Internet Draft
Expiration: May 2001
File: draft-ietf-policy-req-02.txt

Hugh Mahon
Hewlett-Packard
Yoram Bernet
Microsoft
Shai Herzog
IP Highway
John Schnizlein
Cisco Systems
November 9, 2000

Requirements for a Policy Management System

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or made obsolete by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress.''

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Copyright Notice

Abstract

This document describes why policy based management is interesting to people managing IT environments and what is needed to make

policy management address those interests. Work to date is described, as well as usage cases demonstrating how policy-based management would actually work.

The goal for this document is to provide a set of requirements for further development of standards for policy management systems. There has already been work in the area of policy management and the work to date is described as well as additional areas to be defined.

This document is the result of discussions, e-mail, and other communications within the Policy Framework Working Group and among individuals.

Table of Contents

1. Introduction

Policy based management has generated a lot of buzz in the industry lately. Unfortunately hype can create unrealistic expectations. While Policy Based Management won't solve all problems, or make IT administration a trivial task, there is a real need for Policy Based Management. So why are people interested in Policy Based Management?

Policy is essentially a matter of allocating resources in terms of business decisions. It is the translation between business terms and the configuration details necessary to produce those resource allocations that distinguishes policy management from configuration management.

Internet technology based networks are being used for more functions and by more businesses. Their ability to do business is affected by the health and abilities of their networks.

As networks grow the amount of things that need to be managed grows. Not only are there more devices to be managed, but also the number of kinds of things (e.g., capabilities, services, types of interfaces, etc.) is growing. As more kinds of things are introduced, so are more management interfaces the IT administrators must learn and use to manage the environment. In addition, many of those management tools work with individual devices, so that an administrator must duplicate the actions used to manage (configure) one device for each other device, even if they are the same type of device from the same vendor. The problem is exacerbated if the devices are from different vendors, since they must perform different tasks to manage similar capabilities. The same problem exists not just for networking, but for just about anything an IT administrator may need to manage.

In response to this situation, customers (IT administrators) have for many years been asking vendors for tools which better address their needs in managing such large and dynamic environments. Their list of desired features includes:

- centralized management
- abstracted (or simplified) management data
- commonality across devices
- automation of management tasks
- fewer interfaces
- consistency across interfaces

Centralized management requires the ability to perform management tasks via the network. Scalability factors into the requirements since a centralized system is not practical if it doesn't scale well to fit the management needs in the environment.

Abstracted (or simplified) management data fits with the fewer

interfaces objective by abstracting the functions and decision criteria across multiple devices, lending itself to the next desired feature, commonality across devices.

There are two aspects to commonality: the ability to learn how to do something once and apply that across multiple things of the same kind, and the ability to use the same data, not just similar data, across multiple things. By using the same configuration across multiple devices, the administrator can achieve consistent behavior in the managed environment and reduce, or better yet eliminate, duplication of efforts. It is this desire to use the same data across multiple devices that is behind the desire to have fewer interfaces.

Automation of management tasks is the feature that causes a change from most implementations of management tools with existing technologies (e.g., SNMP). One aspect of automation is the desire of customers to be able to re-use management data where that re-use makes sense, and for the tools to support such re-use. In other words, wherever possible, the tools support management information re-use, and do not require the administrator to duplicate information already in the management system, and can automatically get the information where it needs to go and when it is needed rather than require additional intervention by the human administrator. Automation is also key in allowing the network to operate with a minimum of human intervention (once the human administrators have specified, through management data, how the environment is to behave under given circumstances).

The key to providing a solution for these requirements is the data used to manage the environment; what that data represents, how it gets from the administrator to what the data affects, and the functionality that supports reuse and automation.

That data has been called 'policy'. Policy Based Management is the term used to describe the technologies that address the customer requirements described above.

In support of the above features, the efforts for defining Policy Based Management have focused on the data representation and properties of a repository for that information.

The use of a repository is important to support reusability of data across managed things, as well as allowing an administrator to edit existing management data (both are forms of reuse). In addition to being stored in a repository, the data must get to where it will be used (this supports the requirements of centralized management and automation). (Information distributed from a centralized repository also aids in consistency of information throughout the managed environment.)

