Internet Engineering Task Force
Internet-Draft
Expires: April 18, 2005

R. Suryanarayanan
S. Madanapalli
Samsung India Software Operations
K. E. Nielsen
Ericsson
F. Parent
Hexago
J. Palet
Consulintel

October 18, 2004

Zero-Configuration Tunneling Requirements
draft-suryanarayanan-v6ops-zeroconf-reqs-00.txt

Status of this Memo

Copyright Notice

Abstract

This document describes the set of goals to be fulfilled by a

Zero-Configuration Tunneling protocol.

Zero-Configuration Tunneling here denotes an automatic tunneling mechanism that could be used by a Service Provider to offer IPv6 connectivity to its customers in early phases of IPv4 to IPv6 transition.

Table of Contents

1. Introduction

The IETF v6ops Working Group has identified and analyzed deployment scenarios for IPv4/IPv6 transition mechanisms in various stages of IPv6 deployment and IPv6 and IPv4 coexistence.

This work has been carried out for a number of different network environments each with their particular characteristics: Enterprise, ISP, Unmanaged and 3GPP networks, see e.g. [1], [2], [3] and [4].

The work has identified a need for automatic IPv6-in-IPv4 tunneling mechanisms that provide bidirectional IPv6-in-IPv4 tunneled connectivity between dual stack end-nodes located at an IPv4-only access network and dual-stack tunnel servers located at IPv6-IPv4 network boundaries within the Service Providers network.

The term Zero-Configuration Tunneling is used in this document to denote a tunneling mechanism that fulfills the goals as put forward here.

A Zero-Configuration Tunneling mechanism provides a set of minimal features required for automatic establishment of IPv6 connectivity.

For scenarios demanding advanced tunneling features, for example full emulation of native (though tunneled) IPv6 connectivity, a more full-fledged tunneling mechanism is envisaged to be deployed, see [5]. With respect to the latter, an analysis of appropriate mechanisms for automatic discovery of the tunnel endpoint is being done in [6], which will be also useful for the zero-configuration tunneling protocol.

One of the major differences between the zero-configuration tunneling mechanism and the full-fledged tunneling mechanism is that the former does not support user authentication, which should not be an issue because the scope of the users of this mechanism are already users of the Service Provider actually deploying it. Consequently, the users are to be authenticated by other means, which are out of the scope of this document.

It should be emphasized that unless otherwise specified, in this document the reference, IPv6-in-IPv4 encapsulation as defined in [7], refers to the aspects of Protocol-41 encapsulation related to IPv4 header construction (except for source and destination address determination), MTU and Fragmentation, Hop Limits and ICMP handling as detailed in Section 3.1-3.6 of [7]. The particular aspects of Configured IPv6-In-IPv4 Tunneling in the areas of IPv4 source and destination address determination, tunnel link characteristics and IPv6 Neighbor Discovery operation are not intended referred to by the

above reference.

This document only identifies requirements for a zero-configuration tunneling mechanism, based on which solutions can be developed or identified.

2. Terminology

Zero-Configuration Tunneling site: A logical IPv4 network over which IPv6 connectivity is provided to dual-stack nodes by means of Zero-Configuration Tunneling.

Tunnel End-Point (TEP): A dual-stack node performing IPv6-in-IPv4 tunnel encapsulation/decapsulation in accordance with Zero-Configuration Tunneling.

Tunnel Server (TS): A dual-stack server node with IPv6 connectivity and which provides IPv6 connectivity to client nodes by performing IPv6-in-IPv4 tunnel encapsulation/decapsulation to/from client nodes in accordance with Zero-Configuration Tunneling. A Tunnel Server is likely to be a dual-stack router.

Tunnel Client: A dual-stack node that obtains IPv6 connectivity by means of Zero-Configuration Tunneling. A tunnel client relies on IPv6-in-IPv4 tunnel encapsulation/decapsulation to/from Tunnel Servers for IPv6 communications to native IPv6 nodes.

Direct Tunneling: Direct tunnelling here refer to the case where end-hosts located within the same Zero-Configuration Tunnelling site may circumvent the Tunnel Server and communicate directly using the tunnel protocol.

