

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: November 27, 2018

Z. Kahn, Ed.  
LinkedIn  
J. Brzozowski, Ed.  
Comcast  
R. White, Ed.  
LinkedIn  
May 26, 2018

**Requirements for IPv6 Routers**  
**draft-ietf-v6ops-ipv6rtr-reqs-04**

**Abstract**

The Internet is not one network, but rather a collection of networks. The interconnected nature of these networks, and the nature of the interconnected systems that make up these networks, is often more fragile than it appears. Perhaps "robust but fragile" is an overstatement, but the actions of each vendor, implementor, and operator in such an interconnected environment can have a major impact on the stability of the overall Internet (as a system). The widespread adoption of IPv6 could, particularly, disrupt network operations, in a way that impacts the entire system.

This time of transition is an opportune time to take stock of lessons learned through the operation of large-scale networks on IPv4, and consider how to apply these lessons to IPv6. This document provides an overview of the design and architectural decisions that attend IPv6 deployment, and a set of IPv6 requirements for routers, switches, and middleboxes deployed in IPv6 networks. The hope of the editors and contributors is to provide the necessary background to guide equipment manufacturers, protocol implementors, and network operators in effective IPv6 deployment.

**Status of This Memo**

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted 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."

This Internet-Draft will expire on November 27, 2018.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](https://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">1.1.</a>	Contributors . . . . .	<a href="#">3</a>
<a href="#">1.2.</a>	Acknowledgments . . . . .	<a href="#">4</a>
<a href="#">1.3.</a>	Use and Applicability . . . . .	<a href="#">4</a>
<a href="#">2.</a>	Review of the Internet Architecture . . . . .	<a href="#">5</a>
<a href="#">2.1.</a>	Robustness Principle . . . . .	<a href="#">5</a>
<a href="#">2.2.</a>	Complexity . . . . .	<a href="#">7</a>
<a href="#">2.2.1.</a>	Elegance . . . . .	<a href="#">7</a>
<a href="#">2.2.2.</a>	Trade-offs . . . . .	<a href="#">8</a>
<a href="#">2.3.</a>	Layered Structure . . . . .	<a href="#">9</a>
<a href="#">2.4.</a>	Routers . . . . .	<a href="#">10</a>
<a href="#">3.</a>	Requirements Related to Device Management and Security . . .	<a href="#">12</a>
<a href="#">3.1.</a>	Programmable Device Access . . . . .	<a href="#">12</a>
<a href="#">3.2.</a>	Human Readable Device Access . . . . .	<a href="#">13</a>
<a href="#">3.3.</a>	Supporting Zero Touch Provisioning for Connected Devices	13
<a href="#">3.4.</a>	Device Protection against Denial of Service Attacks . . .	<a href="#">15</a>
<a href="#">4.</a>	Requirements Related to Telemetry . . . . .	<a href="#">15</a>
<a href="#">4.1.</a>	Device State and Traceability . . . . .	<a href="#">16</a>
<a href="#">4.2.</a>	Topology State and Traceability . . . . .	<a href="#">16</a>
<a href="#">4.3.</a>	Flow State and Traceability . . . . .	<a href="#">17</a>
<a href="#">5.</a>	Requirements Related to IPv6 Forwarding and Addressing . . .	<a href="#">17</a>
<a href="#">5.1.</a>	The IPv6 Address is not a Host Identifier . . . . .	<a href="#">17</a>
<a href="#">5.2.</a>	Router IPv6 Addresses . . . . .	<a href="#">18</a>
<a href="#">5.3.</a>	The Maximum Transmission Unit . . . . .	<a href="#">19</a>
<a href="#">5.4.</a>	ICMP Considerations . . . . .	<a href="#">20</a>
<a href="#">5.5.</a>	Machine Access to the Forwarding Table . . . . .	<a href="#">21</a>
<a href="#">5.6.</a>	Processing IPv6 Extension Headers . . . . .	<a href="#">22</a>
<a href="#">5.7.</a>	IPv6 Operation by Default . . . . .	<a href="#">22</a>
<a href="#">5.8.</a>	IPv6 Only Operation . . . . .	<a href="#">22</a>



<a href="#">5.9.</a>	<a href="#">Prefix Length Handling in IPv6 Packet Forwarding . . . . .</a>	<a href="#">23</a>
<a href="#">5.10.</a>	<a href="#">IPv6 Mobility Support . . . . .</a>	<a href="#">23</a>
<a href="#">6.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">23</a>
<a href="#">6.1.</a>	<a href="#">Robustness and Security . . . . .</a>	<a href="#">23</a>
<a href="#">6.2.</a>	<a href="#">Programmable Device Access and Security . . . . .</a>	<a href="#">24</a>
<a href="#">6.3.</a>	<a href="#">Zero Touch Provisioning and Security . . . . .</a>	<a href="#">24</a>
<a href="#">6.4.</a>	<a href="#">Defaulting to IPv6 Forwarding and Security . . . . .</a>	<a href="#">24</a>
<a href="#">7.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">25</a>
<a href="#">8.</a>	<a href="#">Conclusion . . . . .</a>	<a href="#">25</a>
<a href="#">9.</a>	<a href="#">References . . . . .</a>	<a href="#">25</a>
<a href="#">9.1.</a>	<a href="#">Normative References . . . . .</a>	<a href="#">25</a>
<a href="#">9.2.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">25</a>
	<a href="#">Authors' Addresses . . . . .</a>	<a href="#">32</a>

## **[1.](#) Introduction**

This memo defines and discusses requirements for devices that perform forwarding for Internet Protocol version 6 (IPv6). The "use and applicability" section below contains more information on the specific target of this draft, and the envisioned use of the draft.

Readers should recognize that while this memo applies to IPv6, routers and middleboxes IPv6 packets will often also process IPv4 packets, forward based on MPLS labels, and potentially process many other protocols. This memo will only discuss IPv4, MPLS, and other protocols as they impact the behavior of an IPv6 forwarding device; no attempt is made to specify requirements for protocols other than IPv6. The reader should, therefore, not count on this document as a "sole source of truth," but rather use this document as a guide.

For IPv4 router requirements, readers are referred to [[RFC1812](#)]. For simplicity, the term "devices" is used interchangeably with the phrase "routers and middleboxes" and the term "routers" throughout this document. These three terms represent stylistic differences, rather than substantive differences.

This document is broken into the following sections: a review of Internet architecture and principles, requirements relating to device management, requirements related to telemetry, requirements related to IPv6 forwarding and addressing, and future considerations. Following these sections, a short conclusion is provided for review.

### **[1.1.](#) Contributors**

Shawn Zandi, Pete Lumbis, Fred Baker, James Woodyatt, Erik Muller, Lee Howard, and Joe Clarke contributed significant text and ideas to this draft.



## **1.2. Acknowledgments**

The editors and contributors would like to thank Ron Bonica, Lorenzo Coitti, Brian E. Carpenter, Tim Chown, Peter Lothberg, and Mikael Abrahamsson for their comments, edits, and ideas on the text of this draft.

## **1.3. Use and Applicability**

The conceived use of this draft is as a reference point. The first part of the draft is designed to help IPv6 implementors and network operators to understand Internet and Internetworking technologies, so they can better understand the context of IPv6. The second part of this draft outlines a common set of requirements for devices which are designed to forward IPv6 traffic. This can include (but is not limited to) the devices described below.

