2

Novel Wireless Communication Protocol for e-Health Applications

A. Zvikhachevskaya and L. Mihaylova
School of Computing and Communications, InfoLab21
Lancaster University
UK

1. Introduction

Evolution from wired to wireless communication systems has brought great advantages to healthcare services. Mobility support function for e-Health applications gives practitioners, medical centres, and hospitals new tools for managing patients' care, electronic records, and medical billing to ultimately enable patients to have a higher control of their own well being. E-Health and health care services are information based, hence better utilisation of information has the potential to make services more integrated, can enhance patient safety and accountability. These will have a positive impact and will increase patient's acceptance of the services. In order to make e-Health applications more integrated and acceptable for the users it is needed to improve their efficiency. All the above motivated us to research within area of wireless standards and their interconnectivity in order to provide efficient, reliable, and robust service and eliminate connectivity boundaries for e-Health applications.
In this chapter, focus is on the development and investigation of novel technologies which would allow efficient and reliable healthcare by utilising the latest wireless technologies. More specifically, research methodology and ideas, which consider the use of wireless broadband systems, commercial (such as WiFi, WiMAX) and military (such as HIDL, Link 11), in real-life healthcare scenarios are proposed and studied.

2. E-Health and Emergency Services applications interconnection

The healthcare industry includes many services, emergency services are among them. During emergency situations communication channels may suffer congestion, errors, call dropping and data loss. In contrary to commercial mobile networks the mobile network technologies for the emergency e-Health services have to be able to provide better connectivity due to sensitivity of the medical applications to data loss, corruption or delay and are expected to provide vital aid for patients.
The most common characteristic of emergency situation is mobility of involved elements (people, devices, etc) and requirements for real-time applications running over the e-Health network have to have stringent requirements in terms of delay, bandwidth, packet loss, jitter and other QoS parameters (Istepanian, et.al., 2009).
ESs must use the most reliable personal safety applications and communication channels. If military services are involved in the emergency case then they will use the military data-links

(www.synthesis.co.uk, 2006) (such as Link11, Link16 and HIDL). Link 11 (www.lm-isgs.co.uk, 2010) is a broadcast digital communications system that was designed for use over UHF or HF frequencies to exchange tactical information between units such as ships, helicopters and submarines. Link 16 (www.lm-isgs.co.uk, 2010) is a tactical data-link that provides a bigger data-rate capability than Link 11 and a more sophisticated network management system. It was designed to meet the different communications needs and a role of units within the emergency places e.g. aircraft, ships, control centres, command posts, and reconnaissance vehicles. While technically Link 16 is the messaging standard that flows over the network, for the purposes of this research it is referred to Link 16 as the data-link system as a whole (Tarter, et.al., 2008). HIDL (www.ultra-cis.com, 2010) is a command and control data-link designed for communicating with unmanned aerial vehicles and distributing situational awareness information to active and passive participants on the ground.

Interoperability between these forces is very difficult, resulting in less than optimal efficiency and effectiveness. As was shown in some well known cases (such as the 9/11 events), this lack of interoperability was the direct cause of significant loss of lives of first responders and of civilians on site.

3. System boundaries and equipment

As we are defining A New Protocol as being a method of transferring digital data from one network to another it is very hard to draw simple system boundaries. There are two main pieces: 'cross-over' nodes and terminal equipment.

The 'cross-over' nodes can be easily represented as 'black boxes' into which the terminal interfaces from multiple data-links are connected. They read the information coming out of one terminal and repackage it into a format that another terminal understands and passes it onto that other terminal. It is possible to think of this device as an operator who reads a message coming in on one radio and typing it into the terminal for transmission on another radio.

The terminal equipment can take many forms, but in essence this is the equipment that users/applications interact with that generate or receive Protocol traffic on. For units operating on a WiMAX network this will be a computer connected via an Ethernet cable to a WiMAX modem. For Link 16 it could take the format of a box/application placed between a computer and the Link 16 terminal that converts the user data generated into Link 16 compatible messages that are sent into the terminal.

Essentially they are theoretical 'bolt on' pieces of equipment that interface with the existing equipment and create this 'network-of-networks'. It should be noted that this research does not address how these 'black boxes' might be designed, manufactured or installed.

4. Data link introduction

4.1 Introduction and types of data

In order for information to be effectively communicated between two users, they must 'speak the same language'. In computing these formats are for the most part already pre-defined; video as MPEG-2/H.264/MPEG-4 (Marpe, et.al., 2006; Chiariglione, 2000), audio as MP3/WMV/AAC (Chandraiah&Domer, 2005; www.microsoft.com, 2010; www.arm.com, 2003), text as ASCII/RTF/WORD (www.asciitable.com, 2010; www.microsoft.com, 1999), etc. Computer networking has also defined protocols for transferring these formats, the ones typically used are the Internet Protocol suite e.g. Internet Protocol version 4 and 6 (IPv4, IPv6), TCP, UDP, RTP.

IPv4 is the most common network layer protocol and uses a 20 byte header for all its packets. While this works for networks such as Ethernet which can communicate packets up to 1500 bytes long, it will not work for networks such as Link 11 which is only capable of sending 6 byte packets. The computers/people generating the information do not know or think about the transmission method or protocol that is used to exchange the information only that they are able to reproduce the source data at the destination. There may be some requirements on the data such as priority, latency or data-rate, but as long as the communications medium is able to support this it does not matter how the information is transported. For the purposes of e-Health service all user data could be arranged into the three categories: Real time traffic, Priority and Best Effort.
While there may be more subcategories that these types of traffic can be divided into for the purposes of this research only these should be addressed. Real time traffic (such as audio or video) has low latency and minimum data rate requirements. If the latency increases or the data rate decreases too much then the information becomes unusable. Priority traffic (such as situational awareness updates) is typically of fixed size and has low latency, high guarantee requirements. Finally best effort traffic (such as email or file transfer) does not have any specific quality of service requirements. Therefore for each data link not only description of how to transfer digital user data is important but also how to try and provide quality of service requirements.
This subsection briefly outlines the characteristics of each data-link and its operation, including the message formats. In the next subsections an explanation is given on why and how to transport digital user data over the various data links. After explaining how each data link works and how to implement a network management system capable of supporting e-Health 'network-of-networks' the translation of information between each network will be provided and ensure compatibility on such matters as addressing and quality of service by creating an overarching network management system (NMS) separate from the individual NMSs on each network .

4.1.1 Internet protocol version 4

IPv4 is presented here before the data links as it is the worldwide standard for packetising digital user data and the message format for exchanging information not only on the Internet but also on WiFi, WiMAX, and HIDL. This means that it is the defacto message format that most PCs, routers and common terminal equipment, that will be connecting to 'network-of-networks', will be applied. Therefore this research is using it as the message format against which all of the others employ will have to be compatible with, i.e. a packet being generated in another network will have to be able to be readdressed as an IPv4 packet and vice-versa (Almguist, 1992).
IPv4 is a network layer protocol, which means it provides a mechanism for source to destination packet delivery. This includes addressing, routing, quality of service and error control. An IPv4 packet consists of a common 20 byte header and a data portion. The header includes information such as a source and destination address, a checksum and details of the underlying packet, packet length, if it has been fragmented, what type of traffic it is etc. IPv4 is being slowly phased out over the Internet in favour of IPv6. IPv6 amongst many other features has a larger address space, more features for prioritization and gives a simplified interface for processing by routers. These features are aimed primarily at large networks, which handle large amounts of traffic at high data rates, these difficulties will not be encountered in this research and thus only IPv4 will be used. This is deemed sufficient as it is possible to translate between IPv6 and IPv4 using well known techniques.

Note that an IP network does not guarantee that packets received at a destination will be received in the same sequence they were sent. It is the responsibility of the transport layer (for those transport layers that do guarantee data order such as TCP) or the application layer (if it is using a datagram protocol such as UDP) to handle mis-ordered packets.

Addressing

The pivotal role of IPv4 is that it provides a standard method of addressing which is used throughout the Internet. In fact, without it, the Internet would probably not exist as we know it today. IPv4 addressing is very similar to postal addressing; everyone has a house number, a street, a city and a country. The only difference in IP is that the information is ordered differently, an IPv4 address consists of 4 bytes which are typically written as AAA.BBB.CCC.DDD with the A's in essence denotes the country, B the city, C the street and D the house number. This subdivision of the address into 4 'octets' allows the Internet to be broken down into lots of networks of networks to facilitate with routing. Simplistically: two computers with the same A, B and C numbers will be on the same small local network, two computers with the same A and B but different C numbers will be in the same larger wide area network but different local networks, and finally two computers with just the same A numbers will probably be in the same country but on physically separated networks. Routers within this 'network-of-networks' can use subnet masks therefore to decide if they need to route a packet internal or external to the network. These subnet masks determine this via checking the source and destination addresses against the mask and if they are different then the packet is for a destination external to the network and if they are the same then it is for somewhere internal to the network. For example a typical IPv4 source address might be 192.168.20.5 and a destination address 192.15.34.140, if the router operates a subnet mask of 255.255.0.0 then the router will compare the first and second octets and if they are the same then route the packet within the network, but if they are different (as in this case) the packet is routed to the external gateway and to the correct network. The octets matching the subnet mask are referred to as the Network ID and the rest of the octets are the Host ID, in the example above the source address has a network ID of 192.168 and a host ID of 20.5.
We will return to this notion of IP addresses and subnet masks later, as a mechanism for subdividing the 'network-of-networks' and thus addressing packets between different data link networks.

Header

The IPv4 header is outlined below in the table 4.1 below.

+	Bits 0–3	4–7	8–15	16–18	19–31
0	Version	Header length	Type of Service	Total Length	
32	Identification			Flags	Fragment Offset
64	Time to Live		Protocol	Header Checksum	
96	Source Address				
128	Destination Address				
160	Options				
160 or 192+	User Data				

Table 1. IPv4 Header

A quick explanation of each field is given below:
Version: This is a fixed value denoting IPv4;
Header Length: This will always be 20 for headers with no optional additions;
Type of Service: This is used to denote any quality of service requirements;
Total Length: This gives the total length of the packet – header + user data;
Identification: This gives a unique identification field and is used in fragmentation;
Flags: These denote settings for fragmentation;
Fragment Offset: Used to reconstruct a fragmented packet;
Time to Live: Gives the number of hops the packet can take from source to destination before it is dropped by the network;
Protocol: Tells the receiver the format of the user data portion is e.g. TCP/UDP/SCTP/OSPF;
Header Checksum: A checksum making sure the header is correct – note it does not protect the user data portion in any way;
Source Address: The IPv4 address of the sending computer;
Destination Address: The IPv4 address of the destination computer;
Options: This field is very rarely used, but some protocols use it to provide more information.
If a piece of information regarding the packet can be inferred without the need of the header then that information is redundant. Thus as we will see later, if we make some assumptions regarding the traffic going over the network then we limit the amount of header information we need to translate between networks.

4.1.2 Description of the military data-links

High Integrity Data Link (HIDL) Description

In Figure 1 the typical topology of the HIDL Supported Network is presented, which includes two HIDL Communities. Each of them has a timing master, Unmanned Aerial Vehicle (UAV) and a Relay terminal. Overview of the HIDL standard and characteristics of the named objects is given below in subsections below.

