INTERNET-DRAFT
Intended Status: Proposed Standard
Expires: September 12, 2012

Rick van Rein
OpenFortress
March 11, 2012

# 6bed4: Peer-to-Peer IPv6 on Any Internetwork
draft-vanrein-v6ops-6bed4-01

## Abstract

The intention of 6bed4 is to support IPv6-only applications, even on IPv4-only networks. A specific area of concern is that of peer-to-peer protocols such as SIP or document exchange during a chat session. Such protocols are designed to run in any environment, which means that they cannot rely on IPv6 for themselves, or their peers. The 6bed4 tunnel mechanism ensures that IPv6 can be assumed on all peers, without a need to configure it explicitly.

The 6bed4 mechanism is meant as a fallback mechanism for IPv6 connectivity on networks that do not support it natively, by running a tunnel over UDP and IPv4. The IPv4 address is used to support traceability of the traffic originator, which means that no user account or other configuration is needed.

The tunnel mechanism builds on existing IPv6 mechanisms; it employs Stateless Address Autoconfiguration [RFC4862] to setup an IPv6 address on a 6bed4 Peer, and Neighbor Discovery [RFC4861] to find the most direct route to a remote 6bed4 Peer.

## Status of this Memo











## Copyright and License Notice



## Table of Contents

















## 1. Introduction

The 6bed4 mechanism is a fallback for networks that have no support for IPv6. It is designed to work without configuration on any network that is not secured beyond common defaults. It works behind any sequence of Network Address Translating (NAT) devices and Firewalls, and it does not require support (such as protocol 41) from any of those devices. The local network may or may not support IPv6 already; and also upstream network providers may or may not support IPv6. The 6bed4 traffic does not mingle with local IPv6 traffic, so it will never raise alarms that monitor native local IPv6 traffic for potential IPv6-specific attacks.

All traffic on the 6bed4 Network is embedded in UDP and IPv4. The 6bed4 Network is thereby layered on top of the IPv4 Internet. Common default configurations for NAT and firewalls support UDP [RFC3022] [RFC4787], as long as the communication is initiated internally. This is precisely what 6bed4 does.

This traffic is exchanged between 6bed4 Peers, which are the endpoints in the communication. They may pass their traffic through a 6bed4 Router. A 6bed4 Router recognizes 6bed4 traffic because it is received on the standard UDP port TBD1, and it recognizes IPv6 addresses as 6bed4 addresses because they start with the /64 prefix that it has dedicated to 6bed4. Such IPv6 prefixes are paired to the IPv4 address and UDP port of the 6bed4 Router.

The IPv4 address and UDP port as seen from the Internet are combined to form a link-layer address, from which an Interface Identifier is constructed in the same way as for Ethernet Networks [RFC2464]. This implies that the IPv4 address and UDP port are shown as part of the IPv6 address. This supports traceback of a traffic originator in case of abuse, without a need to register accounts or login to them. Every 6bed4 Router validate that the IPv4 address and UDP port in the IPv6 address against the actual traffic, so the properties of egress filtering on IPv4 packets are inherited by the 6bed4 Network.

The simplest mode of communication between to 6bed4 Peers is to communicate through a 6bed4 Router. This service relays between 6bed4 traffic and native IPv6 or, if both IPv6 addresses are IPv6 addresses under the same 6bed4 prefix, then a 6bed4 Router relays the traffic between the 6bed4 Peers, using its own IPv4 address and UDP port when forwarding a packet over 6bed4 to the destination. Traffic intended for addresses that the 6bed4 Router does not handle as 6bed4 prefixes are relayed as IPv6 over normal IPv6 routes.

It is normal for a 6bed4 Peer to attempt to bypass the 6bed4 Router by contacting remote peers directly. One such bypass that every

6bed4 Peer MUST attempt is to contact 6bed4 Peers at the IPv4 address and UDP port contained in their Interface Identifier. Other bypassing mechanisms may also be tried; the success or failure of each bypass method depends on co-operation of intermediate network nodes. Rather than devising a model of these intermediate nodes, 6bed4 Peers simply try to send Neighbor Solicitation over a candidate bypass, and when the remote 6bed4 Peer replies with a matching Neighbor Advertisement, then the bypass is fit for use. If the reply does not arrive, the 6bed4 Peer can still fallback to the 6bed4 Router for that remote 6bed4 Peer.

Information about bypasses to a 6bed4 Peer should be cached for some time. A good place to do this is the Neighbor Cache, which retains such information for tens of seconds [RFC4861]. This happens automatically if the IPv6 stack that uses a 6bed4 Link for IPv6 connectivity is used to relay link-layer addresses for the 6bed4 Network; such link-layer addresses would be composed of the IPv4 address and UDP port needed to contact the 6bed4 Peer over the 6bed4 Network. Whenever the IPv6 stack sends a Neighbor Solicitation down the 6bed4 Link, the 6bed4 Peer will know that it should try a number of bypasses for that route, once again using Neighbor Discovery. When no bypasses work, then the link-local address of the 6bed4 Router is passed back up to the IPv6 stack.

[TO BE REMOVED: Code demonstrating this specification is, or will soon be, available on http://devel.0cpm.org/6bed4/ -- please contact the author for details, information and feedback]

## 2. Definitions

The following definitions will be used throughout this document.

A 6bed4 Router is a network node connected to both IPv4 and IPv6 networks, which supports the translation of a particular IPv6 /64 prefix to 6bed4 and back. The IPv6 prefix forms a pair with the router's IPv4 address and UDP port for 6bed4. A 6bed4 Router may translate traffic bidirectionally, or in the case of more advanced routing topologies, it could rest on the assumption that another 6bed4 Router to perform the reverse translation for the same pair of IPv6 prefix and IPv4/UDP coordinates.