With common policy information the administrator can use the same information to configure devices which are supposed to do the same thing (addressing centralized management, commonality across devices, and reducing the number of interfaces required

for multiple devices from different vendors). This policy information can also be abstracted to a higher level, since it will need to be device independent.

Common information does not require a common format (i.e., schema). In other words, it is possible to have common information for QoS management, and common information for security uses, but have completely different formats for the different uses of data. This would cause a duplication of information that could be common (e.g., user information use for access control), and so would be a bad thing because it would lead to greater differences between disciplines than necessary. Therefore, a common format is another requirement to support the desire for automation and fewer interfaces.

To summarize the above: centralized management leads to the need for a repository; scalability requires a means to communicate the data beyond the repository; abstraction requires a common information model; automation requires the abstraction and components to perform actions based on management data and real-time inputs.

The rest of this document describes what is necessary to make a policy-based management system work.

2. QoS Policy usage

The focus of this draft is on the requirements of policy, with an emphasis on network Quality of Service.

Policy control of data (packet) networks coincides with the convergence of voice (and video) calling and business-critical communications with interconnected local area networks (LANs). A strong motivation for users is to protect these forms of communication that are less tolerant of potential delays and congestion than applications usually using in packet networks. Because the application of policy implies unequal treatment, adequate authorization for allocations is essential.

2.1. Voice

Voice over IP requires special network allocations to ensure reasonable quality. Whether using int-serv [RFC 1633] or diff-serv [RFC 2475], priority queuing and traffic shaping are required for real-time traffic. Policy specifies how much of the network is allocated to this real-time traffic and which calls are authorized to use this alloca-

tion. This policy, in conjunction with traffic engineering, determines the configuration of queue schedulers and traffic policers across a variety of network devices.

Because calling parties might connect at different places in the network (unlike plain-old telephone service (POTS) but not roam like cellular mobility), binding personal authorization to which devices to be configured depends on the process resolving personal locations within the network topology. This binding between user identity and location is outside the policy system, but might depend on policy to determine what locations are authorized. Configuration depends on the interaction between calling authorization, location, and the details of network capacity at that location. Since this interaction is more dynamic than policy, separate processes in the configuration environment are suggested.

If the capacity for real-time calls is less than the potential level authorized, some scheduling process, at least call admission control (CAC), is also required. Since CAC is likely to be even more dynamic than station mobility, a process separate from the configuration process above is suggested. RSVP [RFC 2205] anticipates that policy, in addition to local network capacity, will determine CAC for int-serv. Capacity allocation for CAC must be local because availability or congestion is only meaningful for individual links (or queues).

To summarize: Configuration for real-time (voice) calls requires the interaction of policy, traffic engineering, user identity, and location. Call admission control requires all of this plus accounting for local resources along the path of the call.

2.2. Protected classes of traffic

Fear that traffic for critical applications might be displaced by less important traffic when both share the same network motivates interest in policy networking. As a matter of implementation, the only way to protect one class of traffic from the load applied from another class is to queue them separately. Each class must then be allocated a share of the output link capacity. This also has the advantage that each class is protected from the others, which is impossible with priority or precedence approaches; even a class allocated a small share will not be affected by loads in other classes. Priority appears to reflect business interests, "But priority is an implementation mechanism, not a service model." [RFC 1633] Resolving ordering effects among multiple levels of priority also complicates an

already difficult problem of resolving potential conflicts within a set of policy constraints.

Because queues can be scarce resources in network devices, policy should control their allocation and which traffic

sources use which queues. Traffic engineering influences the allocation of traffic classes to queues because queues are necessary only where bottlenecks cause congestion. This interaction between traffic engineering and policy is slightly different than their interaction for real-time traffic; policy creation should be constrained by the ability of the network to protect different traffic classes. Where call admission is analogous to program-language run-time, and configuration is analogous to compile-time, the availability of class separation is analogous to semantically-constrained editing of policy entry.