- o Devices which are primarily designed to forward traffic between more than two interfaces. These are normally referred to by the Internet community as routers or, in some cases, intermediate systems.
- o Devices which are designed to modify packets rather than "just" forwarding them. These are often referred to by the Internet community as middleboxes. See [[RFC7663](#)] for a fuller definition of middleboxes.

This draft is not designed to apply to consumer devices, such as smart devices (refrigerators, light bulbs, garage door openers, etc.), Internet of Things (IoT) devices, cell phones primarily used as an end user device (such as checking email, social media, games, and use as a voice device), and other devices of this class. It is up to each provider or equipment purchaser to determine how best to apply this document to their environment.

The intended use of this document is for operators to be able to point to a common set of functionality which should be available across all IPv6 implementations. Several members of the community have argued there is no common set of IPv6 features; rather each deployment of IPv6 calls for different feature sets. However, the authors of this draft believe outlining a common set of features expected of every IPv6 forwarding device is useful. Specifically:

- o If every IPv6 deployment situation is unique, and requires a different set of features, there will not be a solid definition of what an IPv6 forwarding device is, or performs. This fragments the concept of IPv6 forwarding devices in an unhelpful way, especially as IPv6 deployment is already seen as difficult.



- o It encourages developers and vendors to code a multitude of different IPv6 stacks, one for each possible set of features. This fragments the experience with these stacks, potentially preventing the development of a well designed, fully featured stacks the entire community can rely on.

Because this document is designed to be a reference point rather than a best common practice or a standard, this document does not use [\[RFC2119\]](#) upper case "must" and "should" throughout. Rather, it uses lower case "must" and "should" throughout, anticipating operators will find such guidance clear and useful.

## **[2.](#) Review of the Internet Architecture**

The Internet relies on a number of basic concepts and considerations. These concepts are not explicitly called out in any specification, nor do they necessarily impact protocol design or packet forwarding directly. This section provides an overview of these concepts and considerations to help the reader understand the larger context of this document.

### **[2.1.](#) Robustness Principle**

Every point where multiple protocols interact, is an interaction surface that can threaten the robustness of the overall system. While it may seem the global Internet has achieved a level of stability that makes it immune to such considerations, the reality is every network is a complex system, and is therefore subject to massive non repeatable unanticipated failures. Postel's Robustness Principle countered this problem with a simple statement, explicated in [\[RFC1122\]](#): "Be conservative in what you do, and liberal in what you accept from others."

However, since this time, it has been noted that following this law allows errors in protocols to accumulate over time, with overall negative effects on the system as a whole. [\[RFC1918\]](#) describes several points in conjunction with this principle that bear updating based on further experience with large-scale protocol and network deployments within the Internet community, including:

- o Applications should deal with error states gracefully; an application should not degrade in a way that will cause the failure of adjacent systems when possible. For instance, when a routing protocol implementation fails, it should not do so in a way that will cause the spreading of or continued existence of false reachability information, nor should it fail in a way that overloads adjacent routers or interacting protocols and causing a cascading failure.





- o It is best to assume the network is filled with poor implementations and malevolent actors, both of which will find every possible failure mode over time.
- o It is best to assume every technology will be used to the limits of its technical capabilities, rather than assuming a particular protocol's scope of use will align (in any way) with the intent of the original designer(s). [[RFC5218](#)] defines a wildly successful protocol as one that "far exceeds its original goals, in terms of purpose (being used in scenarios far beyond the initial design), in terms of scale (being deployed on a scale much greater than originally envisaged), or both." Successful implementations attract more functionality, much like a few nodes in a scale free graph eventually become connectivity hubs.
- o Protocols and implementations change over time. A corollary of the assumption that protocols will be used until they reach their technical limits is that protocols will change over time as they gain new functionality. [[RFC5218](#)] points out several problems with "wild success" in a protocol: undesirable side effects, performance problems, and becoming a high value attack target. Protocol and implementation design should take into account use cases that have not yet been thought of by building flexibility into protocols. Protocols should also remained focused on a narrow range of use cases; it is often wise to invent a new protocol than to extend a single protocol into a broad set of use cases.
- o Protocols are sometimes replaced or updated to new versions in order to add new capabilities or features. Updating a protocol requires great care in providing for a transition mechanism between older and newer versions. [[RFC8170](#)] provides sound advice on protocol transition planning and mechanisms.
- o Obscure, but legal, protocol features are often ignored or left unimplemented. Protocols must handle receiving unexpected information gracefully so they do not fail because of incomplete or partial implementations. Protocols should avoid specifying contradictory states, or features that will cause interoperability issues if multiple implementations choose to implement different feature sets.
- o Monocultures are almost always bad. While multiple implementations can represent an interaction surface which increases complexity, particularly if a broad set of protocol capabilities and/or implementation features are used, using the same implementation at every point in a deployment results in a mono-culture. In a monoculture, a single event can trigger a



defect in every router, causing a network failure. Mono-cultures must be carefully balanced against interaction surfaces; often this is best accomplished by using multiple implementations and minimal, widely implemented, and well understood protocol features.

A summary of the points above might be this: It is important to work within the bounds of what is actually implemented in any given protocol, and to leave corner cases for another day. It is often easy to assume "virtual oceans" are easier to boil than physical ones, or for an ocean to appear much smaller because it is being implemented in software. This is often deceptive. It is never helpful to boil the ocean whether in a design, an implementation, or a protocol.

## **2.2. Complexity**

Complexity, as articulated by Mike O'Dell (see [[RFC3439](#)]), is "the primary mechanism which impedes efficient scaling, and as a result is the primary driver of increases in both capital expenditures (CAPEX) and operational expenditures (OPEX)." At the same time, complexity cannot be "solved," but rather must be "managed." The simplest and most obvious solution to any problem is often easy to design, deploy, and manage. It's also often wrong and/or broken. As much as developers, designers, and operators might like to make things as simple as possible, hard problems require complex solutions. See Alderson and Doyle [[COMPLEXHARD](#)] for a discussion of the relationship between hard problems and complex solutions.

The following sections contain observations which apply to the management of complexity in both protocol and network design.

### **2.2.1. Elegance**

Elegance should be the goal of protocol and network design. Rather than seeking out simple solutions because they are simple, seek out solutions that will solve the problem in the simplest way possible (and no simpler). Often this will require:

- o Ensuring the goal is actually the goal. Many times the goal is taken from the operational realm into the protocol design realm before enough thought has been applied to ensure the correct problem is being addressed.
- o Seeing the problem from different angles, trying to break the problem up in multiple ways; and trying, abandoning, and rebuilding ideas and implementations until a better way is found.



- o Sometimes the complexity of the solution will overwhelm the use case; sometimes it is better to leave the apparent problem unsolved, or allow the community time and space to find a simpler solution.

### **2.2.2. Trade-offs**

There are always trade-offs. For any protocol, network, or operational design decision, there will always be a trade-off between at least two competing goals. If some problem appears to have a single solution without trade-offs, this doesn't mean the trade-offs don't exist. Rather, it means the trade-offs haven't been discovered yet. In the area of protocol and network design, these trade-offs often take the form of common "choose two of three" situations, such as "quick, cheap, high quality." In network and protocol design, the trade-offs are often:

- o The amount of state carried in the system and the speed at which it changes, or simply the state. The amount of state required to operate a system as it scales tends to be nonlinear. Some instances of this are described in [\[RFC3439\] section 2.2.1](#), the Amplification Principle.
- o The number of interaction surfaces between the components that make up the complete system, and the depth of those interaction surfaces. Some examples of surfaces are described in [\[RFC3439\]section 2.2.2](#), the Coupling Principle. Layering is essentially a form of abstraction; all abstractions are subject to the law of leaky abstractions, [\[LEAKYABS\]](#) which states: "all nontrivial abstractions leak."
- o The desired optimization, including efficient use of network resources, optimal support for business objectives, and optimal support for a specific set of applications.