A 6bed4 Peer is a communication endpoint that uses 6bed4 to embed IPv6 packets in UDP and IPv4. It has a relationship to one 6bed4 Router which serves as its default router and as a fallback for remote 6bed4 Peers that cannot be connected to directly.

A bypass or bypass route is a connection between two 6bed4 Peers that







does not run through a 6bed4 Router.

A 6bed4 Network comprises of the 6bed4 Router(s) and all 6bed4 Peers. This may include bypass routes that connect over a local network, even using private IPv4 addresses [RFC1918]. The identity of a 6bed4 Network is its IPv6 /64 prefix as supplied by its 6bed4 Router(s).

A 6bed4 Link is a (virtual) network interface that is offered to an IPv6 stack that wants to use it to link to IPv6 nodes. This term is non-normative; it is introduced below as part of the conceptual model for a 6bed4 Peer.

A 6bed4 Tunnel is the connection between a 6bed4 Link and the 6bed4 Network. This term is non-normative; it is introduced below as part of the conceptual model for a 6bed4 Peer.

A 6bed4 Address is a pair of an IPv4 address and UDP port that can be used as an endpoint on the 6bed4 Network, at the very least to its 6bed4 Router and possibly through Bypass Routes from other 6bed4 Peers as well. The IPv4 address and UDP port are the values as seen by the public Internet. This is a conceptual notion, free from any representation in network packets.

A 6bed4 Link-Local Address is a particular representation of a 6bed4 Address. It is employed in network packets for Neighbor Discovery, Router Discovery and Redirect messages.

Plain traffic is defined as IPv6 traffic that is not one of the special ICMPv6 [RFC4443] types Router Solicitation, Router Advertisement, Neighbor Solicitation, Neighbor Advertisement or Redirect.

The Metric, or Communication Cost Metric, of communicating to a given 6bed4 Address is an indication of the relative cost of a number of possible routes; a bypass route to a local subnet ranks as Low, a bypass route directly to the UDP port and IPv4 address of a remote 6bed4 Peer ranks as Medium and a route to the 6bed4 Router ranks as High.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 3. Conceptual Model for a 6bed4 node

A 6bed4 node is a network node that acts as a 6bed4 Router or a 6bed4 Peer. This conceptual model for a 6bed4 nodes is non-normative; the







terms and layers it introduces are intended for illustration purposes, but any equivalent implementation would be equally suitable.

The conceptual model of the internal structure of a 6bed4 node is based on what is called a tunnel in many operating systems; this is a structure which acts as a network interface on one end, and gives programmatic control over packet handling on the other end. An IPv6 stack can access the 6bed4 services through a 6bed4 Link.

The 6bed4 Link is the network side of a 6bed4 Tunnel. This 6bed4 Link acts as a level 2 service to an IPv6 stack. The programmatic side of the 6bed4 Tunnel uses the level 4 service of a UDP/IPv4 stack. The function of the 6bed4 Tunnel is to translate packets between these two interfaces.

```
   :                :
   |   IPv6 stack   |
   +-------++-------+
           ||  <-- 6bed4 Link
           ||  Layer 2 transport: IPv6 over a link-layer protocol
           \/
   +-------++-------+
   |  6bed4 Tunnel  |
   +-------++-------+
           ||
           ||  Layer 4 transport: IPv6 over UDP/IPv4
           \/
   +-------++-------+
   | UDP/IPv4 stack |  <--  6bed4 Network
   :                :
```

The translation of plain traffic is a matter of adding or removing headers into which an IPv6 packet is embedded. On the IPv6 end, some form of header information revealing the 6bed4 Link-Local Address of the next-hop destination must be present, and on the IPv4 end there are UDP/IPv4 headers. Since 6bed4 Link-Local Addresses incorporate the destination's UDP port and IPv4 address, a complete translation between these two prefixed headers is possible. This is what is done with plain traffic.

An exception is made for Router Discovery, Neighbor Discovery and Redirect messages. These were designed for local use, and should therefore be treated differently; the messages are exchanged between the IPv6 stack and the 6bed4 Tunnel, and somewhat independently between the 6bed4 Tunnel and the 6bed4 Network. relayed through a tunnel. These messages are therefore handled explicitly by the 6bed4 Tunnel.







The link-layer address of a 6bed4 Link is not normally known as soon as the 6bed4 Link is created, as it relies on observations made on the Public Internet. To obtain this link-layer address, the 6bed4 Tunnel will send a Router Solicitation to its 6bed4 Router. The 6bed4 Router responds with a Router Advertisement containing the 6bed4 Peer's Interface Identifier as the IPv6 destination address. From this address, the 6bed4 Peer's link-layer address is derived and assigned to the 6bed4 Link; this also serves as the 6bed4 Link-Layer Address when communicating on the 6bed4 Network. This information is usually be exchanged prior to bringing up the 6bed4 Link.

The prefix information from the Router Advertisement is held back until the IPv6 stack on top of the 6bed4 Link sends a Router Solicitation. In response to that, a Router Advertisement is sent to the IPv6 stack that holds the prefix. The IPv6 stack is assumed to reconstruct the Interface Identifier from the link-layer address, prefix it with the Router Solicitation's prefix and assign the resulting address to the 6bed4 Link. Any attempts to perform Neighbor Discovery on its own address should be dropped in the 6bed4 Tunnel, as it is safe to assume that no competition for the link-layer address and its derived full address exists.

The 6bed4 Link is now configured with a /64 prefix and a default route through the 6bed4 Router. Any IPv6 addresses starting with the /64 prefix handled by the 6bed4 Router are treated by the IPv6 stack as local traffic to be contacted through the 6bed4 Link, and any other IPv6 traffic may be forwarded to the 6bed4 Router.