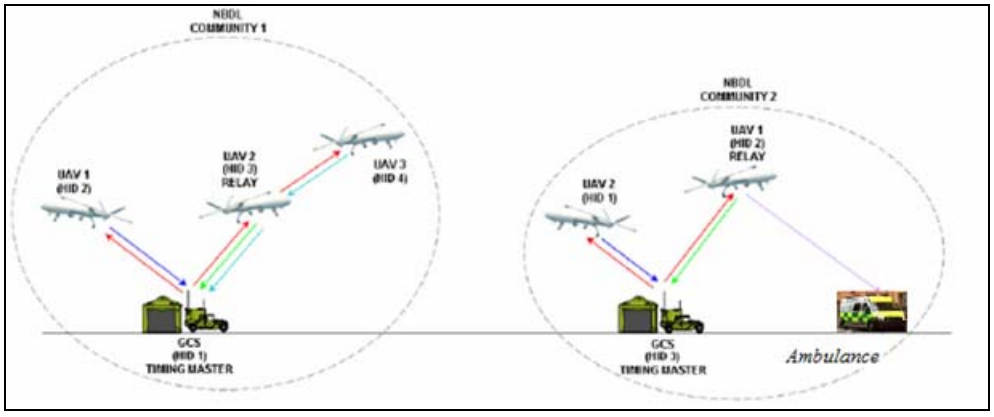

Fig. 1. Architecture of the HILD Supported Network (Tarter, et.al., 2008)

HIDL Overview

HIDL was designed to provide a near real time, high integrity data communications link between multiple nodes within an Unmanned Aerial Vehicle community. It sends command and control information from a ground station to multiple UAVs in the air. It also allows the UAVs to send information from the air to other UAVs or ground receivers.
This network can have a maximum of 5 active transmitters in the network at any one time. This effectively means 1 timing master (base station) and 4 network entrants (client units). However as explained later there can be multiple receive only passive terminals that are capable of one way communication.

Time Architecture

HIDL uses a time division mechanism to packetize the data to be transmitted, i.e. a packet of information is transmitted at a known rate (the period of the time division). The HIDL time structure divides the time domain into contiguous periods of 10ms - termed Timeslots. A group of 100 contiguous timeslots is termed an *epoch*, which is equivalent to a period of one second. These *epochs* are repeated every second, and therefore the timeslot allocation is repeated every second. It is essentially a broadcast architecture and therefore each receiver is capable of receiving every packet transmitted in an epoch as long as it is in range, and therefore while there is only ever one transmitter per timeslot there maybe multiple receivers.
As a result of this scheme multiple QoS schemes cannot be assigned to a timeslot as there is no data packet processing performed within the system, instead only bandwidth (timeslot allocation) is the only variable. Therefore voice, text and video packets are treated identically within the HIDL network; it is up to the operator to provide the required levels of network resources to meet the demands of the application. This is in contrast to Link 16 which can provide contention access as well as the dedicated access scheme which is used in HIDL. Ultimately this will mean that while real-time, priority and best effort traffic will be transported with the same level of QoS the anticipated amount of each type of traffic will be used to calculate the timeslot allocation.

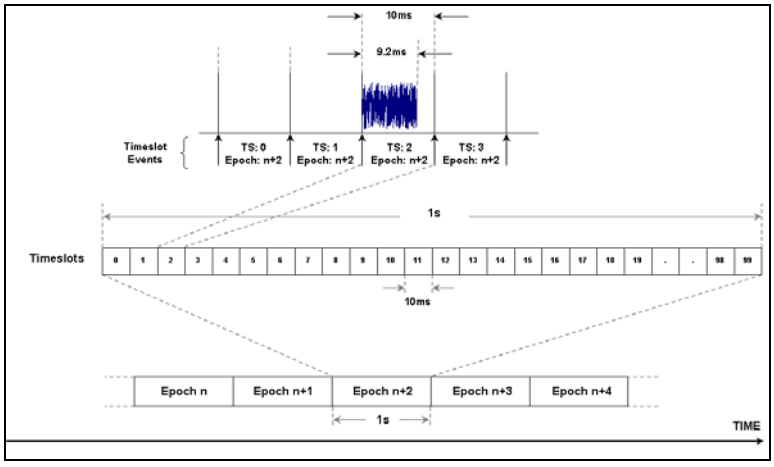

Fig. 2. The HIDL Time Architecture (Tarter, et.al., 2008)

Each timeslot in a HIDL network is assigned a 'circuit'. HIDL supports up to 15 of these 'circuits'. A 'circuit' describes the source terminal, the destination terminal(s), whether the message is to be relayed, and what the destination multicast address of the data packet in the circuit should be. As this is a broadcast radio system the list of destination terminals is really only used to filter the results (if a node is not listed as a receiver then it will not try to capture the transmission) there is no reason why they all couldn't receive the broadcast, however each circuit then need to be defined as broadcast and leave the filtering of the received packet to a higher level protocol outside of the terminal.
There are five timeslots per epoch in which no User data is allowed to be transmitted, leaving 95 timeslots per second for user data. These five timeslots are used by the control station for network management. In each user timeslot a maximum of 422 bytes of user data is allowed to be transmitted, which when Ethernet, IPv4 and UDP headers are added any Ethernet packet of up to 468 bytes can be transmitted. Of course any sized packet below this size maybe transmitted in a time slot, but only at a rate of one packet per timeslot. This gives a theoretical throughput of 355.7Kbps.
To communicate or receive data, each node must synchronise itself in time with a timing master (typically the ground station). This enables each transmitting node to operate within a synchronised global time structure and thus allow each receiving node within range to receive each packet transmitted collision free from the next packet.

Packet Format

Every packet must conform to UDP-IPv4 over Ethernet and be less than 468 bytes in total. HIDL is a very simple radio network that operates by distributing UDP/IP packets over the air. Each packet being sent must conform to UDP-IPv4 over Ethernet and be less than 468 bytes in total (the maximum transmission unit of the radio). If the packet to be transmitted is in a different format or too large (e.g. a TCP packet of 1000 bytes), then it must be fragmented and wrapped in a UDP frame and unwrapped and recreated at the other end.

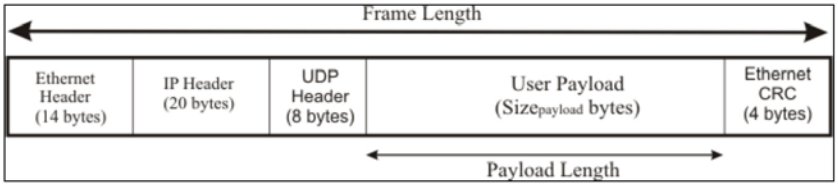

Fig. 3. The HIDL Packet Format

A HIDL terminal accepts user data packets over its Ethernet interface. The terminal recognises the associated circuit for the data via the destination IP address and puts it in the correct buffer. When a timeslot comes around that is allocated to that circuit the user data packet is read from the buffer and sent over the air. The receivers capture the packet and each one outputs it over its Ethernet interface. All circuits use multicast IP addresses for their destination address, this is to overcome the limitation that the transmitter does not know the MAC address of the receiver(s) and to reduce the overhead from the network headers, maximising user data throughput As a result any packet destined for a unicast address must be wrapped in a HIDL UDP/IP multicast packet for transmission over the HIDL network. In order to send packets to different addresses the user could send the correct packet wrapped in a multicast frame and have a receiving unit do the packet

decomposition. Otherwise it could use a Network Address Translation (NAT) router that will convert the traffic to a unicast address via a Port number. While performing NAT over HIDL and 'network-of-networks' is possible, explanation of its functionality is outside the scope of this research.

Relay

HIDL provides the ability for one terminal within the network to act as a relay for other terminals too far away from the source terminal to hear its communication. As all terminals are part of the same network it is up to the network manager to ensure that there are sufficient resources (timeslots) for the relay terminal to pass on any messages destined for terminals out of range of the transmitter. However if there are not enough timeslots available to the relay to pass on the packets within an epoch some packets will get dropped.
The relay unit also provides time synchronization for the nodes out of range of the ground control station, thereby ensuring that all nodes throughout the extended network are operating on the same global time structure.

Receive only units

HIDL allows for portable units to be used in receive only mode, which means that they are capable of receiving all of the messages communicated throughout the network but unable to respond. In an operational environment it is envisaged that there will be multiple ground units with these receive only terminals. This therefore means that when these ground terminals are networked to other networks as part of a larger system there will be more ways of communicating in one direction than the other.

HIDL Network Management System

Each network entrant must first communicate with the timing master in order to fully synchronise itself prior to any node-to-node communication. This process provides a registration mechanism that the network manager can use to ascertain which terminals are actively participating. The five network management timeslots already provide each client with a list of those active client nodes within the network and what their addresses are. This enables all active and passive nodes in the network to continuously have an up to date list of all active participants in the network (obviously the passive nodes are not able to declare their existence).
Resource allocation (time slots) are managed and allocated by the timing master (control station) and are fixed for the duration, unless the timing master issues a new timeslot assignment. This means that any node requiring more bandwidth will have to send a request to the network manager at the base station who will modify the timeslot allocation scheme and issue a new one.
There is no defined protocol inherent within HIDL to accomplish a change in timeslot structure, this must be done by sending over the air data messages to the controlling computer at the timing master who will then provide the timing master HIDL unit with a new timeslot allocation and instruct it to distribute it to all the nodes who will then adopt it. As these messages go over the data interface they must be compatible with the formats of messages being used for 'network-of-networks' traffic over HIDL and be identifiable to the timing master control computer that it is a resource request message. It is the recommendation of this research project to not use a separate or unique message structure for identifying these packets, but instead use a pre-existing mechanism such as UDP port numbers for identification. As long as the length of the packet is less than the maximum

value that can be transmitted in one timeslot it does not matter how big the packet is, as only one packet can be transmitted in any one timeslot regardless of size.
It is also proposed that all circuits denoted for use by 'network-of-networks' compatible terminals be set to the broadcast mode, meaning that all packets transmitted by a 'network-of-networks' HIDL terminal will be received by all of the other 'network-of-networks' HIDL terminals, it will be up to each destination computer/router to decide whether or not to forward or drop the packet. There are two possible methods of implementing 'network-of-networks' over HIDL with regards to resource allocation. The first involves allocating only one circuit to each HIDL terminal for 'network-of-networks' traffic. The second involves allocating 'cross-over' nodes two circuits; the first is used to carry traffic internal to the network and the second for traffic destined for outside the network (effectively 'cross-over' to 'cross-over' communication). The second method will provide the network manager computer with more information that it can use to allocate the timeslots and balance the amount of network-to-network traffic against internal traffic. Discovery of the most effective method and resource allocation algorithm will be investigated in simulation.