CPE: Customer Premises Equipment.

3. Applicability

Zero-Configuration Tunneling is applicable in different IPv6 transition scenarios. The focus of this document is to define the requirements for Zero-Configuration Tunnelling mechanism in the following Service Provider contexts:

- 3GPP scenarios [4].
- Unmanaged network scenarios [3].
- ISP scenarios [2].
- Enterprise scenarios [1].

Zero-Configuration Tunneling does not attempt to provide emulation of the full set of native IPv6 connectivity functions as defined by [8], [9] and [10]

It is possible that the same zero-configuration Tunneling mechanism can be used in various deployment scenarios. However, it is not required that same tunnel set-up protocol be deployable in all scenarios.

4. Limitations

4.1 IPv6 address allocation, Scope and Limitations

It is not explicitly within the scope to support privacy extensions to IPv6 [11].

It is not explicitly within the scope to support usage of IPv6 multicast.

No goals are defined as to how address configuration should be performed. This may be done based on legacy stateless or stateful IPv6 address configuration mechanisms or by some altogether different mechanism particular to the zero-configuration solution.

4.2 IPv6 tunnel link characteristics, Scope and Limitations

Direct tunneling is neither an explicit goal nor explicitly excluded in Zero-Configuration Tunneling.

It is not an explicit requirement for the zero-configuration tunnel link to support IPv6 link-local multicast.

The tunnel protocol should allow for the formation of a link-local address on the tunnel link, though no particular usage of such an address is explicitly demanded by the goals set forward here.

It is an explicit goal that nodes attached to a tunnel link must be able to ascertain the reachability of neighbors with which it is communicating (or wish to start communicate). This may be achieved using IPv6 Neighbor Discovery mechanism ([12]) based on unicast link-local packet exchanges (or link-local multicast if such is supported) but it may also be achieved by altogether different mechanisms.

5. Basic Assumptions and Prerequistes

Zero-Configuration Tunneling is a tunneling mechanism by virtue of which dual-stacks hosts, attached to IPv4-only networks links, can

use IPv6-in-IPv4 encapsulation as defined in [7] to tunnel servers for global IPv6 connectivity.

Zero-configuration Tunneling is a simple mode with no user registration, essentially deployed in a controlled and "authenticated" environment where the service is made available to all the IPv4 customers.

The aim of the document is to define the set of goals to be fulfilled by zero-configured tunneling when the following assumptions are made on the deployment environment:

- The first-hop ISP is providing IPv6 connectivity.
- The Tunneling protocol must not require prior registration, or require registration during the protocol set-up phases.
- The Zero-Configuration Tunneling site is protected from proto-41 encapsulated packets arriving from external IPv4 networks.
- At least one authoritative DNS server is IPv4-enabled and at least one recursive DNS server supports IPv4. Further IPv4 DNS Server discovery is provided by already existing means/means outside the scope of the tunnel protocol.
- The user is being authenticated to the network by means external to the tunneling protocol.

The following assumption is only valid for basic requirements where there is no NAT in the path.

- The Zero-Configuration Tunneling network is fully penetrable for intra-site IPv6-in-IPv4 Protocol 41 traffic.

It is a prerequisite that the tunnel protocol must work in IPv4 network environments where IPv4 multicast is not provided.

6. Requirements for Zero-Configuration Tunneling Mechanisms

6.1 Basic Requirements

The basic requirements described below must be supported by any zero-configuration tunneling protocol. Tunneling protocol satisfying these basic requirements could be used in a deployment scenarios which is NAT-free, does not require IPv6 /64 address or prefix delegation.

6.1.1 Simplicity

The tunnel protocol is easy to implement in the targeted environment. Additionally, the protocol should provide a reasonable,limited set of basic IPv6 connectivity features

6.1.2 Automated IPv6-in-IPv4 tunnel establishment

The Tunnel protocol should provide for the set up of IPv6-in-IPv4 tunnels, based on IPv6-in-IPv4 encapsulation as defined in [7], from dual-stack nodes, attached to IPv4-only networks, to Tunnel Servers.

