Integrating TETRA with Wireless Mesh Networks

Master of Sceince Thesis Report

By

Kiros Siyoum Weldemichael

Supervised by:
Prof. Dr.Ir. Sonia M. Heemstra de Groot
Dr.Ir. Anthony Lo
Ir. Hugo de Graaf

July 2010
Technical University of Delft
Faculty of Electrical Engineering, Mathematics and Computer Science
Telecommunications Track
Wireless and Mobile Communications Group

Acknowledgment

First, I would like to thank my supervisors Prof Sonia Heemstra de Groot and Prof Anthony Lo for their valuable guidance in my thesis work. I am also grateful for Sonia's first proposing the project and providing me a clear description of it. Sonia, I really owe your humble personality in my stay at WMC.

Special thanks to my daily supervisor Hugo de Graaf who has contributed a lot in this project work. I am really grateful for the weekly meetings and warm discussions with him in spite of his tight schedules. He has really offered a consistent guidance by reading all the reports I submitted and suggesting important points in my designs.

It is my pleasure to thank all colleagues at TI-WMC who made this thesis work possible. I would like to thank specially to Michel, Neill and Tom for their voluble support in a number of ways to my thesis work. They have played a great role in helping me understand how the FIGO network works. I am also great full for Jan stoter for editing my design proposal and his important suggestions in the design.

I owe my deepest gratitude to Mekelle Institute of Technology (MIT) and EWI faculty of TU Delft for sponsoring my master's study at TU Delft. Special thanks to the Board of MIT for their full support in realizing my dreams.

My sincere thanks to all my family and friends for their love and support in achieving my goals. Special thanks to my brother Girmay and my cousin Abrhaley for their trust in me and financial support in my high school and university studies. And thanks to all my close friends for your wonderful friendship.

In God the Most High I Believe!

Kiros Siyoum Weldemichael
July 2010

Abstract

Terrestrial Trunked Radio(TETRA) is a narrow band digital cellular network for Private Mobile Radio (PMR) communications mainly designed for voice communications with limited packet data capability because of bandwidth limitations. This limited packet data capability of TETRA is used for the integration of the TETRA network with wireless mesh networks. The wireless mesh network used in this thesis work is part of the FIGO network. FIGO is a robust communication system mainly intended for public safety communications and it consists two main network parts. One of these parts is the wireless adhoc network which can be deployed on incident areas. The other one is the infrastructure component (back office node) that is used to interlink disconnected adhoc nodes and it can also offer access for controlling an incident in the adhoc network. The FIGO network uses different communication technologies to link the adhoc network with the back office node and the main aim of this thesis work is to integrate the TETRA network with the FIGO network so that the TETRA can be used as one of the links from the adhoc network to the back office node of the FIGO network.

The main problem of the TETRA-FIGO integration is that the wireless adhoc network generates a lot of control signals that can easily overload the narrowband packet data channel of the of the TETRA network. In addition to the control overheads, the excessive headers also contribute a lot of overheads to the narrowband TETRA channel. Two architectures are proposed in thesis report to integrate TETRA with the FIGO network and the performance of these two new proposed architectures is analyzed and compared with the existing FIGO network protocols in terms of control traffic and header overheads.

List of Abbreviations

AACH	Access Assignment Channels
AI	Air Interface
ASCCH	Assigned Secondary Control Channel
AT	Attention (Hayes command set)
AWN	Ancillary Wireless Network
BCCH	Broadcast Control CHannels
BF-CBC	Blowfish-Cipher Block Chaining
BON	Back Office Node
CCH	Control CHannel
CCK	Common Cipher Key
CHAP	Challenge Handshake Authentication Protocol
CMCE	Circuit Mode connection Entity
CN	Correspondent Node
CSMA	Carrier Sense Multiple Access
CTS	Clear to Send
DCD	Data Carrier Detect
DCK	Derived Cipher Key
DMO	Direct Mode Operation
DTR	Data Terminal Ready
EAN	Extended Area Network
EMS	emergency medical services
ETSI	European Telecommunication Standards Institute
ETT	Estimated Transmission Time
FA	Foreign Agent
FCS	Frame Check sequence
FEC	Forward Error Correction
Flame	Forward layer meshing
HA	Home Agent
HMAC	Hash-based Message Authentication Code
GW	Gateway
IAN	Incident Area Network
ICMP	Internet Control Message Protocol
IPCP	IP configuration protocol
ISI	Inter System Interface
ISDN	Integrated Services Digital Network
JAN	Jurisdictional Area Networks
KSS	Key Stream Segment
L1	the link from a VN to the BON
LCP	Link Configuration Protocol
LLC	Link Layer Control
LS	Line Station
LSU	Link State Update

MCCH	Main Control CHannel
MIPv4	Mobile IP version 4
MLE	Mobile Link Entity
MM	Mobility Management
MMI	Mobility Management Information
MN	Mobile Node
MS	Mobile Station
MT	Mobile Terminal
NA	Neighbour Acknowledgement
OLSR	Optimized Link State Routing Protocol
OTAR	Over the air re-keying
PAMR	Public Access Mobile Radio
PAN	Personal Area Network
PAP	Password Authentication Protocol (CHAP)
PD	Packet Data
PDCH	Packet Data Channel
PDN	public data network
PDO	Packet Data Optimized
PDP	Packet Data Protocol
PDR	Packet Delivery Ratio
PEI	Peripheral Equipment Interface
PMR	Private Mobile radio
PPDR	public protection and disaster relief
PPP	point-to-point protocol
PPPd	point-to-point protocol daemon
PSCD	Public Safety Communication Device
PSTN	public switched telephone network
PU	Path Update
RD	Receive Data
RTS	Request to Send
RSSI	Received Signal Strength Indicator
SCCH	Secondary Control CHannel
SCK	Static Cipher Key
SDS	Short Data Service
SDU	Service Data Unite
SHA	Secure Hash Algorithm
SNDCP	Sub-Network Dependent Convergence Protocol
STCH	Stealing CHannel
TCH	Traffic Channel
TD	Transmit Data
TE	Terminal Equipment
TEDS	TETRA Enhanced Data Service
TETRA	Terrestrial Trunked Radio
TNP	TETRA Network Protocol
UMTS	Universal Mobile Telecommunications System
V+D	Voice plus Data mode
VN	vehicular Node
VPN	Virtual Private Network

Contents

List of Figures

Chapter 1

Introduction

Terrestrial Trunked Radio (TETRA) is a digital trunked mobile radio standard developed by the European Telecommunications Standards Institute (ETSI). The purpose of the TETRA standard was to meet the needs of traditional Professional Mobile Radio (PMR) user organizations such as public safety, military, governmental and other commercial organizations [2]. The TETRA system is designed to provide different types of services. These services are voice communications, short data service, circuit switched and packet switched data communications. Higher priority is given to voice communications in TETRA as it is crucial for public safety communications. The packet switched data communications service provided by TETRA has very low data rate because of bandwidth limitations in the system. A single packet data channel in TETRA provides on average **1.5 kbps** and this low data rate channel is going to be used to integrate the TETRA with the FIGO network.

FIGO is a robust communication system, which is intended for the public safety communications developed by Twente Institute for Wireless and Mobile Communications (TI-WMC) and it can actually be used to build any secure private network. It provides a fast and easy set-up of a reliable local adhoc communications in the incident area, while providing a reliable connection to other infrastructure networks on the back office [9].

There are two main network parts in FIGO. One is the wireless adhoc network which is deployed on incident areas and it consists wireless adhoc nodes which can reliably communicate among each other without the help of any infrastructure network. The second part is the infrastructure

component (back office node) that is used to interlink disconnected adhoc nodes/networks and it can also be used to offer access for controlling a incident in the adhoc network from a dispatch/control center. A meshing protocol called Forward layer meshing (Flame) is used to mesh all the nodes in FIGO network. Meshing in FIGO is not only in the adhoc network but also the back office node on the infrastructure is part of the mesh. The details of Flame is given in chapter 4.

1.1 Project objective and problem identification

The main aim of this study is to integrate the TETRA network with the FIGO network. Currently different communication technologies (UMTS, WiFi or satellite) are used in FIGO to link the incident area adhoc network part with the infrastructure component (back office node). The main objective of this thesis work is to design an architecture for integrating TETRA as one of the communication links from the adhoc network part to the infrastructure of the FIGO network.

As mentioned at the beginning of this chapter, the packet switched data service of the TETRA system is used for integrating the TETRA with the FIGO network. The packet data channel in TETRA has a limited capacity. There are two problems that need to be addressed in the TETRA-FIGO integration. One of the problems is the control traffic overheads generated by the Flame protocol of FIGO and these can easily overload the narrowband TETRA packet data channel. The other problem is the excessive header overhead introduced due to the Flame protocol and this can consume a lot of the narrowband TETRA capacity.

1.2 Thesis outline

The organization of the thesis report is given as follows. Chapter 2 is an introduction to the public safety communication networks, The overview of the TETRA network and the FIGO network are given in chapter 3 and chapter 4 respectively. Chapter 5 gives the practical aspects of interfacing the FIGO network with the TETRA network. Chapter 6 deals with the analysis of the existing Flame protocol of FIGO for the TETRA packet data channel and problem definitions. Chapters 7 and chapter 8 are about

the architecture design of the TETRA-FIGO integrations and the analysis of control traffic and header overheads of these architectures. From the simulations and measurements undertaken in the literature on the packet data channel of the TETRA network, the services that can be supported by the TETRA-FIGO integration are also identified in chapter 9 as a proof of concept. And finally the conclusion and future work are given in chapter 10.

Chapter 2

Public Safety Communications

Since our aim is to integrate the TETRA network with FIGO for public safety data communications, lets briefly see how the general public safety communications look like before we see the details of the two networks (TETRA and FIGO).

Public safety is a public service that focuses primarily on law enforcement, fire-fighting, emergency medical, and disaster recovery services for protecting the citizens from a likelihood of harm [1]. These services help to ensure the protection and preservation of life and property and they are termed as public protection and disaster relief (PPDR). A Public safety organization is a federal, state, or local organization that helps furnish, maintain, and protect the infrastructures (e.g., highways and utilities) that promote the public's safety and welfare [3].

