Fast Address Configuration Strategies for the Next-Generation Internet

Nick Moore
nick.moore@monash.edu

Greg Daley
greg.daley@monash.edu

Center for Telecommunications and Information Engineering,
Monash University, Melbourne, Australia

August 7, 2003

Abstract

Fast IPv6 Address Configuration is pivotal to the unification of mobile Internet and mobile telephony services. We describe the standard method for IPv6 address configuration and examine its drawbacks, identify the current alternatives and their flaws and suggest two new methods of Duplicate Address Detection which perform Address Configuration more quickly.

1 Introduction

The recent trend towards ubiquitous, mobile Internet access encourages unification of packet and telephony networks to minimize duplication of infrastructure and increase resource utilization. In such a future unified network, voice calls would be carried over IPv6 flows, with Mobile IPv6[9] (MIPv6) providing handovers to support mobility. At present, however, MIPv6 does not provide sufficient performance to allow unobtrusive handovers of voice calls from one cell to another.

MIPv6 builds upon the IPv6 standards to allow hosts to maintain a *Home Address* through which they are always reachable, while using a changable *Care-of Address* to connect to the Internet. In doing so it uses the router discovery and autoconfiguration mechanisms available in IPv6[15, 16]. These mechanisms were designed for initialization, configuration and centralized management of fixed Internet hosts, and do not provide sufficient performance for use in applications where short handover times are necessary.

We have identified several separate delays during handovers in MIPv6[12], the largest of which is the 1000ms delay required to configure a new address. The delay is caused by the time taken to perform Duplicate Address Detection (DAD).

This paper describes the process of DAD, compares the features of some existing methods of reducing this problem and presents two novel methods we have developed to eliminate DAD delay.

1.1 Duplicate Address Detection

IPv6 nodes can statelessly autoconfigure their own addresses on a network, based on information sent by the IPv6 router of that network[16]. When a node wishes to create a new address on an interface, it combines the network prefix obtained from the router with a suffix generated from its 64-bit Interface Identifier. The Interface Identifier can be either obtained from the interface hardware[8, 15] (eg: a MAC address) or generated randomly[14]. This new, untested address is referred to as a 'Tentative Address' (TA). The node joins the appropriate solicited-node multicast group for this address, then sends a Neighbour Solicitation (NS) to the TA. If the TA is already in use by another node, that node will reply with a Neighbour Advertisement (NA) defending the TA.

Once it has sent the NS, the node waits for *RetransTimer* milliseconds to see if a defending NA is forthcoming, and this solicit-and-wait process is repeated *DupAddrDetectTransmits* times. The default

value of *RetransTimer* is 1000ms and by default the process is only done once, resulting in a delay of 1000ms. During this time, the node cannot correspond on the address other than to send NSs and listen for NAs. A router can request a smaller *RetransTimer* on its network, however the 1000ms default is still used for DAD on the Link Local address. If the node does receive a defending NA while the address is tentative, it deconfigures the address and does not attempt to use it again.

This procedure provides a reasonable approach to checking address uniqueness in situations (such as fixed installations, or even mobile web-browsing) where 1000ms is not a significant delay. Obviously, the default *RetransTimer* could be 'tuned' down to a more acceptable value, but the smaller the value the more risk of a defending NA arriving too late.

1.2 Why DAD?

There are serious consequences if two nodes do configure the same address. The Neighbour Discovery[15] mechanisms of IPv6 are quite robust in that a collision will not cause the network to become unstable. However, artifically induced collisions on our testbed network indicate that, if the collision is undetected, the colliding nodes will 'fight' for the address. Both nodes will respond to Neighbour Solicitations, and corresponding nodes must choose arbitrarily between the responses depending on their order of arrival.

Not only will packets not be delivered to the correct node, but the colliding node may send negative acknowledgements such as TCP resets or ICMP 'Destination Unreachable' messages, causing existing connections to be terminated. As neither node will automatically deconfigure the address, this could cause extended and difficult to analyse problems, which would not be easily resolvable beyond the conflicting nodes' local network.

