Internet Engineering Task Force S. Floyd, Editor
INTERNET DRAFT Internet Architecture Board
draft-iab-considerations-03.txt October, 2002

General Architectural and Policy Considerations

Status of this Memo

Abstract

This document suggests general architectural and policy questions that the IETF community has to address when working on new standards and protocols. We note that this document contains questions to be addressed, as opposed to guidelines or architectural principles to be followed.

TO THE RFC EDITOR: PLEASE REMOVE THIS SECTION UPON PUBLICATION.

Changes from draft-iab-considerations-02.txt:

* Updated the section on SIP.

* Added questions about deployment issues.

* Added a paragraph in the Introduction clarifying the purpose of the document.

* Added to the Conclusions.

* Added a paragraph to the section on "Discussion: evolvability" on maintaining evolvability.

* Added a paragraph to the section on "Discussion: deployability" about issues that complicate deployment.

* Added citations.

* Reordered some sections.

* Other minor clarifications.

Changes from draft-iab-considerations-01.txt:

* Added a discussion on overloading.

* Added a discussion on complexity, robustness, and fragility.

* Added a section on Internationalization.

1. Introduction

This document suggests general architectural and policy questions to be addressed in our work in the IETF. This document contains questions to be addressed, as opposed to guidelines or architectural principles to be followed. These questions are somewhat similar to the "Security Considerations" currently required in IETF documents [RFC2316].

This document is motivated in part by concerns about a growing lack of coherence in the overall Internet architecture. We have moved from a world of a single, coherent architecture designed by a small group of people, to a world of a complex, intricate architecture to address a wide-spread and heterogeneous environment. Because individual pieces of the architecture are often designed by sub-communities, with each sub-community having its own set of interests, it is necessary to pay increasing attention to how each piece fits into the larger picture, and to consider how each piece is chosen. For example, it is unavoidable that each of us is inclined to solve a problem at the layer of the protocol stack and using the tools that we understand the best; that does not necessarily mean that this is the most appropriate layer or set of tools for solving this problem in the long-term.

Our assumption is that this document will be used as suggestions (but not a checklist!) of issues to be addressed by IETF members in chartering new working groups, in working in working groups, and in evaluating the output from other working groups. This document is not a primer on how to design protocols and architectures, and does not provide answers to anything.

2. Relationship to "Architectural Principles of the Internet"

RFC 1958 [RFC1958] outlines some architectural principles of the Internet, as "guidelines that have been found useful in the past, and that may be useful to those designing new protocols or evaluating such designs." An example guideline is that "it is also generally felt that end-to-end functions can best be realized by end-to-end protocols." Similarly, an example design issue from [RFC1958] is that "heterogeneity is inevitable and must be supported by design."

In contrast, this document serves a slightly different purpose, by suggesting additional architectural questions to be addressed. Thus, one question suggested in this document is the following: "Is this proposal the best long-term solution to the problem? If not, what are the long-term costs of this solution, in terms of restrictions on future development, if any?" This question could be translated to a roughly equivalent architectural guideline, as follows: "Identify whether the proposed protocol is a long-term solution or a short-term solution, and identify the long-term costs and the exit strategy for any short-term solutions."

In contrast, other questions are more open-ended, such as the question about robustness: "How robust is the protocol, not just to the failure of nodes, but also to compromised or malfunctioning components, imperfect or defective implementations, etc?" As a community, we are still learning about the degree of robustness that we are able to build into our protocols, as well as the tools that are available to ensure this robustness. Thus, there are not yet clear architectural guidelines along the lines of "Ensure that your protocol is robust against X, Y, and Z."

3. Questions.

In this section we list some questions to ask in designing protocols. Each question is discussed in more depth in the rest of this paper. We aren't suggesting that all protocol design efforts should be required to explicitly answer all of these questions; some questions will be more relevant to one document than to another. We also aren't suggesting that this is a complete list of architectural concerns.

DESIGN QUESTIONS:

Justifying the Solution:

* Why are you proposing this solution, instead of proposing something else, or instead of using existing protocols and procedures?

Interactions between Layers:

* Why are you proposing a solution at this layer of the protocol stack, rather than at another layer? Are there solutions at other layers of the protocol stack as well?

* Is this an appropriate layer in terms of correctness of function, data integrity, performance, ease of deployment, the diagnosability of failures, and other concerns?

* What are the interactions between layers, if any?

Long-term vs. Short-term Solutions:

* Is this proposal the best long-term solution to the problem?

* If not, what are the long-term costs of this solution, in terms of restrictions on future development, if any? What are the requirements for the development of longer-term solutions?

The Whole Picture vs. Building Blocks:

* Have you considered the larger context, while appropriately restricting your own design efforts to one part of the whole?

