INTERNET-DRAFT F. Templin
Nokia

Expires 30 April 2003 30 October 2002

Neighbor Affiliation Protocol for IPv6-over-(foo)-over-IPv4

draft-templin-v6v4-ndisc-01.txt

Abstract

This document proposes extensions to IPv6 Neighbor Discovery for IPv6-over-(foo)-over-IPv4 links, where (foo) is either an encapsulating layer (e.g., UDP) or a NULL layer. It is essentially a lightweight, link-layer mechanism for neighbors to establish security associations, discover and dynamically re-adjust maximum receive unit (MRU) estimates, and perform unreachability detection. The protocol makes no attempt to ensure reliable message delivery; this function is performed by higher-layer protocols, e.g. TCP.

1. Introduction

The author anticipates a long-term requirement for [IPv6] operation over IPv6-over-(foo)-over-IPv4 links, where (foo)-over-[IPv4] is treated as a link layer for IPv6). Neighbors that exchange data over such links will need to make the most efficient, robust and secure use of the intervening IPv4 paths possible. The author believes that this will require link-layer extensions to enable a hybrid proactive/on-demand neighbor discovery mechanism using bi-directional links.

IPv6 nodes will need to establish security associations, determine and dynamically re-adjust maximum receive unit (MRU) estimates, and perform neighbor unreachability detection using an IPv4 intranet or internet as the link layer. Although IPv4 fragmentation is considered harmful [FRAG][FOLK], the author believes that strategically adapting to and minimizing fragmentation can provide a superior solution to strict fragmentation avoidance. Central to the design goals is the ability for neighbors to establish and maintain lightweight, link-layer associations to provide a continuous feedback loop. Although these associations take on certain aspects of reliable transport connections, the author prefers to refer to them as "neighbor affiliations".

2. Applicability Statement

- proposes a neighbor affiliation protocol for IPv6 neighbors on IPv6-over-(foo)-over-IPv4 links

- works with either automatic or configured tunnels

- may be useful for IPv6 operations in certain deployment scenarios

- may extend to future IPv6 applications beyond the case of IPv6-over-(foo)-over-IPv4

3. Terminology

The terminology of [IPv4] and [IPv6] apply to this document. The following additional term is defined:

neighbor affiliation:
a lightweight association that enables robust, efficient and secure bi-directional links between IPv6 neighbors

4. Neighbor Affiliation Protocol

When multiple IPv4 hops intervene between nodes, [DISC] provides no trust basis (i.e., neighbors are not on the same connected LAN segment) nor any specification for the nodes to monitor each other's reachability (i.e., multicast is not supported). Moreover, the path between any pair of neighbors is mutually exclusive from other on-link neighbors and may change over time. Thus, the maximum packet size a node A can receive from neighbor B may differ from the amount it can receive from neighbor C; even though all three nodes technically share the same "link". These issues are addressed through lightweight neighbor affiliations that are established on-demand and maintained proactively as long as they are in active use.

[DISC] specifies mechanisms for IPv6 nodes to discover and maintain reachability information for active neighbors. Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages are normally used for this purpose on multiple access, broadcast media. But, when an IPv4 intranet/internet is used as the link layer, the standard multicast mechanisms do not apply. For the purpose of this specification, unicast NS/NA messages are formatted as specified in [DISC, 4.3-4], and ALWAYS include a source/target link layer address option (even though [DISC] does not require these options for unicast operation). Also, as permitted in [DISC, 4.6.1], this document specifies the format of link-layer address options for use in IPv6-over-(foo)-over-IPv4 links. The following subsections describe the neighbor affiliation protocol and specific adaptations of the mechanisms in [DISC] in detail:

4.1. Affiliation Establishment

Neighbor affiliations are established using the following algorithm. The algorithm assumes that a source has a unicast packet to send to a target and invokes address resolution ([DISC, 7.2]):

1) The source sends an NS message to the target with a specially formatted source link layer address option containing a "SYN" indication. The source creates a neighbor cache entry for the target, and transitions from the "CLOSED" to the "SYN-SENT" state

