Prioritising access to scarce resources: network survivability through policy supported quality of service

Luiz A. DaSilva* and Kaustubh Phanse

Bradley Department of Electrical and Computer Engineering,
Virginia Polytechnic Institute and State University, USA
E-mail: ldasilva@vt.edu E-mail: kphanse@vt.edu
*Corresponding author

Abstract: Packet-switched networks (such as those using the internet protocol) now support an increasingly heterogeneous mix of applications, generating the need for performance guarantees for each traffic flow, or what is commonly known as quality of service (QoS). We argue that a QoS architecture, supported by the appropriate set of policies, can not only be used to provide service assurances to performance-sensitive traffic but also to enhance network survivability by prioritising access to scarce resources whenever there is sudden degradation in the underlying network infrastructure. In this paper, we illustrate this idea through representative examples. We also present the taxonomy for various policy based network management architectures and discuss our preliminary results in comparing such architectures through experiments conducted on our network test-bed. The identification of the appropriate policy architectures is an important first step in the implementation of a network system with the desired survivability characteristics.

Keywords: quality of service; network survivability; network management; policy-based management.

Reference to this paper should be made as follows: DaSilva, L.A. and Phanse, K. (2004) 'Prioritising access to scarce resources: network survivability through policy supported quality of service', *Int. J. Critical Infrastructures*, Vol. 1, No. 1, pp.20–37.

Biographical notes: Dr. Luiz A. DaSilva received his PhD in Electrical Engineering from the University of Kansas. In 1998 he joined Virginia Tech as an Assistant Professor at the Bradley Department of Electrical and Computer Engineering having previously worked for IBM for six years. His research interests currently focus on performance and resource management in wireless mobile networks and QoS issues. He is currently involved in funded research projects in the areas of QoS interoperability and network policy, resource management mechanisms for the deployment of smart antennas in 3G systems, and mobile ad hoc networks, among others. Dr. DaSilva has published over 30 refereed papers in journals and major conferences in the communications and computer areas. Current and recent research sponsors include NSF, the Office for Naval Research, the US Customs Services, Intel and Microsoft Research. He is a member of the Center for Wireless Telecommunications (CWT), associated faculty at the Mobile and Portable Radio Research Group (MPRG) and a member of the Governing Board of the NSF-funded Integrated Research and Education in Advanced Networking (IREAN) program at Virginia Tech. Dr. DaSilva is a senior member of IEEE and a member of ASEE.

Kaustubh S. Phanse received his PhD in Electrical Engineering from Virginia Tech. in 2003. He received his BE degree in Electronics and Telecommunication in 1998 from Vivekanand Education Society's Institute of Technology (VESIT), University of Mumbai, India and his MS degree in Electrical Engineering in 2000 from Virginia Tech, Blacksburg, VA. His current research focuses on applying policy based networking for QoS provisioning and management in wireless mobile networks. Dr. Phanse is currently a post-doctoral associate at the Bradley Department of Electrical and Computer Engineering, Virginia Tech. and is a student member of IEEE and ASEE.

1 Introduction

In today's information age, data networks have undeniably become a part of our critical infrastructure. This applies both to private networks operated within a commercial enterprise or government agency (intranets) and the public infrastructure that forms what we know as the internet. From entertainment to dissemination of important news, from electronic commerce to the support of real time communications (voice and video in particular), these networks have dramatically changed the way we work, communicate and play. Despite the best efforts of security practitioners to harden communication networks, such networks remain vulnerable to failure and attacks, and it is clearly important that we develop methods that can enhance, if not ensure, their survivability.

One can view network survivability as depending on two complementary mechanisms: prevention methods that minimise the probability of a network being disrupted by a failure or attack, and mitigation methods that limit the damage when a failure or attack occurs.

Prevention methods must address both hardware and software failures [1]. These will include a variety of network security mechanisms, such as encryption, authentication protocols and firewalls [2] as well as fault-tolerant design and redundancy of network equipment and links. Further, certain communication techniques, especially in a wireless environment, may also fall under the category of prevention. For example, spread spectrum was originally developed to prevent jamming by the enemy in a military scenario; more recently, the use of directional antennas has been investigated to provide both increased robustness and low probability of interference and jamming [3].

Mitigation methods include network design approaches that contain route redundancy and allow for fast rerouting of traffic in case of link failure, as well as traffic management mechanisms that allow classification and differentiation among traffic flows and favour high priority or mission critical flows when resources become scarce. We believe that the use of such traffic management mechanisms, when combined with route redundancy in the network design, can provide a comprehensive solution for network survivability. In this paper, we focus on traffic management associated with the QoS concept [4–6] and the policy architectures that are needed to support it.

To date, quality of service mechanisms, such as admission control, queuing, scheduling, shaping and policing, have been proposed and used mainly to provide service differentiation to an increasingly heterogeneous application mix. However, these mechanisms, with the help of a robust underlying policy framework [7,8], can also be

used in order to support graceful quality degradation when network resources become scarce and can no longer meet demand. In such demanding situations (caused by infrastructure failure, network congestion, etc.), the network must have the means to direct existing resources to mission critical applications (for instance, command/control communications in a military environment or emergency services in a disaster management situation). These means include reliable ways to differentiate offered traffic a robust set of policies to determine which flows should be given access to which resources, and efficient dynamic resource allocation algorithms. This article summarises the main mechanisms used to attain QoS and outlines some ideas regarding how these mechanisms can be adopted in order to increase network survivability.

The remaining portion of this article is organised as follows. In Section 2, we provide an overview of recent and ongoing research in network survivability and promote the need for further research in this area. In Section 3, we discuss the applicability of policy-based QoS mechanisms to network survivability. We begin by briefly describing the various mechanisms used to attain QoS and certain QoS protocols and architectures that have been standardised to date. We provide a brief overview of policy-based networking and how it can be used to impart robustness to the QoS mechanisms. Further, we discuss various architectural approaches that can be adopted to implement a policy framework and what impact each of these approaches may have on network survivability. In Section 4, we elaborate, through examples, on the utilisation of a policy-based QoS framework to enhance network survivability. We also describe the implementation of our test-bed network and experimental evaluation of certain policy architectures. Finally, we offer some concluding remarks in Section 5.