In many cases, traffic rates from server to client are so much larger than from client to server that classification of protected traffic can be made on the basis of server addresses, which are authenticated and authorized over long time periods. Sometimes, it may be necessary to include client authorization beyond what the application performs. Client authorization could interact with policy to imply configuration changes on a time-scale (comparable to user mobility above), or individual sessions on a multi-user (host) client (comparable to admission control above).

2.3. Guaranteed Transfer Time

The business-level specification of quality is often in terms much larger than suitable for configuration, even through consistency resolution and translation from policy. An application that demands transfer of a known (approximately) volume of data within a specified time is attractive for archival storage and content distribution. The value of this application justifies scheduling classes of traffic with capacity configurations across the path of the transfer. This application requires (time of day) scheduling along with the same requirements as for protected classes.

2.4. Policy and Services

Policy management is often discussed in terms of the services that are supported via policy. There are different kinds and levels of information required when managing a networked environment. Service management is a relatively high level view of a system. Many drafts discuss an "Olympic" service, in which there are multiple levels of service, for example: Bronze, Silver, and Gold. In such discussions Gold is better than Silver, and Silver is bet-

ter than Bronze. When actually describing the meaning of the service, though, things become more complex.

Policy is used to implement services in an environment. But policy information may not be the only representation

of the service characteristics. Services may be described in a manner that is higher-level than policy itself; that is, services may be described in a form that describes characteristics from which policy information is then derived. Such a higher level representation may be required to perform some functions in a managed environment. For example, different policies which describe different behaviors may be deployed to two entities traversed by a customer's traffic. It may be impossible to tell if a conflict exists simply by looking at the policies themselves, but it may be possible to determine if a conflict exists with the service(s) to which the customer has subscribed.

Service descriptions are beyond the scope of this draft, but the distinction between services and policies is an important one when discussing what information is used in a management system, and what the administrator needs to know in order to create and deploy policies.

3. Usage Cases

Building on the discussion in section 2, two usage cases will be described.

3.1. Simple Usage Case

A customer, Joe, has subscribed for Gold service. For this example Gold service will simply mean that Joe has higher priority than customers who subscribed for Silver or Bronze service levels. What needs to happen in order for Joe to receive the service to which he is subscribed?

This example will be shown in two contexts: an ISP environment and an enterprise environment.

3.1.1. Simple Usage Case in an ISP Environment

The ISP management personnel must configure the environment to support the Gold, Silver, and Bronze services. Within the core of the network this can be accomplished by putting in place policies which cause the devices to examine the DiffServ mark on the packet and then treat the traffic appropriately. These policies do not change frequently because they are not associated with specific customers. The more difficult part is to have the traffic appropriately marked.

Since Joe is signed up for the service, the RADIUS server could be configured to have the POP at which Joe usually logs in assign Joe a known address when he logs

in. With a known IP address the administrator can author a policy which references Joe's IP address and marks traffic coming from that address as Gold service traffic. This policy would then be deployed to Joe's POP.

In order for any traffic going to Joe's system to receive Gold treatment, that traffic must also be marked appropriately. This means that policy must be deployed to the edge devices so that they will mark traffic going to Joe's system to receive Gold service. This is accomplished by deploying policies to edge devices. These edge policies cause packets going to Joe's address to be marked so that they receive Gold service treatment. This would also be done on internal routers or switches to which the ISP's servers (e.g., mail or news servers) are attached so that traffic internal to the ISP is also appropriately marked.

Customers like Joe want to be able to see if they are getting the service they have paid the service provider for. Service providers should provide the tools which allow their customers to see information relevant to each customer's service. Such tools are beyond the scope of this document.

3.1.2. Simple Usage Case in an Enterprise Environment

As in the ISP example above, the corporate IT administrators will need to configure the core to handle the different services offered to users of the corporate network. The difference is in how the user's traffic is marked to receive the desired service.

One way for the traffic to be marked is for Joe to have a fixed IP address. The policies would then be written to recognize Joe's address and treat the traffic coming to and from that address appropriately.