2) The target receives the NS and notices that it includes a source link layer option containing a "SYN" indication. The target creates a neighbor cache entry for the source and records the physical interface the NS arrived on. The target transitions from the "LISTEN" to the "SYN-RECEIVED" state, then sends a unicast Solicited NA to the source. The NA

includes a target link layer address option that contains a "SYN/ACK" indication and a "Maximum_MRU" option initialized to the maximum receive unit (MRU) of the physical interface recored above

3) The source receives the Solicited NA and notices that it includes a target link layer option containing a "SYN/ACK" indication. It records the physical interface the NA arrived on and places the "Maximum_MRU" value in the target's neighbor cache entry. (This value is also optionally written into the destination cache entry(s) for the IPv6 destination address(es) of any packets waiting for address resolution completion [DISC, 7.2.2]. The "Maximum_MRU" value represents the current upper bound for the maximum packet size the target is able to receive, i.e., the MRU of the target's physical interface

   Next, the source transitions from the "SYN-SENT" to the "ESTABLISHED" state and sends a unicast Solicited NA to the target, i.e., the source treats the received NA as an "NS equivalent". The NA includes a target link layer address option that contains an "ACK" indication and a "Maximum_MRU" option initialized to the MRU of the physical interface recorded above

4) The target receives the Solicited NA and notices that it includes a target link layer option containing an "ACK" indication. It places the "Maximum_MRU" value in the cache for the source and transitions from the "SYN-RECEIVED" to the "ESTABLISED" state. The target and source have now established a bi-directional link using the exact mechanism specified in [TCP, 3.4] and are said to be "affiliated"

4.2. Affiliation Maintenance

Neighbor affiliations are established on-demand in response to packet transmissions, as described above. Once established, the neighbors work together to proactively maintain the affiliation as long as it is being actively used by either or both endpoints. When the affiliation is no longer needed, it is allowed to expire with stale cache entries eventually garbage-collected.

Since the "Maximum_MRU" values exchanged in the affiliation establishment phase only convey information about the maximum packet size each node can receive from neighbors on the same physical link, a mechanism is needed to detect whether an MRU reduction is incurred by the intervening IPv4 path. To satisfy this requirement, each neighbor MUST monitor its IPv4 reassembly cache to detect packets being

fragmented by the network. The size of the largest fragments arriving from a particular neighbor indicates the "Current_MRU" that can be accepted, and this value needs to be conveyed back to the neighbor causing the fragmentation (see below).

Additionally, since the Neighbor Affiliation Protocol does not ensure reliable data delivery, no periodic acknowledgements are sent as in [TCP]. [DISC, 7.3] accepts hints from upper layer protocols of "forward progress" as reachability confirmation, but such indications may not be present when one (or both) of the neighbors are routers or when only "unidirectional" traffic (e.g., UDP-based continuous media streams) is present. Thus, the proactive maintenance process requires periodic transmission of "keepalive" messages.

Of paramount importance is the fact that data traffic arrival conveys unidirectional link state information only. In particular, the fact that node A is receiving packets from node B only guarantees that the unidirectional link B->A is operational (and vice-versa for A->B). In other words, it is insufficient for A to receive packets from B and for B to receive packets from A; in addition, A MUST KNOW THAT B IS RECEIVING ITS PACKETS AND VICE-VERSA.

In order to satisfy the above affiliation maintenance requirements, each node MUST implement the following keepalive algorithm:

- if one or more data packets were received from an affiliated neighbor within the past N seconds, send the neighbor an unsolicited Neighbor Advertisement containing a target link layer address option with a "Current_MRU" option set to the size of the largest fragments arriving from the neighbor. If no fragmentation is taking place, "Current_MRU" is set to "Maximum_MRU". (Note that "Maximum_MRU" may change over time, e.g., due to routing fluctuations, so it too must be included in the NA.)

- if no unsolicited NAs have arrived within the past N seconds even though packets have been sent to the neighbor, mark the affiliation as "stale".

