PREFIX CONTINUITY AND GLOBAL ADDRESS AUTOCONFIGURATION IN IPV6 AD HOC NETWORKS

*Short version**

Christophe Jelger†
Computer Networks Research Group - University of Basel
Bernoullistrasse 16, CH-4056 Basel, Switzerland
Christophe.Jelger@unibas.ch

Thomas Noël
Louis Pasteur University (Strasbourg) - LSIIT - UMR 7005 CNRS-ULP
Boulevard Sebastien Brant, 67400 Illkirch, France
noel@dpt-info.u-strasbg.fr

Abstract

Ad hoc networks are formed by the spontaneous collaboration of wireless nodes when no networking infrastructure is available. When communication to the Internet is desired, one or more nodes must act as gateways for the ad hoc network. In this case, global addressing of ad hoc nodes is required. This article presents a protocol which can be used by an ad hoc node to dynamically select a gateway and create an associated IPv6 global address. The core of our proposal is the concept of *prefix continuity*. By building and maintaining a forest of logical spanning trees, our proposal ensures that there exists, between a node A and its gateway G, a path of nodes such that each node on this path uses the same prefix P as the node A and its gateway G. This concept results in an organized ad hoc network, in the sense that sub-networks (with respect to prefixes) are automatically created and dynamically maintained when multiple gateways are available. Moreover, the concept of prefix continuity ensures that each sub-network forms a connected graph of nodes which all use an identical prefix. In contrast to traditional wired networks, this feature is not trivial in ad hoc networks.

* The initial (long) version of this paper, as originally submitted and accepted for the MedHocNet'05 workshop, is available at http://cn.cs.unibas.ch/pub/

† Part of this work was carried out during the tenure of an ERCIM fellowship (see http://www.ercim.org for details).

1. Introduction

In contrast to current wireless networks, ad hoc networks require no preexisting infrastructure to exist. Because of the inherent limited propagation range of radio transmissions, ad hoc nodes must collaborate to forward (and route) packets within such a spontaneous multi-hop network. Moreover, nodes have to face unpredictable topological changes, which makes routing a challenging task in an ad hoc network. A growing issue with ad hoc networking is Internet connectivity. There is indeed an increasing deployment of community-based *mesh networks* [Draves et al., 2004] [V. Bahl (organizer), 2004], which currently rely on protocols developed for ad hoc networks. For such *species* of ad hoc networks, to be connected to the Internet is of major importance in order to offer Internet services (e.g. email, web access, etc) to their users. Being connected to the Internet requires that each node in the network must have a topologically correct global address in order to be natively reachable from outside the ad hoc network (i.e. without any network address translation mechanism).

In this paper, we present a protocol that can be used to configure the nodes of an IPv6 ad hoc network with a globally routeable address. Our proposal builds a forest of logical spanning trees, where each tree if formed by nodes that share a common global network prefix. As in classical IPv6 wired networks, gateways are responsible for prefix announcement. An inherent feature of our proposal is *prefix continuity*: it ensures that there exists, between a node A and its gateway G, a path of nodes such that each node on this path uses the same prefix P as the node A and its gateway G. When multiple (different) prefixes are available, this concept results in an organized ad hoc network, in the sense that sub-networks (with respect to prefixes) are automatically created and dynamically maintained when multiple gateways are available. As a result, each subnetwork is a connected graph of nodes which all use an identical prefix. In contrast to previous work, prefix continuity is the core element of our proposal.

Following this introduction, we present in Section 2 some of the related proposals. In Section 3 we present our approach and also introduce the concept of prefix continuity in an ad hoc network. We then describe in Section 4 the three algorithms used by an ad hoc node to choose its gateway and its associated prefix. We finally conclude the paper.

2. Related Work

The particular nature of ad hoc networks makes it impossible to use the IPv6 mechanisms used in wired networks in order to propagate prefix information, mainly because they have been designed to work on a shared broadcast link. To overcome this situation, Weniger *et al.* [Weniger and Zitterbart, 2002]

[Weniger, 2005] have proposed to modify the stateless address autoconfiguration (SAA [Thomson and Narten, 1998]) mechanism used in IPv6 networks, and the duplicate address detection (DAD) procedure of the SAA protocol. The interesting point of these proposals is that they try to re-use the protocols designed for classical IPv6 networks. However, the SAA and DAD protocols are inherently mal-adapted to ad hoc multi-hop networks, mainly because their efficiency and simplicity are based on the fact that they have been designed for networks that have a unique layer-3 link. For ad hoc networks, we believe that the use of such techniques should be avoided.

