Home Network Protocol with Device-centric Addressing

X. Ren and M. Fleury
Electronic and Computing Systems Department
University of Essex, United Kingdom
{xren,fleum}@essex.ac.uk

Abstract— Existing plans for home networks are aimed at the short to medium term and postulate that TCP/IP will extend to home networks. This view may be unsustainable in the long term, with increasing network diversity. This paper proposes a network/transport home network protocol (HNP) that supports multiple levels of flexibility. Protocol layer fusion is possible through a combined mode or TCP compatibility is preserved with a split mode. Within combined mode simple or flexible sub-modes are possible. HNP introduces device-centric addressing allowing anycast and broadcast to be easily supported. Relative to IP addressing, header overhead is reduced and short command messages are supported.

I. INTRODUCTION

The home network (HN) is set to transform from a computer/PC-centric to device-centric network in which white goods, entertainment equipment, and control and environment-monitoring sensors are all network-enabled. Because access to the Internet has driven the growth of HNs, it is a natural assumption that the HN will continue with the same TCP/IP protocol suite as the Internet. Misleadingly, the rise of campus and corporation networks, all of which are based solely on computers, has led to a belief in the seemingly infinite extendibility of IP. Unlike a computer, for many purposes in the home the fact that a device offers, say, a refrigerator service is more important than which particular refrigerator it is. Hence, generic addressing of device type is more important than addressing for routing purposes. This suggestion is part of a wider programme, which, for ease of reference, we call Home Network Protocol (HNP). The main contribution of this paper is device addressing within HNP.

HNP is a replacement for IP and possibly TCP or UDP (if protocol layer fusion occurs). Because of the diversity of physical media, HNP does not replace the data-link layer (in the ISO OSI 7-layer protocol model), unlike a scheme from Almeroth *et al*. [1]. The HN will almost certainly be composed of various physically different network segments such as a computer segment (for example 100 Mbps Ethernet), an embedded wireless segment (for example, Ultra Wideband) and an audio/video (A/V) segment (for example through IEEE 1394 [2]). Ethernet's medium access control (MAC) does not provide latency guarantees, whereas this is desirable for both control and A/V communication. Other network types such as phoneline (HomePNA [3]), power line (X.10, LONworks, CEbus [4], HomePlug and others), and cable (CableHome [5], Smart House) may also be present, due to differing market penetration. As a result of network heterogeneity, spanning tree routing across bridges already takes place at the data-link layer.

Undoubtedly, transfer of large messages (for example, A/V streams, file downloads (including peer-to-peer), and reconfiguration code) must still occur both within and from outside the HN. Within HNP, flexible mode and header extensions allow the control of large messages within the home. It is also likely that one source of external traffic will be from remote device configuration and network management [6] and another will be Internet control of devices. For these purposes, HNP split mode retains TCP as a transport protocol but substitutes HNP as a network layer protocol. For external communication, semantic compatibility allows conversion of HNP addressing at the HN edge. Software written for home devices assuming a TCP/IP stack existed can be catered for by means of an adaptation layer above HNP flexible mode. Equally, HNP split mode more directly caters for backwards compatibility.

To ease deployment, it is possible to zone the HN [5] so that a home network protocol is confined to a protocol zone, with QoS and other zones partitioned through bridges. The addressing structure of HNP allows an easy mapping between incoming downlink IP messages and HN messages. To avoid the Residential Gateway (RG) constructing uplink TCP/IP headers, split mode allows the RG simply to remove a header before passing the TCP/IP packet over the external network, while the source itself performs protocol handling. For other uplink packets, hardware offloading is able to reduce the load on the RG, which is anyway reduced by the typical downlink/uplink asymmetry of Web traffic.

II. PROBLEMS WITH IP

Short-term commercial imperatives lead to the advocacy of existing protocols or updated versions of them. The already near-exhausted IP version 4 (IPv4) address space (according to the IETF's Address Lifetime Expectations working group) is not suited to a massive growth in addressable devices within homes. Gokulrangan [7] advocates transitioning to IPv6, currently supported through dual protocol stacks. This suggestion has the merit of avoiding the need for Network Address Translation (NAT), which would impede certain end-to-end applications. However, a more radical suggestion

advocated herein is to abandon addressing of individual devices for routing purposes (though not for configuration purposes).