2.1 Public Safety Requirements

A public safety organization requires an effective command, control, coordination, communication, and sharing of information with numerous public safety agencies. Broadly defined, the public safety community performs emergency first response missions to protect and preserve life, property, and natural resources and to serve the public welfare. The public safety support entities include law enforcement, fire fighter, and emergency medical services

(EMS) and are considered as the first category, while transportation or public utility workers fit the second category in public safety communications [1].

Communication systems to be used in a public safety specially in mission critical emergency, whether fixed or mobile, voice or data, need to fulfil the following essential requirements [21]:

- **Resilience:** available all the time (very high level of reliability);
- **Coverage:** available in all locations;
- **Grade of service:** network access instantly available when required. The network should never be too busy. That can include flexibility managed by the relevant agency that is called hierarchical command and control management;
- **Security and inter-operability:** secure communications between all parties that need to be involved. Radio networks for PPDR should provide high quality end to end encryption with key autonomy for each user group.

The requirements given above are not the only requirements in public safety networks. The requirements can be categorized in terms of services, required features or performances. Exhaustive list of the requirements and facilities are given in reference [1, 2, 3, 4]

Voice communications are critical in public safety communications, but voice communication requirements are not the only issue because these days public safety operations are increasingly dependent on the sharing of data, images, and video. New technologies promote the convergence of information and data communication systems with the result that mobile units are increasingly being viewed as merely wireless nodes within information networks.

SAFECOM is a communications program that provides research, development, testing and evaluation of communications-related issues to local or federal emergency response agencies in the US. It works to improve emergency response through more effective and efficient inter-operable wireless communications. According to the SAFECOM Program, the mode of communications in public safety are categorized into four types of communications [1].

`Interactive voice communications` between public safety practitioners, their supervisors and dispatchers, which is crucial in life and death situa-

tions.

`Non-interactive voice communications` are those that occur when a dispatcher or supervisor alerts members of a group about emergency situations and/or to share information.

`Interactive data communications` will provide practitioners with maps, floor plans, video scenes, etc.

`non-interactive data communications` are a one-way stream of data, such as the monitoring of fire fighter biometrics and location, which greatly increases the safety of the practitioners.

2.2 Public Safety Networks

Many specifications of public safety networks are given in the technical report of the MESA project. Project MESA is an international partnership producing globally applicable technical specifications for digital mobile technology, aimed initially at the sectors of public safety and disaster response. More information about project MESA can be found on the following website: http://www.projectmesa.org/.

According to the MESA project there are 4 types public safety networks that play a great role in providing a reliable public safety communication. Each of the network types have their own requirements to fulfil to be part of the public safety communications [4].

1. **Incident Area Networks (IAN)**: The IAN is generally dedicated to a single incident or event. The key public safety requirement is the ability of communication devices to work in the lack of any supporting infrastructure and the IAN has to fulfil this requirement. An IAN can be pre-deployed for a planned event, or it could be dynamically deployed for an unplanned event or incident.
2. **Jurisdictional Area Networks (JAN)**: The JAN is designed to provide service over a wide area that may include such geographic boundaries as a city, or country. So, it includes and interlinks many IANs.
3. **Extended Area Networks (EAN):** The EAN is a network mainly designed to provide connectivity between various JANs and it can also

include traditional back end networks used to access various databases and information sources.

4. **Ancillary Wireless Networks (AWN):** An AWN is a network that is primarily designed and operated to accommodate commercial services, but is accessible by and available to public safety users for both general purpose and public safety communications needs.

The public safety network architecture includes all the network types and network components. The network components consists of Personal Area Network (PAN), Public Safety Communication Device (PSCD) and semi-stationary mobile terminals. The PAN is comprised of special purpose devices or components of limited scope and transmission radius. There is a great variety of devices that may be deployed in the PAN as a PAN can provide the basis for a distributed implementation of a terminal device. In many cases, devices on the PAN are sensors and these sensors are referred to as public safety sensors. PSCD is generally hand held or mobile communication device and may not include as many capabilities in the interest of conserving power and weight. A semi-stationary mobile terminal is usually vehicle based, allowing it a more dependable power supply, improved antenna placement or reach and improved transmit/receive functionality due to increased power levels. The network architecture proposed by the MESA project is given in figure 2.1

The TETRA network is going to be used as a link between the wireless mesh nodes (vehicular nodes) and the back office node (BON) of the FIGO network in the TETRA-FIGO integration. So, the integration of TETRA with the FIGO network fits in the interface between the adhoc IAN and the infrastructure IAN of the public safety network stated in figure 2.1. In this thesis work, the TETRA-FIGO integration is typically considered as interface 2c of the public safety network depicted in figure 2.1.

Most of the existing public safety communication systems use narrowband digital mobile networks. Across Europe most public safety national agencies have decided to invest into dedicated narrowband digital mobile networks for voice and data communications. TETRA and TETRAPOL are most commonly used narrowband digital mobile networks all over Europe. Due to the intensive use of data applications in the fixed environment, public safety agencies are now using such data applications in the mobile environment as well. However these are usually used over commercial networks like UMTS which can be limiting in critical situations because these commercial networks get overloaded by users in emergency situations.

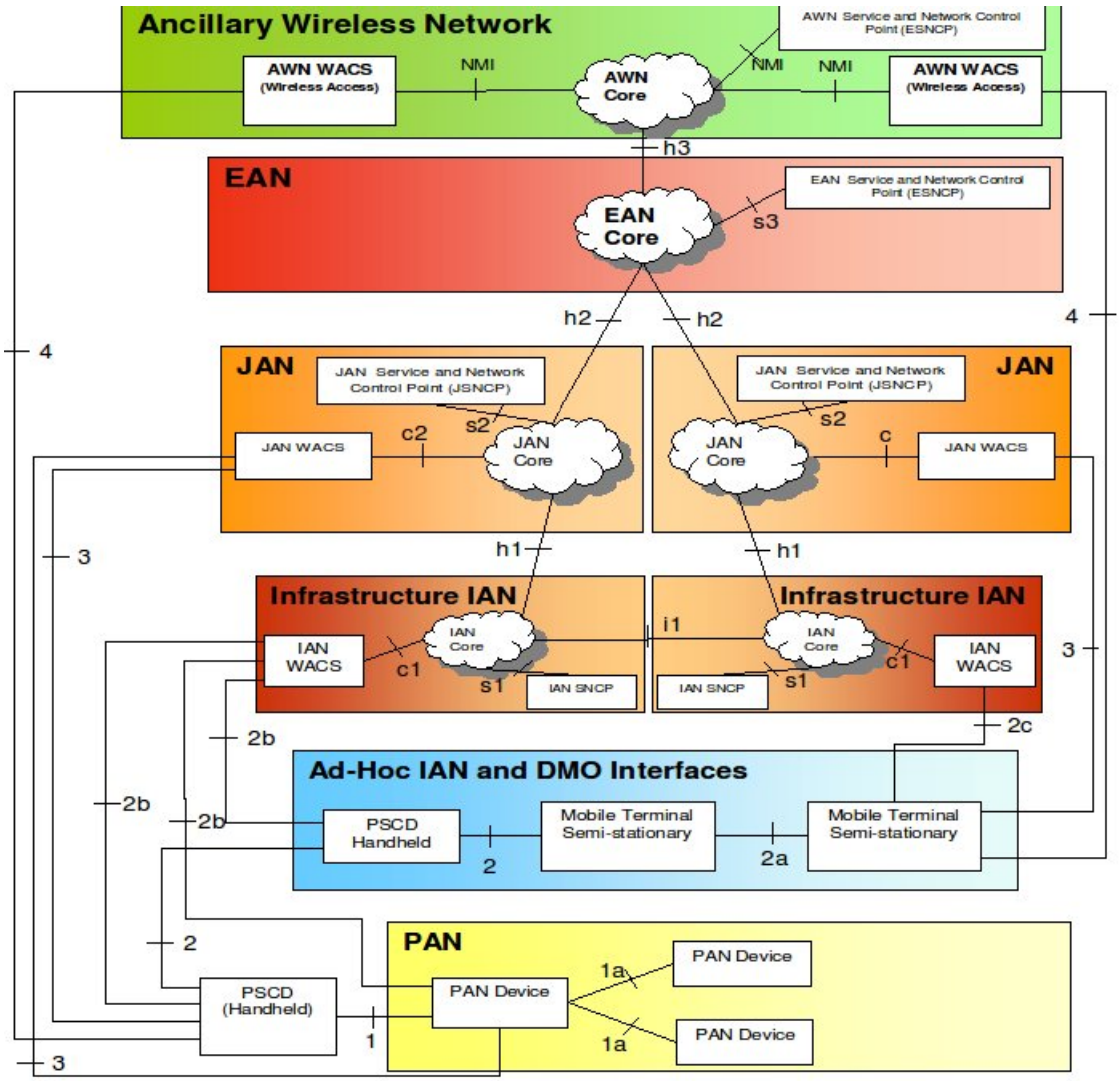

Figure 2.1: Network architecture of public safety communications [4]

Chapter 3

The TETRA Network

3.1 Introduction:

TErrestrial Trunked RAdio (TETRA) is a cellular communications system standardized by the European Telecommunications Standards Institute (ETSI) for critical voice and data mobile communications in advanced Private Mobile Radio (PMR) or Public Access Mobile Radio (PAMR) networks. PMR/PAMR systems are wireless radio sytems set up by a company or group of users to provide a mobile radio services for that group of users only. The overview of the TETRA network is given in this chapter but indeed TETRA is a big wireless communications system and it can't be covered in one chapter. More information about the TETRA can be found in [5, 6, 8]

TETRA provides the capability of trunked, and direct mobile-to-mobile communications. The services offered by TETRA include voice (group or private calls), circuit switched data, packet switched data and Short Data Service (SDS). The main applications of the TETRA system are: utility services, public transport, service fleets, police ambulance and fire brigade (public safety), and other applications like courier services, for construction sites, building security, taxis, and military [5].