Wakikawa *et al.* [Wakikawa et al., 2002] have proposed a reactive method that can be used with any kind of routing protocols. With their proposal, an ad hoc node broadcasts a request to obtain a prefix with global scope. This request propagates within the entire ad hoc network and eventually reaches a gateway. The gateway replies to the originator of the request with a message which contains the prefix. The node receiving this information creates a global address and adds a particular entry in its routing table. Xi *et al.* [Xi and Bettstetter, 2002] extended this model with proactive features (i.e. the periodical transmission of prefix information), and with the possibility for an intermediate node to respond to request messages. These two papers also consider the use of Mobile IPv6 [Johnson et al., 2004] to maintain connections at the transport layer. However, these existing proposals do not consider the unpredictable topological changes that occur in an ad hoc network, in the sense that they do not specify how the prefix information is updated (or changed) over time, a crucial consideration with ad hoc networks.

Our work differs from previous work as follows. First, we define *prefix continuity* as the core element of our proposal. For various reasons detailed later, this feature is highly relevant for the management and daily operation of ad hoc networks. Prefix continuity also prevents node isolation (i.e. after a network partition a node cannot reach its gateway) and avoids the use of an IPv6 routing header (as in other proposals). Second and in contrast with some previous work, our method supports multiple gateways which may announce different global prefixes. And third, this work aims to propose a method that is independent of the underlying routing protocol, as our proposed method can be used with both proactive and reactive routing protocols.

3. Protocol Operation and Prefix Continuity

This section introduces the concept of prefix continuity, and the hop-by-hop propagation technique used to disseminate the control messages that contain gateway and prefix information. We also give some implementation details related to the operation of IPv6.

3.1 Prefix continuity

The core feature of our proposal is what we have already defined as *prefix continuity*. Our proposal ensures that any node A that selected a given prefix P has at least one neighbor with prefix P on its path to the selected gateway G. Recursively, this feature thus ensures that there exists between A and G, a path of nodes such that all the nodes on this path use the same prefix P and gateway G as the node A. Prefix continuity is a inherent consequence of the propagation technique presented in Section 3.2. This technique leads to the creation of a forest of logical spanning trees which are dynamically maintained and updated when unpredictable topological changes occur. Each logical tree is rooted at a gateway, and it is formed by nodes which all use the global network prefix advertised by the gateway. Note that we use the term *logical* tree since the real physical topology of a sub-network is not necessarily a tree: the tree is only used to propagate the prefix information. Note that routing among two nodes of the ad hoc network is still done via a shortest path derived by the routing protocol used in the MANET. Figure 1 shows an ad hoc network with (a) and without (b) prefix continuity. There are 3 gateways, and each color corresponds to a given network prefix. Arrows indicate the orientation of the trees.

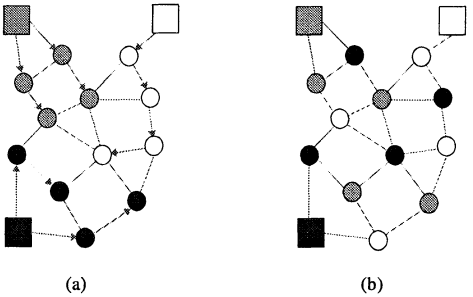

Figure 1. Ad hoc network with (a) and without (b) prefix continuity

A first advantage of prefix continuity is that a node does not need to add an IPv6 routing header in the packets it sends to correspondants located outside the ad hoc network (as in [Wakikawa et al., 2002]). This is because the default route of a node points to its parent in the tree, which necessarily uses the same gateway (recursively, the packet will reach the gateway via nodes that share the same network prefix). Without prefix continuity, a node must indeed specify via which gateway its packets must go through in order to avoid ingress filtering. Our proposal is also natively robust to network partitionning and it does not require any special mechanism in order to handle such situations. If a network partition occurs and if a node becomes isolated from its current gateway, it will quickly receive control messages from a new gateway and will eventually acquire a new global address.

Another advantage of prefix continuity is that it establishes a logical organization within an ad hoc network, i.e. the network becomes divided in sub-networks, each being formed by a contiguous gathering of nodes using the same prefix. This is particularily attractive for network providers that want to deploy specific applications that are meaningless in the absence of sub-networks (e.g. supervision/management systems, billing/accounting, on-demand/pay-per-view multicast streaming).