4.3. Fulfilling Contractual Obligations

When a pair of neighbors engages in an affiliation as specified in the previous subsections, they effectively engage in a mutual contract that requires vigilance to maximize robustness and efficiency while doing no harm to each other or to the intervening network path. Mandatory contractual obligations include:

- the packets a neighbor sends to its peer MUST be no larger than the peer's "Current_MRU" estimate

- all packets SHOULD be sent with the IPv4 DF flag NOT set; regardless of their size

- each node SHOULD attempt to maximize network utilization by periodically increasing the estimate for its neighbor to "Maximum_MRU" as long fragmentation remains below "harmful" levels

- each neighbor MUST take preemptive measures to reduce fragmentation if it reaches "harmful" levels

Preemptive measures to reduce harmful fragmentation (see "What constitutes harmful fragmentation?" below) include sending unsolicited NAs and ICMPv6 "packet too big" messages as appropriate. When a node detects harmful fragmentation, it MUST send a gratuitous unsolicited NA to the offending peer (as described in the previous section) without waiting for the normal affiliation maintenance timeout period. Nodes should employ a strategy to rate-limit such gratuitous NAs, since a RTT "burst" of fragmented packets may be in the pipeline.

When fragments are lost such that a packet cannot be reassembled, the receiver MUST generate an ICMPv6 "packet too big" message, provided the first-fragment was not lost. The "packet too big" message encodes the length of the first-fragment in the MTU field and encapsulates as much of the IPv6 packet contained in the IPv4 first-fragment as possible [ICMPv6, 3.2]. This message provides the peer with a transmit failure indication so lost data can be retransmitted.

Note that [FRAG, 3.4] speaks favorably for the use of "transparent fragmentation", i.e., the use of reliable fragmentation and reassembly at a layer below IP. What is proposed here is essentially transparent link layer fragmentation with judicious and mitigated use so that network utilization is maximized while fragmentation is minimized. In this sense, fragmentation in and of itself is NOT cause for alarm and rash actions - as long as both ends of the neighbor affiliation honor their contractual obligations and adapt their behavior appropriately.

4.4. What Constitutes "Harmful" Fragmentation?

In 1987, [FRAG] made an early assertion that fragmentation was harmful, and in 2002 the quantitative studies in [FOLK] concluded that this assertion is still true today. Even in today's Internet (where nodes by-and-large have correct fragmentation/reassembly

implementations) fragmentation causes "slow-path" processing in routers and network performance degradation when the loss unit (a fragment) is smaller than the retransmission unit (a packet or segment). [FOLK] observed that the principal contributors to fragmentation in the Internet are continuous media applications (e.g., media players and interactive games) and tunnel ingress points with misconfigured MTUs. Both produce UNMITIGATED and PERSISTENT fragmentation when no path MTU discovery feedback occurs, and it is ONLY this sort of fragmentation that this author believes should be considered harmful.

Both [FRAG] and [FOLK] failed to note that the fundamental cause of unmitigated and persistent fragmentation are senders which disobey the robustness principle, i.e., nodes that are not "conservative in what they send". But, the contractual obligations of nodes that participate in the neighbor affiliation protocol specified in this document provide the necessary means for eliminating harmful fragmentation. While fragmentation is an integral mechanism for efficient path probing in the specification, it's use is an appropriate and mitigated application of the receiver's side of the robustness principle, i.e., be "liberal in what you receive".

4.5. Willingness to Affiliate

When a node initiates a neighbor affiliation as described in the previous subsections, it may find that its peer either does not support the protocol or is otherwise unwilling to affiliate. In the former case, the NA that a peer returns in response to the initial NS will either not contain the "SYN/ACK" indication or not contain a target link layer option at all. In this case, the initiating node may:

1) Safely estimate that the peer's MRU is the IPv6 MINMTU of 1280 bytes

2) Bravely estimate that the peer's MRU is something larger than IPv6 MINMTU, e.g., 1480 bytes)