* Are there parts of the overall solution that will have to be provided by other IETF Working Groups or by other standards bodies?

EVALUATION QUESTIONS:

Weighing Benefits against Costs:

* How do the architectural benefits of a proposed new protocol compare against the architectural costs, if any? Have the architectural costs been carefully considered?

Robustness:

* How robust is the protocol, not just to the failure of nodes, but also to compromised or malfunctioning components, imperfect or defective implementations, etc?

* Does the protocol take into account the realistic conditions of the current or future Internet (e.g., packet drops and packet corruption; packet reordering; asymmetric routing; etc.)?

Tragedy of the Commons:

* Is performance still robust if everyone is using this protocol? Are there other potential impacts of widespread deployment that need to be considered?

Protecting Competing Interests:

* Does the protocol protect the interests of competing parties (e.g., not only end-users, but also ISPs, router vendors, software vendors, or other parties)?

Designing for Choice vs. Avoiding Unnecessary Complexity:

* Is the protocol designed for choice, to allow different players to express their preferences where appropriate? At the other extreme, does the protocol provide so many choices that it threatens interoperability or introduces other significant problems?

Preserving Evolvability?

* Does the protocol protect the interests of the future, by preserving the evolvability of the Internet? Does the protocol enable future developments?

* If an old protocol is overloaded with new functionality, or reused for new purposes, have the possible complexities introduced been taken carefully into account?

* For a protocol that introduces new complexity to the Internet architecture, how does the protocol add robustness and preserve evolvability, and how does it also introduce new fragilities to the system?

Internationalization:

* Where protocols require elements in text format, have the possibly conflicting requirements of global comprehensibility and the ability to represent local text content been properly weighed against each other?

DEPLOYMENT QUESTIONS:

* Is the protocol deployable?

Each of these questions is discussed in more depth in the rest of this paper.

4. Justifying the Solution.

Question: Why are you proposing this solution, instead of proposing something else, or instead of using existing protocols and procedures?

4.1. Case study: Integrated and Differentiated Services.

A good part of the work of developing integrated and differentiated services has been to understand the problem to be solved, and to come to agreement on the architectural framework of the solution, and on the nature of specific services. Thus, when integrated services was being developed, the specification of the Controlled Load [RFC2211] and Guaranteed [RFC2212] services each required justification of the need for that particular service, of low loss and bounded delay respectively.

Later, when RFC 2475 on "An Architecture for Differentiated Services" proposed a scalable, service differentiation architecture that differs from the previously-defined architecture for integrated services, the document also had to clearly justify the requirements for this new architecture, and compare the proposed architecture to other possible approaches [RFC2475]. Similarly, when the Assured Forwarding [RFC2597] and Expedited Forwarding [RFC2598] Per-Hop Behaviors of differentiated services were proposed, each service required a justification of the need for that service in the Internet.

5. Interactions between Layers.

Questions: Why are you proposing a solution at this layer of the protocol stack, rather than at another layer? Are there solutions at other layers of the protocol stack as well?

Is this an appropriate layer in terms of correctness of function, data integrity, performance, ease of deployment, the diagnosability of failures, and other concerns?

What are the interactions between layers, if any?

5.1. Discussion: The End-to-End Argument

The classic 1984 paper on "End-To-End Arguments In System Design" [SRC84] begins a discussion of where to place functions among modules by suggesting that "functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level. Examples discussed in the paper include bit error recovery, security using encryption, duplicate message suppression, recovery from system crashes, and delivery acknowledgement. Low level mechanisms to support these functions are justified only as performance enhancements." The end-to-end principle is one of the key architectural guidelines to consider in choosing the appropriate layer for a function.

5.2. Case study: Endpoint Congestion Management.

The goal of the Congestion Manager in Endpoint Congestion Management is to allow multiple concurrent flows with the same source and destination address to share congestion control state [RFC3124]. There has been a history of proposals for multiplexing flows at different levels of the protocol stack; proposals have included adding multiplexing at the HTTP (WebMux) and TCP (TCP Control Blocks) layers, as well as below TCP (the Congestion Manager) [Multiplexing].

However, the 1989 article on "Layered Multiplexing Considered Harmful" suggests that "the extensive duplication of multiplexing functionality across the middle and upper layers is harmful and should be avoided" [T89]. Thus, one of the key issues in providing mechanisms for multiplexing flows is to determine which layer or layers of the protocol stack are most appropriate for providing this functionality. The natural tendency of each researcher is generally to add functionality at the layer that they know the best; this does not necessarily result in the most appropriate overall architecture.