2 Related work

Over the past few years, there has been considerable research effort devoted to network survivability. The subject has been extensively studied with regard to circuit-switched, Asynchronous Transfer Mode (ATM) and fibre optic networks. An overview and some sample references on related work can be found in [1].

The proliferation of packet-switched networks, in particular networks adopting the Internet Protocol (IP), has resulted in the research focus shifting to the survivability of such networks. Further, with the emergence of the concept of quality of service for IP-based networks, there is growing interest in how these QoS mechanisms may be used to enhance network survivability. Dovrolis and Ramanathan [9] have proposed a mechanism, using the resource reservation protocol (RSVP) [10], to minimise disruption of service for critical and real-time applications. This is done by reserving additional bandwidth along a secondary or back-up path that can be used by the real-time applications if the primary path fails. This idea has been extended by Srikitja, Tipper and Medhi [11], providing a comparative study of two RSVP-based restoration schemes. Multi-protocol Label Switching (MPLS) [12,13] has also been found to be useful in providing traffic engineering for multi-class traffic with different survivability or restoration requirements, as shown in [14,15].

Most of the proposed mechanisms mentioned above are based on the fundamental philosophy of allowing certain applications to reserve bandwidth along a secondary path. Although such an approach provides faster restoration times, the redundancy in resource reservation may limit the ability of the network to offer bandwidth guarantees to new

flows. Also, this approach is not practical for some low bandwidth networks (e.g., ad hoc wireless data networks), where it is not possible to lay claim to precious network bandwidth for back-up reservations under normal network conditions. An alternative approach [11] involves the setting up of a primary reservation and attempting traffic restoration on network disruption or failure. However, since this approach is based on resource-based admission control, recovery from failure is not guaranteed.

Further, it must be noted that, while QoS mechanisms such as those mentioned above can be used to enhance survivability, they also impose an additional security risk. The ability to gain privileged access to resources through QoS mechanisms provides the means for malicious users to launch an attack (such as denial of service) by illicitly acquiring network resources, potentially starving authorised users or applications. Therefore, it is extremely important to ensure the security and robustness of such mechanisms.

We believe that the problems or shortcomings mentioned above with respect to network survivability or restoration can be addressed with the help of an underlying policy-based network management framework [7,8]. The key features that a policy framework endows are the ability to implement policy-based admission control enabling authentication and authorisation of users or applications, support for dynamic resource allocation (i.e., the ability to accommodate higher priority or mission critical traffic, even when resources are scarce, unlike purely resource-based first-come first-served admission control) and graceful degradation in the case of network failure, hence ensuring *best possible* performance for the various traffic classes under challenging network conditions. Our aim is to study the various policy architectures and how they can be used to support robust and adaptive QoS and enhance network survivability. Fulp *et al*. [16] present one such policy architecture based on pricing [17] to limit the various QoS requests. We discuss and compare alternative mechanisms and architectural approaches that facilitate efficient implementation of a policy-based QoS framework.

3 QoS policy: an overview

Circuit-switched networks, of which the public telephone network is the prime example, assign a fixed amount of resources to each accepted call. These resources are dedicated to the call for its duration, and, therefore, cannot be shared by other calls. Data networks, on the other hand, adopt a packet-switched approach and achieve increased efficiency through the concept of statistical multiplexing. Initially, packet switched networks worked on a best-effort basis, where packets from all flows competed on an equal footing for the available resources. Nowadays, there is increased interest in providing performance guarantees to different flows in such networks, motivated by a very heterogeneous mix of applications and, more recently, by the desire to effectively support real-time traffic over the internet. In this paper, we focus on packet-switched networks.

What has become known as network QoS generally entails at least one of two aspects: service differentiation and performance assurance. Service differentiation recognises that different traffic flows, supported over the same network infrastructure, may have significantly distinct traffic characteristics (e.g., average and peak data rates, burstiness) as well as distinct needs (e.g., limited delay or jitter, minimum bandwidth requirements). Through appropriate traffic management mechanisms, the network can

offer different services that are tailored to specific classes of traffic. Performance assurance may be associated to service classes; examples include guaranteeing access to a certain amount of bandwidth or treating different classes according to a pre-established set of priorities.

Thus, the underlying principle of QoS can be summarised as the assignment of resources to each flow according to its needs and to the network's capabilities. However, service differentiation or the ability to reserve network capacity for certain flows opens up the possibility of unauthorised usage of available resources. A QoS implementation that does not include the means for authentication and authorisation would be likely to lead to a 'tragedy of the commons' and possibly result in even worse than best-effort performance.

In commercial network services, pricing can be used to ensure proper distribution of resources (some examples of this are discussed in [17]): users who are willing to pay more can request and obtain higher levels of service. However, there are environments in which pricing should not or cannot be used for this purpose; these include intranets run by an enterprise, ad-hoc networks used by the military, and networks established by law enforcement or disaster management entities. In such cases, QoS architecture must be able to implement resource allocation or admission control based on one or more factors, such as the owner of the traffic (identity of the user, application or department the traffic is originating from), temporal elements (time of day or week) etc., in addition to the availability of network resources. This is where policy-based networking becomes critical.

In this section, we discuss the relationship between QoS mechanisms and network survivability, as well as the principles behind policy-based networking.

3.1 QoS for network survivability

When network resources are reasonably plentiful, the main benefit of QoS is to protect performance-sensitive traffic (e.g., real-time traffic) from occasional deterioration in performance due to temporary congestion conditions. It is when these resources (bandwidth, in particular) become scarce that QoS mechanisms take on the more crucial task of ensuring their most efficient use.

