What are yRFCs?

yRFCs are discussion documents on one or more issues related to the design, development or implementation of the Y-Comm architecture. Y-Comm is a new architecture being developed to support heterogeneous networking. yRFCs therefore represent the views of the authors of the document. They are non-binding and do not oblige anyone to agree with or to implement any concepts or details expressed therein. They can also be modified without notice. Finally, yRFCs are public documents and should not in whole or in part be the basis of a patent or copyright claim. Please contact the authors directly to discuss relevant issues.

yRFC1: Network Addressing for Heterogeneous Environments

Authors: Mahdi Aiash (m.aiash@mdx.ac.uk) and Glenford Mapp (g.mapp@mdx.ac.uk)

This document was released to the Y-Comm Website Team on: 27th July 2009.

1.0 Introduction:

The yRFC discusses the issues related to addressing entities in heterogeneous networking environments. Support for heterogeneous networking is now a major challenge for the networking community. In this environment however, mobile nodes may be connected to several networks at the same time. Support for mobility and providing continuous connectivity using efficient vertical handover mechanisms are two key goals. In order to achieve these aims, a new addressing scheme is proposed.

1.1 Background

Addressing is one of the biggest issues in networking systems and can exist on many levels. There is a LAN or MAC address which allows information to be exchanged by nodes connected over a Local Area Network (LAN). So an Ethernet Address is an example of a MAC or LAN address. Objects or devices can also have a network address which allows them to be connected over several networks such as the Internet. At present the Internet uses IP Protocol Suite with IPv4 being widely used. IPv4 uses a 32-bit address field to specify its source and destination addresses. It should be stressed that an IPv4 address is a network interface address. Hence devices with more than one interface will have several IPv4 addresses. Each network interface can be managed independently so that one interface may be unaware of other interfaces even though they are attached to the same device. Such problems, known as multi-homing, used to be confined to routers and switches, but in heterogeneous networking, this is now an issue for end devices as well.

1.1.2 Towards IPv6

IPv6 uses 128-bits for addressing objects and there is now a concerted effort to get IPv6 widely deployed. There are 3 basic types of IPv6 addresses [1]:

Unicast address:

Similar to the concept of unicast in IPv4, this type identifies a single interface. So a packet sent using a unicast address is delivered to that specific interface identified by the address.

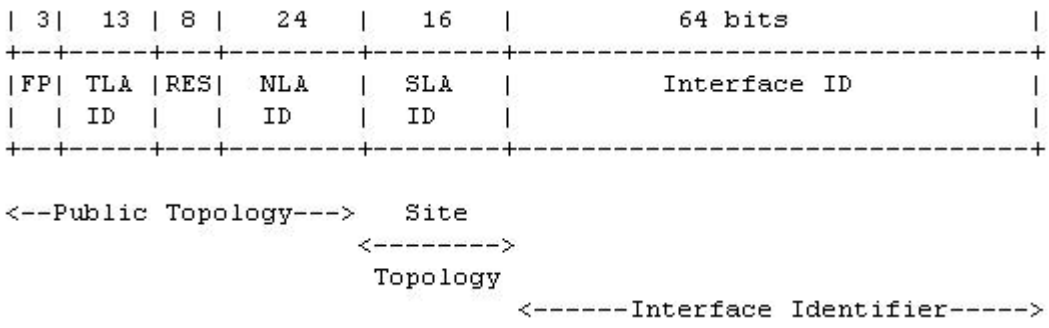
Anycast address:

This is a new type of address proposed by IPv6. It identifies a set of interfaces which may belong to the same node, or to different nodes. A packet sent using an anycast address is delivered to one of the associated interfaces identified by the anycast address.

Multicast address:

This type was originally proposed with IPv4 protocol. It identifies a set of interfaces. Again, these might belong to one node. When sending a packet using a multicast address, the packet is delivered to all interfaces, identified by the address. Another distinct difference in IPv6 is that the broadcast address of the IPv4 has been removed.

The figure (1) shows the format of a global unicast address structure as stated in RFC 2374:



Where

FP: Format prefix.

TLA ID: Top-Level Aggregation identifier.

RES: Reserved for future use.

NLA ID: Next-level aggregation identifier.

SLA ID: Site-level aggregation Identifier.

Interface ID: interface identifier.