5.3. Case study: Layering Applications on Top of HTTP.

There has been considerable interest in layering applications on top of HTTP [RFC3205]. Reasons cited include compatibility with widely-deployed browsers, the ability to reuse client and server libraries, the ability to use existing security mechanisms, and the ability to traverse firewalls. As RFC 3205 discusses, "the recent interest in layering new protocols over HTTP has raised a number of questions when such use is appropriate, and the proper way to use HTTP in contexts where it is appropriate." Thus, RFC 3205 addresses not only the benefits of layering applications on top of HTTP, but also evaluates the additional complexity and overhead of layering an application on top of HTTP, compared to the costs of introducing a special-purpose protocol.

The web page on "References on Layering and the Internet Architecture" has pointers to additional papers discussing general layering issues in the Internet architecture [Layering].

6. Long-term vs. Short-term Solutions

Questions: Is this proposal the best long-term solution to the problem?

If not, what are the long-term costs of this solution, in terms of restrictions on future development, if any? What are the requirements for the development of longer-term solutions?

6.1. Case study: Traversing NATs.

In order to address problems with NAT middleboxes altering the external address of endpoints, various proposals have been made for mechanisms where an originating process attempts to determine the address (and port) by which it is known on the other side of a NAT. This would allow an originating process to be able to use address data in the protocol exchange, or to advertise an external address from which it will receive connections.

The IAB in [UNSAF] has outlined reasons why these proposals can be considered at best as short-term fixes to specific problems, and the specific issues to be carefully evaluated before standardizing such proposals. These issues include the identification of the limited-scope problem to be fixed, the description of an exit strategy for the short-term solution, and the description of the longer-term problems left unsolved by the short-term solution.

7. Looking at the whole picture vs. making a building block.

For a complex protocol which interacts with protocols from other standards bodies as well as from other IETF working groups, it can be necessary to keep in mind the overall picture while, at the same time, breaking out specific parts of the problem to be standardized in particular working groups.

Question: Have you considered the larger context, while restricting your own design efforts to one part of the whole?

Question: Are there parts of the overall solution that will have to be provided by other IETF Working Groups or by other standards bodies?

7.1. Case Study: The Session Initiation Protocol (SIP)

The Session Initiation Protocol (SIP) [RFC2543], for managing connected, multimedia sessions, is an example of a complex protocol that has been broken into pieces for standardization in other working groups. SIP has also involved interaction with other standardization bodies.

The basic SIP framework is being standardized by the SIP working group. This working group has focused on the core functional features of setting up, managing, and tearing down multimedia sessions. Extensions are considered if they relate to these core features.

The task of setting up a multimedia session also requires a description of the desired multimedia session. This is provided by the Session Description Protocol (SDP). SDP is a building block that is supplied by the Multiparty Multimedia Session Control (MMUSIC) working group. It is not standardized within the SIP working group.

Other working groups are involved in standardizing extensions to SIP that fall outside of core functional features or applications. The SIPPING working group is analyzing the requirements for SIP applied to different tasks, and the SIMPLE working group is examining the application of SIP to instant messaging and presence. The IPTEL working group is defining a call processing language (CPL) that interacts with SIP in various ways. These working groups occasionally feed requirements back into the main SIP working group.

Finally, outside standardization groups have been very active in providing the SIP working group with requirements. The Distributed Call Signaling (DCS) group from the PacketCable Consortium, 3GPP, and 3GPP2 are all using SIP for various telephony-related applications, and members of these groups have been involved in drafting requirements for SIP. In addition, there are extensions of SIP which are under consideration in these standardization bodies. Procedures are under development in the IETF to address proposed extensions to SIP, including a category of preliminary, private, or proprietary extensions, and to provide for the safe management of the SIP namespace [MBMWOR02].

8. Weighing architectural benefits against architectural costs.

Questions: How do the architectural benefits of a proposed new protocol compare against the architectural costs, if any? Have the architectural costs been carefully considered?

8.1. Case Study: Performance-enhancing proxies (PEPs)

RFC 3135 [RFC3135] considers the relative costs and benefits of placing performance-enhancing proxies (PEPs) in the middle of a network to address link-related degradations. In the case of PEPs, the potential costs include disabling the end-to-end use of IP layer security mechanisms; introducing a new possible point of failure that is not under the control of the end systems; adding increased difficulty in diagnosing and dealing with failures; and introducing possible complications with asymmetric routing or mobile hosts. RFC 3135 carefully considers these possible costs, the mitigations that can be introduced, and the cases when the benefits of performance-enhancing proxies to the user are likely to outweight the costs.

8.2. Case Study: Open Pluggable Edge Services (OPES)

One of the issues raised by middleboxes in the Internet involves the end-to-end integrity of data. This is illustrated in the recent question of chartering the Open Pluggable Edge Services (OPES) Working Group. Open Pluggable Edge Services are services that would be deployed as application-level intermediaries in the network, for example, at a web proxy cache between the origin server and the client. These intermediaries would transform or filter content, with the explicit consent of either the content provider or the end user.

