IP Version 10.0: A Strawman Design Beyond IPv6

Ken Carlberg
SAIC
McLean, VA, USA
carlbergk@saic.com

Saleem Bhatti
University of St. Andrews
St. Andrews, SCT
saleem@cs.st-andrews.ac.uk

Jon Crowcroft
University of Cambridge
Cambridge, UK
jon.crowcroft@cl.cam.ac.uk

ABSTRACT
After nearly14 years since the first version of IPv6 was defined by the Internet Engineering Task Force (IETF), there is still just a minimal amount of native IPv6 deployment in today's Internet. Clearly, the evolution of IPv6 since its initial roots as the Simple Internet Protocol has turned the next generation IP effort into one lacking any significant "must have" features. This paper revisits the subject of a next generation IP and presents a new design that builds upon previous and on-going research in proposing a strawman design that we term IPv10.0. Our objective is to present a starting point for discussion of a new IP version that is extensible, introduces new architectural features, and prompts new innovative capabilities.

Categories and Subject Descriptors
C.2.2 [Computer-Communication Networks]: Network Protocols

General Terms
Design, Standardization

Keywords
Internet Protocol, IP, IPv6, IPv4

1. INTRODUCTION
The Internet Protocol (IP) has represented an interesting circular inflection point of an engineered solution leading to a vast output of deployed research. As the deployed research matured, new engineering solutions arose that led to more research both at the network layer and the layers and services defined above it. This cumulative success led to more participation to the point where the original design choices of IPv4 were nearing their limit of relevance, and its design needed to be revisited. In 1995, the IETF published the first specification of IPv6; the next generation IP designed to address its current limitations and future evolution.

However, the dearth of significant migration by the Internet community to IPv6 shows an underlying lack of interest (and possibly confidence) in a protocol touted as the definitive successor to IPv4.

ReArch'09, December 1, 2009, Rome, Italy. Copyright 2009 ACM 978-1-60558-749-3/09/12...$10.00.

One reason for this condition is the lack of a significant set of new architectural features in IPv6 that produce a "must-have" interest from the user/customer base. While some may argue that larger address space and multiple optional headers constitute an architectural change, clearly the market and Internet community have shown that such changes are of minor interest, at best, in fostering new innovations and demand.

This paper revisits the subject of a successor to IPv4 (and IPv6) and presents a new strawman design that builds upon previous and on-going research in defining a new Internet Protocol and introduces new features and design philosophies. The following sections provide some context for why a successor to the current Internet architecture is needed, and some of the technical opportunities that have been missed in the design of IPv6. Following this background information, new directions, objectives, and innovative potential features are presented to the reader as a foundation for a successor to IPv6 known as IPv10.0; we use the moniker '10.0' to acknowledge that our proposal is a starting point for further discussions and refinements.

2. BACKGROUND

2.1 IPv4 Header and Address Conservation
In the early '90s, discussions sprung up in the IETF community via the big-internet mailing list [1] on the next generation of the IP due to problems associated with address allocations. At that time, IPv4 addresses were defined as a four-class structure in the form of: A (large networks), B (mid-sized networks), C (small networks), and D (multicast). The most pressing problem at the time was the depletion of class B addresses before the year 2000 [2].

In these initial discussions on IPv4, the address pool was the most immediate issue that captured the attention of engineers and the technical press at the time. However, others pointed to an equally pressing problem of the corresponding increase in routing table size commonly referred to as the Routing Information Base (RIB). The problem also extended to the forwarding table, or Forwarding Information Base (FIB). In both of these cases, the problem is not strictly the size of the table in terms of memory, but the access time in finding an entry (a longest matching address prefix) and inserting entries in these tables in relation to the speed of incoming packets. Memory is cheap, but as all vendors will tell their clients, accessing and touching memory is the slowest part of a router/switch.