1.1.2 Criticisms of IP Addressing for Heterogeneous Networks

1) **Poor support for mobility**: According to [2], IP addresses attempt to achieve two goals. Firstly, it uniquely identifies the network interface to which packets should be

delivered. This means that since IP addresses are broken down into network and host bits, a machine using the Internet must have a valid address on a given network in order to receive packets over that network. Secondly IP addresses are used by the routing mechanism to decide the route which should be used to deliver data to that interface. An IP address can fulfil these two functions while the object/node is static. This also allows the routing mechanism to cache routes to end devices making the system highly efficient. However, the dual function of an IP address breaks down with mobile nodes. Firstly, if the mobile node is on a new network, then it requires a completely new IPv4 address. This allows packets to be routed to the mobile node, but the identity of the network interface has now changed. Hence a new way of finding the new address from the original address must be found. These issues are addressed in Mobile IPv4 by the introduction of Home Agents and Foreign Agents, etc. Mobile IPv6 [3] reduces the complexity, but these solutions are difficult to scale to heterogeneous environments where mobile nodes may be attached to several networks at the same time and may be doing vertical handovers between these networks.

2) **Multi-homing**: As we have stated multi-homing is an end-device issue in heterogeneous environments and so a solution is badly needed. Multi-homing must be addressed in two key situations: Firstly, a single node has multiple NICs with different IP addresses. Thus the node belongs to different networks which might be using different technologies. Secondly, a single node has a single NIC but with multiple IP addresses. In this case if the connection assigned to one address fails, the node can directly switch to the other address. However, all ongoing connections attached to the previous address are terminated causing connection disruption. Therefore, this form of multi-homing is therefore not recommended for mission-critical servers. Many protocols have been proposed to solve the multi-homing problem such as Host Identity Protocol (HIP) which is explained in [4]. HIP is a "3.5" layer entity in the TCP/IP architecture, capable of working with IP v4/6 rather than a modification to IP itself.

   Current IPv6 formats do not directly help with multi-homing because the Interface ID is included as part of the IPv6 address. IPv6 can attempt to deal with this problem by using anycast addresses, allowing packets to be delivered to the mobile node on any available interface, but the use of both anycast and unicast addresses makes the situation very complex for heterogeneous networks.

3) **Vertical Handover complicates things even further:** Handover can be divided into two types; horizontal handover which is used to switch between base-stations of the same technology and vertical handover which is used to switch between base-stations of a different technology. Under the current IPv6 scheme, when the mobile node does a horizontal handover, then the mobile node needs to change the network part of the IPv6 address, but the interface identifier stays the same. This is manageable, since the previous base-stations will be able to forward packets to the new base-station. However, a vertical handover will change both the network and interface identifier

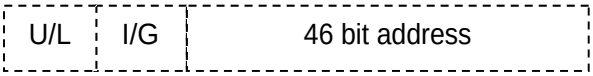
parts of the IPv6 address. In effect, the need to continuously change IPv6 addresses will make the multi-homing problem more acute in IPv6 networks that support vertical handover.

4) **The Internet of Things**: Over recent years, there has been a shift in thinking concerning what should be addressed at the network level. We are beginning to move away from the identity of the network interface as the prime identifier and move towards an address or unique id/name for the object rather than the interface. This new environment, called the Internet of Things, based on RFID technology, will allow new mechanisms to be employed.

1.2 Ethernet addresses:

Ethernet Addresses are LAN addresses which are used in Ethernet networks. Though minimalist in its design, Ethernet addresses achieve high functionality by qualifying addresses using the first two bits of the address.

The figure below [5] shows the format of the 48bit Ethernet address



I/G=0 individual- Unicast- address.

I/G=1 group – multi/ any cast- address

U/L=0 Globally managed address; addressed administered by IEEE

U/L=1 locally managed address; local DHCP server.

Once the data packet reaches the LAN's gateway- based using the IP destination address, it is delivered to the node based on its Layer 2 address. The mapping between L2 and L3 address is done by ARP and related protocols.

2.0 An Enhanced IPv6 solution for Heterogeneous Networking

This yRFC is looking at a new addressing scheme for heterogeneous networks. However, it can be shown that the problems highlighted above can be solved by enhancing the IPv6 addressing scheme with a modest change to how the address bits are used.

Our first change is to divide the IP address into two main components. The first is called the Node ID and is used to identify the device or node and NOT a network interface. The second is called the Location ID which is used to say where the mobile node is located.

Different networks are therefore represented by their location IDs. Handover will therefore involve a change in the Location ID but not a change in Node ID. The benefit of this approach is that both vertical and horizontal handovers require the same change in the

Location ID, since the Node ID identifies the mobile node; it is not changed and so multi-homing is not exacerbated with respect to vertical handovers.