A second way for the traffic to be marked correctly is that when Joe's computer is connected to the network the DHCP system recognizes that it is Joe's computer, or that it is a computer to which Gold service is to be provided, and thus notifies another management component of the event. This causes a policy to be deployed (or more likely causes an existing policy to be modified) that will mark the traffic to and from Joe's computer as receiving Gold service.

A third way for the traffic to be marked correctly is to have a sequence of events be started when Joe logs into the system. This would notify a central authority which would cause the traffic to be marked to receive Gold

service.

Yet a fourth way would be to have policy deployed to the end systems themselves. When Joe uses an application that generates traffic the networking stack on that system would mark Joe's traffic appropriately.

For all of these approaches, the network devices would also need to be configured to appropriately mark traffic going to Joe's system so that it gets the desired treatment. As can be seen, an environment with totally dynamic address assignment would require dynamic configuration changes in order to support QoS. Signaling addresses some of these issues, but introduces other issues as well.

Each of these approaches has advantages and disadvantages. Approaches two and three are the most complex and would require more elaborate management systems than approaches one and four. The fourth approach is the only method described which addresses policy associated with an individual user that would work with a multi-user system. However, the problem of solving marking traffic going to Joe on system X, and treating the traffic going to other users on system X, is not solved here.

As in the ISP example above, knowledgeable users of the network in the enterprise like to be able to review the services they are receiving.

3.1.3. Simple Usage Case - Steps to Implement

In order to implement the policy discussed above, the administrator must enter new policy (or edit old policy). Some interface, whose specification is beyond the scope of this document, is used to accomplish this task. This interface then delivers the policy information to a repository. At this step other functions may be performed, such as validation, verification, conflict detection, etc. The Policy Consumers (see [POLFRAME]) associated with the interfaces to which the policy applies will be notified that the policy has changed (or is newly available). The Policy Consumers will receive the policy information and transform it into a form suitable for the device. During this step the Policy Consumer will use information about the Policy Target to perform the information transformation. Note that noth-

ing is being stated about the architecture of the Policy Consumer or Policy Target. They may be integral, distributed, or in whatever form will accomplish receiving policy information and implementing the behaviors described by the policy information.

3.1.4. Simple Usage Case Requirements

Now let's look at what happened in the above example and see what is necessary to support it.

At least two policies would be written in this case. One is to configure the core devices. In this example a single policy could be written which specifies priority treatment based on DSCP values. The other policy would be to configure the RADIUS server to assign an IP address for Joe. A third policy may be required so that the devices at the POP recognize Joe's IP address and give his traffic the appropriate DSCP mark. The third policy (which could be an action on the RADIUS policy) would require dynamic reconfiguration in real-time in order to provide appropriate service to the user in a timely manner.

In order to perform these tasks the administrator will need to enter or edit policy and have it stored in a central repository. The policy would then be sent to the appropriate devices which must carry out the operations specified by the policies. The administrator would need to be able to associate the policies with the devices in some way. The interface the administrator uses for policy administration is beyond the scope of this document, but there must be a standardized interface for inserting into and retrieving policy information from the repository.

The next step is the repository. The actual repository must be able to support the structured nature of policy information, and support insert, search, and retrieval. The key aspect of the repository is its network accessibility. So far LDAP is the stand out example meeting this requirement. However, the environment described above is dynamic. Policy can be, and should be, relatively static. But when the administrator makes changes in policy, especially to address an existing problem in the network or to correct an incorrect policy, those changes may need to be propagated quickly. This is best done via notification rather than polling. LDAP currently does not provide a notification to LDAP clients of changes.

There may be other functionality which is logically associated with the repository. This functionality may address the notification requirements, and may also con-

tribute to the desire to have validation, verification, and conflict detection performed on new or modified policy information.

The next requirement is the transformation of policy information into device information, followed by the features in the device to enforce policy.

In addition is the need to address policy which is referring to a moving or changing set of needs, primarily users moving around in the network. This issue will only grow in importance. The issue arises not just because the origin of the traffic is moving, but the destination, meaning that more points in the network must be made aware that traffic going to that destination should receive a particular treatment.