These three make up a "triangle problem." For instance, to increase the optimization of traffic flow through a network generally requires adding more state to the control plane, leading to problems in complexity due to amplification. To reduce amplification, the control plane (or perhaps the various functions the control plane serves) can be broken up into subsystems, or modules. Breaking the control plane up into subsystems, however, introduces interaction surfaces between the components, which is another form of complexity. [\[RFC7980\]](#) provides a good overview of network complexity; in particular, [section 3](#) of that document provides some examples of complexity trade-offs.



### 2.3. Layered Structure

The Internet data plane is organized around broad top and bottom layers, and much thinner middle layer. This is illustrated in the figure below.

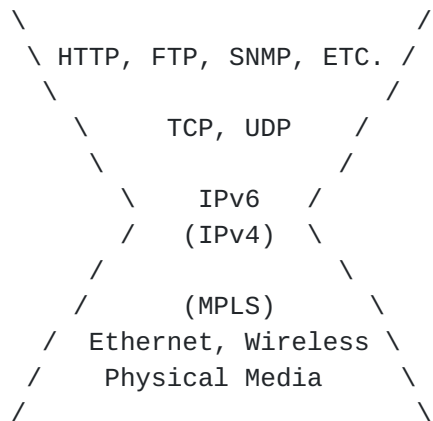


Figure 1

This layering emulates or mirrors many naturally occurring systems; it is a common strategy for managing complexity (see Meyer's presentation on complexity). [COMPLEXLAYER] The single protocol in the center, IPv6, serves to separate the complexity of the lower layers from the complexity of the upper layers. This center layer of the Internet ecosystem has traditionally been called the Network Layer, in reference to the Department of Defense (DoD) [DoD] and OSI models. [OSI] The Internet ecosystem includes two different protocols in this central location.

- o IPv4, an older network protocol that, it is anticipated, will be replaced over time as the Internet ecosystem standardizes on IPv6
- o IPv6, a newer network protocol that is being adopted

MPLS is often used as a "middle" sub-transport layer, and at other times as "middle" sub data link layer; hence MPLS is difficult to classify within the strictly hierarchical model depicted here. These protocols are often treated as if they exist in strict hierarchical layers with a well defined and followed Application Programming Interface (API), data models, Remote Procedure Calls (RPCs), sockets, etc. The reality, however, is there are often solid reasons for violating these layers, creating interaction surfaces that are often deeper than intended or understood without some experience. Beyond this, such layering mechanisms act as information abstractions. It is well known that all such abstractions leak (see above on the law of leaky abstractions). Because of these intentional and





unintentional leakages of information, the interactions between protocols is often subtle.

## 2.4. Routers

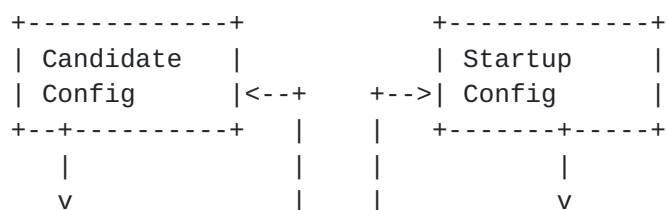
A router connects to two or more logical interfaces and at least one physical interface. A router processes packets by:

- o Receiving a packet through an interface
- o Stripping the data link, physical header, or tunnel encapsulation off the packet
- o Examining the packet for errors, and determining if this packet needs to be punted to another process on the router
- o Looking up the destination in a local forwarding table
- o Rewriting the data link and/or physical layer header
- o Transmitting the packet out an interface

When consulting the forwarding table, the router searches for a match based on:

- o The longest prefix containing the destination address (this is the most common matching element)
- o A label, such as a flow label or MPLS label
- o The source address or other header fields (not common)

The router then examines the information in the matching entry to determine the next hop, or rather the next logically connected device to forward the packet to. The next hop will either be another router, which will presumably carry the packet closer to the final destination, or it will be the destination host itself. The following figure provides a conceptual model of a router; not all routers actually have this set of tables and interactions, and some have many more moving parts. This model is simply used as a common reference to promote understanding.





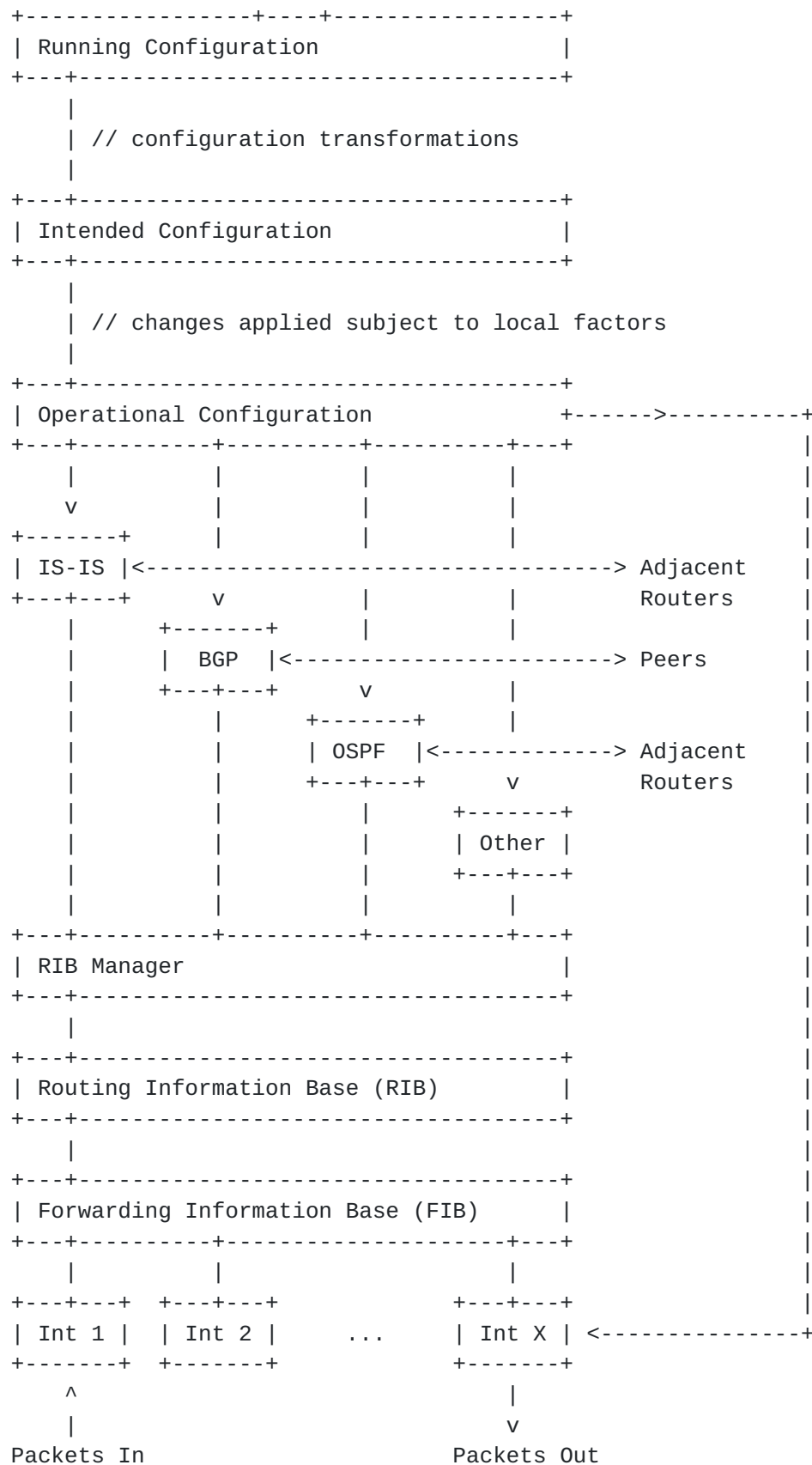




Figure 2

The configuration datastores in this figure follow [[RFC8342](#)].