One of the architectural issues that arose in the process of chartering the OPES Working Group concerned the end-to-end integrity of data. As an example, it was suggested that ''OPES would reduce both the integrity, and the perception of integrity, of communications over the Internet, and would significantly increase uncertainly about what might have been done to content as it moved through the network'', and that therefore the risks of OPES outweighed the benefits [CDT01].

As one consequence of this debate, the IAB wrote a document on "IAB Architectural and Policy Considerations for OPES", considering both the potential architectural benefits and costs of OPES [RFC3238]. This document did not recommend specific solutions or mandate specific functional requirements, but instead included recommendations of issues such as concerns about data integrity that OPES solutions standardized in the IETF should be required to address.

9. General Robustness Questions

Questions: How robust is the protocol, not just to the failure of nodes, but also to compromised or malfunctioning components, imperfect or defective implementations, etc?

Does the protocol take into account the realistic conditions of the current or future Internet (e.g., packet drops and packet corruption; packet reordering; asymmetric routing; etc.)?

9.1. Discussion: Designing for Robustness.

Robustness has long been cited as one of the overriding goals of the Internet architecture [Clark88]. The robustness issues discussed in [Clark88] largely referred to the robustness of packet delivery even in the presence of failed routers; today robustness concerns have widened to include a goal of robust performance in the presence of a wide range of failures, buggy code, and malicious actions.

As [ASSW02] argues, protocols need to be designed somewhat defensively, to maximize robustness against inconsistencies and errors. [ASSW02] discusses several approaches for increasing robustness in protocols, such as verifying information whenever possible; designing interfaces that are conceptually simple and therefore less conducive to error; protecting resources against attack or overuse; and identifying and exposing errors so that they can be repaired.

Techniques for verifying information range from verifying that acknowledgements in TCP acknowledge data that was actually sent, to providing mechanisms for routers to verify information in routing messages.

Techniques for protecting resources against attack range from preventing "SYN flood" attacks by designing protocols that don't allocate resources for a single SYN packet, to partitioning resources (e.g., preventing one flow or aggregate from using all of the link bandwidth).

9.2. Case Study: Explicit Congestion Notification (ECN).

The Internet is based on end-to-end congestion control, and historically the Internet has used packet drops as the only method for routers to indicate congestion to the end nodes. ECN [RFC3168] is a recent addition to the IP architecture to allow routers to set a bit in the IP packet header to inform end-nodes of congestion, instead of dropping the packet.

The first, Experimental specification of ECN [RFC2481] contained an extensive discussion of the dangers of a rogue or broken router "erasing" information from the ECN field in the IP header, thus preventing indications of congestion from reaching the end-nodes. To add robustness, the standards-track specification [RFC3168] specified an additional codepoint in the IP header's ECN field, to use for an ECN "nonce". The development of the ECN nonce was motivated by earlier research on specific robustness issues in TCP [SCWA99]. RFC 3168 explains that the addition of the codepoint "is motivated primarily by the desire to allow mechanisms for the data sender to verify that network elements are not erasing the CE codepoint, and that data receivers are properly reporting to the sender the receipt of packets with the CE codepoint set, as required by the transport protocol." Supporting mechanisms for the ECN nonce are needed in the transport protocol to ensure robustness of delivery of the ECN-based congestion indication.

In contrast, a more difficult and less clear-cut robustness issue for ECN concerns the differential treatment of packets in the network by middleboxes, based on the TCP header's ECN flags in a TCP SYN packet [RFC3360]. The issue concerns "ECN-setup" SYN packets, that is, SYN packets with ECN flags set in the TCP header to negotiate the use of ECN between the two TCP end-hosts. There exist firewalls in the network that drop "ECN-setup" SYN packets, others that send TCP Reset messages, and yet others that zero ECN flags in TCP headers. None of this was anticipated by the designers of ECN, and RFC 3168 added optional mechanisms to permit the robust operation of TCP in the presence of firewalls that drop "ECN-setup" SYN packets. However, ECN is still not robust to all possible scenarios of middleboxes zeroing ECN flags in the TCP header. Up until now, transport protocols have been standardized independently from the mechanisms used by middleboxes to control the use of these protocols, and it is still not clear what degree of robustness is required from transport protocols in the presence of the unauthorized modification of transport headers in the network. These and similar issues are discussed in more detail in [RFC3360].

10. Avoiding Tragedy of the Commons.

Question: Is performance still robust if everyone is using the new protocol? Are there other potential impacts of widespread deployment that need to be considered?