1.3 The problem with DAD

DAD uses a 'Stateless' strategy which attempts to detect duplicate addresses without relying on a centralized server or router to keep track of the state of the network. Instead, DAD relies on the already configured nodes to cooperate in the DAD process.

The advantages to this approach are that there is no central server to be configured, and no centralized state to be lost, and a router reset will therefore not invalidate all addresses on the network.

The disadvantage is that these strategies depend on nodes 'defending' their addresses, and there is no positive acknowledgement to say '*yes, you can have that address*'. This means that these strategies are susceptible to packet loss, which may cause a solicitation or a defence to be lost and thus a collision to go undetected. Also a configuring node must wait for a set time to allow negative messages to be received rather than receiving a positive message and continuing immediately.

1.4 No DAD?

As the problems of DAD delay have become apparent, there have been suggestions to skip DAD altogether[11], or to perform unmodified DAD "asynchronously"[10]. This is not entirely unreasonable, especially for well-distributed addresses such as Cryptographically Generated Addresses[1], or addresses generated from well-distributed random suffixes.

However, as discussed above, an undetected address collision will cause significant disruption – it would be preferable to find a solution which maintained the safety of DAD but eliminated the delay.

2 Existing Alternatives

The following sections outline methods which have been suggested as possible improvements to DAD, and identify possible problems with their design.

These methods are all 'stateful', depending on state held on a router or server to provide positive acknowledgements that an address is available. Therefore, they introduce additional dependence on infrastructure, and in some cases this may impact upon scalability or reliability.

2.1 DHCPv6

Dynamic Host Configuration Protocol for IPv6 (DHCPv6)[6] would seem to solve all the problems of

DAD by offering a stateful, server-based method of negotiating addresses.

However, communication with a DHCPv6 server requires that the Mobile Node has already configured a Link-Local address, which as discussed above requires the use of standard DAD and thus introduces the *RetransTimer* 1000ms delay. Also, standards require DHCP-assigned addresses to be double-checked with DAD[16], as the DAD server will be unaware of nodes which have configured statelessly. This potentially *increases* DAD latency, as first the Link-Local, and then the Global addresses must clear DAD, rather than being able to do both in parallel. As a stateful method, DHCPv6 is dependant on server-stored state, reducing the reliablility of the network to that of the DHCP server.

For these reasons, we believe that DHCPv6, while an appropriate method for providing host configuration information, is not appropriate as a replacement for DAD.

2.2 A-DAD

A new method for supplying addresses quickly to roving mobile nodes, called Advance Duplicate Address Detection (A-DAD)[7], has been proposed. An A-DAD capable router supplies addresses to arriving Mobile Nodes from a pool of addresses which are known to be unique on the link. A host can safely configure this address without performing DAD, as the router has ensured it is unique.

In order to provide addresses for this pool, the router must create addresses based on random suffixes as per RFC 3041[14] and undertake standard DAD on them. This means that the router must configure sufficient addresses *in advance* to ensure that demands are met.

In addition, the configuring node has no choice as to what address is provided. Since Secure Neighbour Discovery (SEND) is likely to rely upon cryptographically generated addresses[1], nodes which are depend on A-DAD rather than generating addresses from their own private keys will be excluded from Secure Neighbor Discovery.

3 New Alternatives

The following sections describe two novel techniques which we have developed. We believe they have potential as compatible modifications to DAD, as both methods reuse standardized signalling, and thus provide interoperability with unmodified, ‘slow’ nodes.

3.1 MLD-DAD

When performing standard DAD, nodes first begin listening to the solicited nodes multicast group appropriate to that address. This allows the nodes to hear advertisements of the address, and thus detect address collisions. Therefore, a system that keeps track of which solicited nodes’ addresses are in use also knows which unicast addresses are available.