TETRA is feature rich which are crucial for public safety and disaster recovery. The following features are the most important features of TETRA:

- Very clear digital speech

- Group calls, personal calls, telephony, direct mode
- Fast call set up times
- Seamless roaming
- Equipment interoperability
- Operational and administrative interoperability and flexibility
- Total system management
- Privacy and security
- Spectrum efficiency- TDMA trunking

The original TETRA standard first envisaged in ETSI was known as the TETRA Voice plus Data (V+D) standard. Because of the need to further evolve and enhance the TETRA, the original V+D standard is now known as TETRA Release 1 and it is one of the narrowband digital mobile communication networks for public protection and disaster relief (PPDR).

TETRA Release 1 is implemented by users from many sectors in over 100 countries throughout the world. In The Netherlands, the C2000 project entails the design, construction and implementation of a countrywide digital radio communication network based on the TETRA standard. The Netherlands' public safety and security organizations, such as the fire brigade, police, ambulance services and the royal marechaussee (gendarmerie) use this network. The TETRA-based radio system, C2000 is built, maintained and monitored by the TETRANED, which is commissioned by the Ministry of the Interior and Kingdom Relations. The TETRA infrastructure of The Netherlands (C2000) is based on the Motorola Dimetra network. According the the 2004 report, there are more than 450 base stations/ antennas-sites and more than 95% nation wide coverage of TETRA network in The Netherlands.

TETRA Release1 is a narrowband wireless communication system with a 25KHz carrier spacing but the TETRA standards have recently been updated to TETRA Release2 which includes the TETRA Enhanced Data Service (TEDS) that can support 50 kHz, 100 kHz and 150 kHz channel bandwidths. In Europe the TEDS capable channels are already taken into account in PMR spectrum decisions even though those channels are not yet readily available for use in all countries [2]. TETRA Release2 is standardized as a wideband digital mobile communication network for PPDR and the broadband version of the PPDR is expected to be built on the existing

technologies as much as possible. TETRA Release2 is not deployed yet and in this document TETRA is to mean TETRA Release1 unless otherwise stated.

3.2 Frequency Allocation and Frame structure in TETRA

TETRA is a cellular system and it requires an uplink and downlink frequencies. The frequency allocation of TETRA can be different in different countries. Generally the TETRA standard was developed to provide optimal performance in the frequency range 300 MHZ to 1000 MHz and outside this range the performance has not been verified. The European Public Safety and Security forces are using the radio frequency band 380-385——390-395 MHz for operation of their TETRA networks [2].

The access scheme in TETRA is TDMA with 4 physical channels per carrier of 25 KHz wide. The basic radio resource is a timeslot lasting 14,167 ms (85/6 ms) transmitting information at a modulation rate of 36 kbit/s. This means that the time slot duration, including guard and ramping times is 510 bit durations. Four time slots make a TETRA frame and 18 frames forms a multi-frame. The 18th frame is always used for control signal communication. 60 multi-frames are grouped to form a hyper-frame. These numbering is used for controlling the resource allocation and generation of key stream segment (KSS) for the air interface encryption. Figure 3.1 shows the structures of hyperframe, multiframe, frame, timeslot, and burst in TETRA.

As TETRA is a wireless communication system, it includes a lot of parameters and standardizations for voice, SMS and packet data communications. The general parameters are summarized in table 3.1.

3.1: General parameters of TETRA [5]

Parameters	**Values**
Carrier spacing	25KHz
Modulation	Pi/4-DQPSK
Carrier data rate	36Kbps
Voice coder rate	ACELP (4.56Kbps, 7.2Kbps gross)
Access method	TDMA with 4 time slots/carrier
User date rate	7.2Kbps per time slot
Maximum data rate	28.8Kbps
protected data rate	up to 19.2Kbps

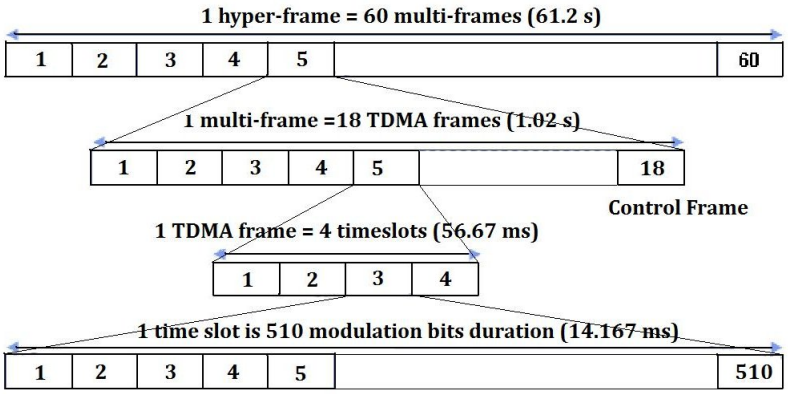

Figure 3.1: Frame structure of TETRA [6]

3.3 TETRA modes and their protocol stacks

TETRA offers services in either of the following three modes. Voice plus Data (V+D), Direct Mode Operation (DMO) and Packet data optimized (PDO).

Voice plus Data (V+D): This is a trunked mode of operation on which the TETRA system provides voice call and data services simultaneously. TETRA provides three types of data services in V+D mode. These are SDS, circuit mode data and packet mode data. This mode of operation is the most commonly used in TETRA systems and in this report TETRA network is to mean the V+D mode unless other wise stated. The protocol stack of the V+D mode in TETRA is given in figure 3.2 .

The TETRA standard is mainly on the lower three layers (physical layer, data link layer and network layer), the upper layers can be different with different types of end user applications. As stated in figure 3.2, the user plane is for speech and circuit mode data and the control plan is for control signals and packet data communications.

Direct Mode Operation (DMO): This mode of operation is when mobile stations in TETRA directly communicate without the need of infrastructure. Only voice call, SDS and circuit mode data are available in this mode. The DMO is also supported under the V+D mode and its protocol stack is the same as the V+D mode without the packet data service and mobility management.

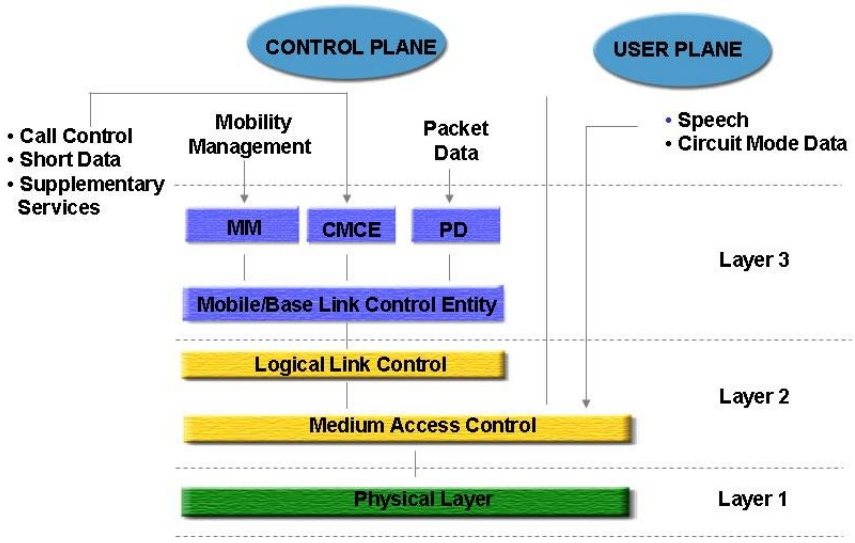

Figure 3.2: protocol stack of V+D [5]

Packet Date Optimized (PDO): This mode is exclusively for higher rate packet data where one full carrier is allotted for packet data communications. Voice or SDS is not supported in this kind of mode of operation. As this mode uses the whole carrier for packet data communication, it is not commonly used in the current deployment of TETRA because higher priority is give to voice communication in the currently deployed TETRA systems.

3.3.1 Overview of TETRA Network Architecture

The TETRA network architecture consists of a number of system entities and interfaces. The main entity of the TETRA system is the Switching and Management Infrastructure (SwMI). The SwMI in TETRA comprises all of the equipment and sub-systems of the TETRA network, including base stations. A typical TETRA network is given in figure 3.3

The Air Interface (AI) is part of the TETRA network which includes the interface between the base station and the mobile terminal (in trunked mode) and the DMO air interface. The Peripheral Equipment Interface (PEI) and and the Inter-System-Interfaces (ISI) are also part of the network to interface other network components.

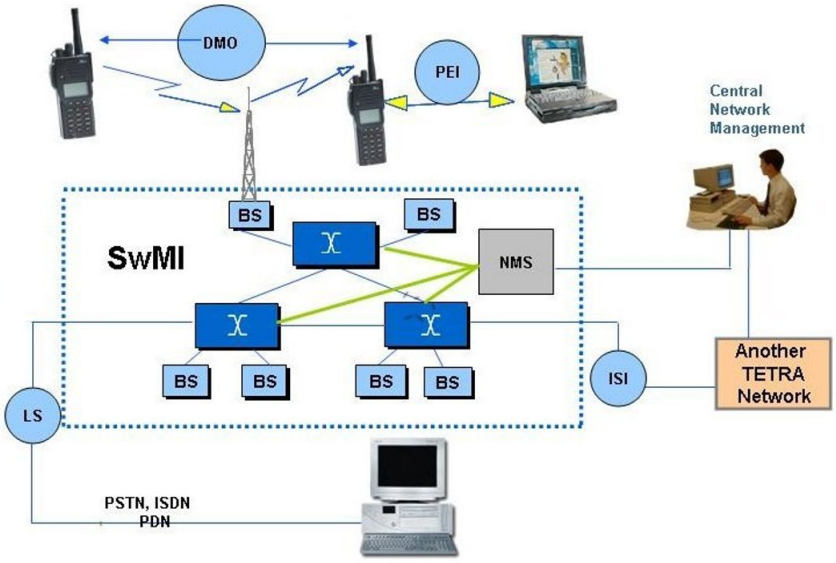

Figure 3.3: TETRA Network architecture [Internet]

Another part of the TETRA network architecture is the Network Management System (NMS) which manages the whole system of a single TETRA network and it is connected to the central network management for coordinating and controlling different TETRA systems. The PSTN, ISDN and PDN interfaces are also part of the Line Station (LS) interface which can also be used for remote dispatchers.