Whenever the IPv6 stack needs to find either the 6bed4 Router or a 6bed4 Peer with the same /64 prefix, it will perform Neighbor Discovery on the 6bed4 Link. The 6bed4 Tunnel handles these especially, in an attempt to create a bypass when possible. To this end, it will first try one or more bypasses before falling back to the 6bed4 Router.

The 6bed4 Tunnel will first try a few times to send Neighbor Solicitation messages with the targeted IPv6 address to the direct IPv4 address and UDP port of the remote 6bed4 Peer. When this results in a matching Neighbor Advertisement, it will relay that knowledge through Neighbor Advertisement to the IPv6 stack, which will note it in its Neighbor Cache. If it fails to receive such a response, but before the IPv6 stack times out on its Neighbor Solicitation, the 6bed4 Tunnel will instead send back a Neighbor Advertisement announcing the 6bed4 Link-Local Address of the 6bed4 Router.

Among the possibly found direct routes may be Pinholes, which relay the traffic back to the local network. These Pinholes, as well as any other bypass routes through a hole in a Network Address Translator or Firewall, are subject to timeouts. It is useful to realize that Neighbor Caches tend to hold a Link-Local Address for "tens of seconds" [RFC4861] and then either drop it or revitalize it through another Neighbor Discovery attempt. These delays are smaller than common expiration times of holes in NAT or Firewalls, so no Neighbor Cache parameter changes should normally be needed to sustain a connection between 6bed4 Peers communicating through any of these bypasses.

It is even find a better bypass than a Pinhole if a direct connection to a locally connected 6bed4 Peer is possible. This may be considered for contact between 6bed4 Peers that exhibit the same public IPv4 address. Those may be sought through multicasting a Neighbor Solicitation over 6bed4 on the local IPv4 network, by sending to the address for all IPv4 nodes on the current subnet, or 224.0.0.1, using the standard UDP port TBD1 to select 6bed4 Peers.

To keep a hole in a Firewall or NAT open while it is in active use, it is generally useful to pass traffic in both directions; this means that two 6bed4 Peers must use the same route to pass traffic between each other. In exceptional cases however, asymmetric routes may be learnt. This may be remedied with a Redirect message that requests a 6bed4 Peer to change to a more efficient bypass. Redirect messages will never be sent to request changing to a route with a higher Metric, but only to a bypass with a lower Metric. In order of rising Metric, the available bypass mechanisms are:

- Low: Traffic sent to locally attached private addresses [RFC1918] (Optionally supported)
- Medium: Traffic sent to the IPv4 address and UDP port in the 6bed4 Link-Local Address
- High: Traffic sent through the 6bed4 Router

Note that these variations can be distinguished easily by observing the IPv4 address targeted. Also note that the third routing mechanism would not add anything to local networks by supporting publicly routed addresses, as the second mechanism provides sufficient support for optimal bypasses on such networks.

## 4. Link-Local Profile for 6bed4

Each 6bed4 Link-Local Address contains the UDP port and IPv4 address as observed on the public Internet. This means that it is 6 bytes in







size. The format is as follows:

```
   0                                               47
  +-------+-------+-------+-------+-------+-------+
  | UDP.1 | UDP.0 |IPv4.0 |IPv4.1 |IPv4.2 |IPv4.3 |
  +-------+-------+-------+-------+-------+-------+
```

The UDP port is encoded in two bytes, where UDP.0 UDP.1 would be the network byte order; note that the bytes in the 6bed4 Link-Local Address are included in the reverse of network byte order. The IPv4 address bytes are represented in the network byte order, IPv4.0 IPv4.1 IPv4.2 IPv4.3.

The UDP port MUST be an even number. The lowest-valued bit of UDP.1 is reserved to indicate a multicast address, when it is set. In situations where multicast addresses are known to be impossible, the bit may alternately be interpreted as an error indication.

The 64 bits of Interface Identifier to follow a /64 prefix can be derived from the 6bed4 Link-Local Address as it is done for 6-byte Ethernet addresses, as follows:

```
   0                                                               63
  +-------+-------+-------+-------+-------+-------+-------+-------+
  | UDP.1'| UDP.0 |IPv4.0 | 0xff  | 0xfe  |IPv4.1 |IPv4.2 |IPv4.3 |
  +-------+-------+-------+-------+-------+-------+-------+-------+
```

In this representation UDP.1' is UDP.1, bitwise xor-ed with 0x02. The result is the EUI-64 address derivation that is also used for Ethernet Networks [RFC2464].

Multicast addresses for IPv6 are also constructed as for Ethernet, resulting in the following format:

```
   0                                               47
  +-------+-------+-------+-------+-------+-------+
  | 0x33  | 0x33  |IPv6.12|IPv6.13|IPv6.14|IPv6.15|
  +-------+-------+-------+-------+-------+-------+
```

The bytes IPv6.12 through IPv6.15 are the last four bytes of the IPv6 address sought over multicast.

## 5. 6bed4 Router Behavior

The 6bed4 Router services 6bed4 Peers with Router Solicitation requests, and by relaying any traffic that it cannot automatically direct. The following definitions specify the behavior in detail.







The description below assumes that the 6bed4 Router acts bidirectionally; note that a split of this function over multiple network nodes is thinkable, in which case their combined behavior should match these specifications.

### 5.1. Address Information

A 6bed4 Router offers its services on an IPv4 address and the UDP port TBD1. The IPv4 address varies between 6bed4 Routers, and may be found through various means; it may be hard-coded into 6bed4 Tunnel software, it may be an Anycast address [RFC4786], or it may be found through IPv4 application protocols such as DNS [RFC1034] or DHCP [RFC2131]. The combined IPv4 address and UDP port form its 6bed4 Address, which will sometimes be represented as its 6bed4 Link-Local Address.