HIDL Node Attrition Strategy

The HIDL network is very similar in format to a WiFi network: it requires a central base station to provide timing and network management but individual client units can talk to each other. All HIDL radio equipment is identical whether the node is to be a timing master, an active node, a relay or a passive node; therefore any node can be chosen to perform the timing master's role. It is advisable to choose a node within range of all other terminals, so as to allow synchronisation. If a node is too far away but covered by a relay node then the relay node must be in the range of the timing master. As any node can take on this role of timing master it is proposed to use the same recovery process as was outlined above in this section. Although the given scheme, will provide the ability for timing master to take over it should be noted that HIDL was designed to be a UAV Command and Control data link. As such nodes could lose contact with the timing master as a result of their location rather than the loss of the timing master. If a node falls out of link there are mechanisms such as a 're-acquisition strategy' that are performed to account for this.
Therefore it is not advised that another UAV automatically assume that the timing master has been lost and adopt its functionality, instead like WiMAX (where the role of the base station is restricted to a few units) the adoption of the timing master role should only be performed by a ground unit who should be more capable of making this assessment.

Link 16 Description

Link 16 Overview

Link 16 is one of the military's Tactical Data Links, which is to say it is primarily used to communicate tactical information between units or platforms in the battle space. This research is not aiming to investigate the benefits to be obtained from changing the equipment, but rather the benefits that could be obtained by modifying the operational use of the Link 16 standard.

Packet Format

Link 16 messages can be transmitted using either Double Pulse (DP) or Single Pulse (SP) encoding. Double pulse operation sends the same symbol packet using two pulses rather than the one used for single pulse operation. This means single pulse packets can send more

data per timeslot than double pulse packets but the probability of reception is reduced. There are 4 different formats a Link 16 message can take Standard, Packed-2 SP, Packed-2 DP and Packed- 4 SP. A standard message can send 225bits/timeslot, both Packed-2 formats can send 450bits/timeslot and the Packed-4 can send 900 bits/timeslot. As there are 128 timeslots per second this gives us a data rate of 28.8, 57.6 and 115.2Kbps respectively. These numbers also depend on whether or not Error Detection Coding (EDC) is used, however this research will not be investigating their use, instead we will only use formats that do use EDC.

Each transmission in a timeslot is preceded by a Link 16 header which tells the receiver how to decode the data portion by identifying the packet format (Packed-2, Packed-4 etc), the message format (free text or fixed format), encoding (i.e. Reed Solomon) the transmitting terminal and if the message has been relayed.

There are two message formats used in Link 16: Free Text and Fixed Format. Free text messages within Link 16 do not need to follow any defined message structure; this is how voice, ASCII text and video are passed over JTIDS. Fixed format messages though need to follow the Link 16 message structure (J-Series Messages).

Access Methodologies

Link 16 operates a Time Division Multiple Access scheme (TDMA), which means that all units operating within a Link 16 network are synchronised in time and transmit and receive at predefined times.

It uses 12.8 minute epoch which is divided into 98,304 timeslots. However, this is a little unwieldy so it is broken down into 64 frames, each 12 seconds long. Each frame contains 1536 timeslots and these are used when allocating timeslots to terminals. All the timeslots in the scheme are allocated by the network manager to individual units for transmission. As all nodes know the timeslot allocation the receivers know when they should listen to receive data from any given transmitter. By increasing or decreasing a unit's timeslot allocation you are effectively changing the maximum transmission bandwidth/data-rate of the unit. Currently timeslots are first labelled according to their Network Participation Group (a mechanism for receivers to use to determine in which timeslots they need to listen) then allocated to units. This mechanism allows us to easily define a new Network Participation Group (NPG) for 'network-of-networks' network data, which will allow 'network-of-networks' to use existing hardware and maintain operational compatibility with existing systems. Those terminals not equipped to take part in the data network will not listen and will not take part in the networked data NPG and therefore will not receive any 'network-of-networks' packets and be unable to decode them. Again at the receiver, messages are output with a header defining in which NPG the packet was received in thereby allowing terminals to clearly identify 'network-of-networks' traffic from other traffic being received from the network. Users interact with Link 16 terminals by sending messages to the terminal with a header defining in which NPG the message is to be transmitted. The terminal is then left to broadcast the message in the appropriate timeslot. An interesting result of using NPGs is that the sender does not necessarily have to know who the receivers are or the route to the destination, and as it is a broadcast system, the sender can take for granted that the same timeslot allocation table has been distributed and therefore that all receivers it wants to talk to are listening in for its transmissions.

Currently Link 16 systems distribute Precise Participant Location and Identification (PPLI) messages to organise sender, receiver and route information (at least once every 12 seconds).

For data networking this concept should be utilized, although the Route Indicator Parameters do not provide enough information for this exact implementation mechanism to be used solely for 'network-of-networks' route planning. Timeslot allocation is performed via the J0.3 and J0.4 messages (TS Assignment and Radio Relay Control), these messages are used to delete assignments, add specific time slot allocations, change a terminal's operation as a relay, add or remove relay time slot allocations. When terminals receive these messages they check if the required change is valid and if so automatically inform the Network Manager that the action has been accepted, thus providing verification. Timeslots are allocated in blocks rather than as individual timeslots and a single terminal can handle up to 64 time slot blocks. These blocks define when a terminal should transmit, receive or relay some data. This places a complexity limitation on the Network Manager who must ensure that in calculating the timeslot allocation there are no more than 64 distinct blocks of timeslots (a block is a collection of timeslots that have the same parameters – e.g. type, NPG, access mode, Tx/Rx). The network manager has some flexibility over this limitation as it can describe a block's access mode as being either dedicated, contention or timeslot reallocation.
In dedicated access a timeslot is given to a single unit for transmission, this is fine when the unit always has data to send but if not then nothing is transmitted and the resource is wasted.
In contention access a block of timeslots are allocated to a number of terminals, these terminals are each given a transmission rate (a given fraction of the total number of timeslots).
The terminals are not required to transmit at this rate, but could do so if required. The terminals then use a pseudo-random function to choose the timeslots in the block that they will transmit in (up to the maximum rate granted to them). This mechanism does not guarantee the sole use of a timeslot and the likelihood of a transmission collision is a factor of the block size, number of terminals and transmission rates. Hopefully the network planning process will have reduced this probability to an acceptable maximum level.
Finally under time slot reallocation the timeslots are put together in a common pool and allocated on expected demand. At the beginning of each period terminals announce their demand using J0.7 Time Slot Reallocation messages. All other units hear these announcements and using a common algorithm in each terminal create the timeslot allocation table for the rest of the period. This allocation will not be exactly replicated across all terminals as some terminals may not have heard all of the demand announcements, even so this could still be acceptable.

Link 16 Network Management System

'Network-of-networks' proposes to use a centralised network management system for control of timeslot allocation within the network, but use a distributed scheme for allocation between 'cross-over' nodes. This means that a centralised network management system will allocate timeslots (either all of them if the system is fully automated, or the 'network-of-networks' subset if the initial allocation is done by an outside source e.g. the data links planning office) using the dedicated and contention access schemes to terminals within the network, thereby allowing current operations to continue with the minimum of impact. 'Cross-over' nodes will share a pool of timeslots which they will distribute according to the timeslot reallocation scheme. If the 'cross-over' nodes require more bandwidth than the pool is capable of supplying then they will have to negotiate with the network management system for dedicated allocation from the rest of the pool. This division of management

functions means that the local network management system has ultimate control over the balance of data over its network but leaves the routing aspects between networks up to the 'cross-over' nodes. The local network management system can increase or decrease the size of the time slot reallocation pool and therefore increase or decrease the amount of utilisation of the network for 'network to network' communications.

A network is formed initially by a terminal acting as a Network Timing Reference and broadcasting a J0.0 'Initial Entry Message', network entrants then uses these J0.0 messages to synchronise in time (typically responding with a PPLI message). Other terminals can use any other active terminal to synchronise with and thus gain access to the network. There is no requirement for registering with a network manager first. This means that an up to date list of network participants is not available intrinsically from the terminals. Instead the network manager is going to have to perform this task by requiring all 'network-of-networks' Link 16 terminals to periodically inform it of their existence. Then, the network manager will then distribute this list of the participants. In order to accommodate new terminals who have yet to be granted dedicated or reallocated slots, it is recommended that the network manager always leave some timeslots in contention mode (allocated to all terminals) for network management functionality such as registration.

For standard IP traffic within the network this project would recommend using a contention access scheme. This is because IP traffic is typically bursts and a dedicated access scheme will end up with an underutilised network. Dedicated access can be used to ensure applications such as video or audio have the required bandwidth to support their use, and should only be granted on demand.

Initially the network management system will grant: a portion of its timeslots to the 'cross-over' nodes for them to use (under the time-slot reallocation scheme), a portion of its timeslots to all of the terminals within the local network under a contention access scheme, and possibly keep a portion of timeslots in reserve for requests for dedicated access. The size of these portions and the amount of timeslots held in reserve will have to be investigated and modelled using a software simulation later in this project. Obviously as the network continues to operate, terminals will request greater contention access rates, dedicated allocations and an increased pool for network-to-network communications. The network manager will have to balance the demands for resources against the utilisation of the network, the priorities of the demands and the types of traffic being sent. The network manager will allocate the timeslots and distribute that information via the current Link 16 method of using J-series messages. This will allow compatibility with non-'network-of-networks' terminals and limit the impact on continued operations.

Finally the reason for using a centralised network manager as opposed to an entirely distributed timeslot reallocation scheme is one of security and robustness. While any terminal can become the network manager and can perform its duties, completely distributing the functionality increases the risk that a mis-used terminal or spoof messaging can disrupt the consistent timeslot allocation table algorithm and thus can heavily impact the network operations

Link 16 Node Attrition Strategy

As in WiFi and HIDL any terminal can act as the Network Timing Reference (NTR) and thus take on the role of the network manager. Therefore a scheme of recovery due to node attrition similar to that outlined in 3.4 for WiFi could be utilised if another scheme has not already been outlined. Such a scheme with attrition nodes has been successfully deployed

for military purposes, including a backup strategy in place that is initiated in the case of loss of a node (especially the NTR). This involves choosing the node with the closest time synchronisation to the original timing master. This thesis proposes a strategy that requires compatibility with any node that could possibly perform as the NTR be 'network-of-networks' compatible. If such an operational strategy is not required or an automatic one is required instead then as each terminal should send a PPLI message at least once every 12 seconds (frequency depends on timeslot allocation) we could use the 12 seconds frequency as a reference value. If the network manager/network timing reference does not transmit a PPLI or Initial Entry message after 48 seconds then it will be deemed to have been lost. The next terminal in the sequence should then take over. In doing so the epoch will have to begin again and units will have to renegotiate with the network manager for timeslot allocations. The reason for the renegotiation is that each node will not have the complete list of timeslot allocations, instead as explained above each node is only notified of its assignment in up to 64 blocks.

Link-11 Description

Link-11 Overview

Link 11 was the precursor to Link 16, and while its operational use is similar to that of Link 16 its technical characteristics and network operation are very different. In essence it operates very similarly to a token ring network. Nodes within the network wait until they are called upon by the Network Control Station to broadcast at which point they begin broadcasting until they have finished, at which point the Network Control Station then calls upon another node. This Roll Call mechanism is controlled by the Network Control Station and it is this NCS that controls the sequence of node transmissions. There are three methods of controlling the roll call:

- Full Roll Call – all nodes are active and are called on one by one;
- Partial Roll Call – some nodes are in Radio Silence and thus do not respond to the NCS;
- Roll Call Broadcast – the NCS broadcasts all data, and any node with new information informs the NCS of this, which the NCS then broadcasts to the rest of the network.