3.2 Forwarding/propagation of prefix information

Our proposal relies on a periodical hop-by-hop exchange of information between each node and its directly connected neighbors. Each gateway is responsible for sending periodical information in order to notify nodes in the ad hoc network about its existence and the prefix it uses. Such messages are denoted GW_INFO (GateWay INFOrmation), and each message contains:

- the distance (in hops) at which the sender is from the gateway
- the global address of the gateway and the length (in bits) of the prefix part of this address
- a sequence number used to disregard outdated information
- an optional DNS server address

This information subsequently propagates in a hop-by-hop manner. Depending on the network topology and on the number of gateways, each node may receive multiple GW_INFO messages. In short, each intermediate node selects the most appropriate information from one of its neighbors which becomes what we define as its *upstream neighbor*. The algorithms used to select the upstream neighbor are detailed in sections 4. The physical interface from which GW_INFO messages sent by the upstream neighbor are received is called the *upstream interface*.

A GW_INFO message must always be sent with a hop limit of 1. Therefore the initial GW_INFO message sent by a gateway is only received by its directly connected neighbors. Also the initial *distance to the gateway* information sent by a gateway must be zero. When a node has selected its upstream neighbor, it immediately forwards an updated version of the GW_INFO message sent by its upstream neighbor (the information sent by other neighbors than the upstream neighbor is not propagated). The updated message must also be sent with a hop limit of 1. The *distance to the gateway* must be increased by one. All other fields of the forwarded message remain unchanged. The prefix information contained in an initial GW_INFO message (sent by a gateway) is therefore propagated in a hop-by-hop manner among a subset of nodes of the ad hoc network which have decided to use this prefix and gateway. This method of propagation naturally leads to prefix continuity, and to the creation of a logical tree for each prefix. In a topology where multiple gateways and prefixes are present, our proposal leads to the creation of a forest of logical trees.

The propagation technique is illustrated by Fig. 2. There are two gateways G1 and G2 with the respective prefixes P1 and P2. Arrows emanating from a node indicate a GW_INFO message, the number represents the value of the *distance to the gateway* information and the color indicates the carried prefix. For clarity, many GW_INFO messages are not represented. It can be seen on the figure that each gateway announces a distance equal to zero. Nodes A and C therefore select the gateway G1 as their upstream neighbor. They in turn send their GW_INFO messages with a distance field set to one. Note that, for example, node A also receives the GW_INFO messages sent by node C but does not use them as the distance field is greater than the one it uses (i.e. from G1). Node D is the only node at equal distance of both gateways, it therefore has to arbitrarily select one of the two prefixes. In this example, because it selected C as its upstream neighbor, node D will not forward the GW_INFO messages sent by nodes E and F.

3.3 Integration with routing protocol

In this paper we present our proposal as a stand-alone mechanism which can be used in parallel with any ad hoc routing protocol. It is however important to node that our proposal can be integrated in the operation of the routing protocol

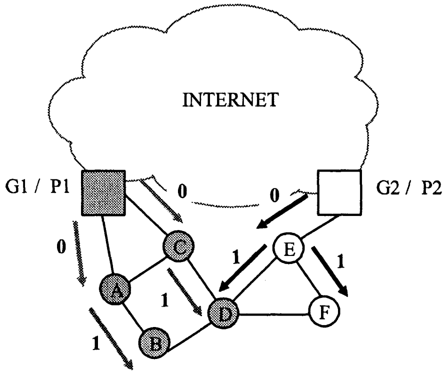

Figure 2. Hop-by-Hop propagation of GW_INFO messages

used in the ad hoc network. For example, if our proposal is integrated as part of the operation of a proactive routing protocol, it can benefit from the mechanisms used by such a routing protocol to maintain a view of its neighborhood (i.e. exchange of HELLO messages). Moreover, GW_INFO messages can be combined with messages sent by the routing protocol (e.g. in messages used to disseminate topological information). With reactive routing protocols, our proposal can simply be added to the normal operation of the routing protocol. We have for example integrated our proposal in an IPv6 version of the OLSR routing protocol. This version is currently available for the Linux operating system (see http://www-r2.u-strasbg.fr/~frey/safari/autoconf.html).