Concerns about the growing size of FIBs/RIBs have been

periodically revisited, but the subject typically took a back seat to address consumption rates. Breakthroughs in the mid to late ‘90s in both hardware and new search algorithms realized in software have allowed current top-of-the-line routers to keep up with the explosive increase in FIB/RIB size. However, continued examination of the Default Free Zone routing table [3] has led to contemplation of short-term proposed research/engineering solutions involving tunnels and a BGP-less core Internet to reduce table size as well as reduce route convergence time of the path-vector routing algorithm of BGP [4].

In parallel to the initial discussions of a new IP header in the early '90s, the Internet community adopted two directions that were meant to be short-term fixes to the immediate issue of address depletion. One approach involved the use of Network Address Translators (NAT) to conserve the advertisement of globally reachable addresses [6]. A second approach involved removing the original class structure of IPv4 addresses into one that was literally classless, which led to Classless Inter-Domain Routing (CIDR)[5]. The objective was to add a measure of aggregation of smaller class C address prefix classes by ISPs as well as a de-aggregation of larger class A and B into blocks assigned to various registries for their redistribution to ISPs and their customers. Initially, the reassignment and aggregation had the positive result of maintaining slower growth in FIBs, while at the same time, new address prefixes were assigned to enterprise networks joining the Internet. However, the growth in FIBs returned to an exponential rate (and later a high linear rate) due to de-aggregation stemming from adding multi-homed and load-balanced address assignments [3].

2.3 IPv6

In the '93-'94 time frame, several candidates for replacing IPv4 emerged. One of the top candidates was Simple IP (SiiP)[1], whose key features were: (a) minimal fixed header with one or more "next" headers, and (b) larger address space that doubled the four byte sources and destination addresses of IPv4 [9]. In the former case, SiiP brought a minimal, simplistic design along with a path to extensibility for future growth with its ability to add a series of "Next" headers at the network layer. In the case of (b), SiiP responded to the address depletion problem, but it did not take on the issue of a separate identifier.

Another top candidate for IPv6 was Paul's IP (PIP), named after its author Paul Francis [8]. PIP was more ambitious in its design than SiiP and took a different approach to the addressing structure in two substantial ways. The first involved a distinct separation of the identifier from the locator. The second was that addresses were both hierarchical and variable in length. The hierarchical feature led itself to a one-to-one correlation with hierarchical distributing routing, while the variable length led itself to an optimal as-needed address length that could reflect a source route. Note that source routes allow the source to decide the path to a destination instead of total reliance on the network

In '94, a compromise was reached between the two primary candidates, SiiP and PIP, that was to be known as Simple IP Plus (SIPP). In this compromise, the majority of the features of SiiP were retained, but the addressing structure would be hierarchical in nature. In addition, the address would retain the same binding of locator and identifier to that of IPv4, thus retaining the problem in supporting mobility and NATs. In '95, the compromise led to the first version of IPv6 [9].

One thing to note in the first version of IPv6 is that instead of doubling the size of the IPv4 address, it was quadrupled to 128 bits. This, of course, added to concerns of bandwidth-constrained communication systems, but that was not considered of great importance in the early to mid ‘90s. A subsequent version of IPv6 was standardized in '98 [10], which removed the hierarchical structure of the address and essentially removed the last bit of compromise between SiiP and PIP.

As a result, a constructive feature for hierarchical routing was also removed from the design. This had a direct impact on the NIMROD link state routing protocol [11] being advanced in the IETF as a potential successor to BGP and its path vector design.

2.3.1 History of Locator and Identifier Split

There have been four moments in ARPAnet/Internet history where researchers/engineers have brought up the issue of the locator and identifier split as separate and distinct fields. The first instance was documented in [12] published in 1977. In this note, the authors observed that TCP would not be able to seamlessly support mobile nodes with IPv4's addressing structure.

The second time the debate occurred was in the early '90s, as mentioned above in the PIP proposal, when the successor to IP was being debated within the IETF. A third debate occurred in 1996, after the first version of IPv6 was defined, but before the second and current day version of IPv6[2]. In this third attempt, Mike O'Dell suggested a format of 8+8, where the first 8 bytes represented the location and the second 8 bytes represented a unique end-point identifier [13].