As we will be passing network data rather than tactical information, such as enemy/friendly positions this research does not recommend Roll Call Broadcast, instead it is proposed to use the Full Roll Call method.

This method, however, is not conducive to real-time traffic as there is no way to determine exactly when is the next time a node may be allowed to transmit (even if there is a maximum transmit window). As the information passed via Link 11 has traditionally been of use to everybody (battlefield situational awareness information) having each node transmit all of its information before releasing transmit token was acceptable. However, as we are transmitting information that might not be of use to everyone within the network this method does not seem prudent. Especially as one node that transmits a lot of data will end up monopolising the network resource. Instead one proposed method is for all 'network-of-networks' terminals within the Link 11 network to operate on a two cycle roll call. During the first roll call each terminal transmits its requirements (amount and type of data) and the network manager coordinates this information and at the end of the 1st cycle broadcasts the amount of data each terminal is allowed to transmit, each terminal then when called upon, during the second cycle, only transmits the amount of data that the network manager has decided upon.

This mechanism relies of all 'network-of-networks' terminals abiding by the allocation granted it by the network manager.
Another method may be to fix the number of Link 11 packets that each 'network-of-networks' terminal is allowed to transmit at once. This means that if a message is longer than the number of Link 11 packets a terminal can transmit at once, then it will have to wait until its turn comes round again before it may continue. In order for this method to work then each receiving terminal will have to know who has transmitted each terminal.

Link 11 Packet Format

Link 11 messages conform to the M-series messages, there is no mechanism for free text as there is with Link 16 and as such 'network-of-networks' terminals will have to conform to the M-series format in order to maintain compatibility with ongoing operations.
M-series packets are divided into two 30-bit messages, with 6 bits each used for error correction, thereby leaving 48 bits in total for the information portion. All M-series messages use the first 4 bits of the first message to denote which message type is being sent. These 4 bits are called the message number and provide for 16 different types of message. Message type 12 can be used by nations for individual systems such as 'network-of-networks'. Messages are subdivided again using a label suffix, which again is 4 bits long; in this case we propose to use M12.14.
Once the message designation has been given the rest of the 40 bits can be used for the actual information. The original use of the data-link is to pass information of use to everyone and as such there is no header field for destination – all transmissions are broadcast in essence. As can be inferred the size and nature of this data-link are orders of magnitude different to WiMAX, and thus careful consideration will have to be made in how to pass information over Link 16 networks.

Link 11 Network Management Strategy

It is proposed to use the link's Network Control Station (NCS) as the Network Manager, the NCS will either determine the maximum number of packets the terminals can send at once or collect in all of the transmission requests from each 'network-of-networks' compatible Link 11 terminal and decide on the maximum number for each terminal in the next cycle.
The NCS can operate either in Net Synchronisation or Roll Call mode. In Net Sync mode the NCS calls upon each terminal in turn to transmit and receive and thus achieve synchronisation in time with it. After network sync has been achieved the NCS moves into the normal Roll Call mode. In this mode, when the NCS polls a 'network-of-networks' terminal, which has nothing to transmit, it should answer with a zero requirement response; this will allow the NCS to determine if the node is still active and allow the NCS to skip it in the 2nd cycle. The NCS should only have to perform Net Sync at network initialisation or on command from a user; there is no automatic mechanism for a new node to register with the terminal without a user first informing the NCS that such a terminal exists and to include it in the polling loop. It is not proposed to circumvent this operation but instead to utilise it, therefore within 'network-of-networks' if a terminal wishes to join the network it must first be added manually at the NCS by an operator.

Link 11 Node Attrition Strategy

As with Link 16 a current operational Link 11 network will have a backup strategy in place that will be used in the event of the loss of a node (especially the NCS). Again it is proposed

not to usurp such a strategy if one is in place. However, if an automatic solution is required the following mechanism could be used.
As this is a roll call network, where each terminal may transmit until it is finished there is not a deterministic frequency to the NCS's transmissions and thus any fixed time between control station transmissions.

5. Architecture and communication protocol of the 'Network-of-Networks'

The 'network-of-networks' is a utility that allows users of any data link network to communicate with users of any other data link network that is within the 'network-of-networks' umbrella. The information generated in one network is able to traverse the inter-networks and be consumed at any destination. This can be especially problematic if two data-link networks are fundamentally different such as HIDL and WiMAX. Each network has a different maximum packet size, data rate and network management scheme.
For instance how will Link 11 address and transport IPv4 traffic, generated in a WiMAX network or HIDL provide the QoS required for an audio stream originating in a Link 16 network? As explained in the introduction, one problem is how to label the packets so that the information is correctly routed to the destination wherever it might be, and in such a way as to enable the recipient computer to accurately reconstruct the data and communicate back in reply. To resolve this problem we propose to create a header for each data-link network so that a packet can be translated into a format compatible with any other data-link network. The conventional method is based on creating a single 'network-of-networks' header that would first wrap any packet before the data-link header (allowing compatibility with the data-link). However, as a result some information (such as a destination address) would ultimately appear twice, which for a data link (e.g. with short packages, like Link 11) might be an unreasonable amount of overhead. A new method is proposed in this dissertation which provides a translation service such that the information in a header for one network could be used to create the header for the other network. This approach would minimise the amount of duplicated network overhead information within each packet and allow packets to be re-formatted into something appropriate for that given data-link. For instance fragment a packet into smaller sizes for transmission over Link 11, but combined into a larger single packet for transmission over WiMAX.
While the header information will be useful to the 'cross-over' nodes (who will be performing the translation service) it is also required to send packetised digital data between two nodes of the same network.

Addressing

One of the main issues within 'network-of-networks' is how to address a packet of information such that it will identify a destination that may be on a different network entirely. On traditional computer networks this is done by an IP address and explained in the section 4.1.1.
It is proposed to utilise the same first two octets for each computer within the 'network-of-networks' thereby reducing the number of bits required to identify a single destination. This means that the address of each computer will only differ in the last two octets of an IPv4 address. We will use the first of these octets to identify the implementation of the data-link network that the computer resides on and the last octet to identify the computer. This ultimately means that by using this mechanism we are limited to 256 networks and 256

computers within each network or 65,536 computers in total. Of course Network Address Translation mechanisms can be employed to increase these numbers but such a technique is outside the scope of this investigation.

Figure 4 presents an example of the 'network-of-networks' where four individual data-link networks are joined together using five 'cross-over' nodes. In order to guarantee efficient communication between the users we apply the proposed translation algorithm, which is demonstrated by the following example. An Example: each computer can be addressed by the addition of two numbers, the data-link address (number in red) with the node address (number in black). Note that a 'cross-over' node has at least two addresses – one for each data link network (in this example each 'cross-over' node has the same lower octet address, though this needs not be the case. In this example if "4.3" wants to talk to "2.2" then it might send the message via "4.2" – "3.6" – "2.2". Note also that where there are two 'cross-over' nodes sharing the same data links there are two ways of communicating, one via each data link.

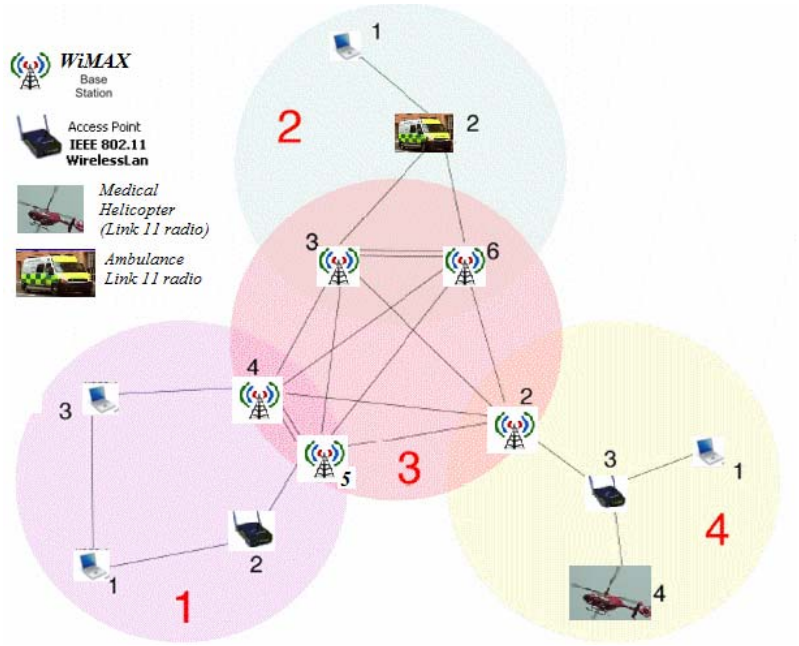

Fig. 4. Four Individual Networks Joined Together into the 'Network-of-Networks'

One of the most useful features of the IP protocol suite is the use of 'multicasting' and 'broadcasting' addressing. Using these techniques a transmitter can send a single packet that will reach multiple destinations thereby reducing the total number of packets sent.

The diversity of the developed algorithm can be another example. 'Broadcast' packets are those addresses using the '255' (or all 1's in binary) as the destination address, e.g. 172.20.255.255. With this designation any node with the same prefix before the 255s will receive the packet (e.g. all nodes with 172.20 as the leading octets of its IP address will receive a packet addressed to 172.20.255.255). A packet sent to 255.255.255.255 is a special case destined only for the local subnet (termed a limited broadcast) and will not be forwarded. This

forwarding of broadcast addresses conforms to that in outlined (Baker, 1995) and an option to prevent broadcast forwarding would be available in 'cross-over' nodes.
Multicast packets are similar to broadcast packets although they use a destination address with the first octet in the range 224 to 239. Nodes wishing to receive these packets send out requests to their routers to forward those packets onto them, who in turn pass the requests back to the sender's router, a router will then only pass on one packet for every common path to the destinations. For instance using the network above if node 1.1 is producing a multicast stream that 2.2, 4.1 and 4.4 want to receive then the following packet streams might be produced:

- 1.1 -> 1.2 -> 1.5
- Two packets are produced: 1.5 -> 3.6, 1.5 -> 3.2
- 3.6 -> 2.2
- 3.2 -> to 4.3
- Two packets are produced: 4.3 -> 4.1, 4.3 -> 4.4

The following conventions are proposed to be followed in the simulation. Firstly broadcast packets will be routed as normal using a node address of 255 to denote a broadcast packet for a given data-link network, and a data-link address of 255 with node address 255 to denote a broadcast packet to all nodes within the 'network-of-networks'. Secondly multicast routing will be done using a data-link address of 224 to 239. Due to 'network-of-networks' header compression constraints (only two byte of IP address supported), the first two bytes of a multi cast address will be repeated in the second two bytes in a 'network-of-networks' (e.g. 224.12.224.12), ensuring the address propagation across the network. The use of reserved subnets apart from theses (e.g. private, APIPA) is not recommended but it is allowed.
This will mean that the total number of data-link networks in the 'network-of-networks' at any one time will be 256 – 15 (number of multicast addresses) – 2 (0 and 255 reserved) = 239. In order to receive the multicast packet a node will have to register its request with a 'cross-over' node.

