upon requests from *policy execution point (PEP)* like a router using the *Common Open Policy Service Protocol (COPS)*, while it is possible to have a combination of PEP and PDP as one single component. The CIM defines generic objects like managed system elements, logical and physical elements, systems, service, and service access point, while the Policy Model defines a policy rule, its component policy conditions, and policy actions, where the policy rule, which can be nested, is assumed to have the format "if <condition set> then do <action list>". Both conditions, which can be a set expressed disjunctive or conjunctive form, and actions, which can be either sequential or in any order, can optionally be stored separately in a policy or reused by multiple rules.

- IETF has also extended PCIM to *QoS Policy Information Model (QPIM)* by containing a set of IntServ-specific or DiffServ-specific management. PCIM does not distinguish between authorization and obligation models, because the emphasis is on the consideration those QoS obligation policies without event triggers. However, simple authorization policy can be affected by means of an action to respond to a message. For example, the term *role* is defined and interpreted as a characteristic of a managed element being used as a means to identify policy-applied elements. In additions, applicable policies are searched after being triggered by an implicit event such as packet arrival at a router, based on the policy conditions like source or destination addresses. In order to prevent the overhead, only situations that infrequent PEP querying PDP decisions, like in IntServ, are practical enough, otherwise PDP has to transfer the policy information and preload that into PEP using COPS policy provisioning mode, or other protocols like SNMP or HTTP, in situations like DiffServ. Currently there are some suggestions to extend QPIM to include explicit events for policies dealing with failures, overloads, and other special situations.
- An extension of the *Unified Modeling Language (UML)* on *ODP Reference Model Enterprise Viewpoint* uses the concept of system abstraction within a defined environment in terms of the system's purpose, scope, and policies that both applying to the system and are defined within the system. Several proposals suggested the additions of design elements and other constraints to UML so that Enterprise Viewpoint concepts can be implemented in UML. For example, an authorization policy can be expressed as a set of objects such as capabilities or certificates and an obligation policy can be realized by the implementation of a particular activity diagram or collaboration.
- Imperial College uses the *Ponder* language as part of the security management projects. Ponder is a declarative object-oriented language which can specify security policies by mapping them onto various access control mechanisms for firewalls, OSs, databases, and even Java, and supports inheritance and element overloading. It also supports event-triggered condition-action obligation policies for network and distributed systems management, such as policies for user registrations, logging, and auditing the events like critical resource accesses or security violations. Objects can be grouped into domains according to policies

applied, geographical boundaries, object type, responsibility, and authority. Management structures and interactions are defined in terms of relationships between roles, which are related to common policies grouped for some positions within the organization and can be viewed as sets of authorizations, obligations, refrains, and delegations having the same subject(s). Therefore, domains can be view as "directory" of the objects they group, and therefore can be nested. In fact they have been implemented as directories in an extended LDAP service. Objects can be added to or removed from the domains without having to modify the policies. A person can be assigned to multiple roles but such person cannot perform action as one role using a right from another role.

Consistency and completeness are requirements for policies. However, since policies include constraints, inconsistency, which results in conflicts, are inevitable. Generally, conflicts are application-specific, so it is necessary to specify the conditions that are expected to result in conflicts to reduce conflicting possibilities. One approach is to specify constraints on the policy sets and to analyze the sets against the constraints to determine if there is any conflict. Sometimes conflicts occur because of parameter settings to the constraints, so conflicts analysis and setting priorities among the conflict policies are helpful. In general, negative authorizations have higher priorities than positive ones, and specific policies may need to override the general ones.

*Future Research in Policy Management*

Even though policies support multiple-level adaptability in a network now, it needs further research to define interfaces for the policy exchange among different levels to provide better efficiency than adaptation within the network. The biggest obstacle is to map different policy semantics among different levels. For example, it is possible that an application is not aware of what components exist in the network, so it still cannot specify related policies after adaptation.

## Conclusion

This article provides a general view of different security areas for Internet QoS. It shows how end-to-end security helps make QoS routing secure. It describes several secure QoS proposals for forwarding, RSVP security, and BGP/MPLS. It gives some details on security infrastructure based on active networks to prevent misuse of resources. It emphasizes the importance of security management in making security effectively offered for QoS. While all these mentioned schemes serve as the protection for Internet QoS, most of them require interoperability in order to prevent attacks effectively. For example, without the use of security infrastructure of policy server in COPS, RSVP protection as a scheme for secure QoS will not be able to be provided. This article offers an overview of various security models focusing on different security areas and how they may interoperate with other security models in the same or different security areas. Experimental results have already proved their functional effectiveness. As we see from this article, all of these schemes are still under continuing research work. One major

focus on the continuing research is to improve the efficiencies so that these proposals can be adopted and integrated into products in the practical industry world.

**References**


- Feiyi Wang, Brian Vetter, Shyhtsun Felix Wu, "Secure Routing Protocols Theory and Practice", http://shang.csc.ncsu.edu/papers/CCR-SecureRP2.ps.gz, May 1997
- Z. Fu, H. Huang, T. Wu, S. F. Wu, F. Gong, C. Xu, I. Baldine, "ISCP: Design and Implementation of an Inter-Domain Security Management Agent (SMA) Coordination Protocol", http://www.cs.ucdavis.edu/~wu/publications/14_1.PDF, IEEE NOMS 2000, page 565 to page 578
- Bob Braden, David Clark, Steve Crocker, Christian Huitema, "Secure QoS Forwarding", http://zvon.org/tmRFC/RFC1636/Output/chapter4.html, RFC-1636: Chapter 4, Report of IAB Workshop on Security in the Internet Architecture February 8-10, 1994
- Errin Fulp, Zhi Fu, Douglass S. Reeves, S. Felix Wu, Xiaobing Zhang, "Preventing Denial of Service Attacks on Quality of Service", http://seclab.cs.ucdavis.edu/papers/2001-01-discexII.pdf, Proceedings of DISCEX II, 2001
- Tsung-Li Wu, S. Felix Wu, Zhi Fu, He Huang, "Securing QoS: Threats to RSVP messages and their Countermeasures", http://arqos.csc.ncsu.edu/papers/1999_10_iwqos99.pdf, Proceedings of IWQOS 1999
- David Durham, Jim Boyle, Ron Cohen, Shai Herzog, Raju Rajan, Arun Sastry, "The COPS (Common Open Policy Service) Protocol", http://www.ietf.org/rfc/rfc2748.txt?number=2748, RFC 2748, IETF
- Yakov Rekhter, Tony Li, "A Border Gateway Protocol 4 (BGP-4)", http://www.ietf.org/rfc/rfc1771.txt?number=1771, RFC 1771, IETF
- "Security of the MPLS Architecture", http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/mxinf_ds.htm, Cisco Systems White Paper
- Gary Alterson, "Comparing BGP/MPLS and IPSec VPNs", http://rr.sans.org/encryption/MPLS2.php, SANS Institute Information Security Reading Room, January 9, 2002
- Andy Heffernan, "Protection of BGP Sessions via the TCP MD5 Signature Option", http://www.ietf.org/rfc/rfc2385.txt?number=2385, RFC 2385, IETF
- D. Scott Alexander, William A. Arbaugh, Angelos D. Keromytis, Steve Muir, Jonathan M. Smith, "Secure Quality of Service Handling: SQoSH", http://www.cs.umd.edu/~waa/pubs/sqosh.pdf, IEEE Communications Magazine, April 2000
- Morris Sloman, Emil Lupu, "Security and Management Policy Specification", http://www.comsoc.org/livepubs/ni/private/2002/mar/sloman.html, IEEE Network, March/April 2002, page 10 to page 19