IP's associated transport protocol also imposes processing complexity and a memory overhead unsuitable for battery-powered processors. TCP's congestion control mechanism is superfluous for short messages, as Arithmetic Increase Multiplicative Decrease (AIMD) rarely reaches peak bandwidth by the time the short message packets have been delivered. The TCP code size is not small: 64 KB in Linux 2.2.20 and 49 KB in FreeBSD 4.4 [2]. The other TCP/IP suite transport protocol, UDP, supports little functionality beyond that given already by IP, requiring reliability and rate control to be added for any application type that requires it.

Universal Plug and Play (UPnP) has announcement, description, control, eventing and presentation message types. In the worst case, for 1-byte messages from a sensor, IPv4's 20 B header would result in a 95.2% overhead, which rises to 97.6% for IPv6's 40 B header. IP header compression schemes [8] offer no amelioration as they are only directed towards persistent connections. However, excessive IP header sizes are not the most serious concern for the HN. For small messages, IP supports unnecessary functionality such as packet fragmentation and a time-to-live hop counter, which requires extra processing and hence extra battery usage for mobile wireless devices.

III. RELATED WORK

A number of alternative transport layer protocols are application oriented: Xpress Transfer Protocol (XTP) [9] for high-speed networks, Versatile Message Transaction Protocol (VMTP) (RFC 1045), Stream Control Transmission Protocol (SCTP) (RFC 2960) for speech control messaging, and Game Transport Protocol [9]. Lightweight Command Transport Protocol (LCTP) [2] is intended for HNs as it provides instant connections and data reliability. However, in [2] LCTP did not present a solution for longer messages. The closest lightweight protocol to the ideas in HNP is Pseudo-IP (PIP) in [1], which was later elaborated as FLIP [10] and aimed at mobile devices. HNP adopts meta-headers from [1] and uses its flexible layer fusion. However, in HNP a common header goes before a meta-header (if present). In HNP combined simple mode, there is no meta-header or IP header and, in HNP split mode, there is no meta-header and TCP acts as the transport layer protocol.

The Plutarch proposal [11] explored the possibility of multiple coexistent networks, which the arrival of sensor network, GPRS, IPv4 and IPv6 networks seem to be leading towards. In a Plutarch world, the HN is a 'context' with its own protocol set. Well-defined 'interstitial' functions govern the interface between each context. Despite its limitations, NAT is a form of interstitial function, which, along with the authors of [11], we feel are an inevitable requirement when interfacing to IP Internets. Sensor and wireless networks, will lead to breaches in IP's homogenization, of which NAT is an imperfect example.

For those comparatively few applications that embed IP addresses in the payload, an Application Layer Gateway (ALG) is an alternative to NAT. An ALG can be dynamically created at a reconfigurable element, either flash ROM for programmable devices or Field-Programmable Gate Array (FPGA) for hardware processing. The authors of [1] also listed a set of contexts for which IP is *unsuitable*: the home, the transport highway, and inhospitable environments, *e.g*. disaster areas. In the home context, the type of device ranges from broadcast-only sensors; through binary-state devices (*e.g*. light switches) able to receive simple commands or broadcasts; and poly-state devices with limited embedded processing (*e.g.* kitchen appliances); to fully capable devices such as PCs and PDAs.

IV. HOME NETWORK PROTOCOL

A. *Home Network Protocol layer fusion*

In this paper, we propose the HNP, a transport and network layer protocol for the HN. HNP can replace the functionality of both the Internet and Transport layer or, in a lightweight variant, only represent the functionality of the Internet layer. Layer fusion is achieved by means of meta-headers [1]. As an HN consists of a variety of different devices from complex computers to simple sensors, the HNP allows them to coexist and interoperate under the same network infrastructure. The upper layer application or protocols choose what functionalities should be included in HNP, depending on the capabilities of a particular device. The functional components can be divided into addressing, management, Quality-of-Service (QoS), security, and protocol operation, though space limitations in this paper prevent a detailed discussion of these components.