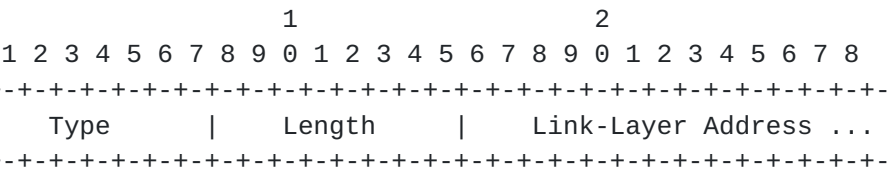
3) Assume that the peer is unreachable, e.g., if no reasonable MRU estimate is possible or if a security association is required

The MRU estimate MUST take into account that tunneled traffic is one of the primary contributers to harmful fragmentation in the Internet today [FOLK]. Nodes MUST honor the robustness principle of "be conservative in what you send and liberal in what you receive" in all cases.

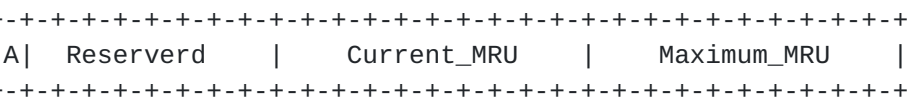
Nodes may employee a strategy for allowing/disallowing particular affiliations, e.g., a router or server may choose not to answer a solicitation from a new host if its state cache is nearly full. Finally, security measures should be taken to ensure that the affiliation protocol described above is not abused by malicious nodes. Candidate mechanisms might be an adaptation of TCP "syn cookies" [reference needed] or a shared secret between the neighbors.

4.6. Source/Target Link Layer Option Format

The NS/NA messages used for the neighbor affiliation protocol always encode the Source/Target Link Layer Address option, as specified by [DISC, 4.6.1]:



The "Type" is encoded exactly as in [DISC, 4.6.1] and the Length is always 1 for the purpose of this specification. But, the link-layer address field has the following special format:



Thus, the "SYN" and "ACK" flags, the "Current_MRU" and the "Maximum_MRU" values can be exchanged between affiliating neighbors, as required by the specification.

5. Operational Considerations

Nodes that establish neighbor affiliations on IPv6-over-(foo)-over-IPv4 links must observe the following operational considerations:

5.1. Default Maximum Transmit Unit (DFLT_MTU)

IPv6-over-(foo)-over-IPv4 interfaces may be configured over one or more underlying physical interfaces for IPv4. The default maximum transmit unit (DFLT_MTU) for an IPv6-over-(foo)-over-IPv4 interface is the maximum MTU of all underlying physical interfaces for IPv4 minus the sum of all header sizes required for IPv6-over-(foo)

encapsulation.

For example, if a multi-homed host has an Ethernet interface and an FDDI interface, the maximum MTU for underlying physical interfaces is 4352. If the encapsulation is IPv6-over-UDP, then:

DFLT_MTU = 4352 - 40 (sizeof(ipv6_hdr)) - 8 (sizeof(udp_hdr)) = 4304

Obviously, the DFLT_MTU value may be overestimated for some traffic on multi-homed hosts with heterogeneous interfaces. But, read further.

5.2. Per-Neighbor Maximum Transmit Unit (NBR_MTU)

As neighbor affiliations are established, the MRUs for "frequently accessed" neighbors are established in the destination cache. The destination cache entry for an active neighbor's MRU is always chosen as the MTU for that neighbor (NBR_MTU). When no destination cache entry exists, the DFLT_MTU is used.

5.3. TCP MSS

When a TCP connection is initiated to an on-link neighbor, the TCP MSS is initialized to (NBR_MTU - 40) if a destination cache entry exists; else (DFLT_MTU - 40). (40 = 20bytes for TCP header plus 20bytes for IPv4 header). The TCP SYN segment carries the MSS option and initiates the neighbor affiliation process if no affiliation currently exists (i.e., the TCP SYN segment is contained in the first packet out.) The neighbor affiliation may reduce the NBR_MTU value in the destination cache while the SYN packet waits on the Address Resolution queue. But, the system will self-correct when the peer responds with an MSS option in the SYN/ACK, since the peer will have up-to-date MRU information from the neighbor affiliation.