### **3. Requirements Related to Device Management and Security**

Network engineering began in the era of Command Line Interfaces (CLIs), and has generally stayed with these CLIs even as the Graphical User Interface (GUI) has become the standard way of interacting with almost every other computing device. Direct human interaction with routers and middleboxes in large-scale and complex environments, however, tends to result in an unacceptably low Mean Time Between Mistakes (MTBM), directly impacting the overall availability of the network. In reaction to this, operators have increased their reliance on automation, specifically targeting machine to machine interfaces, such as Remote Procedure Calls (RPCs) and Application Programming Interface (API) solutions, to manage and configure routers and middleboxes. This section considers the various components of device management.

Across all interface types, devices should provide and use complete, idempotent, stateless configurations. Further, default settings should be accessible in some way, even if they are hidden by default for configuration readability.

#### **3.1. Programmable Device Access**

Configuration primarily relates to the startup, candidate, running, intended, and operational configurations in the router model shown above. In order to deploy networks at scale, operators rely on automated management of router configuration. This effort has traditionally focused on screen scraping and other proprietary methods of "reading" and "writing" configuration information through a CLI. In the future, operators expect to move towards open source/ open standards YANG models, regardless of how these are encoded and/ or carried (or marshaled).

Vendors and implementors should implement machine readable interfaces with overlays to support human interaction, rather than human readable interfaces with overlays to support machine to machine interaction. Emphasis should be placed on machine to machine interaction for day to day operations, rather than on human readable interfaces, which are largely used in the process of troubleshooting. Within the realm of machine to machine interfaces, emphasis should be placed on marshaling information in YANG models.

To support automated router configuration, IPv6 routers and routers should support YANG configuration, including (but not limited to):



- o Openconfig models [[OPENCONF](#)] related to the protocols configured on the device, interface state, and device state
- o [[RFC8343](#)]: A YANG Data Model for Interface Management
- o [[RFC7224](#)]: IANA Interface Type YANG Module
- o [[RFC8344](#)]: A YANG Data Model for IP Management
- o [[RFC7317](#)]: A YANG Data Model for System Management
- o [[RFC8349](#)]: A YANG Data Model for Routing Management (NMDA Version)

### **3.2. Human Readable Device Access**

To operate a network at scale, operators rely on the ability to access routers and middleboxes to troubleshoot and gather state manually through a number of different interfaces. These interfaces should provide current device configuration, current device state (such as interface state, packets drops, etc.), and current control plane contents (such as the RIB in the figure above). In other words, manual interfaces should provide information about the router (the whole device stack).

To support manual state gathering and troubleshooting, IPv6 routers and middleboxes should support:

- o TELNET ([[RFC0854](#)]): TELNET should be disabled by default, but should be available for operational purposes as required or as configured by the operator
- o SSH ([[RFC4253](#)]): SSH should be the default access for IPv6 capable routers
- o All network devices supporting IPv6 must support access through an Ethernet management port

### **3.3. Supporting Zero Touch Provisioning for Connected Devices**

To operate a network at scale, operators rely on protocols and mechanisms that reduce provisioning time to a minimum. The preferred state is zero touch provisioning; plug a new router in and it just works without any manual configuration. The closer an operator can come to this ideal, the more MTBM and Operational Expenses (OPEX) can be reduced -- important goals in the real world. IPv6 routers should support several standards, including, but not limited to:





- o [[I-D.ietf-dhc-rfc3315bis](#)]: Dynamic Configuration Protocol for IPv6 must be supported.
- o [[RFC4862](#)]: IPv6 Stateless Address Autoconfiguration (SLAAC) must be supported, and must be enabled by default on all router interfaces. SLAAC must be able to be disabled by operators who prefer to use some other mechanism for address management and assignment (specifically for customer facing edge ports).
- o [[RFC7217](#)]: Semantically Opaque Interface Identifiers should be supported unless there's a need to embed MAC address.
- o [[RFC7934](#)]: Host Address Availability, the ability to assign multiple addresses to a host, should be supported.
- o [[RFC7527](#)]: Enhanced Duplicate Address Detection should be supported.
- o [[RFC7527](#)]: Enhanced Duplicate Address Detection may be disabled for manually configured interfaces.
- o [[RFC8028](#)]: First-Hop Router Selection by Hosts, specifically [section 2.1](#), which says a router should be able to send a PIO with both the L and A bits cleared.
- o [[RFC3810](#)]: Routers supporting IPv6 must support Multicast Listener Discovery Version 2
- o [[RFC7772](#)]: Routers supporting IPv6 should support Reducing Energy Consumption of Router Advertisements
- o [[RFC8273](#)]: Routers supporting IPv6 should support Unique IPv6 Prefix per Host

The provisioning of Domain Name Systems (DNS) system information is a contentious topic, based on provider, operating system, interface, and other requirements. This document therefore addresses the mechanisms that must be included in IPv6 router implementations, but leaves the option of what to configure and deploy to the network operator. Routers supporting IPv6, and intended for user facing connections, must support:

- o [[RFC3646](#)]: DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6) if DHCPv6 is supported.
- o [[RFC8106](#)]: IPv6 Router Advertisement Options for DNS Configuration. This includes the ability to send Router Advertisements (RAs) with DNS information.



Whether these are enabled by default, or require extra configuration, is left as an exercise for providers and implementation developers to determine on a case by case basis.

### **3.4. Device Protection against Denial of Service Attacks**

Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are unfortunately common in the Internet globally; these types of attacks cost network operators a great deal in opportunity and operational costs in prevention and responses. To provide for effective counters to DoS and DDoS attacks directly on routers:

- o Manufacturers and system integrators should test and clearly report the packet/traffic load handling capabilities of devices with and without various encryption methods enabled
- o Routers should be able to police traffic destined to the control plane based on the rate of traffic received, including the ability to police individual flows, targeted services, etc., at individual rates as described in [[RFC6192](#)]
- o Ideally, devices should be able to statefully filter traffic destined to the control plane

There are other useful techniques for dealing with DDoS attacks at the network level, including: transferring sessions to a new address and abandoning the address under attack, using BGP communities to spread the attack over multiple ingress ports and "consume" it, and requiring mutual authentication before allocating larger resource pools to a connection. These techniques are not "device level," and hence are not considered further here.