A principal feature of HNP is an alternative addressing system, designed to replace IP addressing in the HN. IP addressing is network-oriented, for routing packets through different networks. An HN is an end-point of the Internet. It is a single network with multiple network segments. Instead of using a network identity (id) in an HN we introduce a device id. The device id is used to identify each type of device in the HN such as TVs, stereos, PCs, and fire alarm sensors. The new address system is device-oriented. It makes communication between devices much easier.

B. *Addressing*

An IP address, designed to identify a host on the Internet, has global scope. The address is divided into the network id and host id. As is well known, the network id is used for routing packets to the correct network or sub-network. IPv4 has a number of classes that identify the number of bits employed for the network id and host id on a per-class basis. IPv6 has a uniform address space with 64 bits allocated to the network id and 64 bits to the host ID. The only device that should know the IP network id is the RG. Instead of employing a network id in a HN, we introduce a device id. The device id is unique for each type of device. A device id helps devices to find and communicate with each other, based on their functionality. Furthermore, a device id has the following advantages over an IP address:

- Messages can be anycast to an available device of a particular type.

- Messages can be broadcast to all devices of a particular type.
- A better device-oriented structure is provided to upper-layer service discovery protocols such as UPnP and Jini [12].

There is a subtlety inherent in the term 'device', because it is possible to imagine (say) a refrigerator that offers incompatible services when compared to another refrigerator. There is also the issue of a device that offers a sub-set of the features possible for a device of that type. At some point that device may be deemed to be a device of a different type. This issue is important when discussing anycast. If it is possible for a device not to support a feature then it is possible to make a redundant anycast. In the interests of simplification, this paper assumes that devices are sharply distinguished by their services but accepts that a further round of device service schema design may be necessary.

Here is an (hypothetical) example that shows the benefit of the device id. A home heating device is turned on or off by sending a message to it. All such heating devices have the same device id. Thus, a broadcast message to the heating devices' device id will switch all heating device on or off within a home. A PC, a PDA, or even a mobile phone can send the broadcast message, without being aware of the heating device's network address.

When a heating device is switched on or off, it will need to register its state with a home monitoring device. This can be achieved by sending an anycast message to the device id of monitoring devices. The message will be delivered to one of the monitoring devices in the HN. In most cases, there will be only one monitoring device in an HN but the anycast message will still be effective. Compared to the IP protocol, which has no way to send messages to particularly types of devices and a device has to know the network ID of the home network to perform a network broadcast, the HNP is much simplified.

The addressing structure is also designed in the way that is easy to convert to an IP address. Fig. 1 shows the address structure of the design. The last 8 bits or 64 bits are equivalent to the host id in an IP address. The protocol header indicates whether 8 bits or 64 bits are needed to match respectively IPv4 or IPv6. The RG simply replaces the device id and location id with the network id to form a full-qualified IP address.

The location bits indicate the physical location of a device in the home, the use of which is most likely to indicate which room the device is placed in. Combining the device id and location id provides a powerful method able to broadcast or anycast to a type of device in a particularly room. Location-based addressing may also impede address spoofing and can deter device theft. It is assumed that most devices in a home will have a static location but a specific mobile location address is reserved for portable or handheld devices. Alternatively, mobile devices can implement room pairing, whereby the device identifies with the configuration component (see Section IV.C).

Device id 16 bits	Location id 8 bits	Host id 8/ 64 bits

Figure 1. Addressing structure of HNP

C. *Addressing structure of HNP*

To avoid address conflict between devices of the same type, individual devices must obtain an address from the RG. The RG has no information about individual devices until they request an address. In this respect, HNP works in a very similar way [13] to the Dynamic Host Configuration Protocol (DHCP).