The mechanisms we envision, supported by the appropriate policies, would allow the network to operate in essentially two modes. In 'normal' mode, QoS differentiation would occur to distinguish between the performance requirements of different applications. In 'survival-enhancing' mode, the primary objective of QoS policies would be to ensure delivery of the most sensitive types of traffic.

For example, third generation wireless mobile networks will support both voice and high-speed data over a cellular type of network. During regular operations, traffic management schemes will ensure that both types of users will be allocated the resources that are required for each type of traffic. This can be done, for instance, through radio resource management mechanisms, such as power control in a WCDMA environment. However, suppose an unanticipated event (such as a natural disaster or a terrorist attack) temporarily cripples the wireline infrastructure. In such situations, the wireless network would experience a sudden increase in traffic that would usually flow over the public switched telephone network; this additional load in turn makes it more difficult for law enforcement and disaster relief personnel to carry out their tasks using such networks. If the network already implements a set of QoS policies and associated mechanisms, this set

of policies can be used to keep non-critical traffic out of the network at times of emergency, as well as to pre-empt existing traffic in favour of new traffic of higher priority. Very importantly, the shift from normal to emergency operation can occur without human interaction simply by the enforcement of pre-existing policies.

The actual service differentiation can be based on either a prioritisation scheme, where low-priority traffic is essentially starved when resources are scarce, or on a reservation scheme that coordinates pre-reservation of resources to support sensitive traffic flows and allows pre-emption of existing reservations when in 'survival-enhancing' mode. Priority-based mechanisms are generally simpler to implement, while reservation-based mechanisms allow for more precise quantitative performance assurances but require additional signalling to establish the needed reservations.

3.2 Policy-based network management

Unlike legacy network management, which generally involves configuring and managing each network entity individually, policy based networking involves configuring and controlling the various operational characteristics of a network as a whole, providing the network administrator with a centralised, simplified and automated control over the entire network. QoS policy is one aspect of a wider area of recent interest, namely policy based network management (PBNM) [7,8]. Examples of other aspects that may be handled by a PBNM system are network security, address allocation, routing and content distribution. In [18] a policy is defined as "a definite goal, course or method of action to guide and determine present and future decisions". In general, policies can be seen as the plans of an organisation to achieve its objectives. This may involve a set of rules to govern the behaviour of its network and its components (e.g., resources, users, applications) and the specification of a set of actions to be performed.

Figure 1 Architectural elements of a policy framework

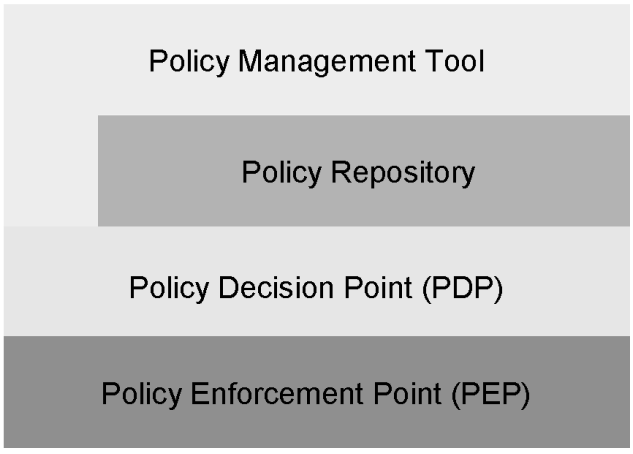
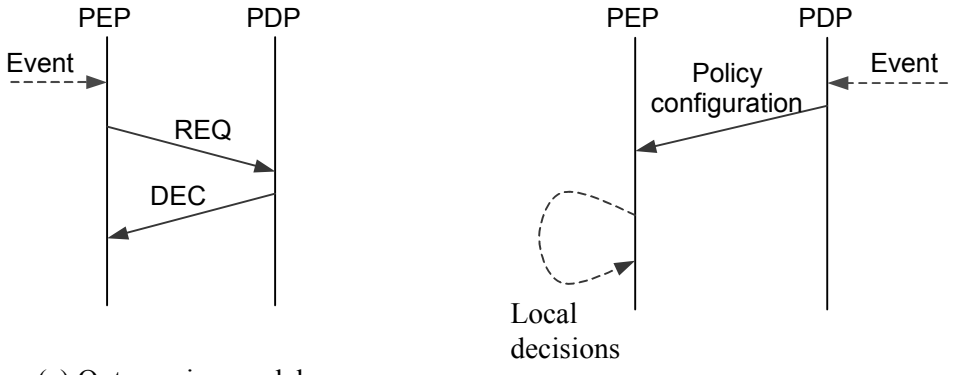


Figure 1 shows the four basic elements that typically constitute a policy framework [7]. A *policy management tool (PMT)* provides the network administrator with an interface via which she/he can interact with the network. A network administrator or operator uses the policy management tool to define the various policies or policy groups. The policies

specified at the PMT are then stored in a *policy repository*. It is typically the function of the PMT to validate the syntactic and semantic correctness of the administrator's input, to ensure consistency among the high-level policies and to check for compatibility of the various policies. The PMT also determines the association between the policies and the network elements where these policies are to be enforced, determines which low-level policies can be used to support the specified high-level policies and ensures that the specified policies are comprehensive enough to cover all the relevant scenarios. For this purpose, the PMT may directly interact with the *policy decision point (PDP)*. The PDP, or policy server, retrieves the policies from the policy repository and performs complex policy interpretation and translation into a format that can then be used to configure one or more *policy enforcement points (PEPs)* or policy clients. The PDP also needs to monitor any changes in the policies that might occur at the policy management tool or repository. The PEP is a network device where the policies are actually executed or enforced. It is noteworthy that, while an administrator would typically have centralised control through the policy management tool, the underlying components, namely the policy repository, the PDP and the PEP, may be numerous and distributed across a network.