## **4. Requirements Related to Telemetry**

Telemetry relates to information devices push to systems used to monitor and track the state of the network. This applies to individual devices as well as the network as a system. Two major challenges face operators in the area of telemetry:

- o Information that is laid out primarily for human, rather than machine, consumption. While human consumption of telemetry is important in some situations, this information should be supplied in a form that focuses on machine readability with an overlay or interpreter that allows human consumption.
- o Software systems that require information to be queried (or polled or even pushed) on a per-item basis. This form of organization can produce a lot of information, and a lot of individual packets,



very quickly, overwhelming monitoring systems and consuming a large amount of available network resources. Instead, telemetry should be focused on bulk collection.

There are three broad categories of telemetry: device state and traceability, topology state and traceability, and flow traceability. These three roughly correspond to the management plane, the control plane, and the forwarding plane of the network. Each of the sections below considers one of these three telemetry types.

#### **4.1. Device State and Traceability**

Ideally, the entire network could be monitored using a single modeling language to ease implementation of telemetry systems and increase the pace at which new software can be deployed in production environments. In real deployments, it is often impossible to reach this ideal; however, reducing the languages and methods used, while focusing on machine readability, can greatly ease the deployment and management of a large-scale network. Specifically, IPv6 routers should support:

- o [[RFC6241](#)] and [[RFC8040](#)]: NETCONF/RESTCONF transporting telemetry formatted according to YANG (see above)
- o [[I-D.ietf-i2rs-yang-l2-network-topology](#)]: An I2RS model for layer 2 topologies
- o [[I-D.ietf-netconf-yang-push](#)]: YANG Datastore Subscription
- o [[RFC5424](#)]: Syslog
- o gRPC based telemetry interfaces [[GRPC](#)]
- o Simple Network Management Protocol (SNMP) MIBs as appropriate

Syslog and SNMP access for telemetry should be considered "legacy," and should not be the focus of new telemetry access development efforts.

#### **4.2. Topology State and Traceability**

IPv6 routers are part of a system of devices that, combined, make up the entire network. Viewing the network as a system is often crucial for operational purposes. For instance, being able to understand changes in the topology and utilization of a network can lead to insights about traffic flow and network growth that lead to a greater understanding of how the network is operating, where problems are developing, and how to improve the network's performance. To support



systemic monitoring of the network topology, IPv6 devices should support at least:

- o [[RFC5424](#)]: North-Bound Distribution of Link-State and Traffic Engineering (TE) Information using BGP
- o [[I-D.ietf-i2rs-yang-l2-network-topology](#)]: An I2RS model for layer 2 topologies
- o [[RFC8346](#)]: An I2RS model for layer 3 topologies
- o [[RFC8345](#)]: A Data Model for Network Topologies

### **4.3. Flow State and Traceability**

Network operators frequently need to observe and understand the types, sources, and destinations of traffic passing through devices. For example, information about traffic flows may be used to identify abuse (such as DDOS attacks) or to plan network expansions based on traffic patterns. To support insight and analysis of this traffic, IPv6 devices should support IPFIX as described in [[RFC7011](#)], PSAMP as described in [[RFC5474](#)], or some other flow state mechanism.

In-situ Operational and Management (iOAM) is a technology that being developed at the time of this writing; see [[I-D.ietf-ippm-ioam-data](#)]. Operators and vendors should consider the deployment of iOAM to provide deeper information about flow and topology information.

## **5. Requirements Related to IPv6 Forwarding and Addressing**

There are a number of capabilities that a device should have to be deployed into an IPv6 network, and several forwarding plane considerations operators and vendors need to bear in mind. The sections below explain these considerations.

### **5.1. The IPv6 Address is not a Host Identifier**

The IPv6 address is commonly treated as a host identifier; it is not. Rather, it is an interface identifier that describes the topological point where a particular host connects to the Internet. Specifically:

- o The IPv6 address will change when a device changes where it connects to the network.
- o A single host can have multiple addresses. For instance, a host may have one address per interface, or multiple addresses assigned





through different mechanisms, or through multiple connection points.

- o A single IPv6 address may represent many hosts, as in the case of a group of hosts reachable through a multicast address, or a set of services reachable through an anycast address.

Because the host address may change at any time, it is generally harmful to embed IPv6 addresses inside upper layer headers to identify a particular host.

## **5.2. Router IPv6 Addresses**

Internet Routing Registries may allocate a network operator a wide range of prefix lengths (see [[RFC6177](#)] for further information). Within this allocation, network operators will often suballocate address space along nibble boundaries (/48, /52, /56, /60, and /64) for ease of configuration and management. Several common practices are:

- o Each multiaccess interface is allocated a /64
- o Point-to-point links are allocated a /64, but should be addressed with a longer prefix length to prevent certain kinds of denial of service attacks ([[RFC6547](#)] originally mandated 64 bit prefix lengths on point-to-point links; [[RFC6164](#)] explains possible security issues with assigning a 64 bit prefix length to a point-to-point, and recommends a /127 instead)
- o Although aggregation is typically only performed to the nibble boundaries noted above, variances are possible
- o Loopback addresses are assigned a /128

Given these common practices, routers designed to run IPv6 should support the following addressing conventions:

- o The default prefix length on any interface other than a loopback should be a /64
- o Configuring a prefix length longer than a /64 on any multi-access interface should require additional configuration steps to prevent manual configuration errors
- o Routers must not assume IPv6 prefix lengths only on nibble boundaries



- o Routers should support any prefix length shorter or greater than /64
- o Loopback interfaces should default to a /128 prefix length unless some additional configuration is undertaken to override this default setting
- o Routers must be able to generate link local addresses on all links and/or interfaces using stateless address autoconfiguration (see [[RFC6434](#)]).

### **5.3. The Maximum Transmission Unit**

The long history of the Maximum Transmission Unit (MTU) in networks is not a happy one. Specific problems with MTU sizing include:

- o Many different default sizes on different media types, from very small (576 bytes on X.25) to very large (17914 bytes on 16Mbps Token Ring)
- o Many different ways to calculate the MTU on any given link; for instance a 9000 byte MTU can be calculated as 8184 bytes on one operating system, 8972 on another, and 9000 on a third
- o The increasing use of tunnel encapsulations in the network; for instance MPLS over GRE over IP over...
- o The wide variety of default MTUs across many different end hosts and operating systems
- o The general ineffectiveness of path MTU discovery to operate correctly in the face of packet filters and rate limiters (see the section on ICMP filtering below)
- o Lower speed links at the network edge which require a lot of time to serialize a packet with a large MTU
- o Increased jitter caused by the disparity between large and small packet size across a lower bandwidth links

The final point requires some further elucidation. The time required to serialize various packets at various speeds are:

- o 64 byte packet onto a 10Mb/s link: .5ms
- o 1500 byte packet onto a 10Mb/s link: 1.2ms
- o 9000 byte packet onto a 10Mb/s link: 7.2ms



- o 64 byte packet onto a 100Mb/s link: .05ms
- o 1500 byte packet onto a 100Mb/s link: .12ms
- o 9000 byte packet onto a 100Mb/s link: .72ms

A 64 byte packet trapped behind a single 1500 byte packet on a 10Mb/s link suffers 1.2ms of serialization delay. Each additional 1500 byte packet added to the queue in front of the 64 byte packet adds an additional 1.2ms of delay. In contrast, a 64 byte packet trapped behind a single 9000 byte packet on a 10Mb/s link suffers 7.7ms of serialization delay. Each additional 9000 byte packet added to the queue adds an additional 7.2ms of serialization delay. The practical result is that larger MTU sizes on lower speed links can add a significant amount of delay and jitter into a flow. On the other hand, increasing the MTU on higher speed links appears to add negligible additional delay and jitter.