3.4 Physical and Logical channels in TETRA

A physical channel in TETRA is defined by a pair of radio carrier frequencies (downlink and uplink) and a time slot number. There are 4 physical channels per pair of radio frequencies. The physical channels defined in TETRA are Control Physical channels (CP) and Traffic Physical channels (TP). The control physical channels in TETRA are used to carry control information and packet data where as the traffic physical channels are used to carry voice or circuit switched data. There are two types of control physical channels, Main Control Channel (MCCH) and Secondary Control Channel (SCCH). A secondary control channel is assigned for the purpose of packet

data communications and it is termed as Assigned Secondary Control Channel (ASCCH).

TETRA burst structure: A burst is a period of RF carrier that is modulated by a data stream. A burst, therefore, represents the physical content of a timeslot or sub-slot. The burst structure of TETRA is designed to meet the narrow bandwidth of the system. Each burst has the following parts: Ramping and linearization, data bit,tail bits, training sequence, and guard period. A burst can be a full time slot or half of the time slot. Most of the control signals use half of the time slots and the traffic channels use the full time slot. Figure 3.4 shows the burst structure of TETRA.

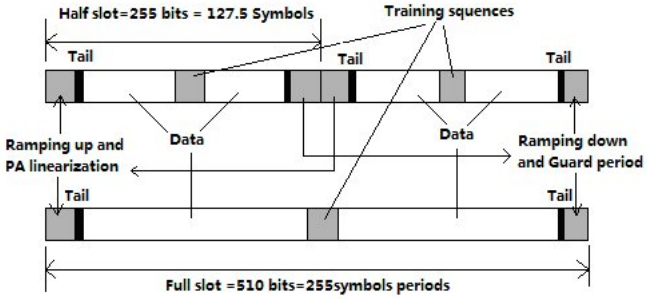

Figure 3.4: Burst stracture of TETRA AI [5]

The logical channels are associated with the data link layer. The data link layer has MAC and LLC sub-sublayers. MAC handles resource scheduling and the MAC layer in TETRA is subdivided into Upper and lower MAC.

Upper MAC: Its main function is for resource scheduling and logical channel multiplexing, like TDMA random access, fragmentation/re-assembling,frame synchronization, measures link quality, addressing managements, air interface encryption/decryption, and power control management.

Lower MAC: The main tasks of this sub layer is channel error protection and mapping a burst from the upper MAC to the physical layer. These are; TDMA channel coding (encoder/decoder), interleaving and scrambling, mapping upper MAC channels with the physical layer slots.

Coding is undertaken by the lower MAC. The coding scheme is different for different logical channels. There are two types of logical channels in TETRA. Traffic CHannels (TCH) and Control CHannels (CCH). The TCH are used to carry voice call or circut switched data and are mapped to the traffic physical channels. The CCH channels are used for control signals

and are mapped to the contraol physical channls. CCH include signaling channels (SCH), STealing CHannels (STCH), Broadcast Control CHannels (BCCH) and Access Assignment CHannels (AACH).

According to the coding used, different traffic channels are defined for speech or data applications and for different data rates. Speech Traffic Channel (TCH/S), Circuit mode traffic channels:(7.2 kbps net rate (TCH/7.2), 4.8 kbps net rate (TCH/4.8), 2.4 kbps net rate (TCH/2.4). Higher net rate up to 28.8 kbps, 19.2 kbps or 9.6 kbps can be used. These higher rates are obtained by allocating up to 4 time slot channels to the same communication.

The Lower MAC in TETRA uses the following coding mechanisms for error detection and error control. Block coding, convolutional coding, puncturing and interleaving. The traffic channels don't use error detection mechanism as it is used for real time data or voice communication and there is no retransmission for this kind of communications but the control channels use error correction and error detection mechanisms because the control information and packet data communications need to be delivered to the upper layers error free.

3.4.1 Random Access in TETRA

The direct mode of operation (DMO) uses CSMA and the trunk mode (V+D) uses slotted Aloha for random accessing mechanisms. In V+D random access takes place on separate control channels so that any collisions don't affect on-going calls. The control burst is used and there are two sub-slots per full slot and this increases the number of access opportunities. The control channel can be on the uplink first slot of the main carrier of every frame.

In order to access the system, a terminal waits for its assigned access slot. There are four different groups of access slots (A, B, C and D) grouped using access codes on the AACH. Mobile terminals in TETRA are allocated to groups in a process called binding and a different grade of service is possible for each group. Group binding is dynamic, operator specific and it is broadcast in an ACCESS DEFINE message.

An access frame is used to control the access of mobile terminals. An access frame may be N number of sub-slots and a mobile terminal may be allowed to request access at the beginning of the access frame. Figure 3.5 shows an access frame in TETRA V+D random access.

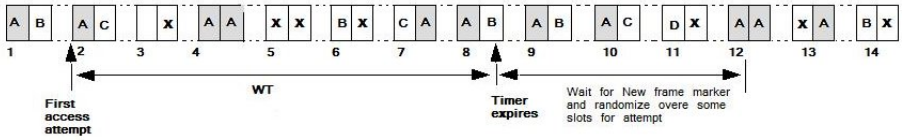

Figure 3.5: Access Frame of the TETRA random Access [5]

The X slot in figure 3.5 indicates that there is no access opportunity. That slot may be used for linearization. The access assignment is so dynamic that if there are collisions the grouping may be changed. There are two access control channels. The ACCESS-DEFINE and ACCESS-ASSIGN.

- ACCESS-DEFINE: It contains slow changing information about the random access parameters like priority and mobile binding to access code, the waiting time (WT), permitted number of retries etc.
- ACCESS-ASSIGN: It has an access field and a usage marker. The access field defines the use of the uplink access slots (access group, access frame number or reserved for CLCH) and the usage marker identifies the traffic in the slot, which for the uplink implies permission for the mobile to use the slot. So, the mobile has to monitor all the AACH if it has requested for access.

The MAC has to wait for an ACCESS-ASSIGN message containing a frame marker for its access code and it is transmitted in the AACH of the corresponding downlink slot. TETRA can assign the uplink and downlink pairs for different purposes. In group call the uplink of all the mobile terminals may not be used so it can be used for other purposes.

Air Interface Encryption(encryption in the MAC)

The air interface encryption scheme encrypts the TETRA MAC Service Data Unit (SDUs) by exclusive oring with key stream segment (KSS) generated from cipher key and offset. Only the data part (SDU) is encrypted but the MAC header is not encrypted and the KSS is restarted for each SDU using different offset value.

The offset (initial value) is constructed from the slot number, frame number, multi and hyper frame numbers and the flag for uplink or downlink. The initial value and the resulting KSS won't repeat for more than 540 hours [5].

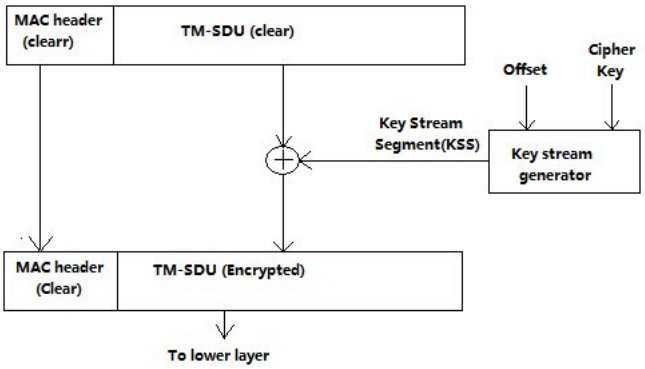

Figure 3.6: Air Interface encryption in TETRA [5]

3.4.2 Advanced link in TETRA

The Link Layer Control (LLC) handles data streams from the MAC and provides error free data to layer3. In order to do this, it adds a frame check sequence (FCS) to the data to be transmitted and checks on reception if the data has been received correctly. MAC uses FEC to protect against channel error but LLC uses ARQ for retransmission and LLC is not used for circuit switched data or speech because of the delay it introduces for retransmission. Broadcast messages are also not processed by the LLC. Only signaling and packet mode data messages are processed by the LLC. LLC provides two types of links. These are:

1. **Basic Link:** In basic link, messages are passed to the MAC to be split into fragments small enough for transmission and then re-assembled at the other end. The message is checked for errors and if error detected, the the entire message must be retransmitted. Used for signalling message(short message).

2. **Advanced Link:** In advanced link, the LLC splits the message small enough for MAC and checks each message part for error and it can retransmit the corrupted segmanet only. It also checks the FCS and if failure, the entire SDU has to be retransmitted. Used for packet mode data transmission (long message). Figure 3.7 shows the advanced link in TETRA.

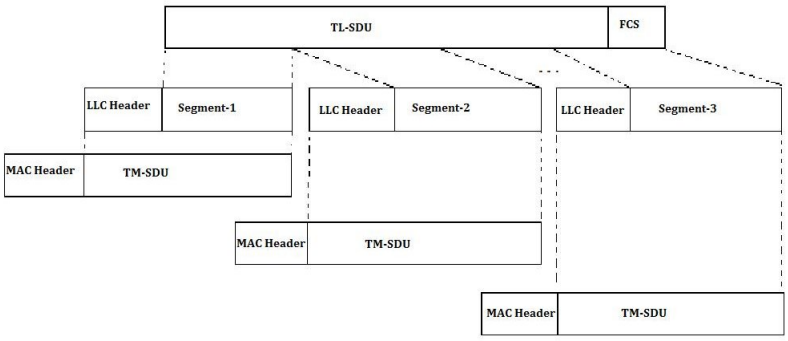

Figure 3.7: Advanced link in TETRA [6]

3.5 The Network Layer (Layer3) of TETRA

The network layer in TETRA provides additional communication functions that extend the functionality of the data link layer into more flexible internetworking facilities. It is generally the level at which equipments of different capabilities and/or different manufacturers inter-operate. The main functions of the network layer are naming and addressing, connection control, and message routing and flow control [5].

The network layer which is part of the control plane as described in figure 3.2 above is divided into two sub-layers. These are, mobile/base link entity (MLE) and the sub-network access functions.

unlike GSM handover decisions in TETRA network are undertaken by the MS not by the infrastructure . The MLE sub-layer within the MS is concerned with the radio link management like measuring signal strength in adjacent cells for handover decisions. The MS relies on the MLE to gain access to all communication resources according to the MM entity's request which activates and deactivates the MLE.