Figure 2 Policy distribution models



(a) Outsourcing model

(b) Provisioning model

The Internet Engineering Task Force (IETF) resource allocation protocol (RAP) [19] working group is active in the field of QoS policy. It has defined, among other standards, the policy-based admission control framework [20] and the common open policy service (COPS) protocol (and its extension for provisioning COPS-PR) [21–23]. COPS is a simple client-server protocol that supports communication between policy clients and remote policy server(s). Two policy control models have been defined: outsourcing and provisioning, illustrated in Figure 2. While COPS supports the outsourcing model, its extension, COPS-PR, integrates both the outsourcing and provisioning models. In the outsourcing model, when the PEP receives an *event* (e.g., a RSVP reservation request [10]) that requires a new policy decision it sends a request (REQ) message to the remote PDP. The PDP then makes a decision and sends a decision (DEC) message (e.g., *accept* or *reject*) back to the PEP. The outsourcing model is thus PEP-driven and involves a direct 1:1 relationship between PEP events and PDP decisions. On the other

hand, the provisioning or configuration model [23] makes no assumptions of such direct 1:1 correlations between PEP events and PDP decisions. The PDP may proactively provision the PEP reacting to external events, PEP events, and any combination thereof (N:M correlation). Thus, provisioning tends to be PDP-driven and may be performed in bulk (e.g., entire router QoS configuration) or in portions (e.g., updating a DiffServ marking filter [24]). This approach allows PEPs to make some decisions locally.

3.3 Taxonomy of policy architectures

We are currently investigating the performance trade-offs involved in deploying a particular policy-based system to manage QoS and provide improved network survivability in bandwidth constrained mobile wireless networks. As a first step in this direction, we have established a characteristics-based taxonomy of policy architectures [25]. Such a taxonomy will help gain a better understanding of the various features that distinguish different architectures and also the strengths and weaknesses of each individual architecture. Further, the taxonomy, when analysed in light of certain constraints (e.g. bandwidth availability, host or network mobility, processing and storage capability of the hosts involved, probability of error or link failure, etc.) that a networking environment may suffer, will help determine the applicability of one or more architectures to that environment. In other words, the taxonomy will facilitate qualitative analysis of the various policy architectures and also the choice of one or more types of architectures that seem most promising for the scenario of interest.

Here, we provide a brief discussion of the taxonomy. A detailed description can be found in [25]. The taxonomy is defined based on the following characteristics:

- locus of control: policy server or policy decision point (PDP), e.g., centralised versus distributed
- locus of information: policy repository or storage, e.g., centralised versus distributed
- policy distribution: outsourcing and provisioning models depicted in Figure 2
- number of tiers: number of levels at which policy-based decisions can be made.

The taxonomy broadly classifies the various policy architectures into three categories:

1 *Outsourced architectures*: as the name suggests, all the PEPs outsource the policy decisions to remote policy server(s). All the policy decisions are made at a single control tier. If more than one PDP is present, then the PDPs operate as peers.

One of the major issues in defining and enforcing policies is policy conflict check: it is important for the network operator to ensure the coexistence of multiple non-conflicting policies. A centralised locus of control and information considerably simplifies the job of checking the consistency of the various policies stored and maintained at a single location. However, a major disadvantage of a purely centralised approach is that the smooth functioning of the entire policy framework is largely dependent on the central policy server and repository. In the event of the policy server becoming non-functional (e.g., shutting down due to a security attack), the entire policy system may break down. This makes the need for the security and integrity of the policy server and its stored information of prime importance. One way to prevent such a failure is to add redundancy and use distributed PDPs.

An outsourced architecture may generate considerable overheads for policy-related signalling. It requires significant signalling between the PEPs and the remote PDP(s), with an associated increase in the requirement for bandwidth. Secondly, since PEPs seek policy decisions from a remote PDP, the policy response time (the time it takes for the PEP to receive a decision after sending a request) tends to be high. The signalling overhead may be excessive for environments with significant bandwidth constraints, as indicated by our preliminary results in Section 4.2.

Thus, this type of policy architecture seems applicable to a small and considerably secure network with enough bandwidth to alleviate the problem of high policy response time and signalling overheads.

2 *Provisioned architectures*: these architectures use provisioning to push the policy information to different nodes in the network. Unlike the outsourced approach, wherein the policy distribution is mainly PEP-driven, the provisioned approach involves a PDP-driven policy distribution. At least two tiers of policy control are involved, i.e., the locus of control and the locus of information are distributed across two or more tiers. This approach involves distributed execution of policy control and reduces the signalling overhead encountered in the outsourced approach.

3 *Hybrid architecture*: A hybrid approach combines features of the outsourced and provisioned architectural approaches. The idea behind this approach is to avoid the pitfalls of the completely outsourced and provisioned approaches, while taking advantage of the benefits of each of those approaches.

   Hybrid architecture typically involves distributed policy execution, wherein either multiple policy servers and/or PEPs have local decision-making capability. However, if a relevant policy to service a request at hand is not found, then the policy decision is outsourced to another remote policy server. Such hybrid architecture seems promising for low bandwidth mobile and wireless network scenarios, since it provides the network devices with the flexibility to adapt to new or varying conditions for which a local policy may not exist.

In general, the distributed and hybrid architectural approaches can be further extended to incorporate a hierarchical control structure, i.e., with more than two tiers of control. The number of levels in the hierarchy may be set according to factors such as the size of the network or number of hosts involved, the processing and storage capability of the various hosts, or bandwidth availability.

4 Network survivability using a policy based QoS framework

As mentioned earlier, policy-based admission control (PAC) not only helps provide a robust QoS framework (by authenticating and authorising the users, applications or hosts requesting quality of service), but also facilitates accommodating higher priority or mission critical traffic when the network resources are already exhausted by lower priority traffic flows. This is unlike many resource based admission control algorithms, which admit traffic flows onto the network on a first-come-first-served basis. In the event of a failure in one or more network elements (nodes, links etc.), PAC can be used to

provide graceful degradation of services for existing flows according to their respective access priorities. Thus, a policy-based QoS framework provides the means for dynamic traffic management and restoration, in turn helping the network adapt to the varying conditions.