The result is that it costs less in terms of overall systemic performance to use higher MTUs on higher speed links than on lower speed links. Based on this, increasing the MTU across any particular link may not increase overall end-to-end performance, but can greatly enhance the performance of local applications (such as a local BGP peering session, or a large/long standing elephant flow used to transfer data across a local fabric), while also providing room for tunnel encapsulations to be added with less impact on lower MTU end systems.

The general rule of thumb is to assume the largest size MTU should be used on higher speed transit only links in order to support a wide array of available link sizes, default MTUs, and tunnel encapsulations. Routers designed for a network core, data center core, or use on the global Internet should support at least 9000 byte MTUs on all interfaces. MTU detection mechanisms, such as IS-IS hello padding, described in [[RFC3719](#)], should be enabled to ensure correct point-to-point MTU configuration. Devices should also support:

- o [[RFC8201](#)]: Path MTU Discovery for IP version 6
- o [[RFC4821](#)]: Packetization Layer Path MTU Discovery

#### **5.4. ICMP Considerations**

Internet Control Message Protocol (ICMP) is described in [[RFC0792](#)] and [[RFC4443](#)]. ICMP is often used to perform a traceroute through a network (normally by using a TTL expired ICMP message), for Path MTU discovery, and, in IPv6, for autoconfiguration and neighbor



discovery. ICMP is often blocked by middleboxes of various kinds and/or ICMP filters configured on the ingress edge of a provider network, most often to prevent the discovery of reachable hosts and network topology. Routers implementing IPv6:

- o Should rate limit the generation of ICMP echo and echo responses by default (for instance, using a token bucket method as described in [\[RFC4443\]](#)). The device should support the configuration of not generating ICMP echo, echo response, and time exceeded packets to prevent topology discovery.
- o Should rate limit the generation of ICMP error messages with a token bucket method as described in [\[RFC4443\]](#). Rate limits should be narrow enough to (a) protect the device's ability to generate packets and (b) reduce the usefulness of ICMP error packets as part of a distributed denial of service attack. Limits should be generous enough to allow successful path MTU discovery and traceroute. For example, in a small/mid-size device, the possible defaults could be bucket size=100, refill rate=100/s. Larger devices can afford more generous rate limits.
- o Should implement the filtering suggestions in [\[I-D.gont-opsec-icmp-ingress-filtering\]](#)
- o Should not filter Destination Unreachable or Packet Too Big ICMP error messages by default, as this has negative impacts on many aspects of IPv6 operation, particularly path MTU discovery.

There are implications for path MTU discovery and other useful mechanisms in filtering and rate limiting ICMP. The trade-off here is between allowing unlimited ICMP, which would allow path MTU detection to work, or limiting ICMP in a way that prevents negative side effects for individual devices, and hence the operational capabilities of the network as a whole. Operators rightly limit ICMP to reduce the attack surface against their network, as well as the opportunity for "perfect storm" events that inadvertently reduce the capability of routers and middleboxes. Hence ICMP can be treated as "quasi-reliable" in many situations; existence of an ICMP message can prove, for instance, that a particular host is unreachable. The non-existence of an ICMP message, however, does not prove a particular host exists or does not.

### **5.5. Machine Access to the Forwarding Table**

In order to support treating the "network as a whole" as a single programmable system, it is important for each router have the ability to directly program forwarding information. This programmatic interface allows controllers, which are programmed to support





specific business logic and applications, to modify and filter traffic flows without interfering with the distributed control plane. While there are several programmatic interfaces available, this document suggests that the I2RS interface to the RIB be supported in all IPv6 routers. Specifically, these drafts should be supported to enable network programmability:

- o [[I-D.ietf-i2rs-fb-rib-data-model](#)]: Filter-Based RIB Data Model
- o [[I-D.ietf-i2rs-fb-rib-info-model](#)]: Filter-Based RIB Information Model
- o [[I-D.ietf-i2rs-rib-data-model](#)]: A YANG Data Model for Routing Information Base (RIB)
- o [[RFC7922](#)]: I2RS Traceability

## **5.6. Processing IPv6 Extension Headers**

(To be added)

## **5.7. IPv6 Operation by Default**

If a device forwards and/or originates IPv4 packets by default (without explicit configuration by the operator), it should forward and/or originate IPv6 packets by default. See the security considerations section below for reflections on the automatic configuration of IPv6 forwarding in parallel with IPv4.

## **5.8. IPv6 Only Operation**

While the transition to IPv6 only networks may take years (or perhaps decades), a number of operators are moving to deploy IPv6 on internal networks supporting transport and data center fabric applications more quickly. Routers and middleboxes that support IPv6 should support IPv6 only operation, including:

- o Link Local addressing must be configurable and usable as the primary address on all interfaces on a device.
- o IPv4 and/or MPLS should not be required for proper device operation. For instance, an IPv4 address should not be required to determine the router ID for any protocol. See [\[RFC6540\]](#) [section 2](#).
- o Any control plane protocol implementations must support the recommendations in [\[RFC7404\]](#) for operation using link local addresses only.



### **5.9. Prefix Length Handling in IPv6 Packet Forwarding**

Routers must support IPv6 destination lookups in the forwarding process on a single bit prefix length increments, in accordance with [\[RFC7608\]](#).

### **5.10. IPv6 Mobility Support**

Mobile IPv6 [\[RFC6275\]](#) and associated specifications, including [\[RFC3776\]](#) and [\[RFC4877\]](#) allow a node to change its point of attachment within the Internet, while maintaining (and using) a permanent address. All communication using the permanent address continues to proceed as expected even as the node moves around. At the present time, Mobile IP has seen only limited implementation. More usage and deployment experience is needed with mobility before any specific approach can be recommended for broad implementation in hosts and routers. Consequently, routers may support [\[RFC6275\]](#) and associated specifications (these specifications are not required for IPv6 routers).

## **6. Security Considerations**

This document addresses several ways in which devices designed to support IPv6 forwarding. Some of the recommendations here are designed to increase device security; for instance, see the section on device access. Others may intersect with security, but are not specifically targeted at security, such as running IPv6 link local only on links. These are not discussed further here, as they improve the security stance of the network. Other areas discussed in this draft are more nuanced. This section gathers the intersection between operational concerns and security concerns into one place.

ICMP security is already considered in the section on ICMP; it will not be considered further here. Link local only addressing will increase security by removing transit only links within the network as a reachable destination.

### **6.1. Robustness and Security**

Robustness, particularly in the area of error handling, largely improves security if designed and implemented correctly. Many attacks take advantage of mistakes in implementations and variations in protocols. In particular, any feature that is unevenly implemented among a number of implementations often offers an attack surface. Hence, reducing protocol complexity helps reduce the breadth of attack surfaces.



Another point to consider at the intersection of robustness and security is the issue of monocultures. Monocultures are in and of themselves a potential attack surface, in that finding a single failure mode can be exploited to take an entire network (or operator) down. On the other hand, reducing the number of implementations for any particular protocol will decrease the set of "random" features deployed in the network. These two goals will often be opposed to one another. Network designers and operators need to consider these two sides of this trade-off, and make an intelligent decision about how much diversity to implement versus how to control the attack surface represented by deploying a wide array of implementations.