2.0.1Defining the Node ID

The Node ID/Locator ID is nothing new and has been proposed by several research groups in various contexts [6-8]. However, most groups have suggested that the split is an equal one. So 64 bits are used to identify the Node ID and 64 bits are used to identify the Location ID. This leaves these proposals compatible with the current IPv6 position. However, insisting on this would, in our view, leave things in an unsatisfactory position as the proposed split would still need to address certain issues.

The first is how do we do multicast and broadcast address in a local network? Do we use part of the Location ID or do we designate a unique set of Device ID address which in combination with the Location ID would signify a multicast address. The second is support for local or site addresses. IPv6 supports these ideas by using local and site address prefixes (hex: FE80) and (hex: FEC0) respectively. However, these are a lot of bits to use from the network part of the IPv6 address and we are not sure that it is necessary to do so. We believe a better combination of Locator ID and Node ID concept could give more efficient bindings to deal with some of the issues discussed above.

Firstly, we define the concept of a Device ID which uniquely identifies the device or mobile node. Device IDs are given to devices when they are manufactured and are not changed during their lifetime. This Device ID concept fits very well with the IEEE-EUI64 guidelines [6] which recommend how a 64-bit global identifier can be constructed.

However, in order to address the issues highlighted above we believe it is better to augment the Device ID rather than using bits from the Location part. Instead, these additional bits form part of the Node ID. We suggest that the following three additional bits be used to qualify the Device ID and thus give it a better network management context:

2.02 The Three Bits

The first bit is called the **Static** Bit or S bit. This is used to indicate how mobile a device is expected to be. If the S bit is set to one then the device should be treated as a static device and so its Location ID would be fixed.

What are the benefits of this? By splitting the Node ID space so explicitly, we can adopt different solutions based on the mobility of the device. For example, if we know that the device is static, then we will know that the Location ID is fixed and so it will be possible to cache routes to these endpoints making access to them quicker. This should certainly benefit distributed computing which is based on client-server models where the servers tend to be at fixed locations. So this will allow server accesses to be optimised.

The second bit is called **Unicast/Multicast** Bit or U/M bit and indicates whether a Node ID is unicast or is multicast. This is similar I/G bit in the Ethernet Address.

Again one may ask: how does this benefit us? It is now possible to use a Device ID to represent multicast data such as a stream or distributed group communication over the network. In addition, it has been decided to use the Device ID, FF:FF:FF:FF, as a broadcast address. This means that if I want to broadcast to all the devices on a given network (i.e. network paging); I use the Location ID to specify which network I am interested in and set the Device ID field to the broadcast value and the packet will be heard by all entities using that network.

The third bit is called the **Global** Bit or G bit and indicates whether a Device ID should be treated as a global or local object. The main idea behind this is object visibility. So even though an object or device has a global Device ID, the owner/system administrator may not want it to be accessible via the Internet. If the G bit is zero then the device is not globally accessible and packets from this device will not be routed by external routers or gateways. In addition, it allows the system administrator to use locally assigned Device IDs provided that they are only used locally. This therefore allows a NAT-based security scheme to be used without having to assign special IPv6 network addresses.

2.0.3 The Interface ID - revisited

Though we have indicated there is no need for full-blown Interface addresses at the network level, we believe it is beneficial to be able to indicate exactly which interface the device may be using for any given series of packet exchanges. The main reason is Quality-of-Service support including security. So from a security viewpoint, it might good to have able to easily set up secure mechanisms such as secure tunnels to devices. These systems can be implemented as pseudo network interfaces (See FreeSwan over Linux) in operating systems. However traditionally it has been hard to integrate such interfaces. So by having a way of indicating explicitly which interface should be used, it would help provide a more stable environment. In addition, as the Quality-of-Service on certain interfaces may be very different, it is necessary for the system to be aware of this in order to deal with the varying QoS in heterogeneous networks.

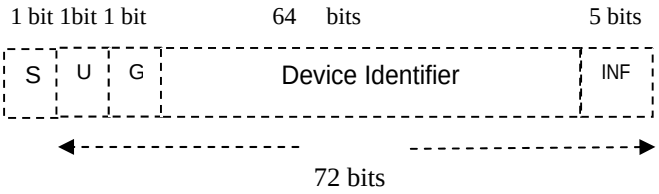
What is being proposed is that 5 bits, called the **Interface Number** (INF), be used to augment Device ID. The INF is a number that indicates which interface is being used. However, if INF value is zero, the packet is delivered to any interface that the mobile is currently using. This is similar to the anycast address in IPv6. In this situation, it is up to the local administration to decide which interface must be used to connect to the mobile. The mobile itself can change the interface it is using at anytime during a connection.