Selection of appropriate signalling mechanism(s) and architectural approaches can enhance the robustness of the policy framework itself against attempts to compromise the network (e.g., with flooding or denial of service attacks). For example, the COPS protocol implements HMAC-MD5 authentication (using shared keys) before a COPS connection is set up [21]. Without the correct key, no connection will be established and subsequent requests from a malicious node will be ignored. Using the provisioned architectures, policy clients can be made capable of making local decisions and overloading of network resources can be avoided.

4.1 Illustration

In this section, our aim is to illustrate the effectiveness of a policy-based framework for admission control and for restoration of high priority or mission critical traffic, in turn enhancing network survivability. Consider a network scenario as shown in Figure 3. The network elements R1 through R4 are routers connecting end hosts H1 through H6. For simplicity, all the links are assumed to have a maximum bandwidth of 1 unit. Four levels of traffic priority are defined for this network: best effort, better than best effort, very high priority and mission critical. These categories, in combination with other factors such as the user(s) or end host(s) involved, current network conditions, time of day, etc., are used to define a set of policies to manage the network resources. As mentioned earlier, the policies are typically stored in a repository accessible to one or more policy servers managing the network. The routers R1 through R4 act as the policy clients (or policy enforcement points).

As depicted in Figure 3, initially a bandwidth reservation of 0.6 is in place for a better than best effort traffic flow from host H4 to host H5. The remaining bandwidth of 0.4 over the links R3-R4-H5 is being used by best effort traffic from host H3. At the same time mission critical traffic is being sent from host H1 to host H6 with a bandwidth reservation of 0.6 along H1-R1-R2-R4-H6. Now, suppose host H3 intends to transmit very high priority traffic to host H5 and requests a bandwidth reservation of 0.5 units for this. Router R3 detects the scarcity of resources to support the potential traffic flows and indicates it to a policy server along with a request (for example, assuming an outsourcing type of policy model) for bandwidth reservation for very high priority traffic from host H3. Based on the defined policies, the policy server then sends a decision message to accept the reservation request for host H3 and, consequently, to slightly degrade the reservation for better than best effort traffic to 0.35. The remaining bandwidth (0.15) is then used for best effort traffic. The revised bandwidth reservations are shown in Figure 4.

Figure 3 Example network scenario; values represent current link utilisation

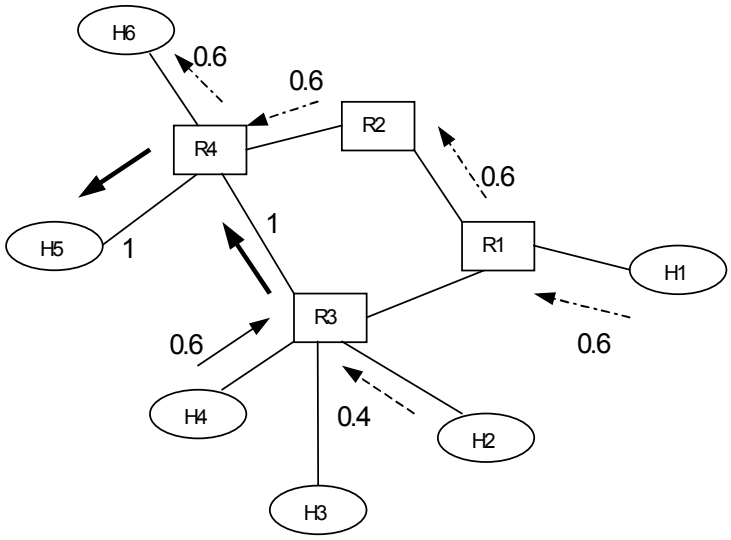
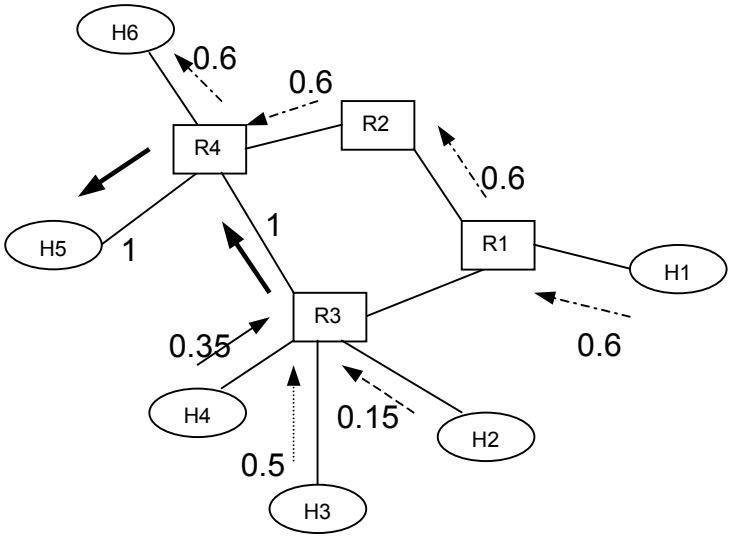


Figure 4 Policy based admission control enables higher priority flows to reserve bandwidth in the presence of scarce resources