### **6.2. Programmable Device Access and Security**

Programmable interfaces, including programmable configuration, telemetry, and machine interface to the routing table, introduce a large attack surface; operators should be careful to ensure this attack surface is properly secured. Specifically:

- o Prevent external access to any administrative access points used for device programmability
- o Use AAA systems to ensure only valid devices and/or users access devices
- o Rate limit the change rate and protect management interfaces from DoS and DDoS attacks

Such interfaces should be treated no differently than SSH, SFTP, and other interfaces available to manage routers and middleboxes.

### **6.3. Zero Touch Provisioning and Security**

Zero touch provisioning opens a new attack surface; insider attackers can simply install a new device, and assume it will be autoconfigured into the network. A "simple" solution would be to install door locks, but this will likely not be enough; defenses need to be layered to be effective. It is recommended that devices installed in the network need to contain a hardware or software identification system that allows the operator to identify devices that are installed in the network.

### **6.4. Defaulting to IPv6 Forwarding and Security**

Operators should be aware that devices which forward IPv6 by default can introduce a new attack surface or new threats without explicit configuration. Operators should verify that IPv6 policies, including filtering, match or fulfill the same intent as any existing IPv4



policies when deploying devices capable of forwarding both IPv4 and IPv6.

## **7. IANA Considerations**

This document has no actions for IANA.

## **8. Conclusion**

The deployment of IPv6 throughout the Internet marks a point in time where it is good to review the overall Internet architecture, and assess the impact on operations of these changes. This document provides an overview of a lot of these changes and lessons learned, as well as providing pointers to many of the relevant documents to understand each topic more deeply.

## **9. References**

### **9.1. Normative References**

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

### **9.2. Informative References**

[COMPLEXHARD]

Alderson, D. and J. Doyle, "Contrasting Views of Complexity and Their Implications For Network-Centric Infrastructures", 2010, <<http://ieeexplore.ieee.org/abstract/document/5477188/?reload=true>>.

[COMPLEXLAYER]

Meyer, D., "Macro Trends, Architecture, and the Hidden Nature of Complexity", 2010, <<http://www.slideshare.net/dmm613/macro-trends-complexityandsdn-32951199>>.

[DoD]

Wikipedia, "The Internet Protocol Suite", 2016, <[https://en.wikipedia.org/wiki/Internet\\_protocol\\_suite](https://en.wikipedia.org/wiki/Internet_protocol_suite)>.

[GRPC]

gRPC, "gRPC", 2016, <<http://www.grpc.io>>.





[I-D.gont-opsec-icmp-ingress-filtering]

Gont, F., Hunter, R., Massar, J., and W. LIU, "Defeating Attacks which employ Forged ICMPv4/ICMPv6 Error Messages", [draft-gont-opsec-icmp-ingress-filtering-03](#) (work in progress), July 2017.

[I-D.ietf-dhc-rfc3315bis]

Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) bis", [draft-ietf-dhc-rfc3315bis-13](#) (work in progress), April 2018.

[I-D.ietf-i2rs-fb-rib-data-model]

Hares, S., Kini, S., Dunbar, L., Krishnan, R., Bogdanovic, D., and R. White, "Filter-Based RIB Data Model", [draft-ietf-i2rs-fb-rib-data-model-01](#) (work in progress), March 2017.

[I-D.ietf-i2rs-fb-rib-info-model]

Kini, S., Hares, S., Dunbar, L., Ghanwani, A., Krishnan, R., Bogdanovic, D., and R. White, "Filter-Based RIB Information Model", [draft-ietf-i2rs-fb-rib-info-model-00](#) (work in progress), June 2016.

[I-D.ietf-i2rs-rib-data-model]

Wang, L., Chen, M., Dass, A., Ananthakrishnan, H., Kini, S., and N. Bahadur, "A YANG Data Model for Routing Information Base (RIB)", [draft-ietf-i2rs-rib-data-model-15](#) (work in progress), May 2018.

[I-D.ietf-i2rs-yang-l2-network-topology]

Dong, J. and X. Wei, "A YANG Data Model for Layer-2 Network Topologies", [draft-ietf-i2rs-yang-l2-network-topology-04](#) (work in progress), March 2018.

[I-D.ietf-ippm-ioam-data]

Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., daniel.bernier@bell.ca, d., and J. Lemon, "Data Fields for In-situ OAM", [draft-ietf-ippm-ioam-data-02](#) (work in progress), March 2018.

[I-D.ietf-netconf-yang-push]

Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "YANG Datastore Subscription", [draft-ietf-netconf-yang-push-15](#) (work in progress), February 2018.



## [LEAKYABS]

Spolsky, J., "The Law of Leaky Abstractions", 2002,  
<<https://www.joelonsoftware.com/2002/11/11/the-law-of-leaky-abstractions/>>.

## [OPENCONF]

OpenConfig, "Openconfig release YANG models", 2016,  
<<https://github.com/openconfig/public/tree/master/release>>.

## [OSI]

Wikipedia, "OSI Model", 2016,  
<[https://en.wikipedia.org/wiki/OSI\\_model](https://en.wikipedia.org/wiki/OSI_model)>.

## [RFC0792]

Postel, J., "Internet Control Message Protocol", STD 5,  
[RFC 792](#), DOI 10.17487/RFC0792, September 1981,  
<<https://www.rfc-editor.org/info/rfc792>>.

## [RFC0854]

Postel, J. and J. Reynolds, "Telnet Protocol Specification", STD 8, [RFC 854](#), DOI 10.17487/RFC0854, May 1983, <<https://www.rfc-editor.org/info/rfc854>>.

## [RFC1122]

Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, [RFC 1122](#), DOI 10.17487/RFC1122, October 1989,  
<<https://www.rfc-editor.org/info/rfc1122>>.

## [RFC1812]

Baker, F., Ed., "Requirements for IP Version 4 Routers", [RFC 1812](#), DOI 10.17487/RFC1812, June 1995,  
<<https://www.rfc-editor.org/info/rfc1812>>.

## [RFC1918]

Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", [BCP 5](#), [RFC 1918](#), DOI 10.17487/RFC1918, February 1996,  
<<https://www.rfc-editor.org/info/rfc1918>>.

## [RFC3439]

Bush, R. and D. Meyer, "Some Internet Architectural Guidelines and Philosophy", [RFC 3439](#), DOI 10.17487/RFC3439, December 2002,  
<<https://www.rfc-editor.org/info/rfc3439>>.

## [RFC3646]

Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3646](#), DOI 10.17487/RFC3646, December 2003,  
<<https://www.rfc-editor.org/info/rfc3646>>.



- [RFC3719] Parker, J., Ed., "Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)", [RFC 3719](#), DOI 10.17487/RFC3719, February 2004, <<https://www.rfc-editor.org/info/rfc3719>>.
- [RFC3776] Arkko, J., Devarapalli, V., and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", [RFC 3776](#), DOI 10.17487/RFC3776, June 2004, <<https://www.rfc-editor.org/info/rfc3776>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", [RFC 3810](#), DOI 10.17487/RFC3810, June 2004, <<https://www.rfc-editor.org/info/rfc3810>>.
- [RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", [RFC 4253](#), DOI 10.17487/RFC4253, January 2006, <<https://www.rfc-editor.org/info/rfc4253>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, [RFC 4443](#), DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", [RFC 4821](#), DOI 10.17487/RFC4821, March 2007, <<https://www.rfc-editor.org/info/rfc4821>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", [RFC 4862](#), DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/info/rfc4862>>.
- [RFC4877] Devarapalli, V. and F. Dupont, "Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture", [RFC 4877](#), DOI 10.17487/RFC4877, April 2007, <<https://www.rfc-editor.org/info/rfc4877>>.
- [RFC5218] Thaler, D. and B. Aboba, "What Makes for a Successful Protocol?", [RFC 5218](#), DOI 10.17487/RFC5218, July 2008, <<https://www.rfc-editor.org/info/rfc5218>>.
- [RFC5424] Gerhards, R., "The Syslog Protocol", [RFC 5424](#), DOI 10.17487/RFC5424, March 2009, <<https://www.rfc-editor.org/info/rfc5424>>.