If the INF value is set to 0xF, this interface number is treated as the broadcast interface, relative to the mobile node. Hence a packet delivered on that interface, will be sent to all the known interfaces on the mobile node. If the INF value is 1, then this is known as the primary interface of the device.

Finally, there needs to be a way to map between the (Device ID, INF) pair to the actual Interface ID or MAC address. This can be done using several locally administered mechanisms.

2.1Node Identifier (Node ID)

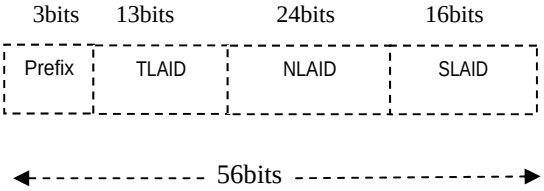
By augmenting these decisions, we can represent the Node ID as shown in the figure below. It uses 72 bits to identify the mobile node along with its interfaces. The following figure shows the Node ID part of the proposed address scheme.



5 bits Interface Number Field (INF): it is used to address up to 16 interfaces on a single node.
64 bits Device Identifier: a unique identifier set by the manufacturer.
1 bit Global/Local (G): similar to the G/L field in the Ethernet address; G=1 → Global
G=0 →Local
1 bit Unicast/Multicast (U) similar to I/G field of Ethernet address; U=0 →Unicast address
U=1 → Multicast address.
1 bit Static (S) field: This is used to determine whether a node is mobile or static one;
S= 0 → Static
S= 1 → Mobile

2.2 Location Identifier (Location ID)

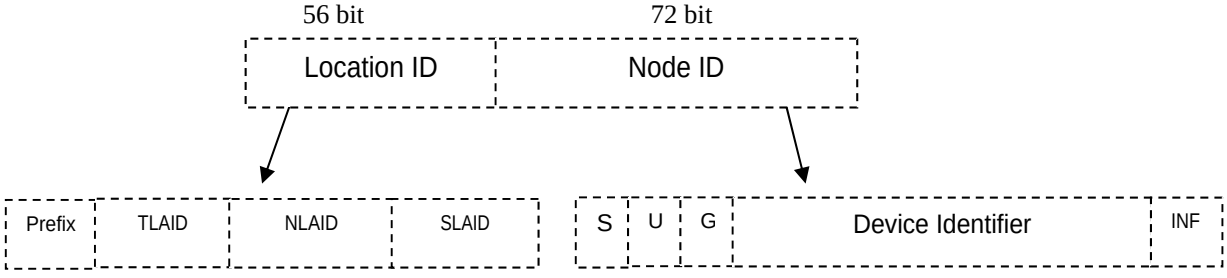
The figure below shows the format of Location ID:



The Location ID is a 56-bit field. This is quite similar to the network part of current IPv6 addresses. The only difference is that we have taken out a reserved 8-bit field between the TLAID and the NLAID fields. So we expect the bits to work in the same as current IPv6 addresses

3.0 Putting it all together

By combining the Location ID and Node ID fields, we will have our proposed structure for heterogeneous networking.



4.0 Conclusion:

This yRFC has looked at a new address scheme for heterogeneous networking. We believe this address scheme can be implemented by enhancing the IPv6 Address Format as shown. We believe that this scheme can be quickly implemented. Mobility schemes based on this type of addressing is being discussed and will be presented in later yRFCs. The authors would value feedback on this document.

5. 0 References:

[1] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture," RFC 2373, Cisco Systems, July 1998.

[2] G.E Mapp, "Is IPv6 the Key to a Global Network Infrastructure?" IPv4 to IPv6 Migration, 10-11 September 2001, Sheraton Hotel Stockholm.

[3] D. Johnson, C. Perkins and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[4]. R. Moskowitz and P. Nikander," Host Identity Protocol Architecture," in work in progress (internet-Draft draft-ietf-hip-arch-02), January 2005.

[5] J. Walrand," Ethernet, ARQ," Available at http://inst.eecs.berkeley.edu/~ee122/sp07/ethernet+arq.pdf [accessed 12 July 2009].

[6] IEEE Standards Association, "Guidelines for 64-bit Global Indentifier (EUI-64)", IEEE 2007.

[7] M. O'Dell, "GSE – An Alternative Addressing Architecture for IPv6", Internet Draft, February 1997.

[8] R. Atkinson, "ILNP Concept of Operation", Internet Draft, June 2008.