In order to join this group, the node is required[5] to send a Multicast Listener Discovery (MLD) Report[4].

We propose that to avoid DAD delay, the router(s) should monitor MLD Reports for the solicited nodes addresses to determine which addresses are being used[3]. As periodic updates are required by the MLD protocol, the router’s state will be kept up-to-date, and can be easily reconstructed if the router is reset. If the router has seen no evidence of the address being used, it can inform the node that the address may be allocated without further delay.

MLD-DAD may only be used on networks where all devices on the network perform MLD properly, as otherwise the router can not be sure of the availability of an address. However, of the systems we tested, only one correctly performed MLD before sending DAD messages (WinXP), others undertook MLD incorrectly or late (Solaris 8, Kame, Usagi) and one didn’t send the required MLD reports at all (Linux-2.4).

On networks where all nodes are known to meet IPv6 Node Requirements, MLD-DAD provides a fast and efficient alternative to DAD.

3.2 Optimistic DAD

The standard DAD strategy can be described as *pessimistic*, since it delays all communications until it is confident that the new address is not a duplicate.

However, if the address is chosen carefully (eg: using CGA or a well-distributed RNG), the odds of a collision are vanishingly small[2]. This suggests that an *optimistic* strategy, where the node assumes that DAD will succeed, would be preferable.

We have developed "Optimistic DAD"[13] as one possible implementation of this approach. It bends the rules of RFC2461/2 to allow communication to be established over a tentative address, while attempting to minimize disruption in the case of collision. We have designed these changes to maintain interoperability with unmodified nodes.

For a node to send or receive packets, it must participate in Neighbour Discovery (ND). However, Neighbour Solicitations (NS), Neighbour Advertisements (NA) or Router Solicitations (RS) sent from a Tentative address risk adverse effects to an exisiting node in the case of an address collision.

To avoid this problem, Optimistic DAD exploits existing flags and options in the ND messages. NAs are sent with the 'Override' bit cleared and NSs and RSs are sent without Source Link-Layer Address Options. The Optimistic node modifies its ND behaviours to work around these restrictions, for example by sending packets for unknown neighbours via the router. The restrictions prevent a Tentative address overriding existing Neighbour Cache entries in the case of a collision, although it does make the ND process less efficient while the address is Tentative. Once the DAD timeout has expired, the address is no longer Tentative, and standard ND behaviour applies.

In the case of an address collision, the Optimistic node is unlikely to be able to properly communicate, since its neighbours will not allow it to complete Neighbour Discovery. As soon as the defending NA is received it will reconfigure a new address in any case. There is still a possibility that the collision will cause connection loss, but the situation will be rapidly resolved, as opposed to the unresolvable problems cause by a collision without DAD.

Optimistic DAD is most suitable for networks on which the transmission of a few extra messages per configuring node is not a significant issue. In addition, because the penalty associated with an address being Tentative is greatly reduced, a node may elect to probe more than once for a duplicate address, greatly decreasing the chance of packet loss causing a collision to go undetected. This makes Optimistic DAD particularly suitable for use on Wireless LAN type networks where packet loss is common[17].

4 Conclusions

Address Configuration on IPv6 networks requires Duplicate Address Detection because of the severe and ongoing consequences of address collision. Since the delays associated with the currently standardized DAD procedures are prohibitive in mobile environments, a way must be found to perform DAD more quickly.

In this paper we have examined existing methods for reducing DAD delay, and discussed their strengths and weaknesses. In addition, we have presented two novel DAD mechanisms which provide interoperable enhancements to DAD, and which we will continue to develop.

Hopefully, our analysis and suggestions will open up a new direction in the development of methods for Fast Address Configuration on IPv6.

Acknowledgements

- Thanks to Ahmet Şekercioğlu for his help with this paper.