Dotted arrows represent high priority traffic

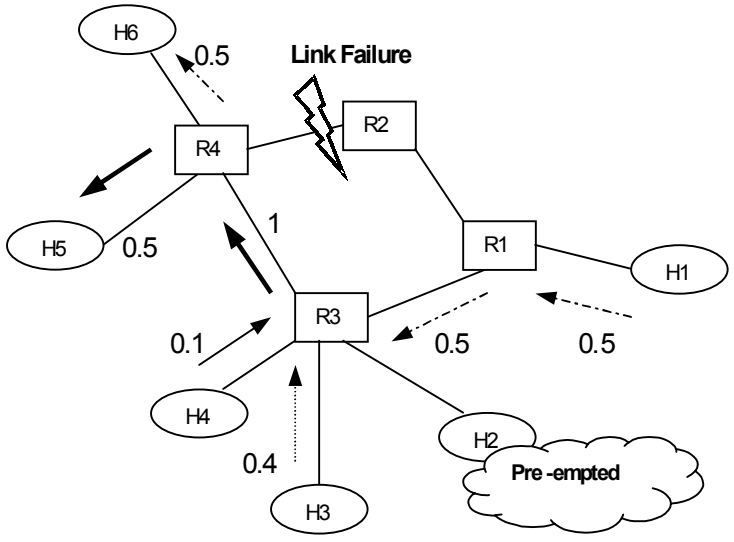
Service classes are represented as:

- best effort: dashed arrows
- better than best effort: solid arrows

- mission critical: dot-dashed arrows
- aggregate traffic belonging to multiple classes is represented by thick arrows.

Now, in the second example, let us assume that the link R2-R4 fails as depicted in Figure 5. This makes it necessary to re-route and restore the mission critical traffic that was being transmitted along H1-R1-R2-R4-H6. The only alternate path available is H1-R1-R3-R4-H6; however, all of its bandwidth is already utilised by existing traffic flows. In the absence of prioritised access, the re-routing of the mission critical traffic would only lead to congestion at router R3, resulting in degraded performance for all the traffic flows. Even in the presence of a QoS framework with purely resource based admission control, the re-routing of mission critical traffic is not feasible as 0.85 units of bandwidth are already reserved (by better than best effort and very high priority traffic) leaving only 0.15 units of bandwidth available for reservation. However, a policy based QoS framework allows the network to adapt to the service requirements of the various traffic flows in turn, enabling graceful degradation and attaining best possible performance for the various flows under the given circumstances. Upon detecting a link failure, router R2 would report the topology change to the policy server. Based on the information regarding the current flows in the network and the modified network topology, the policy server then sends an asynchronous notification to routers R1 and R3 indicating the need for re-configuring the existing bandwidth reservations. Hence, as shown in Figure 5, the mission critical, very high priority and better than best effort traffic flows are respectively allocated 0.5, 0.4 and 0.1 units of bandwidth along the link R3-R4, while the best effort traffic is pre-empted.

Figure 5 Graceful performance degradation enabling restoration of mission critical traffic



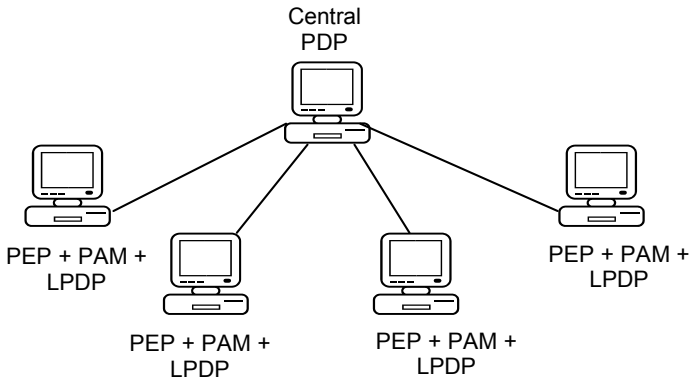
Note that the control traffic for the exchange of policy requests and responses is not explicitly shown in the figures above. From the transport layer point of view, policy requests flow out of band (for instance, over a separate TCP connection when using COPS). However, from a link layer point of view, control traffic will typically flow over the same links as regular data. One can think of the signalling as being logically out of band, physically in band. It is, therefore, important to protect control traffic from starvation in times of high network load. The same QoS mechanisms discussed above may be adopted to differentiate control and data traffic.

4.2 Experimental evaluation

As mentioned in Section 3.3, based on the requirements and constraints of a network, a particular policy-architecture or a combination of more than one architecture can be deployed to efficiently manage the network. We are currently conducting an experimental study of policy-based QoS frameworks and their applicability to traffic restoration/management and network survivability. We are specifically interested in evaluating the effectiveness of our policy-based QoS approach for wireless and mobile networks. Network survivability and restoration of these networks is even more challenging due to various constraints, such as low/variable link bandwidth, higher probability of link failures, mobility of nodes or networks etc. In this section, we briefly describe our Linux-based test-bed network and present some preliminary results comparing the various policy architectures.

A policy framework was implemented onto a Linux-based test-bed, as shown in Figure 6. We used the Intel COPS client software development kit (SDK) [26] to implement the outsourced and provisioned type of policy architectures. We extended the COPS SDK implementation with a policy advisor module (PAM) interface to realise the hybrid architecture. For a given request, the PAM interface detects if relevant policies are available locally, in which case the decision is made using the local decision point (LDP); otherwise, the request is outsourced to a remote PDP.

Figure 6 Test-bed implementation of COPS-based policy framework



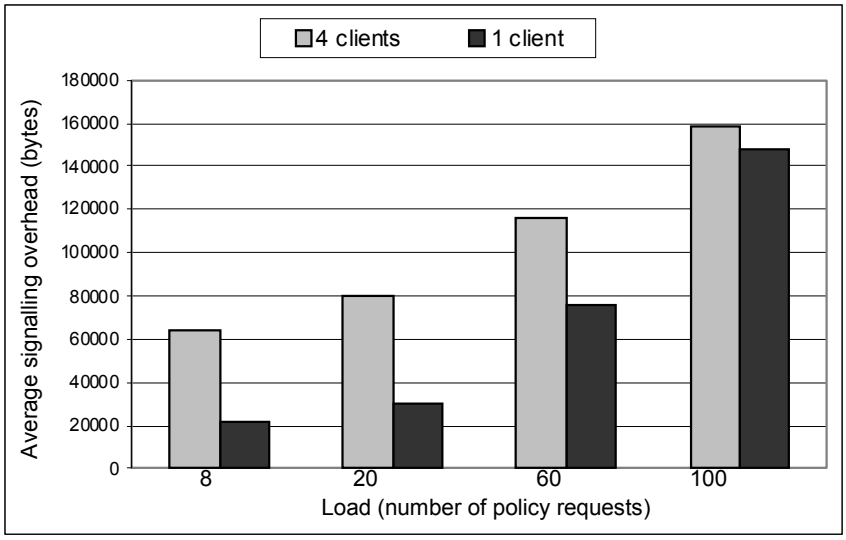
The first set of experiments provides us with an insight into the policy signalling overhead for the outsourced architecture as a function of the load or the number of policy requests, as shown in Figure 7. The experiments involve two cases:

1 a single PEP communicating with the PDP

2 four PEPs simultaneously communicating with the PDP.