A device initially anycasts a message to an RG. As the device does not have an address yet, it uses zero as its source host id. The anycast message should contain the device's device id. Once the RG receives the message, it responds to the device with a host id. If there is a location id set for the device, the RG also responds with the location id to the device. The RG should record every address that it has assigned. However, in some cases an RG may not be available within a home network or the RG device may not have a Dynamic Host Configuration Service Component (DSC). Therefore, the devices cannot get an address from an RG. In these situations, an Address Generator Component (AGC) on each individual device generates a random host id and negotiates with other devices to ensure that each device has a unique host id. The location id is distributed by the DSC component. This mechanism requires a user to pre-set the location id for each device at the RG. If there is no RG or the user does not pre-set the location id in the RG, the location has to be set manually on each device. The AGC component does not automatically generate a location. However, the location id is not a requirement for many purposes and it may be left un-configured.

Address distribution should be automatic without user interaction. However, the location id cannot be easily self-configured. This is because that location id is related to physical location, for example rooms in homes. The only way to reflect a device's physical location to the HNP protocol is by the network topology. Therefore, in order to make location id self-configurable, the network topology has to be forced to a pre-defined type. The Border Beacon Component (BBC) is designed for that purpose. The BBC distributes location ids within a physical area, such as a room. It first connects to the device that has implemented a DSC to request a location id. If such a device is not available, BBC works in a way similar to AGC. It generates a location id and negotiates with every other BBC device to ensure that its id is unique within the home. Mobile devices will not request a location address as they will already be configured with the mobility location address or they can pair with the DSC.

D. *Anycast and broadcast*

Anycast (RFC 1546) is a useful technique in IPv6 for providing redundancy and load sharing to specific types of network services on the Internet. The idea of anycast is also applied to HNP. It is most suitable for poly-state and fully capable devices. Often, there will be more than one device of the same type within a home network. Anycast enable packets to be delivered only to one device of the same type. Whereas IPv6's anycast mechanism is based on application of a distance metric by routers, this is not necessarily appropriate for a HN, as latency may be low. Therefore, HNP is expected to use a load-balancing algorithm, though the method is implementation dependent. There are other differences

between IPv6 anycast and HNP's anycast. IPv6 anycast addressing assigns a common unicast address to multiple interfaces, hosts or services on an Internet. An HNP anycast address is not a unicast address. It is a group of addresses with the same device id and location id. The host id is set to all ones. Like HNP's broadcast address (Section IV.E), it does not have to be in a special address range; any device id can form an anycast address and no administrative control is required.

HNP broadcast is limited to the HN context. There are two types of broadcast designed with the address system: 1) full broadcast and 2) device-limited broadcast. Full broadcast works very much like IP broadcast. There is a special address for all devices in the HN. If a packet is sent to that address then all devices within the home network receive the packet. This is most suitable for binary- and poly-state devices, though fully-capable devices benefit. A device-limited broadcast is delivered to a particular type of device. The broadcast address is the device id for the type of device, while the location id and host id are set to all zeroes.

E. Discussion

To a certain extent the range of anycast and broadcast as stated in Sections IV.D and IV.E is arbitrary. For example, an argument can be made that each of anycast and broadcast should come in two varieties: 1) device-type only and 2) device-type and location combined. This is a secondary issue compared to the need for device-oriented addressing within the home. As multicast will still be needed for personal devices (for example, computers, PDAs and mobile phones) within the group then a mechanism similar to IP multicast with Internet Group Management Protocol (IGMP) (RFC1112) will need to operate through the RG. Equally, compatibility with external multicast groups should be maintained. The mechanism for doing so is beyond the scope of this paper.

V. TRANSPORT AND INTERNET LAYER FUNCTIONALITY

HNP packets are composed of a common header, an optional header, and the payload. The common header includes the fields necessary to deliver the packet to its destination. It also indicates which optional header is present in the packet. The advantage of splitting a packet header into common header and optional header is that the application is able to choose the right optional header for its needs. The optional header is a collection of header fields indicated by a meta-header [1] [10]. A meta-header is a map of the header fields included in the optional header. Specifically it is an array of bits divided into 8-bit words. If a header field is included in the packet, then the corresponding bit in the meta-header is set. The first bit of each word acts as a continuation bit (Fig. 2). If the bit is set, it means that there is an additional word following. The decisions as to which optional fields to include, their size and significance are left to application programmers.