The sub-network access functions have three protocol entities; mobility management (MM), circuit mode connection Entity (CMCE), and packet data (PD). mobility management protocol deals with network signaling aspects of authentication and registration procedures. Circuit mode connection entity (CMCE) provides services to an end user application in call control, supplementary services and short data services. Packet mode data services (PD)

provides packet data communications in TETRA. Connection oriented and connectionless packet data service are provided by the PD protocol.

3.5.1 Sub-Network Dependent Convergence Protocol (SNDCP)

The Sub-network Dependent Convergence Protocol (SNDCP) is a TETRA specific network layer protocol between the MS and SwMI. It has two main functions as described below [6].

1. To negotiate and maintain Packet Data Protocol (PDP) contexts between an MS and the SwMI: A unique PDP context is established for each PDP active on the network.
2. To control PDP data transfer between MS and SwMI. SNDCP allows the service user to select the acknowledged or unacknowledged layer 2 services and it provides mechanisms by which data may be compressed before transmitted over the air interface.

Before a MS may gain access to any SNDCP services, it firstly goes through a packet data registration procedure, called PDP context activation. Context activation is initiated by the MS. PDP context activation involves the negotiation of a PDP address (e.g. an IPv4 address) and other parameters to be used during data transfer. PDP data transfer normally takes place over an assigned secondary control channel (assigned SCCH), termed as Packet Data CHannel (PDCH). An advanced link is set up before data transfer may begin on the PDCH. When a MS has data to transfer, it implicitly requests permission to switch to the PDCH. If accepted, the SwMI responds with a channel allocation, directing the MS to a PDCH.

3.5.2 Network security Management in TETRA

TETRA has specified network security in the standards but the implementation can be different for different network operators. The standard specifies the following security measures: multilevel authentication, air interface encryption (MAC layer), user anonymity , terminal enable/disable, end-to-end encryption (for U-plane), and frequency hopping

There are different types of encryption keys in TETRA. These are: *Derived Cipher Key (DCK)* which is derived during the authentication process, *Common Cipher Key (CCK)* generated by the infrastructure and distributed by

sealing it with DCK (this process is called over the air re-keying (OTAR)), *Group Cipher Key (GCK)*, generated similar to CCK, *Static Cipher Key (SCK)* generated as CCK but is not changed by an authentication procedure and is used for encryption prior to authentication.

Security in TETRA is defined in terms of classes. There are three classes and each class has associated features that are mandatory or optional [16]. In *Class1:* Authentication, OTAR and device enable/disable are optional and there is no encryption in this class. In *Class2:* Authentication, OTAR and device enable/disable are optional and encryption is mandatory in this class. And in *Class3:* Authentication, OTAR and encryption are mandatory and device enable/disable is recommended.

If a cell supports class1 only, it may not be wise to send un encrypted data through this cell as its security is very low because encryption is not supported in class1. It is also possible in TETRA that the SwMI may send D-CK CHANGE DEMAND on the control channels or on the assigned channels to invoke transitions between any of the security classes during communication.

Chapter 4

The FIGO Network

4.1 Introduction

FIGO is a robust communication system, which is intended for the public safety communications and it can actually be used to build any secure private network. It is designed by Twente Institute for Wireless and Mobile Communications (TI-WMC) and this chapter is dedicated to give a brief explanation of the FIGO network.

The beauty of FIGO is that it can use many of the excising communication systems at any time in a secure way to interlink communication devices and this makes it to be a robust communication system. It provides a fast and easy set-up of a reliable local adhoc communications in the incident area, while providing a reliable connection to the infrastructure networks in the back office [9] and it can use the back office to interconnect separated local adhoc networks. There are two main types of nodes in FIGO network and the network is formed by the combination of one or two of these node types. These nodes are:

- Vehicle Node (VN): Is a mobile node that can be mounted on a car or vehicle.
- Back Office Node (BON): Is a fixed node on the infrastructure that acts as a server or gateway in the FIGO network.

The local adhoc communication is formed by the vehicular nodes (VNs) and this adhoc network can operate without any interaction to the BON

in the infrastructure. This adhoc network can also be connected to the BON through many of the existing communication systems using OpenVPN. FIGO nodes form a single mesh network, both the vehicular nodes as well as the back office node are part of the mesh [9]. So, as compared to the standard mesh adhoc network which consists only wireless nodes, a FIGO mesh is extended with the back office node by using secure tunnels between the back office node and the vehicular nodes. The FIGO mesh network is implemented below the network layer (layer 3) and it is a layer 2.5 protocol. In this way, the FIGO mesh is a broadcast segment and becomes a single IP subnet, on top of which existing layer 3 functions can be used. The meshing protocol uses appropriate link quality measurement mechanisms to optimize the QoS while ensuring stable and robust communication [9].

Client devices can get access through the FIGO network where the FIGO nodes act as access point and the FIGO mesh network behaves like a switch for the client device. The FIGO system also contains additional functionality, like a DHCP server to provide other services.

4.2 Network Architecture of FIGO

The FIGO network architecture includes the main FIGO nodes, the VNs and BON, and it is also includes external device interfaces like clients or other networks. The FIGO management device which is dedicated for maintenance and operation of FIGO nodes is also part of the FIGO network. The general FIGO network architecture is given in figure 4.1

The FIGO network architecture consists different internal and external interfaces. These interfaces are explained below [9].

Client devices connecting to VN's (E1-VN): These client devices are devices like laptops, PDA's, or smart phones that are typically used in a mobile environments. Client devices can directly be connect to VN's using wireless or wired IEEE 802-type technologies. FIGO supports at least IEEE 802.11b/g (wireless) or LAN, i.e. IEEE 802.3 (wired Ethernet).

Client devices connecting to the BON (E1-BON): These client devices are devices like desktop PC's, that are typically used in static environments in the back office. They can directly be connect to FIGO using wired IEEE 802-type technologies. FIGO supports at least IEEE 802.3 (wired Ethernet) for these kind of interfaces.

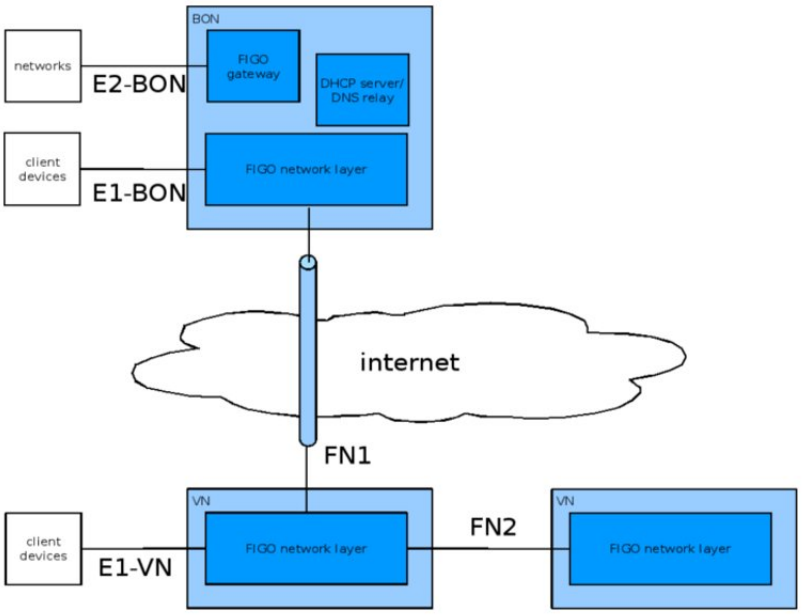

Figure 4.1: Network Architecture of FIGO [9]

The back office gateway interface (E2-BON): The back-office gateway interface will either be configured as a L2 gateway or a L3 gateway to the customer network. When it is configured as L2 gateway, the FIGO back office will act as a switch, connecting the entire FIGO network to the customer infrastructure at L2. Wired Ethernet (IEEE 802.3) is used for this. When it is configured as L3 router, it will act as a default gateway for client devices in the FIGO network. This interface uses a wired Ethernet (IEEE 802.3) connection. Connection to other networks using VPN is also possible when these networks are not located in the same physical location as the FIGO back office.

The link between two FIGO Vehicular Nodes (FN2): A connection between two FIGO adhoc nodes may consist of different types of links; It can be a single link between vehicular nodes, via the back office node, or via other vehicular nodes. When ad-hoc nodes are within communication range of the adhoc mesh interfaces, they automatically form an ad-hoc FIGO network. FIGO supports at least IEEE 802.11b/g in the ad-hoc mode.

Link between a VN and BON (FN1): A connection between a FIGO Vehicular Node and the FIGO back-office typically consists of multiple links

using a different access technology. Vehicular Nodes automatically set up a connection with the back office via VPN tunnels. FIGO supports at least IEEE 802.11b/g , UMTS / HSPA, IEEE 802.3 (wired Ethernet) for this interface. This link can also be narrow band links like the satcom and TETRA links. In this paper the TETRA like is studied to be incorporate it in the FIGO network as one of the interfaces between the adhoc network and the the node in the infrastructure (the BON).

4.3 Flame protocol of the FIGO Network

Forward layer messing (Flame) is a protocol layer for meshing designed and developed at the Twenete Institute for Wireless and Mobile Communications (TI-WMC). Meshing in wireless networks is required when the wireless coverage of a single node in the network is not large enough to reach all other nodes in the network. The mesh in FIGO doesn't not only include the wireless nodes but it extends to the infrastructure to mesh the BON with the adhoc nodes. Flame provides a layer 2.5 meshing protocol and figure 4.2 shows Flame on the TCP/IP layered protocol stack.

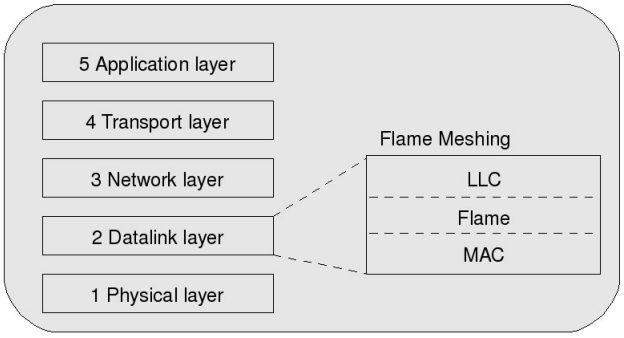

Figure 4.2: TCP/IP protocol stack with flame layer added [10]

There are two types Flame protocols designed for the FIGO network. These are Flame1 and Flame2. Currently Flame1 is implemented in the FIGO network whereas Flame2 is under development. In the subsections given blow, Flame1 and Flame2 protocols are briefly described in order to give a highlight for the analysis of the TETRA link performance for these two protocols.