Header Design for the 'Network-of-Networks'

In section 4.1.1 the IPv4 header is introduced, which is the predominate way of addressing packetised digital data within a computer network, and as such we need to ensure that any header that we create is cross compatible with it and that we are always able to regenerate such a header.
If we take as an assumption that we are only ever going to transport IPv4 and not IPv6 traffic then most of the IPv4 header becomes redundant. A further reduction can be made if we assume that only a few types of protocols will be transported over 'network-of-networks' e.g. TCP, UDP and routing. This means that we can use a reduced protocol field and save space. Below is the list of IPv4 header fields and a description regarding their applicability to 'network-of-networks':

- **Version:** This is a fixed value for IPv4 and therefore can be inferred;
- **Header Length:** This will always be 20 (assuming the use of no protocols with optional headers) and therefore can be inferred;
- **Type of Service**: This gives QoS requirements which will be required;
- **Total Length:** This is the total length of the datagram and can be calculated;
- **Identification**: This provides a unique ID for fragmented datagrams and will be required for IP fragments;

- **Flags:** These are used for fragmentation and in some instances can be inferred;
- **Fragment Offset**: Used to reconstruct a fragmented packet and will be required;
- **Time to Live**: Gives the number of hops the packet can take from source to destination before it is dropped by the network this can be determined by the 'cross-over' nodes;
- **Protocol**: Tells the receiver the underlying protocol which will be required;
- **Header Checksum:** A checksum for the header which can be calculated. Any transmission errors will be detected by the link layer checksum;
- **Source Address**: The IPv4 address of the sending computer which will be required;
- **Destination Address**: The IPv4 address of the destination computer which will be required;
- **Options:** This field is very rarely used and it is assumed that it is not required.

The bold fields are either required in some way or cannot be inferred about the packet, therefore any 'network-of-networks' header for any data-link must include these fields in some way to allow the (re)construction of an IPv4 header for the packet.

Quality of Service

All traffic is divided within the 'network-of-networks' network into three types:

- Real Time,
- Priority,
- Best Effort.

The Type of Service (ToS) field within IPv4 is divided into two sections: the precedence (priority) and the service type. The first three bits denote the importance of the packet and the last three bits denote low delay, high throughput and high reliability respectively. As many of the data-links provide no mechanism to affect the reliability of a packet's transmission, low delay is implicit for real time traffic and the level of throughput will be dictated by the network managers it is proposed to use 3 bits to denote QoS.

Bit 0 and 1: Denote the priority of the packet: 0 being lowest priority, 3 highest (which map to bits 1 and 2 of the ToS field)

Bit 2: 0 Indicates best effort traffic, 1 indicates real-time traffic (which maps to bit 3 of the ToS field).

Identification

Providing a unique identification number for each IP datagram will allow IP fragments to be re-assembled (as it provides a common label for all fragments). When forwarding fragmented IP packets, this *identification* field will need to be included in the compressed IP header.

Fragmentation

IPv4 datagrams are allowed to be up to 65,535 bytes long according to the standard. This is a theoretical limit; however, in the case of many computer networks as the Ethernet, WiFi and WiMAX limits are around 1500 bytes (including Ethernet headers etc). Therefore it is assumed for this simulation that there won't be any single IPv4 packets larger than 1500 bytes to begin with.

'Cross-over' nodes act like IP routers and hence would normally be required to fragment incoming IP datagrams if their length exceeds that of the network they about to traverse (Baker, 1995). However the minimum recommended MTU for IPv4 is set at 68 bytes (www.ietf.org, 1981).

Both Link 11 and Link 16 use much smaller packet sizes that this, so it is intended to fragment and re-assemble IP packets traversing these networks at the data link layer (layer 2) rather than using layer 3 (IP) fragmentation, which relies on the IP destination host to reassemble the IP fragments. A separate layer 2 fragmentation header will be defined where required for each of these 'network-of-networks' data link types. Both Link 11 and Link 16 will maintain packet order, so layer 2 fragment numbering will not be required. Fragmenting packets at layer 2 means that only the 'cross-over' nodes directly connected by the data link are involved in the fragmentation and re-assembly; the transmitting node fragments the IP packet and the receiving node re-assembles the IP packet back to the original packet received by the original node.
Packets fragmented using IP fragmentation (e.g., by an IP router) remain fragmented whilst routed across the IP network until they reach their eventual destination (e.g., an IP host computer) where the fragments will be re-assembled by the IP stack to generate the original IP datagram.
All the 'network-of-networks' must still be able to forward fragmented IP packets across their networks, so if an IP fragment is received on a 'cross-over' node, all the IP fragmentation fields must be included in the 'network-of-networks' compressed IP header. However, if the IP packet is not fragmented, no IP fragmentation information need be sent. For Link 11 and Link 16 'cross-over' nodes a 'network-of-networks' flag bit will be used to indicate if the IP packet is fragmented and the IP fragmentation data included at the end of the 'network-of-networks' IP header if so (giving a variable length header). Note that this is independent of the layer 2 fragmentation described for Link11 and Link 16.
HIDL networks have a much larger MTU (422 bytes), so IP packets over this size will use IP fragmentation before being forwarded across the HIDL network.

Time to Live

This field is used to ensure that a packet does not indefinitely flow around the network never reaching its destination. With every hop, the count is decremented by 1 and when the count reaches 0 the packet is removed. Within IPv4 8 bits are used, allowing a packet to traverse 256 networks before being dropped. It is not anticipated that the 'network-of-networks' will ever be that large, therefore it is assumed that there will never be more than 16 hops between source and destination and thus we only need 4 bits to represent the *Time to Live*. It is then proposed to ignore the most significant part of the IPv4 - *Time to Live* byte.
This seems a reasonable assumption as the latency involved with traversing more than 16 hops could make the communications problematic. (Please note that this does not limit the number of networks within the 'network-of-networks' to 16, only that there will never be more than 16 degrees of separation between two networks).

Protocol

IANA defines around 140 different protocols in (Arko&Brandes, 2008) for use over IPv4, the most common for user data transfer being TCP and UDP. As 'network-of-networks' uses its own network management system and routing algorithms other protocols such as IGP, EGP, RSVP will not be needed. Therefore it is assumed that only TCP and UDP transport protocols will be used for node to node communication within 'network-of-networks'. It is also assumed that the network management functions for 'network-of-networks' (routing, resource reservation, topology discovery) will need to be identified, both for the individual and the overarching network management functions. It is therefore proposed to use 4 bits to represent the protocol field:

- 0: UDP,
- 1: TCP,
- 2: 'Network-of-Networks' internal network management traffic (individual NMS),
- 3: 'Network-of-Networks' external network management traffic (overarching NMS),
- 5: ICMP,
- 6: IGMP,
- 15: protocol defined in 8 bit (optional) header field.

'Network-of-Networks' Header

If other protocols wish to be used over the 'network-of-networks' (such as BGP or RSVP) then the protocol field will contain a special value (15) which indicates that an extra (optional) 8bit IP protocol field will be present after the main 'network-of-networks' header containing the IP protocol, adding an extra byte to the 'network-of-networks' header.

'Network-of-Networks' Headers for Particular Standards

In this Section we describe 'network-of-networks' headers for different communication standards. These headers were designed and optimised, ensuring compliance with the major 'network-of-networks' e-Health requirements.

WiFi and WiMAX 'Network-of-Networks' Header

It is proposed to continue to use the standardised IPv4 headers.

HIDL 'Network-of-Networks' Header

HIDL requires the use of UDP over IPv4 packets with a maximum user data packet size of 422 bytes. The destination addresses are limited to the multicast IP addresses described by the circuit which means that even though the packets technically use an IPv4 header, it is insufficient in its entirety for our purposes. The identification, fragmentation, time to live and source address fields within the IPv4 header can be utilised as normal, but the protocol and destination addresses are going to have to be additionally provided. Therefore all packets going over HIDL will require the following 'H.Network-of-Networks' header to be used:

Bits: 0-7	8-15	16-23
Destination Network	Destination Node	Protocol

Fig. 5. HIDL 'Network-of-Networks' Header

A full 'network-of-networks' over HIDL packet would therefore look like the following:

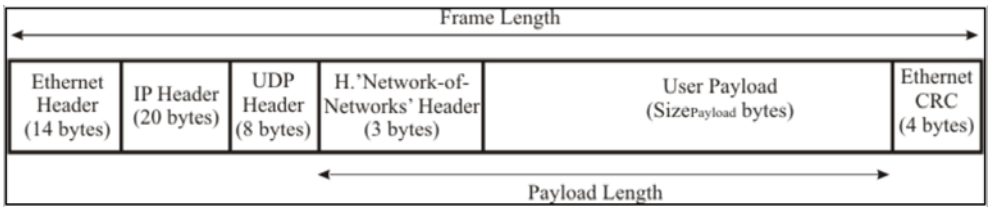

Fig. 6. Packet Format for the HIDL network within 'Network-of-Networks'

Link 16 'Network-of-Networks' Header

We proposed to use the 'Free Text' version of Link 16 which means that there are no J-series message headers for the packets we will be sending, in fact there will be no headers of any kind except to say that this packet is for transmission on the 'network-of-networks' Network Participation Group. Error correction mechanisms such as checksums and cyclic redundancy checks are not needed as they are already provided by the data-link.
Therefore, all the fields identified has to be present in a Link 16 layer 3 'network-of-networks' header. However a full IPv4 at 20 bytes would represent at least 71% of a standard message. The Link 16 'network-of-networks' layer 2 and layer 3 headers combined provide packet overhead which varies between 21 bytes = 75% (fragmented IP datagram requiring Link 16 layer 2 fragmentation) and 2 bytes = 7% (subsequent Link16 layer 2 fragments).
The Link 16 'network-of-networks' headers are formed from a layer 2 (data link) header followed by a layer 3 (compressed IP) header, both of which may contain optional fields (so they are variable length).

Layer 2 header

Optional fields are indicated by a dashed boarder and described below.
Mandatory layer 2 header fields:

- Layer 2 pkt Fragment – flag indicating this packet is a layer 2 fragment;
- Layer 2 first Fragment – flag indicating this packet is first layer 2 fragment in a sequence of fragments (only checked if Layer 2 pkt fragment flag set);
- MS bits Layer 2 fragment sequence number - Most significant 6 bits of layer 2 fragment sequence number (set to 0 if Layer 2 pkt fragment flag clear);

Optional layer 2 header fields included as follows:

- LS bits Layer 2 fragment sequence number - Least significant 8 bits of layer 2 fragment sequence number (only present if Layer 2 pkt fragment flag set);
- Layer 2 number of fragments (2 bytes) – included if Layer 2 pkt fragment flag set AND the layer 2 first fragment flag is set;
- Layer 2 IP datagram checksum (2 bytes) – included if Layer 2 pkt fragment flag set AND the layer 2 first fragment flag is set. Checksum covers the whole IP datagram including compressed IP header.