Zero-configuration tunneling is defined for simple "plug and play" scenarios. In this mode, the tunnel establishment is triggered through the execution of a simple program, without any pre-configuration or pre-registration required from the end-user.

The mechanism must be fully dynamic in the sense that it must not require IP address information such as the IPv4 address of a Tunnel Server and/or the IPv6 address(es) to use for IPv6 connectivity to be configured on the Tunnel Clients beforehand.

6.1.3 IPv6 Address Assignment and Prefix Delegation

Assignment of an IPv6 address to the end-node must be supported.

No goals are defined as to how address configuration should be performed. This may be done based on legacy stateless or stateful IPv6 address configuration mechanisms or by some altogether different mechanism particular to the zero-configuration solution.

Prefix Delegation support is dealt with respect to various deployment scenarios in sections 6, 7, 8 and 9. It is not however required that any tunneling protocol supporting only basic requirements provide support for prefix delegation.

It is preferable that the address assignment provides a stable address, that is, an address that can be used for IPv6 connectivity for a certain amount of time rather than solely one address per higher layer session initiation

6.1.4 Use Native Connectivity When available

The node should not use Zero-Configuration Tunneling when native IPv6 connectivity is available.

The fact that a node should not use Zero-Configuration Tunneling when native IPv6 connectivity is available is not considered to be a

functional requirement on the tunnel protocol.

6.1.5 Tunnel Server End-Point Discovery

In order to offer "plug and play", the implementation should allow a mechanism to discover the address of the tunnel server that will provide the tunnel connectivity. This discovery should be automatic within a Service Provider's network.

6.1.6 Tunnel End-Point Reachability Detection

The tunnel protocol must allow for means for one tunnel end-point to verify the reachability of other tunnel end-points towards which it intends to send packets in a method similar to IPv6 NUD.

It is preferable that a Tunnel Server monitors the reachability of the tunnel client towards which it is sending packets. Full emulation of IPv6 NUD mechanism is however not required to be supported.

6.1.7 Private and public IPv4 addresses

The tunnel protocol must work over IPv4 sites deploying both private and public IPv4 addresses.

Furthermore, the tunnel protocol should work with both dynamic and static IPv4 address allocation.

6.1.8 Scalability and Load-Balancing

The tunnel set-up protocol must be scalable.

Load balancing should be planned in advance during the early phases of deployment. Given adequate planning it should be possible for a Service Provider to seamlessly deploy additional Tunnel Servers in order to support an increased amount of Tunnel Clients.

This may be achieved using load balancing functions provided by the Tunnel Server End-point Discovery mechanism as detailed in [13].

6.1.9 Easy to deploy and Easy to Phase Out

Zero-configuration Tunneling is a transition mechanism to enable Service Provider to jump start IPv6 service without requiring an immediate global upgrade of access networks.

The tunneling protocol should be easy to deploy into the existing network infrastructures.

Once IPv6 is available natively in the access network, it should be easy to phase out the tunneling protocol.

6.1.10 Latency in Set-up Phases

In certain type of networks, keeping tunnels active all the time is not possible. In such environments, the protocol must be able to set-up tunnels on demand when the IPv6 connectvity either native or through tunneling is unavailable. The tunnel will be set-up only once though for the end-node and not per session.

The tunnel set-up protocol must then have a low enough latency to enable quasi-instant configuration. Latency is usually a function of the number of packet exchanges required, so minimizing this parameter is important.

6.1.11 Security

The tunneling Protocol must not introduce any new vulnerability to the network.

6.2 Advanced Requirements

It is not required that the tunneling protocol support one or all of the advanced requirements described below. Support to these advanced requirements by tunneling protocol are driven by the deployment scenarios.

6.2.1 Tunnel Link Sustainability

In certain environments, like in 3GPP, to minimize the overhead and latency associated with tunnel initialisation, it is highly desirable that tunnels remain active for a large amount of time, ideally infinitely. In such environments, the tunnel protocol must not mandate keep-alive messages to be transmitted by the host simply in order to sustain tunnel link connectivity.