Traffic arriving at this UDP/IPv4 combination is said to have arrived from the 6bed4 Network. Similarly, whenever the 6bed4 Router sends traffic over the 6bed4 Network, so when addressing a 6bed4 Peer, it MUST send from that same IPv4 address and UDP port.

The IPv4 address and UDP port are paired with a IPv6 /64 prefix for which the 6bed4 Router serves as a router. The prefix is assumed to be assigned indefinitely to the 6bed4 Router.

### 5.2. Relaying Plain Unicast Traffic

The 6bed4 Router relays plain unicast traffic between 6bed4 Peers, as well as between 6bed4 Peers and the general IPv6 Internet. This section describes how this is done.

Upon arrival of plain unicast traffic over the 6bed4 Network, the 6bed4 Router MUST validate that the sender's IPv6 address is correct; this means that the /64 prefix MUST match the prefix serviced by the 6bed4 Router, and that the IPv4 address and UDP port as shown in the Interface Identifier MUST match the parameters of the incoming 6bed4 message. This certifies the IPv6 address of the 6bed4 Peer as well as its IPv4 address is certified. The value of the bytes 0xff and 0xfe in the default Interface Identifier MUST NOT be validated, as these may vary when the sending 6bed4 Peer acts as a local router.

If only the Interface Identifier fails validation, then the 6bed4 Router MUST proceed as if it had received a Router Solicitation over the 6bed4 Network. The 6bed4 Router MUST drop all other 6bed4 traffic that fails validation. The remainder of this section applies only to plain unicast traffic that passed validation.

When the destination address starts with the /64 prefix that the







6bed4 Router announces to 6bed4 Peers, then the destination is considered to be on the 6bed4 Network. Otherwise, it is considered a general IPv6 address, and MUST be routed through a general IPv6 network to which the 6bed4 Router is connected.

To route a network packet to the 6bed4 Network, the 6bed4 Router MUST extract the IPv4 address and UDP port from the Interface Identifier part of the IPv6 destination address, and forward the packet over UDP and IPv4 to those coordinates.

When relaying traffic to the IPv6 network, the Differentiated Services field [RFC2474], contained in the Traffic Class in the IPv6 header, SHOULD be retained insafar as it cannot interfere with normal network operation. When relaying traffic to the 6bed4 network, the Differentiated Services field in the Traffic Class in the IPv6 header SHOULD be used to fill the Type Of Service field in the IPv4 header, insafar as normal network operation permits.

### 5.3. Relaying Plain Multicast Traffic

This specification places no restrictions on handling of plain multicast traffic. As a result, a 6bed4 Router MAY drop all plain multicast traffic, but extensions to this specification MAY be implemented to support carrying multicast across uncooperative networks.

When relaying traffic to the IPv6 network, the Differentiated Services field [RFC2474], contained in the Traffic Class in the IPv6 header, SHOULD be retained insafar as it cannot interfere with normal network operation. When relaying traffic to the 6bed4 network, the Differentiated Services field in the Traffic Class in the IPv6 header SHOULD be used to fill the Type Of Service field in the IPv4 header, insafar as normal network operation permits.

### 5.4. Handling Router Solicitation from the 6bed4 Network

The 6bed4 Router MUST accept Router Solicitation messages from the 6bed4 Network. In addition, certain validation errors on the origin of plain unicast messages are handled as if a Router Solicitation had been received.

In response to a Router Solicitation, the 6bed4 Router MUST send a Router Advertisement, which is sent back to the originator over the 6bed4 Network, so to the IPv4 address and UDP port over which the incoming message arrived. The Router Advertisement sent MUST at least include the Neighbor Discovery Option for Prefix.

The Prefix Option communicates the /64 prefix that the 6bed4 Router has paired to the IPv4 address and UDP port combination over which the incoming message was received; the Valid and Preferred Lifetime fields are set to infinite by setting all bits to 1.

The IPv6 destination address of the Router Advertisement MUST be set to fe80::/64 plus an Interface Identifier constructed from the UDP port and IPv4 address from which the Router Solicitation was received.

If the UDP port from which the Router Solicitation was received is an odd number, then the lowest-value bit of the first byte in the Interface Identifier of the IPv6 destination address of the Router Advertisement MUST be set as an error indication. This informs the 6bed4 Peer that its UDP port is invalid, and that it should try again from another port. Note that an odd UDP port number on a private address [RFC1918] makes it more likely that an odd UDP port number is assigned on the public Internet, and also that an odd UDP port number can find its way into a 6bed4 Link-Local Address on a bypass route over the local network; for these reason, an odd UDP port number SHOULD be rejected when it is assigned locally, even in a private address range.

5.5. Handling Router Advertisement from the 6bed4 Network

The 6bed4 Router MUST drop all Router Advertisements arriving over the 6bed4 Network.

5.6. Handling Neighbor Solicitation from the 6bed4 Network

The 6bed4 Router accepts Neighbor Solicitation messages over the 6bed4 Network. Incoming packets are subject to the same validation as before relaying plain unicast traffic, and will be dropped if the validation fails.

The only validating Neighbor Solicitation messages to which the 6bed4 Router responds are those directed at its IPv6 address composed of the /64 prefix used for 6bed4 plus its Interface Identifier, its IPv6 address composed of fe80::/64 plus its Interface Identifier, and finally its IPv6 address fe80::/128. To all these, it will respond with a Neighbor Advertisement that provides its 6bed4 Link-Local Address in a Target Link-Local Address Neighbor Discovery Option.