The fourth attempt was initiated by the Internet Architecture Board (IAB) [14] and has been revisited by the IRTF for the past two years, with several proposals being put forth stemming from an IAB report [15]. There is no clear cut "winner" in this latest attempt, but the continual revisiting of the subject shows dissatisfaction with IPv6's addressing structure and its impact on FIB size, mobility, NATs, and distributed systems that make use of shared file systems.

3. BYPASSING IPv6

There is an old saying in the IETF community concerning the relevance of protocols -- "let the market decide". At a macro level of the Internet, the "market" has shown a great reluctance, at best, to adopt IPv6. It has reached a point where mandates have been issued to push at the micro-level agencies like the DoD to migrate their systems to IPv6. Various reasons exist for the lack of migration to IPv6 since its first inception in 1995. One opinion is that the class-less IPv4 and NATs have been a resounding success in extending the life of IPv4. However, it is understood that this success only delays the inevitable need for change.

---

[1] The actual acronym for the Simple IP design was (SIP), but this document uses the acronym SiiP in order to avoid confusion with the Session Initiation Protocol (SIP) used today for Voice over IP.

[2] While [16] requires 64 bits of the address be used as interface identifiers, these identifiers may or may not have global scope.

In terms of recent deployment, Arbor Networks issued a report[3] in August of 2008 on the current state of IPv6 from the perspective of service providers. The data collection involved samplings taken over a one-year period. At its peak (in October 2007), IPv6 traffic represented just under one hundredth of one percent of Internet traffic.

A more fundamental argument in the lack of migration to IPv6 is a lack of a desirable "must have" feature in the protocol that draws users and operators to it and that cannot be reasonably supported in IPv4. Other than drastically larger address space and a path to multiple optional headers at the network layer, IPv6 is quite wholly backwards-compatible to IPv4.

3.1 Outside the Mainstream

Constrained connectivity, such as those found in low bandwidth MANET/sensor networks and high-delay geosynchronous SATCOM, are atypical to the vast majority of networks connected to the Internet. As a result, there were very few vocal proponents advocating the separation of locator and identifier in addresses, or prudence in the size of the new IP header. And this latter aspect, the overhead of header efficiency, becomes of strong interest for low bandwidth systems. As an example, IPv4 traffic is generally accepted to be bi-modal: 44% of packets are of lengths 40 to 100 bytes, and 37% of packets are between 1400 and 1500 bytes [18]. Given IPv4 headers of 20 bytes, the worse case scenario implies an overhead of 50%. In the case of IPv6 and its minimal header size of 40 bytes, the overhead is doubled.

4 IPv10.0

The most significant design choice for IP that separated it from circuit-switched architectures was its use of globally unique source and destination addresses versus locally significant circuit identifiers. This simple element led to inter-packet multiplexing, soft-state paradigms, and realization of the end-to-end principle. It can be argued that there have been no fundamental architectural changes from IPv4 to IPv6, thus contributing to its lack of popularity. In recognizing this lack of fundamental change in designing a successor to IPv4, we have guided our design of IPv10.0 on three distinct new features.

Separate Identifier from Locator: We borrow this design from previous research and discussions. The feature facilitates approaches to reducing the routing table size compounded by multi-homing. It also simplifies support for mobile hosts and networks, leverages use of localized addressing using NAT-like functions, and provisioning of end-to-end state at the transport layer. A trade-off in making this separation is that an additional mapping is required to bind and disseminate current location with an identifier. One approach to support this mapping is the addition of Resource Records (RR) to the Domain Naming System (DNS) correlating to Locator and Identity values as presented in [17]. In this approach, an 'address' in the IPv10.0 sense is a concatenation of the current Locator and Identifier value, which has a level of temporal stability/permanence indicated by the DNS RR by the use of the DNS time-to-live (TTL) value.