In other environments, the Tunnel Server may perform some garbage collection if it is configured to do so. The keep-alive messages can enable the tunnel server to perform garbage collection of its resources when tunnels are not in use anymore.

To enable this functionality, the tunnel set-up protocol must include the transmission of keep-alive messages and time interval.

Implementations, where keep-alive messages are used, must provide facility to turn-off transmission of keep-alive messages. In such cases the tunnel server might use other metrics to perform garbage

collection.

The tunneling protocol should be able to restart the connectivity establishment process if the tunnel no longer is available.

6.2.2 NAT Traversal

The Tunnel set-up protocol must be able detect the presence of one or more NATs in its path. It must be able to adapt to the following cases, by choosing the most optimal tunnel encapsulation depending on the presence of a NAT.

a single node,

a leaf network,

using a globally routable IPc4 address,

behind a NAT (customer or ISP owned),

using dynamic IPv4 address (internally or externally to the NAT).

6.2.3 Firewall Traversal

Even if no NAT is in the tunnel path, there may be a firewall which prohibits proto-41. In such case, the tunnel encapsulation selection based on NAT detection will select a tunnel that will not work.

The implementation must allow a user to explicitly specify the desired tunnel encapsulation, regardless of the NAT detection process.

6.2.4 Extensibility

The protocol must be extensible to support tunnel encapsulation other than IPv6 in IPv4 and IPv6 in transport in IPv4. In particular, encapsulation of IPv4 in IPv6 or IPv6 in IPv6 could be defined.

6.2.5 IPv6 Address Stability

[This section shall be removed after getting more opinions from others.]

The IPv6 address is "transient" and may change, but the protocol should offer a mechanism to provide IPv6 address stability (e.g. cookie mechanism). The implementation of this mechanism must allow this feature to be turned off.

7. 3GPP Specific Requirements

The 3GPP goals of zero-configuration tunnelling covers the basic requirements in section 6.1 and advanced requirement in section 6.2.1.

Any zero-configuration tunneling protocol satisfying the above mentioned sections must take into account constrained conditions of the 3GPP environment. For details see [14]

8. Unmanaged Networks Specific Requirements

An unmanaged network is where no network manager or staff is available to configure network devices. Zero-Configuration Tunneling Protocol is quite useful in this context where automation of IPv6 connectivity to first-hop ISP and prefix assignment is handled.

Unmanaged Networks [3] may or may not be behind a NAT.

A zero-configuration tunneling mechanism should satisfy the basic requirements (see section 6.1) and should take into account the specific requirements described below.

8.1 Address Assignment and Prefix Delegation

In unmanaged networks, assignment of an IPv6 address (/64) to the end-node must be supported.

Prefix Delegation must also be supported.

8.2 NAT Traversal

Zero-configuration Tunneling must work with the existing infrastructure, in particular it must be compatible with the various customer premise equipments available today. This means that, in particular, the tunnels must be able to traverse one or many NAT boxes of different kinds. Hence, Tunneling through IPv4 NAT must be supported.

There are actually two cases where the IPv4 address of the customer tunnel end point can be dynamic, and both must be supported:

- The device used as tunnel end point is using a dynamic IPv4 address provided by the ISP.

- The device used as tunnel end point is located behind a customer owned NAT box that is also acting as a local DHCP server. In that case, the device IPv4 address may change after a reboot.

There is no requirement for any particular NAT traversal technology. However, as NAT traversal usually requires an extra layer of encapsulation, the tunnel set-up protocol should be able to detect automatically the presence of one or more NAT boxes in the path during the set-up phase.

The implementation must provide an option to to turn on extra encapsulation manually.In order to assure interoperability, at least one common tunnel encapsulation type must be supported.

8.3 Firewall Traversal

As indicated in section 6.2.3, the tunneling protocol must be able to work in networks where the firewall prohibits proto-41 packets.

8.4 Tunnel Link Sustainability

The keep alive messages can enable the ISP tunnel end point to perform garbage collection of its resources when tunnels are not in use anymore.