3.2. Complex Usage Case

A more complex usage case would involve managing a particular kind of traffic across the network. For example, say that a corporate IT group decides that no more than 40% of all network traffic can be video. This will further be defined that over any link the traffic can only take up to 40% of the bandwidth of that link.

This presents many different problems to be solved. Traffic identification is a necessary component in order to enforce policy of this type, but such identification is beyond the scope of this draft. The ability to specify conditions which are used to identify traffic is a requirement for policy itself.

One way is to use signaling via RSVP to identify traffic. But this may not always be feasible, especially in this example where the intent is not to guarantee a QoS for the video traffic, but to limit its use of the corporate network bandwidth. Those who are generating video traffic may not always want to have their traffic identified as video, and so using a signal may be avoided.

The traffic must be identifiable and must be able to be specified in conditions used within the policy rule.

Policies would be deployed in the core and at the edges of the network to enforce the utilization limits. The need in the core is that multiple flows from different sources with different destinations may end up traversing the same link. Per the definition above, their aggregate bandwidth usage can be no more than 40% on any link, so the policing must occur everywhere.

For this example consider a simple policing action type

which limits bandwidth usage. This action may use shaping or dropping to police the traffic to ensure it doesn't take more than the permitted bandwidth. Because of this restriction, traffic would end up with no more bandwidth

than 40% of the slowest link it traverses. Issues of jitter and latency should be addressed in some form, possibly by other action types deployed to the same interfaces.

This leads to another topic that must be resolved for a usable policy system: the interaction and relationships between multiple policy rules, particularly of different types, on a single managed entity. For example, how to express policy rules in a way that is obvious to the administrator and device/policy translator that multiple actions are to be taken and are to work together? Later revisions of this draft will include examples in this area.

Either the policy management system must have information about the bandwidth abilities of each link, or the Policy Consumers (which convert policy into device information) must be able to translate percentage into device specific values.

3.3. Complex Usage Case - Requirements

As with the simple usage case, there must be:

- A standard interface to the policy repository.
- Network access to the policy repository.
- A way to notify components which use policy that there is new or modified policy.
- A way to transform the policy information to a form usable by devices.
- Mechanisms to enforce policy on network traffic.

In addition, this example points out the need for well defined semantic relationships between multiple policies and/or rules within the same policy, especially if they are of different action types.

4. Security Considerations

For QoS related Policy, the security needs of a Policy Management System require authentication at a minimum.

The Policy Management System contains components which send messages and read and write data.

The interactions which involve writing of data MUST ensure authentication of both parties. In other words, when a Policy Consumer connects to a Policy Management Repository, in which

the Policy Consumer writes status and configuration information to the Policy Management Repository, the Policy Consumer must authenticate itself to the Policy Management Repository, and vice-versa. The reason for this is that either end of the

communication could be false. If a true Policy Consumer wrote data to a false Policy Management Repository, the Administrator will not see the true data. If a false Policy Consumer wrote data to a true Policy Management Repository, the Administrator will see false data. Either situation means that the Administrator does not know the true state of Policy configuration in the networked environment. Similar requirements exist for the connection of the Policy UI to the Policy Repository and Policy Management Repository.

Authentication also allows ensuring the party is authorized to perform the actions taken (reading and/or writing policy and status information).

There is need to limit access (either read or write) to portions of the policy information (and status information). The policy management system (or data repository if it is to be accessed directly rather than through the policy management system) must allow establishing multiple users (or identities) in order to allow authorization of which subsets of the information the user (or component) is allowed to access.

Policy information should also be shipped with information verifying its integrity, that is, demonstrating that it has not been tampered with during transit from a trusted server or client.

When Policy is used for security purposes, it MUST be encrypted when being transported over the network.

Repositories must be as secure as reasonably possible. If a Repository resides on a general purpose host, access to the Repository data should be controlled and monitored. If the data cannot be so secured, other means, such as encryption of data in the repository, or other methods ensuring integrity should be employed.

5. Summary

Policy Based Management is not just a buzz word, or a solution looking for a problem. There is a genuine need for allowing network Administrators to be more effective by managing the network as a collective, not as a collection of individual devices each requiring a separate set of knowledge.