The load is modelled as simultaneous requests to reflect a network scenario with several application flows trying to access network resources at any given time. As one would expect, for a given load (number of policy requests sent simultaneously by each policy client), the presence of four separate connections with the policy server in case of four PEPs results in greater signalling overhead as compared to a single PEP. However, it is seen that the difference in the overhead between the two cases considerably reduces with increasing load. This is attributed to the fact that the COPS protocol uses TCP as the underlying transport protocol. We observed that several COPS messages were encapsulated and transmitted in a single TCP segment. This reduced the TCP overhead per COPS message per connection, thus reducing the overall signalling overhead required for larger number of policy requests.

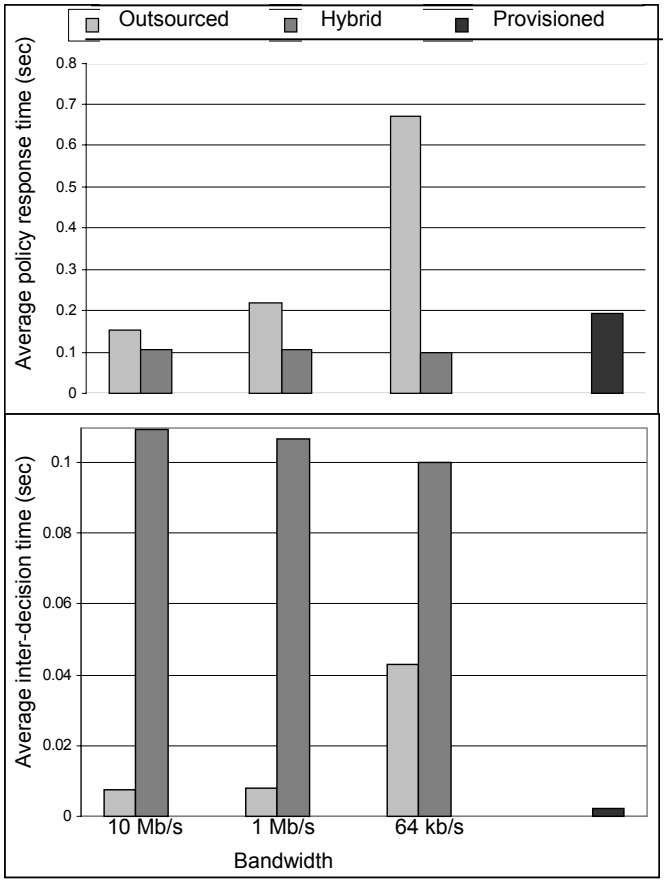
Figure 7 Policy signalling overhead for outsourced architecture as a function of load



A second set of experiments compares the three architectures being considered. The metrics used for this set of experiments are: the policy response time (the difference between the time at which the policy request was sent by a PEP and the time at which the corresponding policy decision was received at the PEP); and the inter-decision time (the time difference between consecutive decisions received by the PEP). The average policy response time and the inter-decision time were measured as a function of the available bandwidth and for a load of 25 policy requests per PEP (i.e., in the outsourced approach, the central PDP services 25 x 4 = 100 simultaneous requests). The constrained-bandwidth

links were implemented over an Ethernet network by using a token bucket filter implementation in the *Diffserv on Linux* or *tc* tool [27]. For illustrative purposes, the hybrid architecture in these experiments was configured to have about 60% of the requests processed locally, while the remaining 40% of the requests are outsourced to the central PDP. We wrote a small utility tool to run multiple iterations and process results in an automated fashion. We conduct our experiments on a real network, which by itself provides randomness in the results; multiple iterations were considered to ensure tight 95% confidence intervals. The results are shown in Figure 8.

Figure 8 Policy response time and inter-decision time as a function of bandwidth



We note that the average policy response time per request for the outsourced architecture increases considerably in the case of a low bandwidth of 64 kb/s. For an outsourced approach, the constrained bandwidth is the major bottleneck and, in this case, renders the approach impractical. The response time for the provisioned approach is essentially independent of the available bandwidth. Even the performance of the hybrid architecture is not much affected by available bandwidth, resulting in a considerably lower policy response time as compared to the outsourced case for low values of bandwidth. This

illustrates the performance advantage of distribution of policy information and control for networks with constrained bandwidth.

In addition, we also recorded the average inter-decision time for the three architectures, reflecting the average service rate of the policy requests. Due to the additional processing caused by interacting with the remote PDP interspersed with local decision-making, the average inter-decision time is greater in the case of hybrid architecture for all three scenarios. The inter-decision time is found to be at a minimum for the completely distributed case, while it increases for the centralised case as bandwidth decreases. Thus, we can conclude that the additional processing required in the hybrid architecture seems to be the main bottleneck, while bandwidth is the major bottleneck for the outsourced approach.

5 Concluding remarks

It is clear that voice and data networks have become an integral part of our critical infrastructure. Protecting these networks from both natural and man-generated disasters must be a strong priority. Until now, the primary mechanisms for survivability of computer networks have employed route redundancy to ensure speedy recovery from link failures. Although link redundancy is an effective and needed approach to increase survivability, it should be complemented by appropriate traffic mechanisms that allow service differentiation, both at times of regular operation and when resources are stretched to their limits.