- This work was supported by the ATCRC Next Generation Internet project. http://www.telecommunications.crc.org.au/

References

[1] Tuomas Aura. Cryptographically generated addresses (CGA) *Internet Draft – work in progress*. URL reference: http://www.watersprings.org/pub/id/draft-ietf-send-cga-00.txt, 2003.

[2] M. Bagnulo, I. Soto, A. Garcia-Martinez, and A. Azcorra. Random Generation of Interface Identifiers *Internet Draft – work in progress* . URL reference: http://www.watersprings.org/pub/id/draft-soto-mobileip-random-iids-00.txt, 2002.

[3] Greg Daley and Richard Nelson. Duplicate address detection optimization using IPv6 multicast listener discovery *Internet Draft – work in progress*. URL reference: http://www.watersprings.org/pub/id/draft-daley-ipv6-mcast-dad-02.txt, 2003.

[4] S. Deering, B. Fenner, and B. Haberman. RFC 2710 Multicast Listener Discovery (MLD) for IPv6, 1999. URL reference: http://www.ietf.org/rfc/rfc2710.txt.

[5] John Loughney (Ed.). IPv6 node requirements. URL reference: http://www.ietf.org/internet-drafts/draft-ietf-ipv6-node-requirements-04.txt, 2003.

[6] R. Droms et al. Dynamic host configuration protocol for IPv6 (DHCPv6) *Internet Draft – work in progress*. URL reference: http://www.watersprings.org/pub/id/draft-ietf-dhc-dhcpv6-28.txt, 2002.

[7] Y. Han et al. Advance Duplicate Address Detection *Internet Draft – work in progress* . URL reference: http://www.watersprings.org/pub/id/draft-han-mobileip-adad-01.txt, 2003.

[8] B. Hinden and S. Deering. RFC 2373 IP Version 6 Addressing Architecture, 1998. URL reference: http://www.ietf.org/rfc/rfc2373.txt.

[9] David B. Johnson, Charles Perkins, and Jari Arkko. Mobility support in IPv6 *Internet Draft - Work in progress*. URL reference: http://www.watersprings.org/pub/id/draft-ietf-mobileip-ipv6-20.txt, June 2002.

[10] David B. Johnson, Charles Perkins, and Jari Arkko. Mobility support in IPv6 *Internet Draft - Work in progress*. URL reference: http://www.watersprings.org/pub/id/draft-ietf-mobileip-ipv6-18.txt, June 2002.

[11] N. Montavont and Thomas Noël. Handover management for mobile nodes in IPv6 networks. *IEEE Communications Magazine*, pages 38–43, August 2002.

[12] Nick Moore. Non-Predictive Handovers *Presentation at IETF 56*. URL reference: http://www.ietf.org/proceedings/03mar/slides/mobileip-8.pdf, 2003.

[13] Nick Moore. Optimistic duplicate address detection *Internet Draft – work in progress*. URL reference: http://www.watersprings.org/pub/id/draft-moore-ipv6-optimistic-dad-02.txt, 2003.

[14] T. Narten and R. Draves. RFC 3041 Privacy Extensions for Stateless Address Autoconfiguration in IPv6, 2001. URL reference: http://www.ietf.org/rfc/rfc3041.txt.

[15] T. Narten, E. Nordmark, and W. Simpson. RFC 2461 Neigbhour Discovery for IP Version 6 (IPv6), 1998. URL reference: http://www.ietf.org/rfc/rfc2461.txt.

[16] S. Thomson and T. Narten. RFC 2462 IPv6 Stateless Address Autoconfiguration, 1998. URL reference: http://www.ietf.org/rfc/rfc2462.txt.

[17] A. Willig, M. Kubisch, and A. Wolisz. Measurements and stochastic modeling of a wireless link in an industrial environment. Tech Rpt TKN-01-001, Telecommunication Networks Group, Technical University Berlin, 2001. URL reference: http://citeseer.nj.nec.com/willig01measurements.html.