5.4. Adaptation to Overestimated DFLT_MTU, NBR_MTU

When a packet is sent to a neighbor based on an overestimated DFLT_MTU or NBR_MTU, an ICMPv6 "packet too big" message is generated locally and the too-big packet is dropped. Upper layers will retransmit the data based on the reduced MTU specified in the packet too big message.

5.5. Implementation Alternatives

Obviously, the cleanest implementation would entail in-kernel instrumentation of the IPv4 reassembly cache and intervention in the specific IPv6-over-(foo)-over-IPv4 device drivers that will use neighbor affiliation. But, an alternative is to write an application that uses the Berkeley Packet Filter (libpcap) to monitor the fragments that enter the host and generate the NS/NA messages necessary to support the protocol outside of the context of the kernel. In this way, the neighbor affiliation protocol can be easily deployed on nodes with existing "vanilla" IPv6-over-(foo)-over-IPv4 tunnel drivers.

6. Rationale for this Approach

One might reasonably ask why the approach in this document is recommended instead of the current practices specified in [PMTUDv4],[PMTUDv6]. When IPv6 uses (foo)-over-IPv4 as a link layer, ICMPv6 "Packet Too Big" messages can only be produced by the tunnel encapsulator and decapsulator (i.e., the two IPv6 neighbor nodes), while ICMPv4 "Fragmentation Needed" messages are produced by the intervening IPv4 routers. But, the ICMPv4 messages are not readily translated into ICMPv6 since they are only guaranteed to include up to 8 bytes of the too big packet's data (i.e., not enough information to determine the IPv6 header).

One possible solution is to cache the IPv6 packets at the encapsulator and match them up with any ICMPv4 "frag needed" messages that are delivered by the IPv4 network. But, this creates a state scaling issue - especially when the encapsulator is an IPv6 router. Also, an encapsulating router would need to retransmit the data in the too-big packets on behalf of the final destination, i.e., act as a "proxy" for the final destination - but, this would require the router to understand the semantics of the packetization layer of the original source when it receives ICMPv4 "frag needed" messages from the IPv4 network.

Finally (and most importantly) IPv4 routers in the path between the encapsulator and decapsulator cannot be trusted to reliably deliver ICMPv4 "frag needed" messages - they can be lost due to network congestion or filtering firewalls, and they can be forged by an attacker since the end nodes have no trust basis with the IPv4 routers. The only acceptable means is to engage the IPv6 endpoints in a neighbor affiliation as described in this document.

7. IANA considerations

N/A

8. Security considerations

This document provides a potential platform for integrating secure neighbor discovery mechanisms.

Acknowledgements

The proposal herin is nearly identical to some presented on the TCP-IP discussion group [TCP-IP] and IETF Path MTU Discovery Working Group [MTUDWG] mailing lists, roughly between the period of May 1997 through May 1990. The earliest proposal that most closely matches the one herein was offered by Charles Lynn on November 17, 1987. Others (e.g., Fox, Bohle, 1989, etc.) proposed combining the basic mechanism described by Lynn with transport layer protocols, e.g., TCP. To the best of the author's knowledge, this document presents the first suggested combination of the Lynn proposal with Neighbor Discovery.

Earlier works from SRI International proposed a "router affiliation" protocol. The term "affiliation" as used in this document was directly derived from its use in those earlier works. Two SRI researchers who participated in this effort were Barbara Denny and Bob Gilligan.

The author acknowledges those who participated in discussions on the NGTRANS and V6OPS mailing lists between August and October 2002 for helpful insights.

The author would finally like to acknowledge the founding architects of the DARPA Internet protocols, who created the technologies used by the millions of nodes on the Internet today and the billions more to come in the forseeable future.

Normative References

[ICMPv6] Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 2463.

[IPv4] Postel, J., "Internet Protocol", RFC 791.

[IPV6] Deering, S., and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460.