5.7. Handling Neighbor Advertisement from the 6bed4 Network

The 6bed4 Router MUST drop all Neighbor Advertisements arriving over the 6bed4 Network.

5.8. Handling Redirect from the 6bed4 Network







A 6bed4 Router MUST drop all Redirect messages arriving over the 6bed4 Network.

5.9. Handling Empty UDP Packets from the 6bed4 Network

Packets arriving over the 6bed4 Network may consist of nothing but an IPv4 header and a UDP header, so without any UDP data. Such messages may have been sent to keep an open connection to the 6bed4 Router, especially by 6bed4 Peers behind Network Address Translators and/or Firewalls. For this purpose, it is usually the outgoing traffic that keeps a hole open, and no return traffic is required. Therefore, such packets MAY be dropped.

5.10. Dealing with Packet Fragmentation

The IPv6 specifications define a minimal MTU that exceeds that of IPv4; as a result, it is always possible for packets on the 6bed4 Network to be subjected to fragmentation. The 6bed4 Router SHOULD NOT set the Don't Fragment flag [RFC0791] on IPv4 traffic and it SHOULD be prepared to reconstruct packets from fragments.

The 6bed4 Router MAY reject packets in excess of the 1280 byte MTU for IPv6 traffic; it MUST then respond with an ICMPv6 message of type 2, Packet Too Big.

6. 6bed4 Peer Behavior

Every 6bed4 Peer has a single 6bed4 Router that it uses as its gateway to the general IPv6 network, and possibly to other 6bed4 Peers on the same 6bed4 Network to which it cannot create a bypass route. The 6bed4 Router has a fixed IPv6 /64 prefix setup for its 6bed4 Network. The following details the behavior of the 6bed4 Peer over the 6bed4 Network defined by that prefix, both in its relation to its 6bed4 Router and to other 6bed4 Peers.

The following gives some background information about a possible implementation of a 6bed4 Tunnel from the aforementioned conceptual model. This involves the behavior of the 6bed4 Tunnel and its link to the IPv6 stack over the 6bed4 Link. All such information should be considered as non-normative illustrations; any equivalent implementation is acceptable. This specification is normative where it concerns the behavior of the 6bed4 Peer over the 6bed4 Network.

6.1. Filtering Incoming Traffic

As a general precaution against spurious senders, the 6bed4 Peer MUST filter traffic reaching it over the 6bed4 Network. The IPv4 address







and UDP port over which traffic originates should arrive over an approved (bypass) route:

- Traffic from the 6bed4 Router's IPv4 address and UDP port TBD1 are always acceptable;
- Traffic with an IPv6 source address under the /64 prefix that is either fe80::/64 or the 64 prefix of the 6bed4 Network is acceptable, provided that the IPv4 address and UDP port from which the 6bed4 Network traffic originated match the Interface Identifier of the IPv6 source address.
- Traffic that originated from a directly attached IPv4 subnet MAY be accepted if the implementation and configuration of the 6bed4 Peer support it. Note that the destination address MAY in this case be the multicast address 224.0.0.1 for all nodes on the local subnet, provided that it is used with UDP port TBD1.

Any other incoming traffic over the 6bed4 Network MUST be dropped.

Since other 6bed4 Peers also implement these tests, sending behavior must match it. Any traffic that a 6bed4 Peer sends over the 6bed4 Network MUST originate from the 6bed4 Link-Local Address, even the Router Solicitation sent to the 6bed4 Router. Normal sending of UDP/IPv4 traffic from a socket would implement that automatically, even if the 6bed4 Peer does not know its origin yet. IPv6 traffic that is not a Router Solicitation MUST additionally use an IPv6 source address comprising of the IPv6 /64 prefix for the 6bed4 Network, plus an Interface Identifier derived from the 6bed4 Link-Local Address of the 6bed4 Peer.

6.2. 6bed4 Tunnel and Link-Layer Address Setup

Every 6bed4 Peer MUST have an assigned 6bed4 Router that it can use as its default route to the general IPv6 network. This 6bed4 Router MAY be configured, or it MAY be determined using some automated mechanism during initialization of the 6bed4 Peer. Furthermore, the 6bed4 Peer MAY iterate over a number of options until it finds one that it finds acceptable. It SHOULD not change its 6bed4 Router after it has decided to join its 6bed4 Network.

To be able to communicate on the 6bed4 Network attached to its 6bed4 Router, the 6bed4 Peer MUST obtain a 6bed4 Link-Local Address. To obtain one, it sends a Router Solicitation to its 6bed4 Router. The 6bed4 Router responds with a Router Advertisement that is processed as specified below, among others to derive the 6bed4 Link-Local Address for the 6bed4 Peer. If this response is not received for







some time, then another Router Advertisement SHOULD be attempted, observing an exponential fallback scheme to avoid overloading the 6bed4 Network.

In terms of the non-normative conceptual model, an accepted 6bed4 Link-Local Address can be assigned to the 6bed4 Tunnel, and the 6bed4 Link can be brought online.

When bringing the 6bed4 Peer online, he 6bed4 Peer MAY setup a default route over the 6bed4 Network with the 6bed4 Router as a default router; in terms of the non-normative conceptual model, the router's address may be setup as fe80::/128 or as fe80::/64 plus the Interface Identifier of the 6bed4 Router.

In addition to listening for traffic on the UDP port and IPv4 address from which the 6bed4 Peer sends its traffic, it MAY also subscribe to multicast traffic sent to IPv4 address 224.0.0.1 for all hosts on local subnets, and UDP port TBD1. Subscribing to this port and processing messages on it as though they arrived over the usual UDP port and IPv4 address is optional, but for reasons of symmetric routing it MUST be done if the 6bed4 Peer wants to be able to setup bypass routes with the Low Metric.