10.1. Case Study: End-to-end Congestion Control.

[RFC2914] discusses the potential for congestion collapse if flows are not using end-to-end congestion control in a time of high congestion. For example, if a new transport protocol was proposed that did not use end-to-end congestion control, it might be easy to show that a individual flow using the new transport protocol would perform quite well as long as all of the competing flows in the network were using end-to-end congestion control. To fully evaluate the new transport protocol, it is necessary to look at performance when many flows are competing, all using the new transport protocol. If all of the competing flows were using the more aggressive transport protocol in a time of high congestion, the result could be high steady-state packet drop rates and reduced overall throughput, with busy links carrying packets that will only be dropped downstream. This can be viewed as a form of tragedy of the commons, when a practice that is productive if done by only one person (e.g., adding a few more sheep to the common grazing pasture) is instead counter-productive when done by everyone [H68].

11. Balancing Competing Interests

Question: Does the protocol protect the interests of competing parties (e.g., not only end-users, but also ISPs, router vendors, software vendors, or other parties)?

11.1. Discussion: balancing competing interests

[CWSB02] discusses the role that competition between competing interests plays in the evolution of the Internet, and takes the position that the role of Internet protocols is to design the playing field in this competition, rather than to pick the outcome. The IETF has long taken the position that it can only design the protocols, and that often two competing approaches will be developed, with the marketplace left to decide between them [A02]. (It has also been suggested that "the marketplace" left entirely to itself does not always make the best decisions, and that working to identify and adopt the technically best solution is sometimes helpful. Thus, while the role of the marketplace should not be ignored, the decisions of the marketplace should also not be held as sacred or infallible.)

An example cited in [CWSB02] of modularization in support of competing interests is the decision to use codepoints in the IP header to select QoS, rather than binding QoS to other properties such as port numbers. This separates the structural and economic issues related to QoS from technical issues of protocols and port numbers, and allows space for a wide range of structural and pricing solutions to emerge.

There have been perceived problems over the years of individuals adding unnecessary complexity to IETF protocols, increasing the barrier to other implementations of those protocols. Clearly such action would not be protecting the interests of open competition. Underspecified protocols can also serve as an unnecessary barrier to open competition.

12. Designing for Choice vs. Avoiding Unnecessary Complexity:

Is the protocol designed for choice, to allow different players to express their preferences where appropriate? At the other extreme, does the protocol provide so many choices that it threatens interoperability or introduces other significant problems?

12.1. Discussion: the importance of choice

[CWSB02] suggests that "the fundamental design goal of the Internet is to hook computers together, and since computers are used for unpredictable and evolving purposes, making sure that the users are not constrained in what they can do is doing nothing more than preserving the core design tenet of the Internet. In this context, user empowerment is a basic building block, and should be embedded into all mechanism whenever possible."

As an example of choice, "the design of the mail system allows the user to select his SMTP server and his POP server" [CWSB02]. More open-ended questions about choice concern the design of mechanisms that would enable the user to choose the path at the level of providers, or to allow users to choose third-party intermediaries such as web caches, or providers for Open Pluggable Edge Services (OPES). [CWSB02] also notes that the issue of choice itself reflects competing interests. For example, ISPs would generally like to lock in customers, while customers would like to preserve their ability to change among providers.

At the same time, we note that excessive choice can lead to "kitchen sink" protocols that are inefficient and hard to understand, have too much negotiation, or have unanticipated interactions between options. For example, [P99] notes that excessive choice can lead to difficulty in ensuring interoperability between two independent, conformant implementations of the protocol.

The dangers of excessive options are also discussed in [MBMWOR02], which gives guidelines for responding to the "continuous flood" of suggestions for modifications and extensions to SIP (Session Initiation Protocol). In particular, the SIP Working Group is concerned that proposed extensions have general use, and do not provide efficiency at the expense of simplicity or robustness. [MBMWOR02] suggests that other highly extensible protocols developed in the IETF might also benefit from more coordination of extensions.

13. Preserving evolvability?

Does the protocol protect the interests of the future, by preserving the evolvability of the Internet? Does the protocol enable future developments?

If an old protocol is overloaded with new functionality, or reused for new purposes, have the possible complexities introduced been taken into account?

For a protocol that introduces new complexity to the Internet architecture, does the protocol add robustness and preserve evolvability? Does it also introduce unwanted new fragilities to the system?

13.1. Discussion: evolvability.

There is an extensive literature and an ongoing discussion about the evolvability, or lack of evolvability, of the Internet infrastructure; the web page on "Papers on the Evolvability of the Internet Infrastructure" has pointers to some of this literature [Evolvability]. Issues range from the evolvability and overloading of the DNS; the difficulties of the Internet in evolving to incorporate multicast, QoS, or IPv6; the difficulties of routing in meeting the demands of a changing and expanding Internet; and the role of firewalls and other middleboxes in limiting evolvability.