Layer 3 header

The illustration below shows the proposed Link 16 layer3 header and how it maps to the IPv4 header.
Optional fields are indicated by a dashed boarder.
Optional header fields:
Note if more than one option is present, they must be in the order shown below (shown with all optional fields present).
Optional header fields included as follows:

- IP Identification, Fragmentation Flags and Fragmentation offset (4 bytes) – included if IP pkt fragmentation flag set. Values are the same as in the original IP header.
- IP Full Protocol – included if value of protocol field is 7. Value the same as in the original IP header.

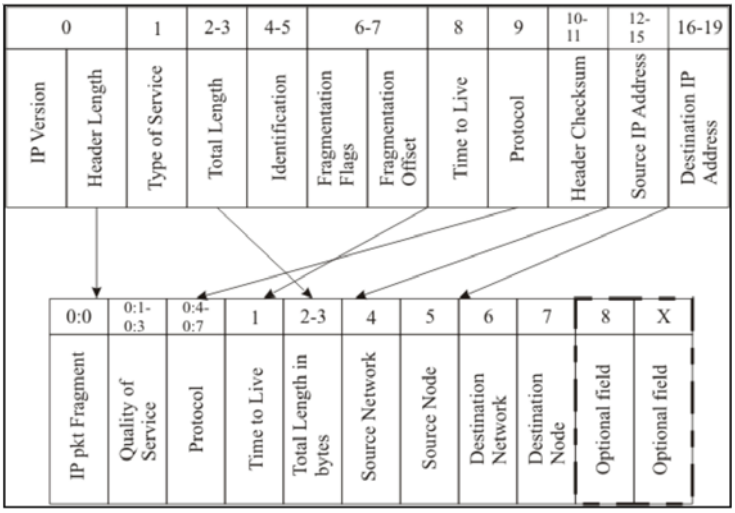

Fig. 7. Mapping Link 16 Layer3 Header to the IPv4 Header

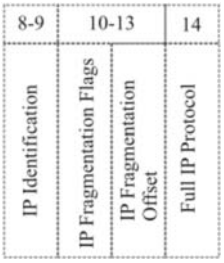

Fig. 8. Optional Header Fields

Link 11 'Network-of-Networks' Header

Because Link 11 uses a roll call mechanism, in which the transmitter carries on transmitting until all data is delivered and connection is completed, it means that it is not needed to provide a header to every packet we transmit, but rather send the header first and stream the data portion afterwards so the entire packet arrives all in one sequential stream. This stream may not be continuous (if other nodes transmit between portions of it), but by stitching together the transmissions from each node separately a terminal will be able to recover all packets. If the transmit window allows for multiple transmissions then it may subsequently send more header then data packet sequences. As a Link 11 packet is only 5 bytes long, the header needs to be split into 2 (or 3) packets as shown below (the destination address information arrives first thereby allowing a node to immediately determine if they should capture for the rest of the transmission or ignore it):

The layer 2 addressing for Link 11 'network-of-networks' information is mapped directly from layer 3 (the IP destination address) due to the broadcast nature of Link 11, so layer 2 and layer 3 header information are mixed together. Two bytes of layer 2 fragmentation information are always included, to assist in the identification of the first 5 byte message (which could be lost due to reception errors).

Fig. 9. Link 11 Destination Identification Message

Mandatory layer 2 header fields:

- Layer 2 pkt Fragment – flag indicating this packet is a layer 2 fragment;
- Layer 2 first Fragment – flag indicating this packet is first layer 2 fragment in a sequence of fragments (only checked if Layer 2 pkt fragment flag set);
- Layer 2 fragment sequence number - 14 bits of layer 2 fragment sequence number (set to 0 if Layer 2 pkt fragment flag clear).

The optional fields consist of layer 2 optional fields followed by layer 3 optional fields.

Fig. 10. Layer 2 Optional Fields

- Layer 2 number of fragments (2 bytes) – included if Layer 2 pkt fragment flag set AND the layer 2 first fragment flag is set;
- Layer 2 IP datagram checksum (2 bytes) – included if Layer 2 pkt fragment flag set; AND the layer 2 first fragment flag is set. Checksum covers the whole IP datagram including compressed IP header.

Link 11 layer 2 fragmentation works in a similar way to that described for Link 16, the major difference being that a single layer 2 fragment consists of a "stream" of Link 11 messages (as each message is only 5 bytes long). The first layer 2 fragment will have both the layer 2 header and layer 3 (compressed IP) header. Subsequent layer fragments will just have the layer 2 header bits.

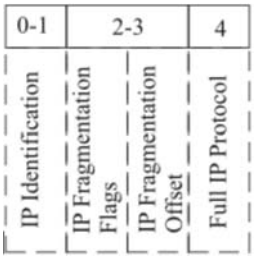

Fig. 11. Layer 3 Optional Fields are as Described for the Link 16

The layer 2 MTU size for link 11 is determined such that a 'network-of-networks' fragment will not take an excessive time to transmit, allowing other Link 11 traffic to be sent, but not too small such that the compressed 'network-of-networks' IP header is a too large fraction of the datagram. A MTU size of around 50 would take about 150ms to transmit

6. 'Cross-Over' nodes

We propose a new 'cross-over' node solution, which will ensure communication across the systems described above. However this is only half of what it is meant to do, the 'cross-over' nodes also perform an overarching network management system (O-NMS).

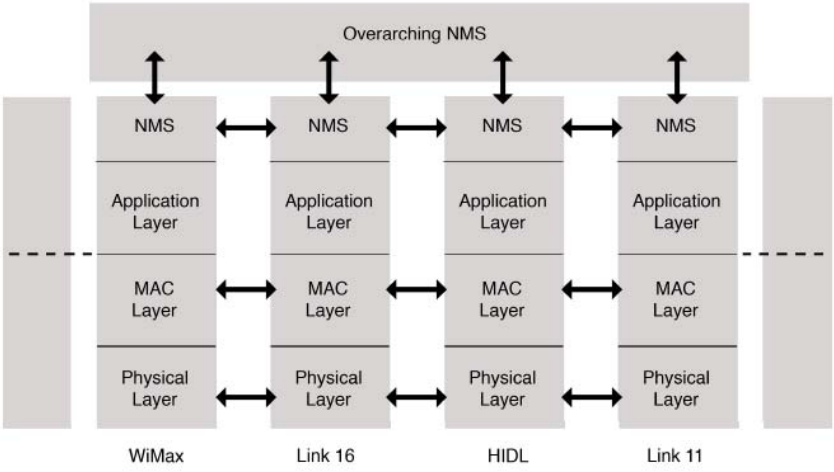

Fig. 12. Overarching Network Management System

It is shown previously how each data-link network in turn manages its own network and allocates resources, but these are narrow views of the 'network-of-networks' as a whole.
While each individual NMS maintains the allocation of resources within its own domain it is up to the Overarching NMS to try and balance the utilization and capacity of the network-of networks as a whole.
The Overarching NMS should try and make sure that not only there is no single point of failure within the large 'network-of-networks' (such as might be the case if all external network traffic is routed through the WiMAX network) but that routing information is kept

up to date and in the event of a change in the network topology (either someone joining or leaving) the routing of packets within the network reflects it.
The reason for separating the functions of the individual network management systems from the overarching network management system is that first and foremost the individual data link networks need to be able to continue functioning as they have been and provide the services they were designed for. Thus apart from a capacity utilisation impact the current data-link networks should not be further impacted. By separating the functions we are ensuring that the overarching NMS should be lost or a data-link network becomes cut off from the rest of the 'network-of-networks' it can continue operating as it has done with no noticeable effect from the point of view of non-'network-of-networks' terminals.

Centralised Overarching NMS

One method of accomplishing the task of the O-NMS is to centralise the process so that only one 'cross-over' node (per group of networks) centrally collects all the external network traffic together and re-distributes it according to the current network conditions and demands. Such a set up is shown below:

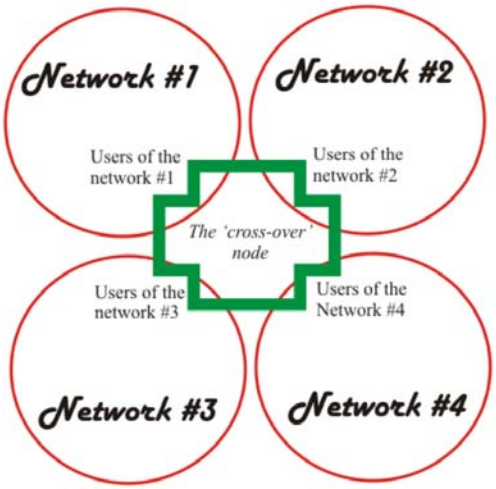

Fig. 13. Centralised Overarching Network Management System

This implementation would allow a centralised management system to request and effectively organise individual network resources so that real time and priority traffic were routed efficiently. There would also be no additional network overhead in this implementation as the O-NMS can hear all of the individual NMS's broadcasts and thus infer all the information it requires.
This central 'cross-over' node would also have a broadband link (such as WiMAX) linked to another 'cross-over' node elsewhere that controlled another separate group of networks, thereby enabling multiple groups of mini 'network-of-networks' to communicate with each other.
While there are advantages to centralised authority for communicating between networks the disadvantages are that the 'cross-over' node must be in a position to communicate with everyone within the local 'network-of-networks', and that there is a single point of failure

within the system that leaves the 'network-of-networks' implementation vulnerable to node attrition.

Distributed Overarching NMS

The other option is to distribute the functionality of the O-NMS to many disparate systems and have them cooperatively perform duties such as load balancing and traffic routing. This would require multiple 'cross-over' nodes between networks at different points; there could even be the possibility of multiple 'cross-over' points between two networks. Such a set-up is shown in figure 14.

This implementation would provide a robust architecture that has no single points of failure. If one 'cross-over' node is lost there are still many other routes a packet could take from source to destination. However, in order to produce a balanced load across the 'network-of-networks' and to effectively route packets through this large network the 'cross-over' nodes will have to communicate with each other, which will mean an increased network management overhead.

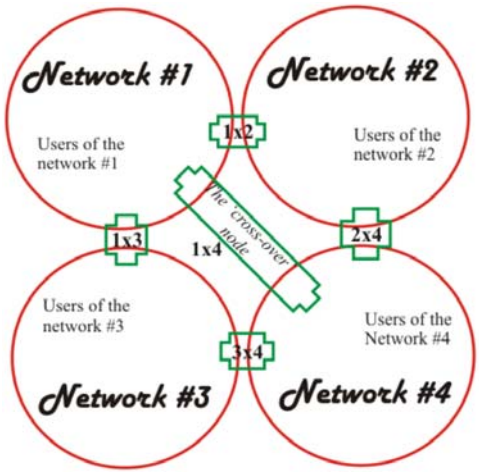

Fig. 14. Distributed Overarching Network Management System

As there are multiple routes that a packet could take and the best route for a given type of packet will depend on the current loading of the network there won't be a fixed route from source to destination. This dynamic nature may provide robustness to changing network topologies, but from the time the topology changes until the whole network is informed of this fact the network will remain in a state of flux. How changes in the topology are distributed around the network and how they will affect the routing choices of the 'cross-over' nodes will depend on the routing algorithms implemented, which will be investigated further within this research.