6.3. Relaying Plain Unicast Traffic

The 6bed4 Peer relays between the 6bed4 Network and IPv6 stack, such as over the 6bed4 Link proposed in the non-normative conceptual model.

When plain unicast traffic passes from IPv6 to the 6bed4 Network, it is assumed that a 6bed4 Link-Local Address is provided as the next hop destination. From this next hop address, the IPv4 address and UDP port are retrieved, and the IPv6 packet is sent to those coordinates over the 6bed4 Network.

When relaying traffic to the 6bed4 network, the Differentiated Services field in the Traffic Class in the IPv6 header SHOULD be used to fill the Type Of Service field in the IPv4 header, inasfar as normal network operation permits.

6.4. Relaying Plain Multicast Traffic

This specification places no restrictions on handling of plain multicast traffic. As a result, a 6bed4 Router MAY drop all plain multicast traffic, but extensions to this specification MAY be implemented to support carrying multicast across uncooperative networks.

When plain multicast traffic arrives over the 6bed4 Network, the

6bed4 Peer MAY accept it for processing; in terms of the non-normative conceptual model, that would mean to relay it over the 6bed4 Link to the IPv6 stack.

A 6bed4 Peer MAY also support sending plain multicast traffic; to that end, it MAY utilize the destination address 224.0.0.1 to reach all IPv4 hosts on the local subnet, with UDP port TBD1 to select 6bed4 Peers. Multicast through the 6bed4 Router and to remote 6bed4 Peers are not covered by this specification.

When relaying traffic to the 6bed4 network, the Differentiated Services field in the Traffic Class in the IPv6 header SHOULD be used to fill the Type Of Service field in the IPv4 header, inasfar as normal network operation permits.

6.5. Handling Router Solicitation from the 6bed4 Network

The 6bed4 Peer is normally not a router, because that would require a /64 prefix under which to offer addresses. Therefore, incoming attempts of Router Solicitation over the 6bed4 Network are usually a mistake and SHOULD be dropped.

There is one possible exception though. A 6bed4 Peer MAY provide leases over an additional protocol, such as DHCPv6. To that end, it would substitute the middle two bytes of the Interface Identifier with other values than the default 0xff,0xfe values and offer a thusly constructed address; it would also serve as a router for any such traffic. To make this possible, the 6bed4 Peer MAY send back Router Advertisements with the Managed Address Configuration flag set.

6.6. Handling Router Advertisement from the 6bed4 Network

Incoming Router Advertisement from the 6bed4 Network MUST be dropped if its origin 6bed4 Address is not that of the 6bed4 Router assigned to the 6bed4 Peer.

To be accepted, a Router Advertisement must have been sent by the 6bed4 Router, it MUST NOT have a set Managed Address Configuration flag [RFC4861], it MUST have a Prefix Option with a /64 prefix, and it MUST have an IPv6 destination address that starts with fe80::/64 and that has bytes 0xff,0xfe in the middle of the Interface Identifier. This Interface Identifier MUST NOT have the lowest-value bit in the first byte set, as that would indicate an error condition; this is usually due to an odd UDP port as seen by the public Internet. Since this is not permitted, the 6bed4 Peer SHOULD retry the Router Solicitation if that happens, after switching to another UDP port. It MUST NOT continue to communicate on the 6bed4 Network with





an UDP port that flagged this error.

If no MTU settings are provided as part of the Router Advertisement, then the 6bed4 Peer MUST not assume more than the guaranteed minimum MTU of 1280 for IPv6 networks without performing Path MTU Discovery.

When sent by the 6bed4 Router, the message MUST be treated as a new announcement of a prefix and a 6bed4 Link-Local Address, possibly replacing previously set values. The 6bed4 Peer MUST either remove any old settings that do not match the new announcement, or deprecate them and then it SHOULD remove them at a later time. When the 6bed4 Link-Local Address changed, then the hole punched in Network Address Translators and/or Firewalls appears to have changed, so connections through the 6bed4 Router are not likely to be able to continue; but some connections through bypass routes may continue to work.

6.7. Handling Router Solicitation from the 6bed4 Link

This subsection is an illustration of a possible implementation; it is non-normative.

The assumption in the conceptual model is that the 6bed4 Link has been setup with a 6bed4 Link-Local Address before it is brought online; this information is obtained with Router Discovery initiated by the 6bed4 Peer. As a result, the 6bed4 Link's attempts to Router Discovery are a bit late, but should nonetheless be handled.

When a Router Solicitation enters over the 6bed4 Link, it could be responded to with a Router Advertisement holding a Prefix Option with the /64 prefix obtained from the 6bed4 Router. The Default Router Preference Field [RFC4191] may be set to Low, so as to prefer default routing over native links, except for traffic targeted at the /64 prefix offered in this Router Advertisement.

6.8. Handling Router Advertisement from the 6bed4 Link

This subsection is an illustration of a possible implementation; it is non-normative.

Any Router Advertisement arriving over the 6bed4 Link must be dropped.

6.9. Handling Neighbor Solicitation from the 6bed4 Network

When a Neighbor Solicitation destined for one of its IPv6 addresses enters a 6bed4 Peer over the 6bed4 Network, then a reply MUST be formulated in the form of a Neighbor Advertisement. In terms of the non-normative conceptual model, the message could be replicated to





the IPv6 stack through the 6bed4 Link.

6.10. Handling Neighbor Advertisement from the 6bed4 Network

When a Neighbor Advertisement arrives over the 6bed4 Network, then the message will be normally processed; in terms of the non-normative conceptual model, the message would be replicated to the IPv6 stack over the 6bed4 Link.