[CWSB02] suggests that among all of the issues of evolvability, "keeping the net open and transparent for new applications is the most important goal." In the beginning, the relative transparency of the infrastructure was sufficient to ensure evolvability, where a "transparent" network simply routes packets from one end-node to another. However, this transparency has become more murky over time, as cataloged in [RFC3234], which discusses the ways that middleboxes interact with existing protocols and increase the difficulties in diagnosing failures. [CWSB02] realistically suggests the following guideline: "Failures of transparency will occur - design what happens then," where examples of failures of transparency include firewalls, application filtering, connection redirection, caches, kludges to DNS, and the like. Thus, maintaining evolvability also requires mechanisms for allowing evolution in the face of a lack of transparency of the infrastructure itself.

One of the ways for maintaining evolvability is for designers of new mechanisms and protocols to be as clear as possible about the assumptions that are being made about the rest of the network. New mechanisms that make unwarranted assumptions about the network can end up placing unreasonable constraints on the future evolution of the network.

13.2. Discussion: overloading.

There has been a strong tendency in the last few years to overload some designs with new functionality, with resulting operational complexities. Extensible protocols could be seen as one of the tools for providing evolvability. However, if protocols and systems are stretched beyond their reasonable design parameters, then scaling, reliability, or security issues could be introduced. Examples of protocols that have been seen as either productively extended, or as dangerously overloaded, or both, include DNS [K02,RFC2915], MPLS [A02a], and BGP [B02]. In some cases, overloading or extending a protocol may reduce total complexity and deployment costs by avoiding the creation of a new protocol; in other cases a new protocol might be the simpler solution.

We have a number of reusable technologies, including component technologies specifically designed for re-use. Examples include SASL, BEEP and APEX. TCP and UDP can also be viewed as reusable transport protocols, used by a range of applications. On the other hand, re-use should not go so far as to turn a protocol into a Trojan Horse, as has happened with HTTP [RFC3205].

13.3. Discussion: complexity, robustness, and fragility.

[WD02] gives a historical account of the evolution of the Internet as a complex system, with particular attention to the tradeoffs between complexity, robustness, and fragility. [WD02] describes the robustness that follows from the simplicity of a connectionless, layered, datagram infrastructure and a universal logical addressing scheme, and, as case studies, describes the increasing complexity of TCP and of BGP. The paper describes a complexity/robustness spiral of an initially robust design and the appearance of fragilities, followed by modifications for more robustness that themselves introduce new fragilities. [WD02] conjectures that "the Internet is only now beginning to experience an acceleration of this complexity/robustness spiral and, if left unattended, can be fully expected to experience arcane, irreconcilable, and far-reaching robustness problems in the not-too-distant future." Citing [WD02], [BFM02] focuses on the ways that complexity increases capital and operational expenditures in carrier IP network, and views complexity as the primary mechanism that impedes efficient scaling.

14. Internationalization.

Where protocols require elements in text format, have the possibly conflicting requirements of global comprehensibility and the ability to represent local text content been properly weighed against each other?

14.1. Discussion: internationalization.

RFC 1958 [RFC1958] included a simple statement of the need for a common language:

"Public (i.e. widely visible) names should be in case-independent ASCII. Specifically, this refers to DNS names, and to protocol elements that are transmitted in text format."

The IETF has studied character set issues in general [RFC 2130] and made specific recommendations for the use of a standardized approach [RFC 2277]. The situation is complicated by the fact that some uses of text are hidden entirely in protocol elements and need only be read by machines, while other uses are intended entirely for human consumption. Many uses lie between these two extremes, which leads to conflicting implementation requirements.

For the specific case of DNS, the Internationalized Domain Name working group is considering these issues. As stated in the charter of that working group, "A fundamental requirement in this work is to not disturb the current use and operation of the domain name system, and for the DNS to continue to allow any system anywhere to resolve any domain name." This leads to some very strong requirements for backwards compatibility with the existing ASCII-only DNS. Yet since the DNS has come to be used as if it was a directory service, domain names are also expected to be presented to users in local character sets.

This document does not attempt to resolve these complex and difficult issues, but simply states this as an issue to be addressed in our work. The requirement that names encoded in a text format within protocol elements be universally decodable (i.e. encoded in a globally standard format with no intrinsic ambiguity) does not seem likely to change. However, at some point, it is possible that this format will no longer be case-independent ASCII.

15. Deployability

Question: Is the protocol deployable?

15.1. Discussion: deployability

It has been suggested that the failure to understand deployability considerations in the current environment is one of the major weakness of the IETF. As examples of deployment difficulties, RFC 2990 [RFC2990] discusses deployment difficulties with Quality of Service (QoS) architectures, various documents of the MBONE Deployment Working Group address deployment problems with IP Multicast, and the IPv6 Working Group discusses a wealth of issues related to the deployment of IPv6. [CN02] discusses how the deployment of Integrated Services has been limited by factors such as the failure to take into account administrative boundaries. Additional papers on difficulties in the evolution of the Internet architecture are available from [Evolvability].