Routing choices

One of the most important features of IPv4 packets is that they look identical if they are addressed to a computer within the same network or to an external computer on the other side of the world. This unification of communication should be emulated within 'network-of-networks', such that a node within one network should be able to communicate in the

same manner with another node regardless of its location. The only difference should be in the QoS experienced (the greater the number of hops, the greater the latency), the format should be the same.

'Cross-over' nodes as the gateways between different networks have the responsibility to forward packets between different networks, which imply that they are also capable of deciding which packets need forwarding, and onto which other network. If a 'cross-over' node only sits on two networks then it only needs to know if it needs to forward it onto the other network, however if it sits on three or more networks then it needs to also make the decision as to which interface should be used for the next hop. The goal of the 'cross-over' nodes routing therefore is twofold; first to satisfy the QoS requirements for every admitted packet/stream, and second to achieve global efficiency in resource utilisation.

A simple methodology would be to make 'cross-over' nodes forward all externally addressed packets. This approach would ensure that a packet reaches its destination, but in the process it would be replicated numerous times and would make an inefficient use of the network resources (not to mention that multiple copies of each packet would end up reaching the destination), thereby failing to meet the second goal. While this might seem reasonable for a small all-informed network the magnitude data rate differences between WiMAX and Link 11 could mean that Link 11 is swamped by external WiMAX traffic.

Another methodology would be to coordinate the actions of the 'cross-over' nodes so that they have some knowledge of the topology of the network and its current utilisation and therefore forward it on to the most appropriate next hop. Having a simple knowledge of the network topology will allow each 'cross-over' node to easily calculate the route with the least number of hops and to forward the packet onto the next hop in the sequence. However, this does not take into consideration the appropriateness of each hop in the sequence. If the traffic is real time and of a high data rate, then it does not make much sense to route it over a Link 11 network even if it may be the most direct method, instead a route with a greater number of hops may be able to provide a traffic stream with the QoS it requires. Not only does a 'cross-over' node need to make an intelligent decision regarding the routing of a packet, but it also needs to coordinate its actions with other 'cross-over' nodes within the network.

QoS Routing and 'Cost' of transmission

The routing choice a 'cross-over' node will have to make will depend on the QoS requirements of the packet and the current utilisation of the 'network-of-networks'. The 'cross-over' nodes, in their role as the O-NMS need to ensure that the network as a whole is properly load balanced, so when making their decision regarding routing they may consider the following sort of parameters:

- Type of packet (Real-time, Priority, Best Effort);
- Impact of latency on packet;
- Impact of blocking on packet;
- Individual packet or part of a stream;
- Size of packet;
- Each network's current capacity utilisation;
- Each network's data rate throughput;
- Each network's jitter;
- Each network's ability to provide QoS;
- Each network's possible Bit Error Rate;

- Each 'cross-over' node's utilisation (spare buffer capacity);
- Current traffic route patterns.

These parameters, together with a weighting value as to the importance of each parameter, can be put into a function to determine the most appropriate route for each packet type at that instant. As a result a priority labelled packet will be routed differently to a Best Effort labelled packet, which could be different again to the route for a real-time packet. The calculated weighted sum is called the 'cost' of transmission:

$$C = \alpha \bullet C1 \bullet W1 + \beta \bullet C2 \bullet W2 + ... + \gamma \bullet Cn \bullet Wn, \quad (1)$$

where C – indicates the 'cost' of transmission; W*i*, *i*=1,...*n* – are the importance Weight and α, β, ..., γ are various possible QoS parameters (latency, end-to-end delay, throughput, etc.) This 'cost' function (1) is only meaningful for the current state of the network and the type of packet to be transmitted.

=For example, it is needed to transmit the delay sensitive information and there is two or more way of transmission. The cost of transmission through the first path includes

$$C_{path1} = \alpha \bullet C_1 \bullet W_1 + \beta \bullet C_2 \bullet W_2 + \xi \bullet C_n \bullet W_3 = C_1 \bullet delay \bullet 0,5 + C_2 \bullet blocking\_probabolity \bullet 0.1 + ... \quad (2)$$

$$C_{path2} = \alpha \bullet C_1 \bullet W_1 + \beta \bullet C_2 \bullet W_2 + \xi \bullet C_n \bullet W_3 = C_1 \bullet delay \bullet 0,4 + C_2 \bullet blocking\_probabolity \bullet 0.1 + ... \quad (3)$$

The parameter of weight should be chosen in respect to the type of transmitting information. When the scheme of the cost of transmission is known by the decision making mechanism, then it makes the decision about which way to transmit and initiates the transfer.

If such a cost function can be calculated for each hop through the network then a 'cross-over' node will be able to: work out the most effective route (the one with the minimum cost), ensure that it meets the load balancing requirements of the network as a whole, and route the packet accordingly.

In order for to calculate a meaningful figure the information used needs to be correct, and that means current. Using out of date information could negatively impact the network, such as route more information into an already over-congested network. 'Cross-over' nodes will therefore need to share information with each other at regular intervals. This information sharing will cause increased network overhead, which will need to be carefully balanced against the benefits the information sharing will produce. Therefore, an investigation will need to be carried out to determine which parameters and weightings will be required to effectively calculate the 'cost' of transmission for each packet type, and what update interval is most appropriate.

'Cross-Over' Node Communication Protocols

The actual message exchange will not strictly conform to a pre-existing IP protocol scheme (e.g. BGP, OSPF), this is due to the unique nature of 'network-of-networks'. 'Network-of-networks' is not trying to implement the internet, as they are orders of magnitude different and the network resources these protocols were designed around (i.e. maximum data rates, packet sizes, latencies) are very different. Instead, while the functionality may be similar the exact message structures will be different.

As described in the previous sections it is proposed to use two separate 'network-of-networks' messaging protocols: internal and external. Internal messages are used by nodes

to communicate with their individual network managers, external messages are used by the 'cross-over' nodes to communicate with each other.

Internal Messages

Networks such as WiMAX and Link 16 have their own network management system communications protocols, and it is anticipated that 'network-of-networks' will continue to use them where possible. But for other messages that the current NMS protocol does not support, such as the current list of active nodes, new 'network-of-networks' NMS messages will need to be sent.

The formats of the messages should conform to the data-link network formats defined previously in this Chapter, with the protocol defined as either '2' (for 'network-of-networks' headers) or '222' for IPv4.

The following messages will be used:

Hello: Used to inform the NMS that they are present

- Node Address,
- Is it a 'cross-over' node (or not),
- The network addresses of the other networks they are attached to.

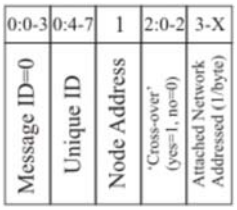

Fig. 15. 'Network-of-Networks' NMS Internal Messages (hello message)

Hello Reply: Used by the NMS in reply to a Hello message

- 'Hello' Node Address,
- NSM Node Address,
- Network Address.

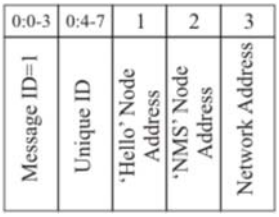

Fig. 16. 'Network-of-Networks' NMS Internal Messages (hello reply message)

Active Node List: Used to distribute the list of current active nodes

- Network Address,
- Total number of Active Nodes,
- List of Active Node Addresses,
- Total number of 'cross-over' nodes,
- List of 'cross-over' node addresses,

0:0-3	0:4-7	1	2	3	4-X	(X+1)-Y
Message ID=3	Unique ID	Network Address	Number of Active Nodes	Number of 'cross-over' nodes	List of Active Nodes (1/byte)	List of 'cross-over' nodes (1/byte)

Fig. 17. Active Node List: Used to Distribute the List of Current Active Nodes

External Messages

In order to perform the required O-NMS functions the 'cross-over' nodes require two elements: some knowledge of the network topology and some knowledge of the 'cost' of traversing the network. The 'cost' can be determined by an algorithm and information exchange, but in order to know which nodes to contact and what networks are available the 'cross-over' nodes need to know the topology of the network.

There are two main methods of undertaking this, the first is for all 'cross-over' nodes to know the entire topology of the network and the other is for them to know a local portion of the network and how to route traffic to for more remote portions of the network. These two methods are born out in two styles of routing protocols; interior and exterior routing protocols. Each version requires a differing amount of network overhead, and ends up with different strengths and weaknesses.

Therefore, the two major external message types anticipated are distribution of 'cost' information, and distribution of network topology.

The exact structure of these messages and their sizes will depend on factors such as the routing algorithm, simulation implementation and network complexity. The protocol used however, should be defined as either '3' (for 'network-of-networks' headers) or '223' for IPv4.

External and Internal Messages: Resource Reservation

In order to route real-time and possibly some priority traffic some QoS requirements will need to be met for each hop from source to destination. If the route is contained within one network then this should just involve a request for resources from the Individual NMS (I-NMS). If the route involves multiple networks then each individual NMS along the route will have to be contacted and resources reserved. If resources are not available across one hop in the route, then a new route will have to be calculated and any unused reserved resources released back to their network managers. In order to fulfil our requirement that a node should be able to communicate with an external destination in the same manner as an internal one the mechanisms for the resource reservation should also be identical. This will mean that the local I-NMS will be contacted by the source requesting resources to send a traffic stream to a remote destination. The I-NMS should identify that the destination is not local to this network and allocate the necessary resources for the first hop (if possible) before sending the request to a 'cross-over' node. The 'cross-over' node will then have to decide on the appropriate route and request resources for each hop along the way. Once a route has been reserved the source will need to be informed and the stream can begin. Once the communication has been completed the source will need to inform its local I-NMS that it no longer needs the resource. Once the resources are released, the local I-NMS should inform the 'cross-over' node which will release the reserved resources along each hop. Therefore this process utilizes both internal and external messaging.

The resource reservation for each hop can either be controlled from the first 'cross-over' node or handed off to the next 'cross-over' node in turn. Which method will be more appropriate will depend on the topology and routing algorithms chosen. If the topology algorithm does not allow for complete knowledge of the network topology then the reservation process cannot be centrally managed, although if it does then the complexity of the reservation process is greatly reduced.
All individual network managers will need to monitor the utilization of all allocated resources, so that should a crucial 'cross-over' node or source node drop off the network the resources are not then reserved indefinitely.

Internal Messages

Resource Request: used to request resources from the NMS:

- Requesting Node Address,
- Destination Address (Network and Node),
- Data size in KiloBytes (or Bytes depending on the network) per frame,
- Frequency of frames per timebase (the timebase is link dependent, or for Link 11 refer to a transmission cycle),
- QoS of traffic,
- Utilisation time (units are link dependent, not required for Link 11).