4.3.1 Flame1 protocol

Flame1 is a proactive table driven routing protocol similar to the Destination-Sequenced Distance Vector (DSDV) routing protocol and it makes sure that, to a certain limit, node pairs in a network can communicate with each other, even if the network is not fully connected [10]. Flame1 does this by building routing tables in each node, just like normal layer2 switches do. In order for a Flame1 to work properly it needs to create a routing table on every FIGO node. The routing table is developed by the following mechanisms.

Path Discovery algorithms: Path discovery is done by broadcasting packets. Every node sends out a broadcast Path Update (PU) packet to its neighbor at every PU-interval. When another node receives this packet, it knows it can access the node. It also forwards the PU packet, so that neighbors (which are two hops away) discover a route to the node which initialized the PU packet [10]. After all the nodes broadcast PU, the best path between two nodes is selected using the routing algorithm. There are some link quality measurement criteria for the routing algorithm to consider in Flame1. These criteria are Packet Delivery Ratio (PDR), Received Signal Strength Indicator (RSSI) and Transmission rate.

A message called Neighbor Acknowledgement (NA) is sent between neighbors to measure the cost of the link for link quality measurement. The routing algorithm has to find paths between two nodes with the highest quality. Therefore, the algorithm has to measure the quality of each link in the network. All different paths between two nodes should be taken into consideration, where the path with the highest quality should be used for data transfer. A link quality algorithm metrics called Estimated Transmission Time (ETT) based on PDR is used in the current FIGO network to compare different paths between two nodes [17].

Bridged nodes: Flame can be seen as an advanced tunnel. It hooks into an existing network, accepts packets from other networks, encapsulates it in Flame packets and routes it to other flame nodes, where the packets are decapsulated again and injected into network outside the flame domain [10]. Flame takes care of the routing in the FIGO mesh network and the nodes in the Flame domain are called Flame nodes. The nodes outside the Flame domain are called bridged nodes. Bridged nodes are discovered when they are trying to send packets (broadcast or unicast). The PU packet is used to report bridged nodes to other Flame nodes so that they will update their routing tables.

OpenVPN tunnels: OpenVPN tunnels are used to connect VN's to the BON. Because all nodes need to be able to determine and use the best link to the BON, the Flame routing protocol needs to be used in the BON as well. The VNs have some knowledge about the link used to send OpenVPN traffic to the BON (UMTS, WiFi, Ethernet) but at the BON endpoint of the VPN tunnel it is just broadband connection to the Internet and the BON doesn't know which VN is connected to which communication system.

4.3.2 Flame2 protocol

In Flame1, all nodes attempt to communicate topology information about themselves to every other node in the network. This causes a significant amount of control information to pass between the Vehicle Nodes (VNs) and the Back Office Node (BON) through the OpenVPN. This is a problem, because the wireless (mostly cellular) link will be overloaded as the number of VNs in the network increases.

Flame2 is a hierarchical routing protocol that isolates local traffic as much as possible avoiding to pass through the OpenVPN, which will improve both the scalability and performance of the network. It differentiates the network into different levels and this allows to tailor a specific routing solutions to each level. Flame2 introduced the idea of gateways and cluster-heads that play a great role in isolating the local communication. Gateways (GW) in Flame2 are VNs or Personal Nodes (PNs) which are one-hop connected to a BON. A GW advertises about its link (link to the BON) to the other neighboring nodes with some cost indicator and a node selects a particular GW as its default route to the BON if it finds out that the link to the BON through this GW is the best.

Because of the hierarchical routing in Flame2, nodes are required to register at the BON, GWs or cluster-heads. A cluster-head is a FIGO node which acts as a routing level exit point for a cluster of nodes. All the nodes under the cluster-head can communicate with each other at least through the cluster-head and these cluster-heads will select their default GWs if their direct link to the BON is not available or the link through the GW is better than their direct link to the BON. Because of the GWs and cluster-heads, there will be FIGO nodes which don't require the BON to have a connectivity among each other and these FIGO nodes are called Basic Mesh Set (BNS) in Flame2. A FIGO node which has another FIGO node as its default route to a higher level routing is called a child node.

Node Registration in Flame2

When a FIGO node is first deployed in the adhoc network, it exchanges routing information with its adhoc neighbor nodes through Path Update and Neighbor Acknowledgement control messages and checks if a direct link to the BON is available. Its neighbors can be cluster-heads, GWs, child nodes, or clients. If this FIGO nodes finds out that its direct link to the BON is better than using a other cluster-head or GW as an exit to the BON, it will register itself at the BON as a GW. Once it is registered as a GW, it will advertise about its direct link to all its neighbors with some cost function indicator. If the neighbors (cluster-heads, GWs, or child nodes) find out that the best link to the BON is through this GW, they will register at this GW and they have to inform this new GW about the other nodes which are registered through them. The new GW will also inform the BON about the nodes which are reachable through it. The list of nodes which are reachable through a cluster-head or a GW is called the Node Base (NB). A node which was registered at the cluster-heads or other GWs can directly register at the new GW if that is the best link to the BON. The clients are not part of the FIGO network and they are bridged nodes directly connected to a cluster-head or a GW which is directly accessible but they are included in the NB of the cluster-heads or the GWs.

Hierarchical routing in Flame2

The BON is the top in the hierarchical routing in Flame2. There can be multiple BONs in FIGO but there is only a single BON in the currently implemented FIGO network. The next level of routing is between the GWs through the BON. Then the next level of routing is between cluster-heads. A cluster-head can be registered under another higher layer cluster-head and the routing level continues until it reaches the child nodes.

As depicted in figure 4.3, **Level 0 (L0)** is the routing between the BONs of the network, **Level 1 (L1):** is the routing between the GW nodes of Basic Mesh Sets (BMSs), via the BONs, and **Level 2 (L2):** is the routing between the cluster-heads belonging to a single BMS, which provides support for inter-cluster routing. Because the could be cluster-heads registered under different cluster-head, the level of routing may increase until the last cluster-head with only child nodes and/or clients [13].

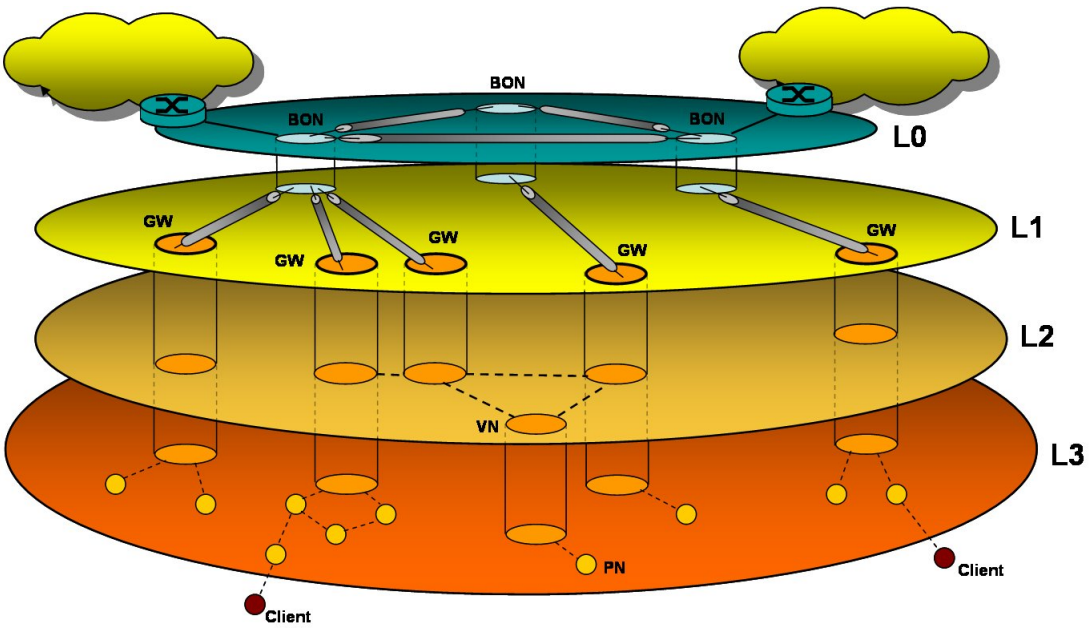

Figure 4.3: Hierarchical Routing Levels in Flame2 [13]

Chapter 5

Interfacing FIGO with TETRA

TETRA and FIGO are two completely different communication systems. How should this two different systems be interfaced and some of the practical aspects are given in this chapter.

5.1 Introduction to TETRA Interfaces

TETRA supports different types of interfaces. These interfaces can be categorized as inter-system interface (ISI) and subscriber access interface. ISI is standardized to interlink two independent TETRA networks. Subscriber access interface is an interface in the TETRA network for subscribers and it can be a Line Station (LS) or Mobile station (MS) interfaces.

The ISI interface is only for interfacing independent TETRA systems and it can't be used for interfacing with the FIGO network. The subscribers access interface is going to be used for interfacing with the FIGO network by integrating TETRA cards on the FIGO nodes. TETRA offers access to external data terminals in the air interface through Mobile Terminal (MT) and fixed line interface through the Network Terminal (NT). The Interface between an external data terminal termed as terminal equipment (TE) and a TETRA MT is standardized as the Peripheral Equipment Interface (PEI). Based on their capability there are different types of TE, MT, and NT defined in the TETRA Network.

The FIGO nodes are data terminal equipment type2 (TE2) and they will require mobile terminal type 2 (MT2) TETRA cards to get access to the TETRA network. TE and MT in this document is to mean TE2 and MT2 unless otherwise stated. The fixed line interface of TETRA is not required in the FIGO network as the TETRA gateways can be accessed from the BON similar to the Internet gateway routers.