Issues that can complicate deployment include whether the protocol is compatible with pre-existing standards, and whether the protocol is compatible with the installed base. For example, a transport protocol is more likely to be deployable if it performs correctly and reasonably robustly in the presence of dropped, reordered, duplicated, delayed, and rerouted packets, as all of this can occur in the current Internet.

16. Conclusions

This document suggests general architectural and policy questions to be addressed when working on new protocols and standards in the IETF.

The case studies in this document have generally been short illustrations of how the questions proposed in the document have been addressed in working groups in the past. However, we have generally steered away from case studies of more controversial issues, where there is not yet a consensus in the IETF community. Thus, we sidestepped suggestions for adding a case study for IKE (Internet Key Exchange) as an possible example of a protocol with too much negotiation, or of adding a case study of network management protocols as illustrating the possible costs of leaving things to the marketplace to decide. We would encourage others to contribute case studies of these or any other issues that may shed light on some of the questions in this document; any such case studies could include a careful presentation of the architectural reasoning on both sides We would conjecture that there is a lot to be learned, in terms of the range of answers to the questions posed in this document, by studying the successes, failures, and other struggles of the past.

We would welcome feedback on this document for future revisions. Feedback could be send to the editor, Sally Floyd, at floyd@icir.org.

17. Acknowledgements

This document has borrowed text freely from other IETF RFCs, and has drawn on ideas from [ASSW02], [CWSB02], [M01] and elsewhere. This document has developed from discussions in the IAB, and has drawn from suggestions made at IAB Plenary sessions and on the ietf general discussion mailing list. The case study on SIP was contributed by James Kempf, an early case study on Stresses on DNS was contributed by Karen Sollins, and Keith Moore contributed suggestions that were incorporated in a number of places in the document. The discussions on Internationization and on Overloading were based on an earlier document by Brian Carpenter and Rob Austein. We have also benefited from discussions with Noel Chiappa, Karen Sollins, John Wroclawski, and others, and from helpful feedback from members of the IESG. We specifically thank Senthilkumar Ayyasamy, John Loughney, Keith Moore, Eric Rosen, and Dean Willis and others for feedback on various stages of this document.

18. Normative References

19. Informative References

[A02] Harald Alvestrand, "Re: How many standards or protocols...", email to the ietf discussion mailing list, Message-id: <598204031.1018942481@localhost>, April 16, 2002.

[A02a] Loa Andersson, "The Role of MPLS in Current IP Network", MPLS World News, September 16, 2002. URL "http://www.mplsworld.com/archi_drafts/focus/analy-andersson.htm".

[ASSW02] T. Anderson, S. Shenker, I. Stoica, and D. Wetherall, "Design Guidelines for Robust Internet Protocols", HotNets-I, October 2002.

[BFM02] Randy Bush, Tim Griffin, and David Meyer, "Some Internet Architectural Guidelines and Philosophy", internet draft, work in progress, July 2002.

[B02] Hamid Ould-Brahim, Bryan Gleeson, Peter Ashwood-Smith, Eric C. Rosen, Yakov Rekhter, Luyuan Fang, Jeremy De Clercq, Riad Hartani, and Tissa Senevirathne, "Using BGP as an Auto-Discovery Mechanism for Network-based VPNs", draft-ietf-ppvpn-bgpvpn-auto-03.txt, internet-draft, work-in-progress, August 2002.

[CDT01] Policy Concerns Raised by Proposed OPES Working Group Efforts, email to the IESG, from the Center for Democracy & Technology, August 3, 2001. URL "http://www.imc.org/ietf-openproxy/mail-archive/msg00828.html".

[Clark88] David D. Clark, The Design Philosophy of the DARPA Internet Protocols, SIGCOMM 1988.

[CN02] Brian Carpenter, Kathleen Nichols, "Differentiated Services in the Internet", Technical Report, February 2002, URL "http://www.research.ibm.com/resources/paper_search.shtml".

[CWSB02] Clark, D., Wroslawski, J., Sollins, K., and Braden, R., "Tussle in Cyberspace: Defining Tomorrow's Internet", SIGCOMM 2002. URL "http://www.acm.org/sigcomm/sigcomm2002/adprog.html".

[Evolvability] Floyd, S., "Papers on the Evolvability of the Internet Infrastructure". Web Page, URL "http://www.icir.org/floyd/evolution.html".

[H68] Garrett Hardin, "The Tragedy of the Commons", Science, V. 162, 1968, pp. 1243-1248. URL "http://dieoff.org/page95.htm".