[3] http://asert.arbornetworks.com/2008/08/the-end-is-near-but-is-ipv6/

Headers and Trailers: Since its inception, IP has been based strictly on the definition of a Header. Prior to this approach, some previous network designs in the '60s included Tails appended to the end of a data stream as a means of indicating the termination of a message as well as updating any current state [19]. Our design for IPv10.0 revisits the use of Tails and expands on their responsibilities. Specifically, in our design of IPv10.0, we divide responsibility of the two overhead components so that the Header primarily contains FIB-related data (e.g., Locators), and information to interpret the Header and body of the IP packet. The Tail predominantly contains end-to-end or hop-by-hop information that would typically not be used to find a next hop entry, examples being: Explicit Congestion Notification (ECN) bits, digital signatures, and the hop limit.

Minimal and Extensible Design: One of the strengths of IPv6 is its design principal of a minimal and extensible header. Over time and subsequent specifications, the extensibility feature of IPv6 was weakened with a relatively small size of the Next Header field and a completely backwards-compatible Differentiated Services field. With respect to minimalism, IPv10.0 follows the design choice of IPv6 and removes the fragmentation and header checksum fields found in IPv4. IPv10.0 also reduces the size of the Locator to 32 bits and maintains the size of the identifier to 64 bits to allow for IEEE EUI-64 values. As a result, the default size of an IPv10.0 Header and Tail is 288 bits, compared to 320 bits for IPv6 and 160 bits for IPv4.

Concerning the subject of extensibility, both the IPv10.0 Header and Trailer have a *Next* field whose cumulative size exceeds that of IPv6's Next Header field. IPv10.0 also allows optional Tail(s) to be appended at the end of a packet by a node or router along the path toward the destination. This feature permits transit domains to bind temporary state to packets, which can be removed as the packet is forwarded to the next administrative domain. More importantly, the ability to efficiently append temporary Headers/Trailers contributes to the evolutionary migration of purely an end-to-end architecture to one that optionally involves middle components.

4.1 Packet Design

This section presents a more in-depth view of various Headers and Tails for IPv10.0. Figure 3 below shows a default Header. Unlike other designs of IP, this header contains a Forwarding Identifier *(FI)* field of value "00", indicating that the fields used to access the FIB are Locators that have no connotation associated with Identifiers. It is conceivable that the *Next Hdr* field would point to another IPv10 Header that contained information populated in a FIB. But in most cases, the *Next Hdr* field would point to a header at a layer above IP. This minimal responsibility and focus on predominantly FIB based information accentuates fast path processing and pushes non-FIB-related information to the Tail portion of an IPv10 packet.

The *Ver* field identifies the version of IP used for the packet, and the *Payload Length* field points to the end of the payload portion of the packet, excluding the Tail. Since all IPv10.0 packets have a tail of at least 12 bytes, each examination of a packet will point to the default Tail to determine if there are additional optional Tails to be examined and processed.

The *Source Locator* field contains unicast values, meaning that it identifies a single, logical fate-sharing location for end-to-end communication. The *Destination Locator* field may contain unicast, multicast, or anycast values – the particular choice is determined by the value stored in the four-bit *FI* field. In the case of Figure 3 below, the *FI* field is set to "00", indicating that both Source and Destination locators are fixed length unicast fields.

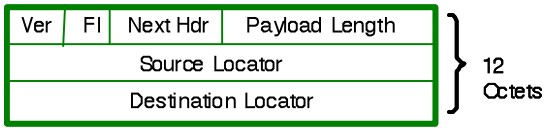

Figure 3: IPv10 Header, with FI="00"

The format of each field is hierarchical and correlates to the hierarchical allocation structure in use today by Internet Assigned Numbers Authority (IANA) and the various regional registries. Routing protocols outside of IPv10.0 provide more specific rules for defining the actual hierarchical structure of a Locator.

In the case of Figure 4, the *FI* field is set to "01", indicating that the field used to access the FIB is a Flow Identifier – a shortcut to quickly access a FIB much in the way Multi-Protocol Label Switched (MPLS) routers accomplishes its fast lookup operation.

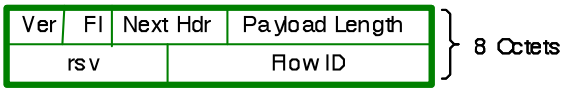

Figure 4: IPv10 Header, with FI="01"

It is expected that this optional header would be pre-pended to an IPv10.0 packet by an ingress router that has established cut-through route along the path towards the destination host. An egress router at the boundary of an administrative domain would then remove the pre-pended header and forward the original IPv10.0 packet to the next hop.