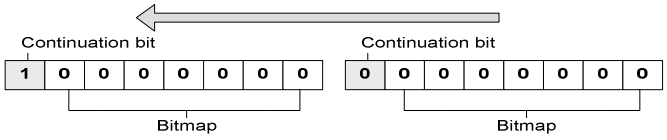

Figure 2. Division of each meta-header optional word into continuation bit and bitmap, after [1] [10], showing the direction of transmission.

The Transport layer and Internet layer can be split or combined. In split mode, HNP can work with TCP and UDP (and other lightweight transport protocols). Therefore, split mode provides an adaptation layer for any application written supporting a TCP/IP protocol stack. Whether split or combined mode is in use is indicated by a version field within the common header. There are two sub-modes in combined mode, one of which is the simple sub-mode administered by the Simple-mode Protocol Component (SPC) shown in Table I.

In the simple sub-mode, HNP provides limited QoS reliability. It does not include end-to-end reliability and packet ordering. The issue of re-ordering of packets from the same stream, when these take different paths across the Internet, does not arise in a HN. In most cases within the Internet, routers cause loss of packets. In the HN, the RG is likely to be the main bottleneck, as a bridge will be dimensioned by the two network segments it connects. Simple sub-mode is also connectionless and does not have congestion control. It therefore relies on hardware to perform reliability checks, ordering and congestion control. Consequently, the simple sub-mode HNP stack is small and can be implemented on most devices within the home. The footprint of the stack can be regulated according to the number of optional headers, thus limiting the stack's size for simple devices.

The flexible form is the other sub-mode, administered by the Flexible Protocol Component (FPC). It includes a meta-header as well as a common header. Within flexible mode, priority-based QoS [5] provides reliability and ordering for important flows as they traverse the RG. The common header of simple and flexible sub-modes is shown in Table II. The definition of each header field is as follows:

- **Version** is 4 bits in length, Table III. It is designed to match the version field of the IP protocol. It also indicates whether combined or split mode is present in the packet. The high order 2 bits represent the version field; the current version of the HNP is 0. The 2 lower order bits indicate the mode field.
- **Header compress** is a flag that indicates whether the optional header is compressed.
- **Address length** is two bits in length. It determines the size of the address. The bitmap settings are shown as Table IV. The 88-bit source address is designed to match IPv6.
- **QoS bit** indicates if priority-based QoS is in use.
- **Source Address** is a variable-length field, with its length determined by 2 bits in the address length.
- **Destination Address** is similar to the source field.
- **Function ID** is similar to port number in TCP/IP protocol, but only employs one byte. The first 4 bits are for the source port number and last 4 bit are for the destination port number.

Example header fields are considered in [10] and in this paper, we simply assume examples taken from the work of

Solis and Obraczka [10]. Checksums may be in use and in which case a bit indicates that there is a checksum header field to follow the optional header. Fragmentation might be in use and in which case a 'more fragments' bit and a fragment offset header field bit will be required.

In split mode, HNP only consists of a protocol field followed by the common header. The HNP protocol field indicates the next level protocol used in the message. This is mainly used for devices that need to communicate with hosts on the Internet and do not understand HNP. It is also very useful for porting existing TCP/IP applications to HNP.

Data can be transmitted in either connection-oriented or connection-less fashion. As all devices implement the network layer part of the HPN protocol stack, data can be transmitted from one device to another device by using only the network layer protocol stack without establishing a connection. Devices that do not need a flow-control mechanism usually use connectionless transmission. The SPC is designed for this purpose. On the other hand, some devices request a flow control mechanism to be implemented within the transport layer part of the protocol stack. These devices need to establish a connection before they can send data. This is because not all devices implement the transport layer protocol part of the stack. HNP within the FPC must use a three-way handshake method to check whether the transport layer part of the stack is implemented at the destination.

TABLE I. PACKET FORMAT OF THE SIMPLE SUB-MODE OF THE HNP COMBINED MODE

Field:	Common header	Payload length	Function id	Payload
Length:	24 bits	8 bits	8 bits	255 bytes max.

TABLE II. COMMON HEADER FIELDS ACCORDING TO BIT POSITION.