Today's tools allow Administrators to configure the devices which enable traffic, but the view they present to Administrators is limited, and the management of a device is the focus of the activities with those tools.

Policy information, as described in [INFOMODEL] allows that abstraction, but additional information is needed to make Policy useful. Information such as the targets of Policy,

attributes about those targets, and the association between Policy and the targets must be further defined.

Additionally, the actual architecture of a Policy Management System must be further defined in order to allow multiple vendors to have interoperable implementations. The details of such an architecture include making the Policy information available in a timely manner, and providing the Administrator (and, in the future, tools) with information about the characteristics of Policy Targets in order to allow validation of Policy and conflict detection. Additionally, Administrators need to know if Policy deployment was successful in order to know if the network will work as expected so they don't have to wait for users of the network to tell them there's a problem.

New requirements not already documented elsewhere are also documented here, such as security, and timely delivery of Policy Data.

Another requirement which is probably best addressed through a combination of data organization, techniques, and architecture, is that of dealing with a mobile (dynamic) set of clients.

To finish the summation of this document, below are bullet lists of the requirements of a Policy Management System. The items marked with an asterisk are yet to be fully defined.

Policy Data

- A way to state actions to be taken by the policy managed entity
- A way to specify under what conditions the above actions are to take place
- A way to specify to what the policy (combination of action and prerequisite conditions) pertains or is to control *
- Status information about the policy managed entity *
- Properties of policy managed entity describing capabilities *
- Semantic relationship of policy actions, both of same action type and dissimilar action types. *
- Security information (integrity, authentication, etc.) *
- A way to limit access to policy contents based on security information.

The following are tentative derivations from the requirements to be considered further.

- Policy repository communication (e.g., LDAP)
- Policy repository (may be settled by above question, e.g., if communication is LDAP)
- Notification to Policy Consumer of new/changed policy *
- Versioning of Policy Data *
- Status reporting mechanism *

6. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11.

Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

7. References

[TERMS] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", Internet RFC 2119, March 1997.

[RFC 1633] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", June 1994.

[RFC 2205] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification",

September 1997.

[RFC 2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", December 1998.

[TERMINOLOGY] J. Strassner, E. Ellesson, "Terminology for describing network policy and services", Internet Draft draft-strassner-policy-terms-01.txt, February 1999.

[IANA] Internet Assigned Numbers Authority, http://www.isi.edu/in-notes/iana/assignments/port-numbers .

[INFOMODEL] B. Moore, E. Ellesson, J. Strassner, "Policy Framework Core Information Model", Internet Draft draft-ietf-policy-core-info-model-03.txt, January 2000.

[POLFRAME] M. Stevens, W. Weiss, H. Mahon, B. Moore, J. Strassner, G. Waters, A. Westerinen, J. Wheeler, "Policy Framework", Internet Draft draft-ietf-policy-framework-00.txt, September 1999.

[COPS] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan, A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.

8. Acknowledgements

Special thanks to Mark Stevens, Bob Moore, Andrea Westerinen, Avri Doria, Cheh Goh, Ken Owens, Rick Roeling, and Brian O'Keefe for input and feedback during the development of this draft. Thanks also go to Ed Ellesson and Bert Wijnan for their guidance on what should be discussed in this document.

9. Author Information

Hugh Mahon
Hewlett-Packard Co.
3404 East Harmony Road, MS A2
Fort Collins, CO 80528-9599
USA
Phone: +1 970 898 2487
EMail: hugh_mahon@hp.com

Yoram Bernet
Microsoft
1 Microsoft Way
Redmond, WA 98052
USA
Phone: +1 206 936 9568
EMail: yoramb@microsoft.com

Shai Herzog
IPHighway
Parker Plaza, 16th Floor
400 Kelby St. Fort-Lee NJ 07024
USA
Phone: +1 201.585.0800
EMail: herzog@iphighway.com

John Schnizlein
Cisco Systems
9123 Loughran Road
Fort Washington, MD 20744
USA
Phone: +1 301 567 7126
EMail: john.schnizlein@cisco.com

10. Full Copyright Statement

developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.