Initially, there will be four values defined for the *FI* field. Beyond those described above, "02" indicates a unicast Source Locator and a multicast Destination Locator. An *FI* value of "03" indicates a unicast Source Locator and an Anycast Destination Locator. If in the future it is determined that a variable length or larger Locator space is needed, then the remaining *FI* values will be allocated to reflect these changes.

4.2 Tail

Figure 5 shows a default IPv10 Tail with additional optional Tails. As in the case of the header(s), a *Ver* field is used to identify its version, which is set to 00. The *Diff-Serv* field is comprised of 10 bits; four more than that assigned to IPv4 and IPv6. The *"C"* and *"E"* flags correspond to the ECN Congestion Experience and ECN Capable Transport bits, respectively.

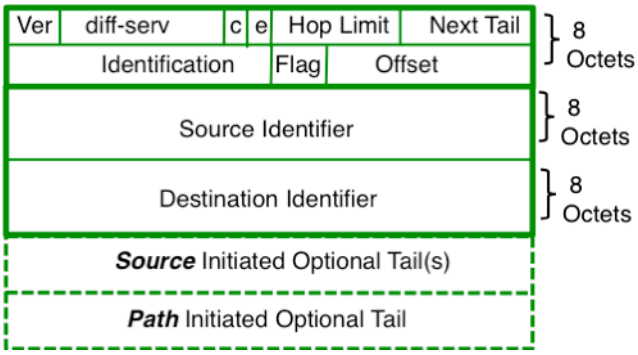

Figure 5: IPv10 Tail with Optional Tails

The *Next Tail* field identifies optional appended Tails, if any. A value of "00" in the field indicates there are no more additional Tails. If additional Tails are added to the packet, those appended by the source remain with the packet on an end-to-end basis. Routers or intermediate nodes may also append optional temporary Tail(s) to any packet. However, it is assumed that a downstream node/router will remove it either because it has fulfilled its purpose, has reached an administrative boundary, or is not recognized or supported.

The IPv10 Tail also has an Identification/Flag/Offset collection of fields that mimic the same fields and responsibilities of IPv4 to address the potential for fragmented packets. The precise definition of these fields may change from future discussions on the topic, but a placeholder is added to the current design of IPv10.0.

4.2.1 Optional Tails

Local Security Tail: One of the strengths of IPv6 is its ability to carry a series of Next sub-headers. However, these sub-headers are only affixed by the source and remain so from end-to-end. A new approach to consider is the adding/removal of Tails that denote security connotations. IPv4 and IPv6 already support strong encryption on an end-to-end basis, but these features are rarely applied on a hop-by-hop basis because of computational and management constraints. On the other hand, local and less stringent security techniques could be applied to reduce the impact of in-path security attacks.

4.3 Strengths and Weaknesses of Tails

The inclusion of Tails is probably the most controversial design choice of IPv10.0. Its most glaring weakness is that it is incompatible with today's Application Specific Integrated Circuit (ASIC) used in routers due to their inability to perform deep packet inspection beyond 50-100 octets (depending on the manufacturer). In addition, current ASIC deployments do not incorporate multi-core/multi-threaded processing of packet headers. However, the authors view these real-world constraints as current engineering limitations that will change over time and as a need arises.

A benefit of Tails is that it presents a design approach that efficiently adds or removes field(s) not originally inserted by the source host. The current practice of adding new types of information along a path involves encapsulation of a new header. This approach is commonly applied when packets transit different

versions of IP and different IP layer addressing schemes are deployed.

However, in cases where the paths of packets only transits IPv4 or IPv6 nodes, then the use of encapsulation to insert a new field, and thus new in-band state, is grossly inefficient. And this inefficiency is compounded if several nodes along the path choose to add new fields that may only be relevant to segment(s) of a path.