When a tunnel has to cross a NAT box, the mapping established by the NAT must be preserved as long as the tunnel is in use. This is usually achieved by sending keep-alive messages across the tunnel.

A client may choose not to send those messages (for example on ISDN type links). In this case, the client should be able to handle a tunnel disconnect event and be able to restart the set-up phase to re-establish the tunnel.

8.5 Extensibility

8.5.1 IPv4-in-IPv6 Tunneling

Unmanaged networks [3], calls for providing a connectivity solution when the first-hop ISP no longer supports v4. In such a scenario, the connectivity has to be provided by a third party ISP using assisted/managed methods, and hence it is out of scope of this document.

8.6 Scalability

Typically, this protocol should be scalable for deployment in broadband ISP.

9. Enterprise Network Requirements

The zero-configuration Tunneling protocol is not applicable for

managed enterprise networks to get external IPv6 connectivity. So the scope of this document is restricted to dealing with zero-configuration requirements for internal connectivity in enterprise networks.

In an enterprise network where IPv4 is dominant, a tunneled infrastructure can be used to provider IPv6 services to the IPv6 islands (hosts or networks) inside the enterprise, before a full IPv6 native infrastructure is built. Zero-Configuration tunneling protocol can be used to give IPv6 connectivity and prefix information for the islands. This gives to the enterprise a basic deployment of IPv6 while maintaining automation and permanence of the IPv6 assignments to the islands.

There can also be a scenario where the remote users use IPv4 VPN to connect to the enterprise, where they are assigned an IPv4 address from the enterprise address space. In such case, zero-configuration tunneling mechanism is applicable since the IPv4 source address is already authenticated for use. But typically, this is the same case where the node is inside the enterprise network.

In cases where the network administrator is sure about the absence of internal NAT and firewalls in the network, and end-nodes will need only IPv6 /128 address, a tunneling protocol satisfying only the basic requirements will suffice.

A zero-configuration tunneling mechanism should satisfy the basic requirements (see section 6.1) and should take into account the specific requirements described below.

9.1 IPv6 Address Assignment and Prefix Delegation

Assignment of an IPv6 address (/64) to the end-node must be supported. Prefix Delegation must be supported.

9.2 NAT Traversal

Tunneling through IPv4 NAT must be supported. The protocol should detect if an IPv4 NAT is in the path during the set-up phase If a NAT is present, an extra level of encapsulation is necessary to tunnel IPv6 across the NAT. If no NAT is detected, IPv6-over-IPv4 tunneling (protocol-41) is enough.

9.3 Firewall Traversal

The tunneling Protocol must be able to handle scenario where firewall is in the path. See Section 6.2.3.

9.4 Extensibility

9.4.1 IPv4-in-IPv6 Tunneling

The tunneling Protocol should be able to handle automatic establishment of IPv4-in-IPv6 tunnels. It must be able to handle assignment of temporary IPv4 address and other tunnel parameters as required.

10. ISP Network Specific Requirements

In some scenarios the ISP has IPv4-only customer connection networks and a backbone that supports both IPv4 and IPv6.

If the customer connections might not yet been upgraded, a tunneling mechanism has to be used to provide IPv6 connectivity through the IPv4 customer connection networks. The customer can terminate the tunnel at the CPE (if it has IPv6 support) or at some set of devices internal to its network. That is, either the CPE or a device inside the network could provide global IPv6 connectivity to the rest of the devices in the customer's network.

Zero-configuration tunneling mechansim is very useful in such scenarios.

NATs might be present at the ISP, if ISP provides IPv4 connectivity using private IPv4 address. In many cases, the customer also has a NAT of his/her own. So its required that the tunneling protocol be able to work when one or more NATs are present in the path.

These scenarios are very similar to unmanaged networks, the zero-configuration tunneling requirements described in section 7.1, 7.2, 7.3, 7.4 and 7.6 holds good in ISP networks as well.