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Version				Header compress	Address length		QoS bit
Source Address (4-11 bytes)							
Destination Address (4-11 bytes)							

Refer to Table IV for source and destination address lengths.

TABLE III. VERSION FIELD VALUES (1,2,3 LOW-ORDER BITS, 4 & 6 HIGH ORDER BITS)

Value	**Function**
1	HNP -Combined mode without meta header
2	HNP -Combined mode with meta header
3	HNP -Split mode
4	Internet Protocol version 4
6	Internet Protocol version 6

TABLE IV. ADDRESS LENGTH CODING VALUES

Value	**Length**
0	32 bit source and 32 bit destination
1	32 bit source and 88 bit destination
2	88 bit source and 32 bit destination
3	88 bit source and 88 bit destination

VI. CONCLUSION

With the advent of wireless, sensor networks and mobile devices in general, the easy homogeneity presented by the IP world is being undermined from the edges. In the home, power line, cable and phone networks are all possible. And here it is especially difficult to justify the need for complex TCP/IP stacks, except for reasons of commercial convenience. Routing is hardly required within the home context, except to make external connections. Therefore, a device-centric addressing system is specified in the HNP. Device addressing allows location-aware broadcast and anycast to occur. As others have perceived, a variety of physical networks and their respective MAC methods requires a means of varying the protocol components deployed. This is performed in HNP in various ways. The active components present on each networking or application device, including the RG, vary. While a common header is included in all HNP communication, the header is either simplified or a variety of application controllable header fields can be included. The whole is an example of the flexible structure that will be required for HNs, spatially small networks but actually containing more complexity than the massive monolithic computer networks that now dominate. Work has commenced on comparative simulation of HN QoS scenarios to evaluate the HNP. Evaluation can also take place within the pioneering intelligent home environment at the University of Essex, where this work is based.

REFERENCES

[1] K. C. Almeroth, K. Obraczca, and D. de Lucia, "A Lightweight Protocol for Interconnecting Heterogeneous Devices in Dynamic Environments", IEEE Int. Conf. on Multimedia Comp. and Systems, pp. 420-425, 1999.

[2] M. Nakagawa, H. Zhang, and H. Sato, "Ubiquitous Homelinks Based on IEEE 1394 and Ultra Wideband Solutions", IEEE Comms., 41(4): 74-82, 2003.

[3] Home Phoneline Networking Alliance, "HomePNA 3.0 Specification, ITU Recommendation G.9954", 2005.

[4] D. Strassburg, "Home-Automation buses: Protocols Really Hit Home", EDN, 40(8): 69-80, 1995.

[5] G. T. Edens, "Home Networking and the CableHome Project at CableLabs", IEEE Communications, 39(6): 112-121, 2001.

[6] A. E. Nikolaidis et al., "Management Traffic in Emerging Remote Configuration Mechanisms for Residential Gateways and Home Devices", IEEE Communications, 43(5):154-162, 2005.

[7] V. R. Gokulrangan, "Internetworking Using IPv6 Technology Inside and Outside the Home", Intel Technology Journal, 6(4): 69-77, 2002.

[8] J. Lilley, J. Yang, H. Balakrishnan, and S. Seshan, "A Unified Header Compression Framework for Low-Bandwidth Links", Int. Conf. on Mobile Computing and Networking, pp. 131-142, 2000.

[9] R. M. Sanders and A.C. Weaver, "The Xpress Transfer Protocol—A Tutorial", Computer Communication Review, 20(5): 67-80, 1990.

[10] I. Solis and K. Obraczka, "FLIP: A Flexible Interconnection Protocol for Heterogeneous Internetworking", Mobile Networks and Applications., 9, 347-361, 2004.

[11] J. Crowcroft et al., "Plutarch: An Argument for Network Pluralism", ACM SIGCOMM Future Directions in Network Architecture, 2003.

[12] W.K. Edwards, Core Jini, Prentice Hall, Upper Saddle River, NJ, 1999.

[13] A. Wils, F. Matthis, Y. Berbers, T. Holvoet, and K. De Vlamink, "Device Discovery via Residential Gateways", IEEE Trans. on Consumer Electronics, 48(3):478-483, 2002.