3.4 Implementation issues and DAD procedure

In this section we give some details about implementation issues since we have implemented our address autoconfiguration mechanism on the FreeBSD and Linux operating systems. First, GW_INFO messages are sent with a link-local source address and the destination address is ff02::1 (all nodes). The fields *prefix length* and *gateway address* are used to derive the gateway's prefix. The prefered prefix length is 64 bits. Finally, *sequence numbers* are used to avoid the propagation of outdated messages and to detect the loss of messages.

Once it has selected its upstream neighbor, a node generates its IPv6 global address with the prefix and prefix length contained in the GW_INFO message sent by its upstream neighbor. With proactive routing protocols, the node also creates a default routing table entry with its upstream neighbor as next hop. Note that the default route entry does not prevent direct routing between ad hoc nodes, as there should be a host entry (i.e. /128) in the routing table for each ad hoc node, even for nodes that use a different prefix. With reactive routing protocols, the GW_INFO information is not used to add a default route towards the gateway. We indeed believe that the reactive nature of such protocols avoids the need of keeping a default route which, by nature, prevents such a protocol from being reactive (the gateway of a node can reply to route requests for destinations outside the ad hoc network).

As stated earlier, the prefered prefix length should be 64 bits. If the prefix length advertised by a gateway is shorter than 64 bits, it must be padded with zeros until it reaches a length of 64 bits. To create an IPv6 global address with SAA, a node normally appends the EUI-64 (Ethernet Unique Identifier) of the upstream interface to the prefix sent by the upstream neighbor. This normally relies on the verification, via a duplicate address detection (DAD) procedure, of the uniqueness of the EUI-64. As mentioned earlier, traditional DAD cannot be used in ad hoc networks, and therefore either a specific DAD mechanism must be used, or the probability of an address collision should be null. In fact, the probability of an IPv6 address collision is already extremely low when EUI-64 are used since they are based on EUI-48 (e.g. Ethernet MAC addresses) which are supposed to be unique. To furthermore reduce the probability of an address collision when generating an EUI-64 from an EUI-48, we propose to replace the added **ff:fe** 16-bit pattern by a randomly generated 16-bit number. It means that in the very rare case where two nodes have a common EUI-48, they will generate a 64-bit host identifier with a collision probability of 1/64536 (1.5e-5). As a result, we think that it is unnecessary to add the overhead and the complexity occured by a DAD procedure.

4. Upstream neighbor selection

We propose three different algorithms in order to select a prefix/gateway pair. The first algorithm ensures that a node always selects the closest gateway, whatever prefix it uses. We call this algorithm the *distance* algorithm. In contrast, the second and third algorithms ensure that a node keeps its current prefix as long as it has neighbors with the same prefix, whatever distance it is from its current gateway. If a node becomes isolated (in the prefix sense), it is allowed to acquire a new prefix. We therefere call these two algorithms the *stability* algorithms. The difference between the two *stability* algorithms is in the way they select a new prefix (and upstream neighbor).

We also consider that the global address acquired by an ad hoc node should be used as the Mobile IPv6 [Johnson et al., 2004] care-of address of the node. MIPv6 is used with mobile nodes to maintain connections at the transport layer. Each change of global address in the ad hoc network will therefore trigger the sending of at least one binding update message.

To maintain prefix continuity, each node must ensure that it does not become isolated from other nodes which share the same prefix. Each node thus permanently checks its neighborhood in order to detect the loss of neighbors which share the same prefix. To do so, each node maintains a list of its current neighbors, whatever prefix they use. Neighbors are discovered via the reception of the GW_INFO messages they send. When a node receives a GW_INFO message from a node that is not yet in its neighbors list, it adds this node to its neighbors list, records the sequence number of the GW_INFO message and starts a timer associated to the entry. Upon expiration of the timer associated to it, an entry is removed from the neighbors list. When a node receives a GW_INFO message from a node that is already in its neighbors list, it restarts the timer associated to the entry if the sequence number is greater than the one recorded for this neighbor. Note that we assume that all wireless links are bi-directional.

4.1 The distance algorithm

This algorithm is very simple: a node simply chooses as its upstream neighbor the node that advertises the shortest distance to a gateway. The main advantage of this algorithm is therefore that the path between a node and its gateway is a topological shortest path. Moreover, in particular circumstances, this algorithm can also lead to the creation of well-balanced sub-networks, in the sense that all sub-networks will have an equal size (statistically speaking). This is for example the case if the area formed by the gateways is symmetrical, and if the ad hoc nodes are uniformely distributed in this geographical area. This is because each node selects the closest gateway. If we assume that the radio characteristics are similar for each ad hoc node, the distance in hops between two nodes in the network is indeed strongly linked to the geographical distance that separates them. The main drawback of this algorithm is that a node may frequently change its global address as topological changes occur. In particular, the distance algorithm does not prevent a node from joining a new sub-network even if the node still has neighbors which are in its previous sub-network.