In a scenario where connection to customer from ISP supports both IPv4 and IPv6, but the customer has IPv6-only network and the ISP backbone is IPv4-only, IPv6 packets from customer needs to be tunneled over the IPv4 backbone to the next upstream ISP. Zero-configuration tunneling mechanism can be used in such scenario as well.

11. Security Considerations

11.1 Access Control

Zero-configuration Tunneling does not require explicit authentication of the user. This essentially offers the IPv6 service to any of the provider IPv4 customers.

Should an Operator/Administrator wish to implement additional access control, e.g., limiting the service to certain customers, then in the case where IPv4 source spoofing prevention is performed within the Operators network, mere filtering on the IPv4 address could give this. Such mechanisms, however, would be external to the tunnel protocol itself and are outside the scope of this document.

In any case, the service should be limited to the provider network and the assumption that the Zero-Configuration Tunneling site is protected from protocol-41 encapsulated packets arriving from external IPv4 networks, should indeed effectively prevent access to the service from outside the provider network.

If for some reason an Operator/Administrator deviates from the above assumption or if additional security measures are wanted (just in case) then proper ingress filtering in the ISP core network together with IPv4 source address filtering would limit the access to internal customers only.

If the mentioned filtering is not in place in the ISP core network, anyone on the Internet could start using its tunneling infrastructure to get free IPv6 connectivity, transforming effectively the ISP into a IPv6 transit provider.

11.2 General Threats

The following have been identified as potential threats applicable to the network and infrastructure nodes within a Zero-Configuration Tunneling site regardless of whether the individual node implements Zero-Configuration Tunneling or not:

- It may be possible to use a tunnel server to reflect tunneled packets into the network, similar to the 6to4-reflection attacks identified in [15].

- In the case of no internal Firewalls or NATs and no interaction with such being performed by the tunnel protocol, the Zero-configuration site must be kept penetrable for intra-site IPv6-in-IPv4 protocol-41 encapsulated packets. This may open up for threats to end-hosts that rely on the network infrastructure to filter out Protocol-41 encapsulated packets.

- Zero-configuration tunneling may open up threats to other mechanisms in the network that rely on Protocol-41 encapsulation.

Detailed analysis of the validity of these threats will have to depend on the particular Zero-Configuration solution. In general it could be noted that attacks based on the above threats largely should

be preventable if the end-hosts in the network implement appropriate drop policies, either simple drop all protocol-41 policies or more differentiated policies based, e.g., on source addresses.

11.3 Threats to nodes implementing Zero-Configuration Tunneling

The following considerations apply to the situation where Zero-Configuration Tunneling is deployed in between tunnel servers and end-hosts only.

Special security considerations for the usage of Zero-Configuration Tunneling for direct tunneling in between end-hosts is given in Section 12.4.

11.3.1 Threats to end-hosts

In current IPv6 networks hosts need to trust on the benevolence of their default routers as well as hosts must trust that anyone impersonating as a router is indeed one, see, e.g., the trust models and threats described in [16].

Future multi-access IPv6 networks may rely on SEND mechanisms, i.e., mechanisms developed in the SEND WG inorder to mitigate the threats described in [16], to establish a trust relationship in between host and routers.

In this context is it constructive to look at the following three categories of Zero-Configuration Tunneling sites:

1. Environments with IPv4 source address spoofing prevention, e.g. 3GPP environments and "filtered" ISP and Unmanaged environments.

2. Open, un-trusted environments without IPv4 source address spoofing prevention, e.g., ȼun-filteredȭ ISP and Unmanaged environments.

3. Closed, trusted, environments without IPv4 source address spoofing prevention, e.g., Enterprise environments.

In all environments, but in open environments in particular, it is assumed a prerequisite that a trustworthy Zero-Configuration tunnel server end-point discovery mechanism is implemented.

Given this, then in a Zero-Configuration Tunneling Site of the first category (1.), end-host can trust that packets they perceive as stemming from Tunnel Servers (identified by IPv4 address) do actually stem from such and further they can trust on the benevolence of these Tunnel Servers.