In this paper, we outline some ways in which policy-based QoS mechanisms can be used to enhance the survivability of a network. This is a work-in-progress, and our contributions to this point are to establish a taxonomy for policy-based networking approaches and to present preliminary results comparing some of the leading candidates for a policy-based networking architecture. In our considerations, we focus on wireless and mobile networks, which are more prone to link degradation and where design for route redundancy is less likely.

Based on the issues discussed in this paper, the authors are currently conducting further research in the following areas:

- the definition of a set of policies that will be used to differentiate traffic based on application requirements (real versus non-real time, low and high data rates, etc.), content (critical versus non-critical information) and user
- the establishment of a robust policy networking architecture that minimises signalling overhead when resources are scarce and possesses enough flexibility to dynamically handle changing policies and mobile users
- the development of middleware that can interface with signalling protocols needed to establish QoS and that can mark traffic according to its performance requirements
- assessment of the impact of policies on QoS interoperability among dissimilar networks across multiple administrative domains.

Acknowledgement

This work was partially funded by the Office for Naval Research (ONR) under the Navy Collaborative Integrated Information Technology Initiative (NAVCIITI).

References

1 Medhi, D. and Tipper, D. (2000) 'Multi-layered network survivability – models, analysis, architecture, framework and implementation: an overview', *Proceedings of the DARPA Information Survivability Conference and Exposition*, Hilton Head Island, SC, January.

2 Amoroso, E. (1994) *Fundamentals of Computer Security Technology*, Prentice Hall PTR.

3 DaSilva, L., Annamalai, A., Woerner, B., Srivastava, V. and Dham, V. (2002) 'Beamforming and ad hoc networking for future combat systems: investigation and simulation model', *Technical Report*, Virginia Tech, January.

4 Armitage, G. (2000) *Quality of Service in IP Networks*, Pearson Higher Education, April.

5 Xiao, X. and Ni, L.M. (1999) 'Internet QoS: a big picture', *IEEE Network*, Vol. 13, No.2, pp.8–18, March/April.

6 Zhao, W., Olshefski, D. and Schulzrinne, H. (2000) 'Internet quality of service: an overview', *Technical Report, CUCS-003-00*, Columbia University, February 2000. Available at: http://www.cs.columbia.edu/~library/TR-repository/reports/reports-2000/cucs-003-00.pdf

7 Verma, D. (2000) *Policy-Based Networking: Architecture and Algorithms*, New Riders Publishing, November.

8 Kosiur, D. (2001) *Understanding Policy-based Networking*, John Wiley & Sons, January.

9 Dovrolis, K. and Ramanathan, P. (1998) 'Resource aggregation for fault tolerance in integrated services packet networks', *Computer Communication Review*, Vol. 28, No. 2, April.

10 Braden, R., Zhang, L., Berson, S., Herzog, S. and Jamin, S. (1997) 'Resource ReSerVation Protocol (RSVP) – Version 1 Function Specification', *IETF RFC 2205*, September.

11 Srikitja, A., Tipper, D. and Medhi, D. (1999) 'On providing survivable QoS services in the next generation internet', *Proceedings of IEEE Military Communications Conference*, Vol. 2, pp.902–907.

12 Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M. and McManus, J. (1999) 'Requirements for traffic engineering over MPLS', *IETF RFC 2702*, September.

13 Rosen, E., Viswanathan, A. and Callon, R. (2001) 'Multiprotocol label switching architecture', *IETF RFC 3031*, January.

14 Thirumalasetty, S.R. and Medhi, D. (2001) 'MPLS traffic engineering for survivable book-ahead guaranteed services', *CST Technical Report*, University of Missouri-Kansas City, July.

15 Kodialam, M. and Lakshman, T.V. (2000) 'Dynamic routing of bandwidth guaranteed tunnels with restoration', in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Vol. 2, pp.902–911, March.

16 Fulp, E., Fu, Z., Reeves, D., Felix, W.S. and Zhang, X. (2001) 'Preventing denial of service attacks on quality of service', *Proc. of the DARPA Information Survivability Conference and Exposition II*, Vol. 2, pp.159–172.

17 DaSilva, L.A. (2000) 'Pricing for QoS-enabled networks: a survey', *IEEE Communication Surveys and Tutorials*, Vol. 3, No. 2, Second Quarter, pp.2–8.

18 Westerinen, A. *et al.* (2001) 'Terminology for policy-based management', *IETF RFC 3198*, November.

19 (2000) *IETF resource allocation protocol (RAP) working group*, Available at: http://www.ietf.org/html.charters/rap-charter.html

20 Yavatkar, R., Pendarakis, D. and Guerin, R. (2000) 'A framework for policy-based admission control', *RFC 2753*, January.

21 Durham, D. *et al.* (2000) 'The COPS (Common Open Policy Service) protocol', *IETF RFC 2748*, January.

22 Herzog, S. *et al.* (2000) 'COPS usage for RSVP', *IETF RFC 2749*, January.

23 Chan, K. *et al.* (2001) 'COPS usage for policy provisioning (COPS-PR)', *IETF RFC 3084*, March.

24 Fine, M. *et al.* (2002) 'Differentiated services quality of service policy information base', *IETF Internet-Draft, draft-ietf-diffserv-pib-06.txt*, work in progress, March.

25 Phanse, K., DaSilva, L. and Midkiff, S. (2002) 'A taxonomy and experimental evaluation of policy architectures for bandwidth-constrained networks', *Technical Report*, Virginia Tech. Available at: http://www.ee.vt.edu/~kphanse/unpublished.html

26 Intel COPS SDK. Available at: http://developer.intel.com/ial/cops/

27 Diffserv over Linux. Available at: http://diffserv.sourceforge.net/