5.2 The Peripheral Equipment Interface (PEI)

The Interface between an external data terminal and a TETRA mobile system/terminal (MS/MT) is standardized as the Peripheral Equipment Interface (PEI). The TETRA PEI provides a link between a data terminal (TE2) such as a Personal Computer (PC) or specialized data terminal and a TETRA mobile terminal (MT2). The PEI provides external data devices access to the services offered by a TETRA network. The TETRA PEI supports the following functionalities [8]. Transmission and reception of packet and circuit data, transmission and reception of short data, set-up and control of speech calls, access to general information of MT2 and the network, and access to user applications located in MT2.

PEI is not designed to send voice calls. Only the call set up, maintenance and clear down signalling for speech calls are sent on the PEI. The actual voice packets go directly from the MT codec. The FIGO terminals are assumed to be packet data terminals but all the capabilities of the PEI is given here for general information and if required the FIGO node can be configured to send other types data. The PEI interface consists of three components. These are: The AT command, packet data, and TNP1. The PEI interface is given in figure 5.1

5.2.1 The AT Commands in TETRA

The main purpose of AT commands is controlling modems from a PC or other intelligent terminal. AT commands have been adopted by many wireless systems as a means for accessing data services and are therefore used as a basis in the TETRA PEI to give access to TETRA services. The TETRA services available using AT commands include call control, mobility management and SDS. In addition to these, there are commands to access the radio configuration and storage parameters.

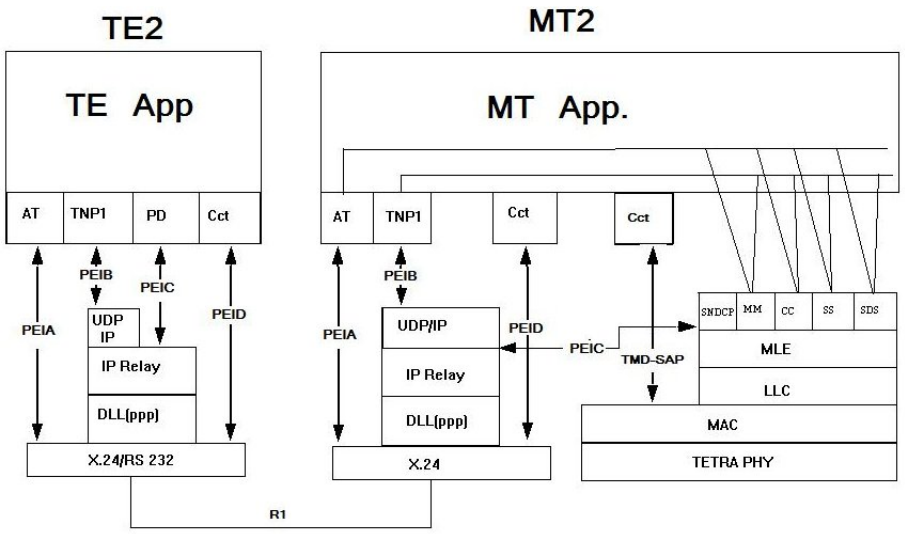

Figure 5.1: The Peripheral Equipment interface in TETRA [8]

There are also AT commands that can be used to set up voice calls. With reference to figure 5.1, the AT commands use the PEIA interface for commands. Subsequent data calls will use the PEID interface for circuit mode data or PEIC interface for packet data.

The PEI AT command has three main states and the MT effectively operates in one of the three states, namely "AT Command" state, "AT Circuit mode data" (On-line data) state, and the "TNP1 or Packet Data" state [8]. The outline state diagram is shown in figure 5.2.

AT command state: Both TE and MT enter this state on initialization or PEI link establishment. It is always entered from circuit mode data state when any ongoing call is cleared or when the TE sends a recognized escape sequence. With reference to figure 5.1 the signalling is sent on the interface PEI A. In this state all commands and responses will be accepted and acted on. This is the state where typically the MT operating modes are set, Mobility Management Information (MMI) is received, SDS messages are sent and received and circuit mode calls are established [8]. The AT command state uses the V.24 or RS232 interface.

AT circuit mode data state: A Circuit Mode Data call is in progress whilst in this state and all AT commands are ignored. Signalling and circuit mode data received from the TE are forwarded as data to the appropriate

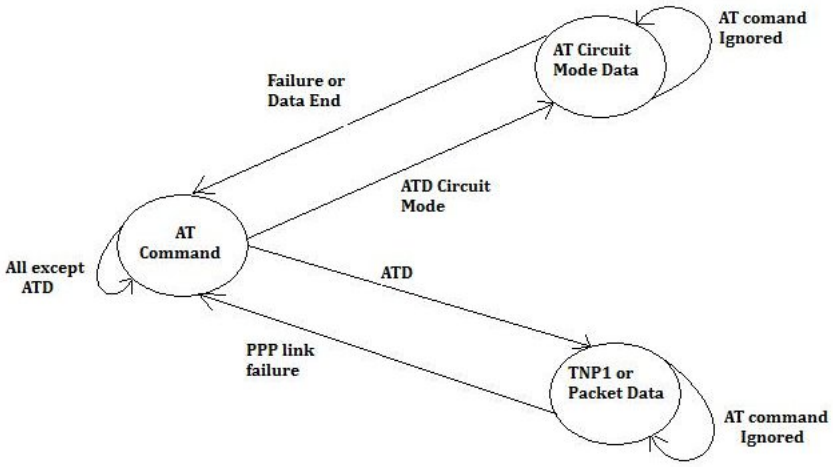

Figure 5.2: AT Command state [8]

destination. With reference to figure 5.1 the signalling is sent on the interface PEID. The MT will monitor the air interface for call maintenance and clear down signalling.

TNP1 or packet data state: A TNP1 or Packet Data session is in progress whilst in this state and all AT commands are ignored. The correct destination for signalling in either direction is determined by the UDP/IP addressing. Signalling or data received from the TE by the MT is forwarded to the appropriate destination . With reference to 5.1 the signalling is sent on the interface PEIB or PEIC. This state has two sub-states called "local" and "wide". Further explanation is given in the following sub-sections.

5.2.2 TETRA Network Protocol type1 (TNP1) Services

The TNP1 specifies a protocol to be used over the TETRA PEI designed to allow the TE to have control over the TETRA services. This includes mobility management; call control, SDS and supplementary services. In addition there are commands to access the radio configuration and storage parameters. TNP1 itself is based on a connectionless, point-to-point, unreliable network layer protocol. The difference between TNP1 and AT command is that TNP1 primitives can share the data link with the packet data services

[8]. The TNP1 services are depicted in figure 5.3

In order to transfer messages over the PEI, TNP1 uses the services of UDP/IP, IP Relay and the PEI DLL (PPP). The MT should be able to support all options to allow for different TE connections. For TNP1 to operate, an IP connection between MT2 and TE2 has to be established. The network layer establishment is based on the PPP and UDP/IP combination. After receiving an administrative open event from the service user the TNP1 entity, as a kind of IP application, should ask for a socket from the IP service task. From this point the TNP1 entity is ready for service.

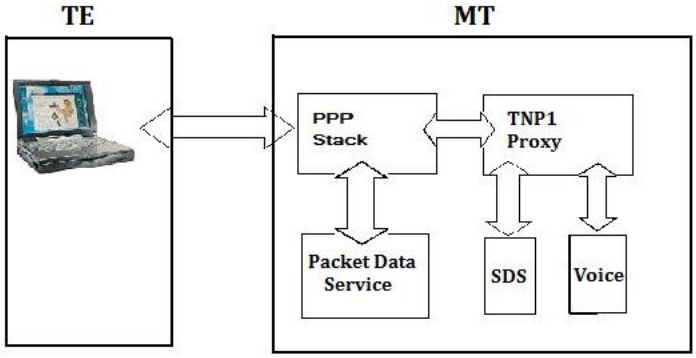

Figure 5.3: TETRA Network Protocol Services (TNP1) [18]

IP addressing in PEI

There are two modes of IP operation supported by the PEI. Each mode has different addressing requirements. In total the TE/MT combination needs three different addresses. One is “TE IP” and this is a fixed IP address used by all TEs and its value is 10.0.0.100. The second is “MT IP” and it is a fixed IP address used by all MTs and its value is 10.0.0.101. The third is “MS IP” and it is a wide address given to the MT by the SwMI on context activation. These addresses are used differently in the two modes. In both modes IP packets from the TE2 to MT2 for internal applications (local communications) have to use the ”MT IP” address and the default mode for the MT is wide address (MS IP) [8].

In the local mode TNP1 runs over the PEI and is used for TNP1 services but not for packet data transfer. The “TE IP” and “MT IP” used in this

mode. In the wide mode all TNP1 services and packet data transfer towards the SwMI are also possible. The "MS IP" is used in this mode.

5.2.3 Packet data services in TETRA PEI

Packet data service in the TETRA PEI can be offered in the wide mode of the TNP1 in parallel with the TNP1 primitives and it is also possible to set the terminal to be packet data only using AT commands. Generally a user can access packet data service in TETRA at two points. One is at the PEI which is the interface for a TE2 with MT2 terminal and the other is the Line Station(LS) interface of the fixed TETRA gateway. The PEI is standardized RS232 connection to the MT with a PPP link over it. The Fixed line TETRA gateway is commonly an Ethernet based interface. The protocol stack of the TETRA packet data communication is given in figure 5.4.

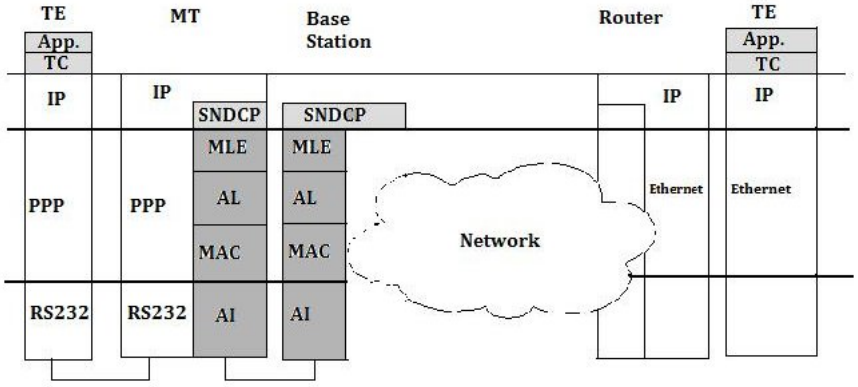

Figure 5.4: protocol stack of packet data in TETRA [18]

When a TE is connected to a mobile terminal (MT), it assumes as if it is connected to a wire line modem but there are some difference between a TETRA PEI and a wire line modem [7]. One of the difference is that a wire line modem provides a circuit mode connection, whereas the Packet Data service in TETRA is entirely packet switched. Another major difference is that, the PPP link in PEI is terminated in the MS but not at the network which is the case in wired line.