- [RFC5474] Duffield, N., Ed., Chiou, D., Claise, B., Greenberg, A., Grossglauser, M., and J. Rexford, "A Framework for Packet Selection and Reporting", [RFC 5474](#), DOI 10.17487/RFC5474, March 2009, <<https://www.rfc-editor.org/info/rfc5474>>.
- [RFC6164] Kohno, M., Nitzan, B., Bush, R., Matsuzaki, Y., Colitti, L., and T. Narten, "Using 127-Bit IPv6 Prefixes on Inter-Router Links", [RFC 6164](#), DOI 10.17487/RFC6164, April 2011, <<https://www.rfc-editor.org/info/rfc6164>>.
- [RFC6177] Narten, T., Huston, G., and L. Roberts, "IPv6 Address Assignment to End Sites", [BCP 157](#), [RFC 6177](#), DOI 10.17487/RFC6177, March 2011, <<https://www.rfc-editor.org/info/rfc6177>>.
- [RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", [RFC 6192](#), DOI 10.17487/RFC6192, March 2011, <<https://www.rfc-editor.org/info/rfc6192>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", [RFC 6275](#), DOI 10.17487/RFC6275, July 2011, <<https://www.rfc-editor.org/info/rfc6275>>.
- [RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", [RFC 6434](#), DOI 10.17487/RFC6434, December 2011, <<https://www.rfc-editor.org/info/rfc6434>>.
- [RFC6540] George, W., Donley, C., Liljenstolpe, C., and L. Howard, "IPv6 Support Required for All IP-Capable Nodes", [BCP 177](#), [RFC 6540](#), DOI 10.17487/RFC6540, April 2012, <<https://www.rfc-editor.org/info/rfc6540>>.
- [RFC6547] George, W., "RFC 3627 to Historic Status", [RFC 6547](#), DOI 10.17487/RFC6547, February 2012, <<https://www.rfc-editor.org/info/rfc6547>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, [RFC 7011](#), DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.





- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", [RFC 7217](#), DOI 10.17487/RFC7217, April 2014, <<https://www.rfc-editor.org/info/rfc7217>>.
- [RFC7224] Bjorklund, M., "IANA Interface Type YANG Module", [RFC 7224](#), DOI 10.17487/RFC7224, May 2014, <<https://www.rfc-editor.org/info/rfc7224>>.
- [RFC7317] Bierman, A. and M. Bjorklund, "A YANG Data Model for System Management", [RFC 7317](#), DOI 10.17487/RFC7317, August 2014, <<https://www.rfc-editor.org/info/rfc7317>>.
- [RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local Addressing inside an IPv6 Network", [RFC 7404](#), DOI 10.17487/RFC7404, November 2014, <<https://www.rfc-editor.org/info/rfc7404>>.
- [RFC7527] Asati, R., Singh, H., Beebee, W., Pignataro, C., Dart, E., and W. George, "Enhanced Duplicate Address Detection", [RFC 7527](#), DOI 10.17487/RFC7527, April 2015, <<https://www.rfc-editor.org/info/rfc7527>>.
- [RFC7608] Boucadair, M., Petrescu, A., and F. Baker, "IPv6 Prefix Length Recommendation for Forwarding", [BCP 198](#), [RFC 7608](#), DOI 10.17487/RFC7608, July 2015, <<https://www.rfc-editor.org/info/rfc7608>>.
- [RFC7663] Trammell, B., Ed. and M. Kuehlewind, Ed., "Report from the IAB Workshop on Stack Evolution in a Middlebox Internet (SEMI)", [RFC 7663](#), DOI 10.17487/RFC7663, October 2015, <<https://www.rfc-editor.org/info/rfc7663>>.
- [RFC7772] Yourtchenko, A. and L. Colitti, "Reducing Energy Consumption of Router Advertisements", [BCP 202](#), [RFC 7772](#), DOI 10.17487/RFC7772, February 2016, <<https://www.rfc-editor.org/info/rfc7772>>.
- [RFC7922] Clarke, J., Salgueiro, G., and C. Pignataro, "Interface to the Routing System (I2RS) Traceability: Framework and Information Model", [RFC 7922](#), DOI 10.17487/RFC7922, June 2016, <<https://www.rfc-editor.org/info/rfc7922>>.
- [RFC7934] Colitti, L., Cerf, V., Cheshire, S., and D. Schinazi, "Host Address Availability Recommendations", [BCP 204](#), [RFC 7934](#), DOI 10.17487/RFC7934, July 2016, <<https://www.rfc-editor.org/info/rfc7934>>.



- [RFC7980] Behringer, M., Retana, A., White, R., and G. Huston, "A Framework for Defining Network Complexity", [RFC 7980](#), DOI 10.17487/RFC7980, October 2016, <<https://www.rfc-editor.org/info/rfc7980>>.
- [RFC7991] Hoffman, P., "The "xml2rfc" Version 3 Vocabulary", [RFC 7991](#), DOI 10.17487/RFC7991, December 2016, <<https://www.rfc-editor.org/info/rfc7991>>.
- [RFC8028] Baker, F. and B. Carpenter, "First-Hop Router Selection by Hosts in a Multi-Prefix Network", [RFC 8028](#), DOI 10.17487/RFC8028, November 2016, <<https://www.rfc-editor.org/info/rfc8028>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [RFC 8040](#), DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", [RFC 8106](#), DOI 10.17487/RFC8106, March 2017, <<https://www.rfc-editor.org/info/rfc8106>>.
- [RFC8170] Thaler, D., Ed., "Planning for Protocol Adoption and Subsequent Transitions", [RFC 8170](#), DOI 10.17487/RFC8170, May 2017, <<https://www.rfc-editor.org/info/rfc8170>>.
- [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, [RFC 8201](#), DOI 10.17487/RFC8201, July 2017, <<https://www.rfc-editor.org/info/rfc8201>>.
- [RFC8273] Brzozowski, J. and G. Van de Velde, "Unique IPv6 Prefix per Host", [RFC 8273](#), DOI 10.17487/RFC8273, December 2017, <<https://www.rfc-editor.org/info/rfc8273>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", [RFC 8342](#), DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", [RFC 8343](#), DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", [RFC 8344](#), DOI 10.17487/RFC8344, March 2018, <<https://www.rfc-editor.org/info/rfc8344>>.



- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", [RFC 8345](#), DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8346] Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", [RFC 8346](#), DOI 10.17487/RFC8346, March 2018, <<https://www.rfc-editor.org/info/rfc8346>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", [RFC 8349](#), DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.

#### Authors' Addresses

Zaid Ali Kahn (editor)  
LinkedIn  
CA  
USA

Email: [zaid@linkedin.com](mailto:zaid@linkedin.com)

John Brzozowski (editor)  
Comcast  
USA

Email: [John\\_Brzozowski@comcast.com](mailto:John_Brzozowski@comcast.com)

Russ White (editor)  
LinkedIn  
Oak Island, NC 28465  
USA

Email: [russ@riw.us](mailto:russ@riw.us)