Any work scheduled to find the target in the received Neighbor Advertisement MUST be descheduled as a result of processing the Neighbor Advertisement.

6.11. Handling Neighbor Solicitation from the 6bed4 Link

This subsection is an illustration of a possible implementation; it is non-normative.

Neighbor Solicitation can be used by the IPv6 stack to avoid collisions. This is not required on the 6bed4 Network, where addresses are already unique if they are assigned through 6bed4 Link-Local Addresses. Therefore, an attempt to verify the prior existence of an IPv6 address based on this is not required. In terms of the non-normative conceptual model, this means that a Neighbor Solicitation arriving over the 6bed4 Link and inquiring after the 6bed4 Network's /64 prefix plus the Interface Identifier of the 6bed4 Peer should be dropped.

In general however, Neighbor Solicitation serves attempts to contact a remote 6bed4 Peer with an unknown 6bed4 Link-Local Address. In terms of the non-normative conceptual model, that could be recognized if the IPv6 stack sent a Neighbor Solicitation through the 6bed4 Link.

If the request is for an address that starts with fe80::/10 or fe80::/64, then a Neighbor Advertisement must be provided by retrieving the 6bed4 Link-Local Address from the Interface Identifier. The only exception would be for address fe80::/128, which should be interpreted as a reference to the 6bed4 Router; if it is requested, then the 6bed4 Link-Layer Address of the 6bed4 Router must be provided in a Neighbor Advertisement sent back over the 6bed4 Link.

What remains is to process Neighbor Solicitation for remote 6bed4 Peers on the 6bed4 Network. The process run to obtain the 6bed4 Link-Local Address for these IPv6 addresses is to attempt all candidate bypasses in turn, by sending a Neighbor Solicitation to the candidate 6bed4 Address of the bypass. There may be a few attempts for each candidate bypass to compensate for network losses. When a





Neighbor Advertisement is received in reply to one of these Neighbor Solicitations, then the 6bed4 Link-Local Address is accepted as the way to contact the requested target address, which is then relayed back in the form of a Neighbor Advertisement to the IPv6 stack over the 6bed4 Link.

Multiple bypass routes may be candidates for a single IPv6 target. The search starts with the nearest-by targets, and ends with the farthest away.

The first bypass route is only useful if the local and remote 6bed4 Peers exhibit the same IPv4 address, and if the 6bed4 Peer subscribes to (and also sends to) the multicast address 224.0.0.1 and UDP port TBD1 for all 6bed4 nodes on the local network. If these conditions apply, then the first candidate bypass to consider is found by sending a Neighbor Solicitation for the target IPv6 address to this multicast address and port.

The second bypass route can be used anywhere, even if the local and remote 6bed4 Peer exhibit the same IPv4 address. It is tried by sending a Neighbor Solicitation to the IPv4 address and UDP port as found in the 6bed4 Link-Local Address of the remote 6bed4 Peer. This may work through a Pinhole or a direct connection, but in general it would attempt contact through the outside of any Network Address Translation device used by the remote 6bed4 Peer. Whether this works depends on the types of NAT and Firewall in the path between the 6bed4 Peers. If both use symmetric NAT, then it will never work; if only the remote side is symmetrical, then the initiative to find a direct route must be initiated by the remote 6bed4 Peer, and a future Redirect would be needed to restart the Neighbor Solicitation from the local 6bed4 peer. But in many other situations, the direct connection is likely to work.

Neighbor Advertisements sent in reply are handled as described above; plus, they would cancel the search for bypass routes for the advertised target, since a bypass appears to have been found.

Only if all bypass routes fail to respond within a reasonable time, would it be necessary for the 6bed4 Peer to fallback to the 6bed4 Router. In terms of the non-normative conceptual model, it would send a Neighbor Advertisement over the 6bed4 Link mentioning the 6bed4 Link-Local Address of the 6bed4 Router for the targeted remote 6bed4 Peer; in terms of that same model, the success in one of the foregoing steps would result in sending a Neighbor Advertisement with the succeeded 6bed4 Link-Layer Address for the targeted remote 6bed4 Peer.

Depending on the original Neighbor Solicitation, the bypass routes attempted and reporting on it may be varied; for instance, for unicast versions it may suffice to attempt only the bypass routes up to and including the nearness of the requested form, and there is no need to report the 6bed4 Router's address.

6.12. Handling Neighbor Advertisement from the 6bed4 Link

This subsection is an illustration of a possible implementation; it is non-normative.

If the IPv6 stack relays a unicast Neighbor Advertisement message over the 6bed4 Link, then the packet must be replicated and sent over the 6bed4 Network. If it is a multicast version, then it may be dropped.

6.13. Handling Redirect from the 6bed4 Network

It is necessary to send and receive 6bed4 traffic between two 6bed4 Peers over the same route; so either both send over the local network, or both send to direct 6bed4 Link-Local Addresses, or both send through the 6bed4 Router. This can be achieved by ensuring that the same Metric applies in both directions.

If a route is established by one 6bed4 Peer, it knows that traffic can pass bidirectionally, so it can safely assume that the remote 6bed4 Peer can use it also. If it receives traffic from a remote 6bed4 Peer over a bypass with a higher cost metric, then it SHOULD send a Redirect message to the 6bed4 Peer, instructing it to continue sending over the more direct route. In this Redirect message, it MUST include as much of the causing message as it can fit in. The Destination Address included MUST be made from the fe80::/64 prefix plus the Interface Identifier of the 6bed4 Peer. The message MUST be sent to the 6bed4 Peer over the bypass for which it indicates a preference.