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

[FRAG] Kent, C., and J. Mogul, "Fragmentation Considered Harmful", December, 1987

[FOLK] Shannon, C., Moore, D. and k claffy, "Beyond Folklore: Observations on Fragmented Traffic"

[PROBE] Mogul, J., Kent, C., Partridge, C., and K. McCloughrie, "IP MTU Discovery Options", RFC 1063, July 1988.

[PMTUDv4] Mogul, J. and S. Deering, "Path MTU Discovery", RFC 1191, November 1990.

[PMTUDv6] McCann, J., Deering, S. and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[MTUDWG] IETF MTU Discovery Working group mailing list, gatekeeper.dec.com/pub/DEC/WRL/mogul/mtudwg-log, November 1989 - February 1995.

[TCP] Postel, J., "Transmission Control Protocol", RFC 793.

[TCP-IP] TCP-IP Mailing list archives, http://www-mice.cs.ucl.ac.uk/multimedia/misc/tcp_ip, May 1987 - May 1990.

Informative References

Authors Addresses

Fred L. Templin
Nokia
313 Fairchild Drive
Mountain View, CA, USA
Phone: (650)-625-2331
Email: ftemplin@iprg.nokia.com

APPENDIX A: Historic Evolution of PMTUD

The topic of Path MTU discovery (PMTUD) saw a flurry of discussion and numerous proposals in the late 1980's through early 1990. The initial problem was posed by Art Berggreen on May 22, 1987 in a message to the TCP-IP discussion group [TCP-IP]. The discussion that followed provided significant reference material for [FRAG]. An IETF Path MTU Discovery Working Group [MTUDWG] was formed in late 1989

with charter to produce an RFC. Several variations on a very few basic proposals were entertained, including:

1. Routers record the PMTUD estimate in ICMP-like path probe messages (proposed in [FRAG] and later [PROBE])

2. The destination reports any fragmentation that occurs for packets received with the "RF" (Report Fragmentation) bit set (Steve Deering's 1989 adaptation of Charles Lynn's Nov. 1987 proposal)

3. A hybrid combination of 1) and Charles Lynn's Nov. 1987 proposal (straw RFC draft by McCloughrie, Fox and Mogul on Jan 12, 1990)

4. Combination of the Lynn proposal with TCP (Fred Bohle, Jan 30, 1990)

5. Fragmentation avoidance by setting "IP_DF" flag on all packets and retransmitting if ICMPv4 "fragmentation needed" messages occur (Geof Cooper's 1987 proposal; later adapted into [PMTUDv4] by Mogul and Deering)

Option 1) seemed attractive to the group at the time, since it was believed that routers would migrate more quickly than hosts. Option 2) was a strong contender, but repeated attempts to secure an "RF" bit in the IPv4 header from the IESG failed and the proponents became discouraged. 3) was abandoned because it was perceived as too complicated, and 4) never received any apparent serious consideration. Proposal 5) was a late entry into the discussion from Steve Deering on Feb. 24th, 1990. The discussion group soon thereafter seemingly lost track of all other proposals and adopted 5), which eventually evolved into [PMTUDv4] and later [PMTUDv6].

In retrospect, the "RF" bit postulated in 2) is not needed if a "contract" is first established between the peers, as in proposal 4) and a message to the MTUDWG mailing list from jrd@PTT.LCS.MIT.EDU on Feb 19. 1990. These proposals saw little discussion or rebuttal, and were dismissed based on the following the assertions:

- routers upgrade their software faster than hosts

- PCs could not reassemble fragmented packets

- Proteon and Wellfleet routers did not reproduce the "RF" bit properly in fragmented packets

- Ethernet-FDDI bridges would need to perform fragmentation

(i.e., "translucent" not "transparent" bridging)

- the 16-bit IP_ID field could wrap around and disrupt reassembly at high packet arrival rates

The first four assertions, although perhaps valid at the time, have been overcome by historical events leaving only the final to consider. But, [FOLK] has shown that IP_ID wraparound simply does not occur within several orders of magnitude the reassembly timeout window on high-bandwidth networks.