[K02] John C. Klensin, "Role of the Domain Name System", draft-klensin-dns-role-03.txt, internet-draft, work in progress, June 2002.

[Layering] Floyd, S., "References on Layering and the Internet Architecture", Web Page, URL "http://www.icir.org/floyd/layers.html".

[Multiplexing] S. Floyd, "Multiplexing, TCP, and UDP: Pointers to the Discussion", Web Page, URL "http://www.icir.org/floyd/tcp_mux.html".

[MBMWOR02] A. Mankin, S. Bradner, R. Mahy, D. Willis, J. Ott, B. Rosen, "Change Process for the Session Initiation Protocol (SIP)", draft-tsvarea-sipchange-03.txt, internet-draft, work-in-progress, August 2002.

[M01] Tim Moors, A Critical Review of End-to-end Arguments in System Design, 2001. URL "http://uluru.poly.edu/˜tmoors/".

[P99] Radia Perlman, "Protocol Design Folklore", in Interconnections Second Edition: Bridges, Routers, Switches, and Internetworking Protocols, Addison-Wesley, 1999.

[RFC1958] B. Carpenter, "Architectural Principles of the Internet", RFC 1958, June 1996.

[RFC2211] Wroclawski, J., "Specification of the Controlled Load Quality of Service", RFC 2211, September 1997.

[RFC2212] Shenker, S., Partridge, C., and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997.

[RFC2316] Bellovin, S., "Report of the IAB Security Architecture Workshop", RFC 2316, April 1998.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[RFC2481] K. K. Ramakrishnan and S. Floyd, A Proposal to add Explicit Congestion Notification (ECN) to IP, RFC 2481, January 1999.

[RFC2543] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol", RFC 25434, March 1999.

[RFC2597] Heinanen, J., Baker, F., Weiss, W. and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597. June 1999.

[RFC2598] Jacobson, V., Nichols, K. and K. Poduri, "An Expedited Forwarding PHB", RFC 2598, June 1999.

[RFC2915] M. Mealling, R. Daniel, "The Naming Authority Pointer (NAPTR) DNS Resource Record", RFC 2915, September 2000.

[RFC2990] G. Huston, "Next Steps for the IP QoS Architecture", RFC 2990, November 2000.

[RFC3124] H. Balakrishnan and S. Seshan, "The Congestion Manager", RFC 3124, June 2001.

[RFC3135] Border, J., Kojo, M., Griner, J., Montenegro, G. and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", RFC 3135, June 2001.

[RFC3168] K.K. Ramakrishnan, S. Floyd, and D. Black, The Addition of Explicit Congestion Notification (ECN) to IP, RFC 3168, Proposed Standard, September 2001.

[RFC3205] K. Moore, "On the use of HTTP as a Substrate", RFC 3205, February 2002.

[RFC3221] G. Huston, "Commentary on Inter-Domain Routing in the Internet", RFC 3221, December 2001.

[RFC3234] B. Carpenter and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

[RFC3238] S. Floyd and L. Daigle, "IAB Architectural and Policy Considerations for Open Pluggable Edge Services", RFC 3238, Informational, January 2002.

[RFC3360] Floyd, S., "Inappropriate TCP Resets Considered Harmful", RFC 3360, August 2002.

[SCWA99] Stefan Savage, Neal Cardwell, David Wetherall, Tom Anderson, "TCP Congestion Control with a Misbehaving Receiver", ACM Computer Communications Review, October 1999.

[SRC84] J. Saltzer, D. Reed, and D. D. Clark, "End-To-End Arguments In System Design", ACM Transactions on Computer Systems, V.2, N.4, p. 277-88. 1984.

[T89] D. Tennenhouse, "Layered Multiplexing Considered Harmful", Protocols for High-Speed Networks, 1989.

[UNSAF] L. Daigle, "IAB Considerations for UNilateral Self-Address Fixing (UNSAF)", draft-iab-unsaf-considerations-02.txt, internet-draft, work in progress, June 2002.

[WD02] Walter Willinger and John Doyle, "Robustness and the Internet: Design and Evolution", draft, March 2002, URL "http://netlab.caltech.edu/internet/".

20. Security Considerations

This document does not propose any new protocols, and therefore does not involve any security considerations in that sense. However, throughout this document there are discussions of the privacy and integrity issues and the architectural requirements created by those issues.

21. IANA Considerations

There are no IANA considerations regarding this document.

AUTHORS' ADDRESSES

Internet Architecture Board
EMail: iab@iab.org

IAB Membership at time this document was completed:

Harald Alvestrand
Ran Atkinson
Rob Austein
Fred Baker
Leslie Daigle
Steve Deering
Sally Floyd
Ted Hardie
Geoff Huston
Charlie Kaufman
James Kempf
Eric Rescorla
Mike St. Johns

This draft was created in October 2002.