Upon receiving a Redirect, as much as possible MUST be verified; the causing packet if possible, the match between the 6bed4 Address over which the packet arrived and the Interface Identifier of the Destination Address; whether the target is currently known to the local 6bed4 Peer; and whether the 6bed4 Link-Local Address in the Interface Identifier would actually have been considered as a candidate 6bed4 Link-Local Address for the target. If any of these tests fails, then the Redirect MUST be dropped. Otherwise, its processing SHOULD consist of sending a Neighbor Solicitation to the proposed 6bed4 Address. If this leads to a Neighbor Advertisement that would also have been acceptable to the normal process of Neighbor Discovery, then it MUST be accepted as a replacement of the current 6bed4 Link-Local Address for the target. In terms of the non-normative







conceptual model, this may be implemented with a Neighbor Advertisement over the 6bed4 Link that has its Override flag set.

6.14. Handling Redirect from the 6bed4 Link

The 6bed4 Link is part of the non-normative conceptual model. If this interface would generate a Redirect message, it must be ignored.

6.15. Dealing with Packet Fragmentation

The IPv6 specifications define a minimal MTU that exceeds that of IPv4; as a result, it is always possible for packets on the 6bed4 Network to be subjected to fragmentation. The 6bed4 Peer SHOULD NOT set the Don't Fragment flag [RFC791] on IPv4 traffic and it SHOULD be prepared to reconstruct packets from fragments.

The 6bed4 Peer MAY reject packets in excess of the 1280 byte MTU for IPv6 traffic; it MUST then respond with an ICMPv6 message of type 2, Packet Too Big.

7. Impact of Firewalls and Network Address Translation

The 6bed4 mechanism has been carefully designed to work behind any sequence of Firewalls or Network Address Translators. The reason that it handles these well has two causes: First, it uses UDP, which is an opaque carrier that is generally supported by these devices. Second, the traffic is internally initiated, by sending a Router Solicitation to a 6bed4 Router. This is the direction in which both Firewalls and NAT open a hole that permits return traffic.

Since UDP is stateless, all holes in Firewalls and NAT are subject to timeouts, and therefore regular traffic is needed to keep such holes open. To this end, a message may be sent to the 6bed4 Router at a regular interval, if no other traffic has been sent. In cases where this fails, an outgoing message will be corrected with a new Router Advertisement revealing the new 6bed4 Link-Local Address of the 6bed4 Peer. If this happens, any communication partners would conclude that the 6bed4 Peer had gone offline; regular traffic may be preferred to avoid this.

It may be beneficial to support 6bed4 on servers or routers, even if they are connected through native IPv6. This could help to have more direct connectivity to popular 6bed4 Networks. If server nodes are located behind a sequence of Firewalls and/or NAT, then Port Forwarding may be setup explicitly to fixate an IPv6 address, by avoiding UDP port changes. When this is done, the IPv6 address could safely be entered in DNS as a server address.







8. Security Considerations

The 6bed4 protocol uses configuration packets that were designed for local network use, but it employs them on the public Internet. This is not a change to be made lightly. Furthermore, 6bed4 is a tunnel protocol, which itself implies certain responsibilities.

All tunnel protocols should reject forged inner addresses; the 6bed4 Peer and 6bed4 Router tackle this by validating the address information upon entry; the IPv6 source address must match with the IPv4 address and UDP port from which the packet originated. The only exception is the Router Advertisement sent to the 6bed4 Router, but this is a mindful choice, and the packet is not permitted to travel beyond the 6bed4 Router.

The link between an IPv4 address (which is hopefully subjected to egress filtering) and the tunnel's inner IPv6 address means that it is not easier to forge an IPv6 address than it is to forge an IPv4 address. Furthermore, the use of an IPv4 address as part of the IPv6 address means that network attackers are as traceable on a 6bed4 Network as they are on the IPv4 network.

This specification does not introduce anything that could cause a 6bed4 Peer to contact a wrong 6bed4 Router; the mechanism for determining the IPv4 address for that service falls outside the scope of this specification.

Special attention is warranted for the Neighbor Advertisement and Router Advertisement messages, as these announce information that influences message redirection of a 6bed4 Peer. These concerns are addressed by the precaution to validate source addresses. In addition, the information claimed is validated as per the behavioral specifications given above.

Another specific concern relates to the ability to send Redirect packets. Once again, validation on source addresses addresses these concerns. In addition, the packet is only a hint and not an update instruction. When it is received, a Neighbor Solicitation process will only commence if the targeted 6bed4 Peer is already known to the 6bed4 Peer. This process is predictable, so a Neighbor Advertisement could be crafted ahead of time; however, these Neighbor Advertisements are also subject to matching of the source address.

In summary, the risks caused by the tunneling technique and the use of Neighbor Discovery style packets on the 6bed4 Network are limited to situations where it is possible to send traffic from forged IPv4 address. In such situations, the problems are much more general and serious.







9. IANA Considerations

This specification allocates one resources from one existing registry.

9.1. Registered UDP port: 6bed4

The registered port for the 6bed4 protocol is TBD1. This port is used to select 6bed4 traffic sent to a 6bed4 Router, as well as to the multicast address 224.0.0.1 for all nodes on the local subnet.

[TO BE REMOVED: This UDP-only port number registration should take place at the following location: http://www.iana.org/assignments/port-numbers -- The port number MUST be an even number. I would like to request port number 25788 because it shows up in IPv6 addresses as a recognizable ...:be64:...]

## 10. References

### 10.1. Normative References

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.





### 10.2. Informative References

[RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, December 2006.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

## Author's Address

Rick van Rein
OpenFortress Digital signatures
Haarlebrink 5
7544 WP Enschede
The Netherlands

email: rick@openfortress.nl