From a broader architectural perspective, Tails represent a paradigm shift in the selection and placement of fields in the overhead segments of data packets. Instead of the classic end-to-end model in the design of a header (where no intermediate node is expected to augment a header's construct), IPv10.0's tails represent a design choice to consider in-path changes to overhead information. This feature not only opens the door for more creative features along a path, but also helps revisit previous research efforts in routing (e.g., NIMROD) and services (e.g., active networking).

5. RELATED WORK

The Forwarding Identifier field of IPv10.0 shares similarities with OSI Network Services Access Point (NSAP) and its hierarchical structure. NSAPs are 20 octets in length and are initially divided into the Initial Domain Part (IDP) and the Domain Specific Part (DSP). The IDP specifies the format and authority responsible for assigning the DSP part of the NSAP address. And both the IDP and DSP are further subdivided into additional authority and identifier fields [20]. One of the drawbacks of NSAPs is its complexity and embedded administrative fields that have no relevance in forwarding packets toward the destination. The *FI* field in IPv10.0 provides flexibility with minimal responsibilities.

6. SUMMARY

This paper presents a new version of IP that introduces new architectural designs beyond that of IPv4 and IPv6, thus leading to new, innovative research in network design, operations, and routing. We retain the spirit of IPv6 in terms of stressing simplicity and minimalism in the selection of fields for IPv10. However, we also introduce new architectural elements that offer new capabilities and a path for future growth beyond that defined and available in both IPv4 and IPv6.

7. ACKNOWLEDGEMENTS

The authors appreciate additional feedback from Isidor Kouvelas, Atanu Ghosh, and Randall Atkinson.

8. REFERENCES

[1] Big-Internet mailing list archive, http://www.sobco.com/ipng/archive/big-i/
[2] Postel, J., "Internet Protocol", RFC 791, IETF, September 1981
[3] Huston, G., "Growth of the BGP Table – 1994 to Present", http://bgp.potaroo.net, 2008
[4] Summary of Proposals, http://trac.tools.ietf.org/group/irtf/trac/wiki/RoutingResearchGroup, 2008
[5] Rekhter, Y., T. Li, "An Architecture for IP Address Allocation with CIDR", RFC 1518, IETF, Sep 1993
[6] Egevang, K., P. Francis, "The IP Network Address Translator", RFC 1631, IETF, May 1994
[7] Deering, S. "Simple IP", Presentation in Proceedings of the 25'th IETF, Nov, 1992
[8] Francis, P., "Pip Header Processing", RFC 1622, IETF, May 1994
[9] Hinden, R., S. Deering, "IP Version 6 Addressing Architecture", RFC 1884, IETF, Dec 1995
[10] Deering, S. R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, IETF, Dec 1998
[11] Castineyra, I., et al, "The Nimrod Routing Architecture", RFC 1992, IETF, Aug 1996
[12] Bennet, C, et al, "Issues in the Interconnection of Datagram Networks", Internet Engineering Note #1, July 1977
[13] O'Dell, M., "8+8 – An Alternate Addressing Architecture for IPv6", Internet Draft, Work-In-Progress, Oct 1996
[14] Kaat, M., "Overview of 1999 IAB Network Layer Workshop", RFC 2956, IETF, Oct 2000
[15] Meyer, D., L Zhang, K Fall, "Report from the IAB Workshop on Routing and Addressing", RFC 4984, IETF, Sep 2007
[16] Hinden, R., S. Deering, "IP Version 6 Addressing Architecture, RFC 4291, IETF, Feb 2006
[17] Atkinson, R., S. Bhatti, "A proposal for Unifying Mobility with Multi-Homing, NAT, and Security", 5'th ACM International Workshop on Mobility Management and Wireless Access, Oct, 2007
[18] John, W., S. Tafvelin, "Analysis of Internet Backbone Traffic and Header Anomalies Observed", Internet Measurement Conference, ACM SIGCOMM, Oct 2007
[19] Campbell, K., "DCS AUTODIN Interface and Control Criteria", DCA Circular 370-D175-1 Standard, Dec 1984
[20] Colella, R., et al, "Guidelines for OSI NSAP Allocation in the Internet", RFC 1237, IETF, July 1991