4.2 The stability algorithms

We have proposed two alternative algorithms whose objective is to maximize the time during which a node keeps its current global address. In other words, with these algorithms a node remains a member of its current sub-network as long as possible, i.e. until it cannot find an upstream neighbor that uses the same network prefix. In practise, a node ignores GW_INFO messages sent by neighbors of a different sub-network as long as it has neighbors from its own sub-network, i.e. as long as there exists a path of nodes using its current prefix between itself and the gateway. In contrast to the previous algorithm, the distance to the gateway is no longer the main criteria when selecting an upstream neighbor. However, a node must select its upstream neighbor in order to find the shortest possible path to its current gateway. The path between a node and its gateway is therefore a shortest path within the sub-network, but it might not be a topological shortest path. For example in Fig. 1(a), the leaf node of the white sub-network/tree has a 4-hops path to its gateway. This path is the shortest path with respect to the sub-network, but it is not a topological shortest path (i.e. 3 hops via the light-grey node above it). For example, if the distance algorithm was used, the leaf node of the white sub-network would decide to join either the light-grey or the dark-grey sub-network as in both cases there is a closer gateway (i.e. 3 hops).

The two stability algorithms behave differently when a node becomes isolated from its current subnetwork. With the first variant named *stability-nowait*, the node selects as its new upstream neighbor the first node from which it receives a GW_INFO message. The node discards it previous global address and creates the new global address with the new prefix. With the second variant called *stability-slow-start*, the node will first gather neighboring information during a short amount of time (e.g. 3 seconds, hence its name *slow-start*). The idea is to select the upstream neighbor among a large set of neighbors in order to increase the probability to remain a member of the new subnetwork as long as possible. Note that the node also selects its new upstream neighbor such that it finds the shortest possible path to a gateway. The main advantage of the two stability algorithms is that they minimize the number of prefix changes. This greatly reduces the overhead induced by the sending of MIPv6 binding update messages when a node changes its global address. The main drawback of these algorithms is however that the path between a node and its gateway is not necessarily a shortest-path (with respect to the entire topology). However, within a sub-network, this path will always be a shortest-path.

5. Conclusions

In this paper, we have presented an address autoconfiguration scheme for IPv6 ad hoc networks. The core of our proposal is the concept of *prefix continuity*, which ensures that there always exists a path between a given node and its gateway such that all nodes on this path share the same network prefix. Moreover when there exist multiple gateways, our protocol builds and maintains a forest of logical spanning trees, where each tree if formed by nodes that share a common global network prefix. This concept results in an organized ad hoc network, in the sense that sub-networks (with respect to prefixes) are automatically created and dynamically maintained when multiple gateways are available.

References

Draves, R., Padhye, J., and Zill, B. (2004). Routing in Multi-radio, Multi-hop Wireless Mesh Networks. In *Proceedings of ACM Mobicom 2004*. Philadelphia, PA, USA.

Johnson, D., Perkins, C., and Arkko, J. (2004). RFC-3775 - Mobility Support in IPv6.

Thomson, S. and Narten, T. (1998). RFC-2462 - IPv6 Stateless Address Autoconfiguration.

V. Bahl (organizer) (2004). Wireless Community Mesh Networks - Hype or the Next Big Frontier? In *Panel Discussion at ACM Mobicom 2004*. Philadelphia, PA, USA.

Wakikawa, R., Malinen, J., Perkins, C., Nilsson, A., and Tuominen, A. (2002). Internet Connectivity for Mobile Ad hoc Networks. *Wirel. Comm. and Mobile Computing*, 2(5):465–482.

Weniger, K. (2005). PACMAN: Passive Autoconfiguration for Mobile Ad hoc Networks. *IEEE JSAC*, 23(3):507–519.

Weniger, K. and Zitterbart, M. (2002). IPv6 Autoconfiguration in Large Scale Mobile Ad-Hoc Networks. In *Proceedings of European Wireless Conference 2002*. Florence, Italy.

Xi, J. and Bettstetter, C. (2002). Internet Connectivity for Mobile Ad hoc Networks. In *Proceedings of 3GWireless*. San Francisco, CA, USA.