0:0-3	0:4-7	1	2	3	4	5	6:0-2	6:3-7
Message ID=4	Unique ID	Node Address	Destination Network Address	Destination Node Address	Data Size	Frequency	QoS Traffic	Utilisation Time

Fig. 18. Resource Request Message: Used to Request Resources from the NMS

Resource Granted: Used to inform the node that the resource has been granted

- Requesting Node Address,
- Unique ID (used in the initial resource request),
- Destination Address (Network and Node),
- Data size granted (same as requested or less if full amount not available),
- Frequency granted (same as requested or less if full amount not available),
- Utilization time (same as requested or less if full amount not available).

0:0-3	0:4-7	1	2	2	3	4
Message ID=5	Unique ID	Destination Network Address	Destination Node Address	Data Size	Frequency	Utilisation Time

Fig. 19. Resource Granted message: Used to Inform the Node that the Resource has been Granted

Resource Denied: used when a route that can support the requested level of QoS cannot be found

- Requesting Node Address,
- Unique ID (used in the initial resource request),
- Destination Address (Network and Node),
- Reason request was denied:

1 – insufficient BW,
2 – QoS type not supported,
3 – resource temporarily unavailable.

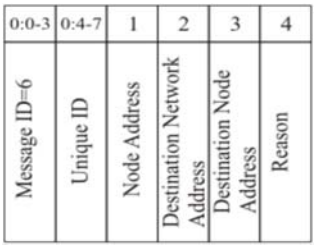

Fig. 20. Resource Denied: Used when a Route that can Support the Requested Level of QoS cannot be Found

Resource Release: used by a node when it has finished with the resource

- Requesting Node Address,
- Unique ID (used in the initial resource request),
- Destination Address (Network and Node).

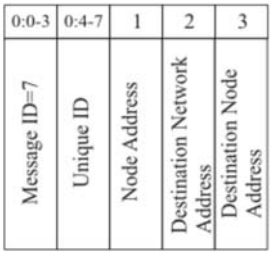

Fig. 21. Resource Release Message: Used by a Node when it has finished with the Resource

External Messages

Resource Request: used by the NMS to a 'cross-over' node to begin reserving resources along a route

- Unique ID (used in the initial resource request),
- Requesting Address (Network and Node),
- Destination Address (Network and Node),
- Data size in KiloBytes per frame,
- Frequency of frames per timebase (the timebase is seconds),
- QoS of traffic,
- Utilisation time (the timebase is 10 seconds),

0:0-3	0:4-7	1	2	3	4	5	6	7:0-2	7:3-7
Message ID=1	Unique ID	Request Network Address	Request Node Address	Destination Network Address	Destination Node Address	Data Size	Frequency	QoS of Traffic	Utilisation Time

Fig. 22. Resource Request External Message: Used by the NSM to a 'Cross-Over' Node to begin Reserving Resources along a Route

Resource Request: used between 'cross-over' nodes to reserve resources along a route

- Unique ID (used in the initial resource request),
- Requesting Address (Network and Node),
- Destination Address (Network and Node),
- Data size in KiloBytes per frame,
- Frequency of frames per timebase (the timebase is seconds),
- QoS of traffic,
- Utilisation time (the timebase is 10 seconds),
- Previous reserved networks (List of Network Addresses).

0:0-3	0:4-7	1	2	3	4	5	6	7:0-2	7:3-7	8-X
Message ID=2	Unique ID	Request Network Address	Request Node Address	Destination Network Address	Destination Node Address	Data Size	Frequency	QoS Traffic	Utilisation Time	Reserved Networks (1/byte)

Fig. 23. Resource Request External Message: Used between 'Cross-Over' Nodes to Reserve Resources along a Route

Resource Granted: used between 'cross-over' nodes to indicate a resources has been reserved

- Unique ID (used in the initial resource request),
- Requesting Address (Network and Node),
- Destination Address (Network and Node),
- Data size granted (same as requested or less if full amount not available),
- Frequency granted (same as requested or less if full amount not available),
- QoS of traffic,
- Utilisation time (same as requested or less if full amount not available),
- Reserved Network Address.

0:0-3	0:4-7	1	2	3	4	5	6	7:0-2	7:3-7	8
Message ID=3	Unique ID	Request Network Address	Request Node Address	Destination Network Address	Destination Node Address	Data Size	Frequency	QoS of Traffic	Utilisation Time	Reserved Network

Fig. 24. Resource Granted External Message: Used between 'Cross-Over' Nodes to Indicate a Resource has been Reserved

Resource Denied: used by a 'cross-over' node to indicate that such a request cannot be granted

- Unique ID (used in the initial resource request),
- Requesting Address (Network and Node),
- Destination Address (Network and Node),
- Denied Network Address,
- Reason (see internal message for values).

0:0-3	0:4-7	1	2	3	4	5	6
Message ID=4	Unique ID	Request Network Address	Request Node Address	Destination Network Address	Destination Node Address	Network Address	Reason

Fig. 25. Resource Denied External Message: Used by a 'Cross-Over' Node to Indicate that such a Request cannot be Granted

Resource Release: used by a 'cross-over' node release a resource:

- Unique ID (used in the initial resource request),
- Requesting Address (Network and Node),
- Destination Address (Network and Node),
- Released Network Address.

0:0-3	0:4-7	1	2	3	4	5
Message ID=5	Unique ID	Request Network Address	Request Node Address	Destination Network Address	Destination Node Address	Network Address

Fig. 26. Resource Release External Message: Used by a 'Cross-Over' Node Release a Resource

Multicast Request: used by both a node and a 'cross-over' node to register its request to receive a multicast stream:

- Unique ID,
- Requesting Address (Network and Node),
- Destination Multicast Address (Network and Node),
- Current Network Address.

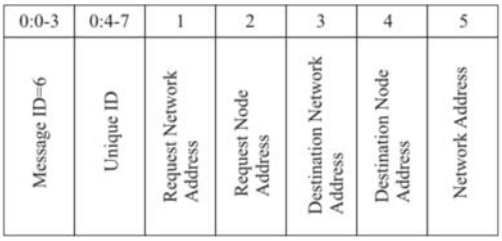

Fig. 27. Multicast Request External Message: Used by both a Node and a 'Cross-Over' Node to Register its Request to Receive a Multicast Stream

Multicast Release: used by node to register its request to receive a multicast stream:

- Unique ID (used in the initial resource request),
- Requesting Address (Network and Node),
- Destination Multicast Address (Network and Node),
- Current Network Address.

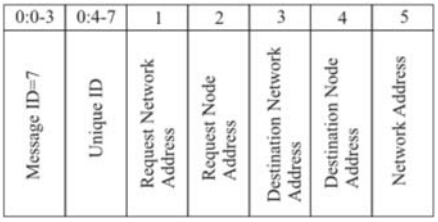

Fig. 28. Multicast Release External Message: Used by a Node to Register its Request to Receive a Multicast Stream

7. Summary

This Chapter develops a wireless cross-standard communication protocol and describes the concept of 'network-of-networks' in conjunction with e-Health applications. Analysis of the legacy communication systems and their integration into a single 'network-of-networks' communication protocol is presented. Based on this analysis, the developed concept was implemented in the CLAHNS project for MOD which was supported by Lancaster University.

8. References

Arkko J., Brander S. (2008). *IANA Allocation Guidelines for the Protocol Field*. Network Working Group. February 2008. [Online]. Available: http://tools.ietf.org/html/rfc5237 [Accessed: January 2009].

Internet Protocol. Darpa Internet Program. Protocol Specification. September 1981. [Online]. Available: http://www.ietf.org/rfc/rfc0791.txt [Accessed: May 2009].

Lockheed Martin UK- Integrated Systems and Solutions. *Tactical Data Links – MIDS/JTIDS Link 16, and Variable Message Format - VMF*. [Online]. Available: http://www.lm-isgs.co.uk/defence/datalinks/link_16.htm [Accessed: March 2010].

Almquist P. (consultant) (1992). Type of Service in the Internet Protocol Suite. Network Working Group. [Online] Available: http://tools.ietf.org/html/rfc1349 [Accessed: March 2010].

ARM Technical Specification. *ARM, MPEG-4, AAC, LC Decoder Technical Specification*. Document Number: PRD10-GENC-0012884.0. Date of Issue, 19 June 2003. ARM Limited 2002-2003. [Online]. Available: http://www.arm.com/files/pdf/PRD10-GENC-001288-4-0.pdf [Accessed: March 2010].

ASCII Table and Description. www.AskiiTable.com. [Online]. Available: http://www.asciitable.com/ [Accessed: March 2010]. Available: http://iphome.hhi.de/wiegand/assets/pdfs/h264-AVC-Standard.pdf [Accessed: March 2010].

Baker F. (editor) (1995). *RFC 1812 – Requirements for IP Version 4 Routers*. Cisco Systems. Network Working Group. [Online]. Available: http://www.faqs.org/rfcs/rfc1812.html [Accessed: March 2010].

Chandraiah P., Domer R. (2005). Technical Report CECS-05-04. *Specification and Design of a MP3 Audio Decoder*. [Online]. Available at: http://www.cecs.uci.edu/technicalreport/TR05-04.pdf [Accessed: September 2009].

Chiariglione L.(2000). *MPEG-2. Generic coding of moving pictures and associated audio information. Start MPEG-2 description.* International Organisation for Standardisation. ISO/IEC JTC1/SC29/WG11 coding of moving pictures and audio. [Online] Available at: http://mpeg.chiarglione.org/standards/mpeg-2.htm [Accessed: January 2010].

Istepanian R. S. H., Philip N., Martini M. (2009). Medical QoS Provision Based on Reinforcement Learning in Ultrasound Streaming over 3.5G Wireless Systems. *IEEE Journal on Selected areas in Communications*. Vol.27, 4, pp.566-574.

Lockheed Martin UK- Integrated Systems and Solutions. *Tactical Data Links – Link 11 and 11B*. [Online]. Available: http://www.lm-isgs.co.uk/defence/datalinks/link_11.htm [Accessed: March 2010].

Marpe D., et al (2006). The H.264/MPEG4 Advanced Video Coding Standard and its Applications. Standards report. *IEEE Communications Magazine.* [Online]. Available: http://iphome.hhi.de/wiegand/assets/pdfs/h264-AVC-Standard.pdf [Accessed: March 2010].

Microsoft Corporation (1999). *Rich Text Format (RTF) Specification, version 1.6.* [Online] Available: http://msdu.microsoft.com/en-us/library/aa140277(office.10).aspx [Accessed: February 2010].

Microsoft Media. *Windows Media Video 9 Series Codecs*. [Online] Available at: http://www.microsoft.com/windows/windowsmedia/forpros/codecs/video.aspx [Accessed: February 2010].

SyntheSys. Military Systems. *UK Tactical Data Systems Reference Guide*. 2006 [Online]. Available: http://www.synthesys.co.uk/UK_Tactical_Data_Systems_Reference_Guide.htm [Accessed: September 2009].

Tarter A., et al (2008). *CLAHNS Protocol Description Document*. PD-650-50004. Issue 2. Ultra Electronics.

ULTRA Electronics Limited. Communication and Integrated Systems. *Report: High Integrity Data Links (HIDL). Mission Critical Secure Networks*. 2010. [Online]. Available: http://www.ultra-cis.com/resourses/HIDL%200909v1.pdf [Accessed: March 2010].