In Zero-Configuration Tunneling Sites of the latter two categories (2 and 3), then due to possibility of IPv4 address source spoofing, this is not possible even when a trustworthy Zero-Configuration Tunnel Server end-point discovery mechanism is implemented.

For trusted Zero-Configuration Tunneling Sites (category 3), the threats may be considered to be manageable, as the environment itself is assumed to be trusted. End-hosts in Zero-Configuration Tunneling Sites of category 2, however, are exposed to the same threats as hosts in non-SEND multi-access IPv6 networks.

11.3.2 Threats to Tunnel Servers

Zero-Configuration Tunneling may be deployed over very large IPv4 sites with low density of active tunnel clients but with a very high number of dormant, but potential tunnel clients. Therefore Denial-of-Service prevention by strict over provisioning of Tunnel Server capacity is unlikely to be performed.

11.3.2.1 Tunnel State related risks

If the Tunnel Server relies on state to be kept per tunnel client that it serves, the server risks resource exhaustion.

In this situation it is a security prerequisite that no node, whether located within or outside the Zero-Configuration Tunneling site, cannot initiate initialization of tunnel state for other entities than itself (identified with IPv4 address).

But even in this case, then in situations where:

- IPv4 address spoofing is possible.
- An unlimited number of tunnels may created per node, e.g. in NAT traversal environments it may still be possible for one or a limited number of nodes to exhaust the resources of the server.

Such attacks, however, may be mitigated by performing IPv4 return routability checks as an intrinsic part of tunnel initialization (first case) or/and by limiting the number of tunnels that may be created per node (second case).

11.3.2.2 Traffic related risks

Tunnel encapsulation is recognized as being more resource demanding than mere packet forwarding. Given the same traffic load a Tunnel Server must thus be more generously provisioned that a corresponding router for it not to be more likely to get overthrown by large

unexpected amounts of traffic than the router.

The authors have found no plausible treats to the tunnel service, due to large unexpected amounts of traffic needing encapsulation, which can be classified as a security threat rather than a case of under-provision. This regardless of whether the traffic is due to a surge in the density of active tunnel clients or due to a surge in the traffic streams set-up by active clients.

11.3.2.3 Packet Delivery related threats

One potential risk related to packet delivery has been identified. This risk is the equivalent of the threat to routers in multi-access environments described in [16] Section 4.3.2.

The risk is associated with the special case where the tunnel protocol requires special resource demanding and/or temporary state creation actions to be taken by the Tunnel Server for delivery of packets destined for not recently addressed Tunnel Clients. The situation where such actions must be performed for all packets at all times is considered to be unlikely. The actions required could be buffering of packets while the reachability of the destined node is being verified.

In case a malicious node (located either within or outside the zero-configuration site) is able to continuously send packets to continuously changing nodes, which by the Tunnel Server is perceived as being existing or potential client nodes, the malicious node may be able to exhaust the Tunnel Servers capability of delivering packets by saturating the packet buffering mechanism and the reachability state table as well as by keeping the Tunnel Server busy determining the reachability state of the ever changing client nodes.

The above threat will not be relevant if reachabilityis performed as an intrinsic part of the, thus stateful, tunnel protocol, e.g., by relying on periodically transmitted keep alive messages.

11.4 Implications of Direct Tunneling

In case direct tunneling in between end-hosts is provided by the tunneling protocol, it will not (as described in Section 1.3.1) be possibly for end-hosts to filter out received Protocol-41 encapsulated packets based on whether the IPv4 source is an address belonging to a trusted or perceived Tunnel Server as such behavior evidently would break direct tunneling.

As other end-hosts generally are non-trusted, direct tunneling may thus open up for attacks against IPv6 ingress filtering.

Detailed analysis of the validity of this threat will have to depend on the particular zero-configuration solution.

12. Acknowledgements

The work done by the authors on the zero-configuration tunneling requirements for 3GPP ([14]) and on assisted-tunneling ([5]), has been the main inspiration for the Zero-Configuration Tunneling requirements work.

This work has benefited from input and comments provided by IPv6 Team in Samsung India Software Operations (India) for the initial phase of the work.

The authors would like to acknowledge also the inputs from Pekka Savola and the European Commission support in the co-funding of the Euro6IX project, where this work is being developed.

13. References

13.1 Normative References

13.2 Informative References

[1] Bound, J., "IPv6 Enterprise Network Scenarios", draft-ietf-v6ops-ent-scenarios-05 (work in progress), July 2004.

[2] Lind, M., Ksinant, V., Park, S., Baudot, A. and P. Savola, "Scenarios and Analysis for Introducing IPv6 into ISP Networks", draft-ietf-v6ops-isp-scenarios-analysis-03 (work in progress), June 2004.

[3] Huitema, C., "Evaluation of Transition Mechanisms for Unmanaged Networks", draft-ietf-v6ops-unmaneval-03 (work in progress), June 2004.

[4] Wiljakka, J., "Analysis on IPv6 Transition in 3GPP Networks", draft-ietf-v6ops-3gpp-analysis-10 (work in progress), May 2004.

[5] "Requirements for assisted tunneling", draft-ietf-v6ops-assisted-tunneling-requirements-00 (work in progress), June 2004.

[6] Palet, J. and M. Diaz, "Evaluation of v6ops Auto-discovery for Tunneling Mechanisms", draft-palet-v6ops-tun-auto-disc-01 (work in progress), June 2004.

[7] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", draft-ietf-v6ops-mech-v2-06 (work in progress), September 2004.

[8] Wasserman, M., "Recommendations for IPv6 in Third Generation Partnership Project (3GPP) Standards", RFC 3314, September 2002.

[9] Loughney, J., "IPv6 Node Requirements", draft-ietf-ipv6-node-requirements-11 (work in progress), August 2004.

[10] IAB and IESG, "IAB/IESG Recommendations on IPv6 Address Allocations to Sites", RFC 3177, September 2001.

[11] Narten, T. and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, January 2001.

[12] Narten, T., Nordmark, E. and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[13] Palet, J., "IPv6 Tunnel End-point Automatic Discovery Mechanism", draft-palet-v6ops-solution-tun-auto-disc-00 (work in progress), September 2004.

[14] Nielsen, k., "Goals for Zero-Configuration Tunneling in 3GPP", draft-nielsen-v6ops-3GPP-zeroconf-goals-00 (work in progress), October 2004.

[15] Savola, P., "Security Considerations for 6to4", draft-ietf-v6ops-6to4-security-04 (work in progress), July 2004.

[16] Nikander, P., Kempf, J. and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.

Authors' Addresses

Radhakrishnan Suryanarayanan
Samsung India Software Operations
No. 3/1 Millers Road
Bangalore
India

Phone: +91 80 51197777
EMail: rkrishnan.s@samsung.com

Syam Madanapalli
Samsung India Software Operations
No. 3/1 Millers Road
Bangalore
India

Phone: +91 80 51197777
EMail: syam@samsung.com

Karen Egede Nielsen
Ericsson
Skanderborgvej 232
8260 Viby J
Denmark

Phone: +45 89 38 51 00
EMail: karen.e.nielsen@ericsson.com

Florent Parent
Hexago
2875 boul. Laurier, bureau 300
Sainte-Foy, QC G1V 2M2
Canada

EMail: florent.parent@hexago.com

Jordi Palet Martinez
Consulintel
San Jose Artesano, 1
Alcobendas - Madrid
E-28108 - Spain

Phone: +34 91 151 81 99
Fax: +34 91 151 81 98
EMail: jordi.palet@consulintel.es

Appendix A. Out of Scope

[Editor's Note: This appendix can be removed in a future revision of this document]

The following issues have been considered as being out of scope of this work:

o DNS: DNS registration of the IPv6 addresses allocated to dual

stack nodes while deploying Zero-Configuration Tunneling for IPv6 connectivity.

- Mobile IPv6: Support of Mobile IPv6 usage over the tunnel-link; here under potential mechanisms required to support MIPv6 movement detection as well as fast tunnel set-up for Mobile IPv6 session survivability.

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.