In this study the TETRA packet data service is used to interface FIGO

with TETRA network but the other services can also be supported in parallel if required when a TETRA mobile terminal is used as a TETRA modem. There are also TETRA modems in the market which are meant only for packet data communication in the TETRA network and these may not support the other services like the TNP1 primitives.

5.3 Practical aspect of the TETRA-FIGO interface

Currently the packet data communication in TETRA network is based on the Motorola Dimetra network which is most commonly used in most of the TETRA networks and the practical aspects given in this document are based on Motorola Dimetra network.

The PEI interface between the FIGO node (TE2) and the TETRA modem (MT2) is used for packet data transfer communications and it uses different protocols. These are RS232, PPP, and IPCP. The overview of these protocols is given below:

RS233: This is very common physical layer, 8-wire connection. The transmit data (TD) and receive data (RD) are used to carry the data between the MT2 and TE2. The request to send (RTS) and clear to send (CTS) are used for flow control and the data set ready (DSR), data terminal ready (DTR) and data carrier detect (DCD) are used for state management between the MT2 and TE2.

PPP Link: Generally the Point-to-Point Protocol is designed for simple link which transports packets between two peers. It uses the link configuration protocol (LCP) to establish a PPP link. The LCP is also used to negotiate what type of authentication to be used between the TE2 and MT2. The full document of LCP is specified in RFC 1661 and RFC 1662. There are two types of authentication protocols supported in the PPP link of the PEI. These are: Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP)

IPCP: The IP Control Protocol (IPCP) is responsible for configuring,enabling, and disabling the IP protocol modules on the TE2 and MT2 ends. This is all about assigning IP addresses. There are two types of addressing in the wide mode of PEI link.

Static addressing: The TE2 can be configured with an IP address and configured to request that static address when it is connected to the MT2. The MT2 will then request the SwMI a permission to use that address. If the address matches with the provisioned addresses, the SwMI will accept this request if not it will reject it.

Dynamic Addressing: The TE2 specifies that it wants to use an IP address 0.0.0.0 then the MT2 will ask the SwMI for a dynamic address. Then the SwMI will either extract UCS provisioned address and hand that address to the MT2 or obtain an address by the use of DHCP server in the customer network.

The practical implementation of TETRA interface in the FIGO network will require a small script and some commands. The establishment of a link through the TETRA network is accomplished with PEI standard and the PEI in TETRA supports a PPP link over an RS232 physical interface to external data terminals. The FIGO system is using Linux operating system and Linux supports a Point-to-Point Protocol daemon (pppd) which is designed for creating a PPP link with external devices. The pppd in Linux handles the PPP link and IPCP protocols described above.

The pppd daemon works together with the kernel PPP driver to establish and maintain a PPP link with another system (called the peer) and to negotiate Internet Protocol (IP) addresses for each end of the link. The pppd can also authenticate the peer and/or supply authentication information to the peer. The pppd demon can use either PAP or CHAP authentication mechanism. The pppd demon undertakes the Link configuration protocol (LCP) and IP configuration protocols (IPCP).

The pppd demon by itself has many options and a script is required to handle these options. Before the establishment of the PPP link, AT commands are sent to the modem and this is done using the chat program in Linux. The chat program defines a conversational exchange between the terminal equipment and the modem (MT).

Chapter 6

Performance Analysis of the TETRA PDCH for Flame protocols

Before we come to the architectural design of the TETRA-FIGO integration, lets see the performance of the TETRA packet data channel (TETRA PDCH) for the current Flame protocols. As described in chapter 4, there are two types of meshing protocols for the FIGO network. These meshing protocols are named as Flame1 and Flame2. Flame1 is implemented on the current FIGO network and Flame2 is still under development. This chapter analysis the performance of the TETRA PDCH for the two Flame protocols in terms of the control message traffic overheads and header overheads. In order to give a brief understanding on the capacity of the TETRA PDCH, the highlight of packet data communications in TETRA is first given in the following sub-section.

6.1 Packet data communication in TETRA

TETRA is a narrow band wireless communication system which is mainly designed for voice communications and it provides low packet data rate applications on the Assigned Secondary Control Channel (Assigned SCCH) termed as Packet Data CHannel (PDCH). Packet data communication in TETRA is provided through a PPP link with the TETRA modem on the

Peripheral Equipment Interface (PEI), which is one of the subscribers' interfaces of the TETRA standard. Messages on the air interface are sent using the TETRA specific Sub-network Divergence Control Protocol (SNDCP) service. SNDCP is a protocol for packet switched data communication between the SwMI on the TETRA infrastructure and the mobile station. The fixed line interface with TETRA is through a TETRA gateway and it supports different protocols. The fixed line is similar to the Internet world and it should support a TCP/IP protocol stack. It is commonly an Ethernet interface. The protocol stack of packet data communication in TETRA is given in figure 5.4.

Let's first see the capacity of the TETRA PDCH to send IP packets through it. It is assumed that a single TETRA PDCH is available and there is only a single user on the available TETRA PDCH [7]. Note that a single PDCH in TETRA is a channel that uses a single time slot out of the four time slots on a single carrier. A message in TETRA is sent over the air interface using the SNDCP service. The message is split into segments (in MAC and LLC) and are mapped into the slots or air interface bursts. Since each slot carries a fixed amount of bits, the number of slots required for an IP datagram to be sent through the TETRA air interface is given in table 6.1. From the table, the maximum IP datagram size in the uplink and downlink for a given number of slots is different. This is because the burst strictures of the uplink and downlink in TETRA are different. For more information, refer to [7, 6].

Additional slots	Max IP datagram size (Uplink/Downlink)	Additional slots	Max IP datagram size (Uplink/Downlink)
-	50/49	7	250/243
1	79/76	8	278/271
2	107/104	9	307/298
3	136/132	10	335/326
4	164/160	11	364/354
5	193/187	12	392/382
6	221/215	13	421/409

Table 6.1: Additional slot requirement for IP datagram on TETRA air interface [7]

For example, in the downlink, any IP datagram from 21 (minimum IP datagram size) to 49 bytes will consume the same amount of resources. Note that the IP datagram size includes the IP and transport protocol header. For UDP, the IP and UDP overhead is 28 bytes. In order to send 130 bytes

of user data, an IP datagram of length 158 bytes must be sent and this requires 4 additional downlink slots. A single TETRA PDCH will in a good condition offer a throughput of slightly above **3000 bps** including the IP headers [18].

D.I. Axiotis and D.Xenikos have undertaken measurements on the performance of UDP/IP over the TETRA PDCH and they have come with a result of higher throughput, less than 2 sec delay and less datagram loss when the IP datagram size is in the range of 150-250 bytes [7]. This measurement is for a single TETRA PDCH. Currently no measurement is undertaken for TCP/IP datagram over TETRA PDCH in the literature and it is not recommended to be used for TETRA PDCH as it generates a lot of control signals for connection establishment, congestion control and other control messages. There are no measurements undertaken for IP packet over multi-slot TETRA system because multi-slot capable TETRA terminals are only recently coming to the market.

According to the measurement undertaken on TETRA PDCH, a single PDCH in TETRA can provide a mean UDP/IP throughput of **1.5 Kbps**. This is the throughput of a single TETRA PDCH link excluding the UDP/IP overheads of the TETRA system. So, there is a 1500 bps capacity TETRA link available over which we can send our own data or protocol messages.

6.2 TETRA PDCH performance analysis for Flame1

In Flame1, all nodes attempt to communicate information about themselves to every other node in the network and it treats the OpenVPN link between the Vehicle Nodes (VNs) and the Back Office Node (the BON) almost similar to the local adhoc links (mostly Wi-Fi) between the vehicular nodes and this causes a significant amount of control information to pass through the narrowband link to the BON. The traffic overheads analyzed in this section are control messages generated by the Flame protocol, other control messages like ARP and header overheads.

Before analyzing the traffic overheads generated by Flame1, lets see the control messages and the packet format of Flame in the FIGO network. Flame1 is a layer 2.5 protocol and the packet format of Flame1 is given in figure 6.1. The IV field is an initial value counter for encryption purpose.

There are three types of Flame1 payloads defined in the FIGO network.

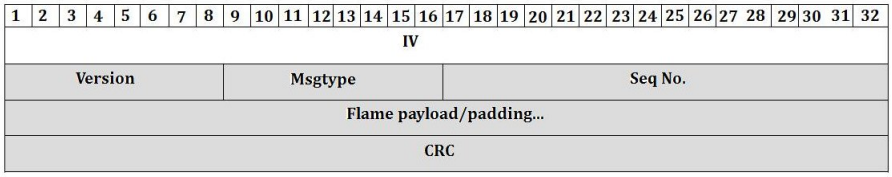

Figure 6.1: Flame frame format [14]

These payloads are: Type0-Data, Type1-Path Update (PU), and Type3-Neighbor Acknowledgement (NA) [14]. The data part (Type 0) of Flame is not generated by the Flame itself as the FIGO nodes act as switches (access points) and it is generated from clients which get access through the FIGO network. The packet size of the type0 payload depends on the MAC headers and upper layer payload. The PU and NA are Flame control messages generated by Flame itself and their traffic load to the network depends on the packet size and on how often these control messages are sent through the narrowband link for updating the routing information.

In this section the traffic overhead over the narrowband TETRA PDCH link between the VNs and the BON is analyzed as this is the bottle neck for the network. The traffic overheads analyzed in this section are given below.

1. Header Overheads: The Flame frame in the FIGO network is carried as payload of a MAC frame as it is a layer 2.5 protocol and this MAC frame is tunneled through the OpenVPN. For a MAC frame from clients, the first field (MAC address) is repeated for a Type0 (data) Flame payload. The packet format of Type0 Flame payload is given in figure 6.2:

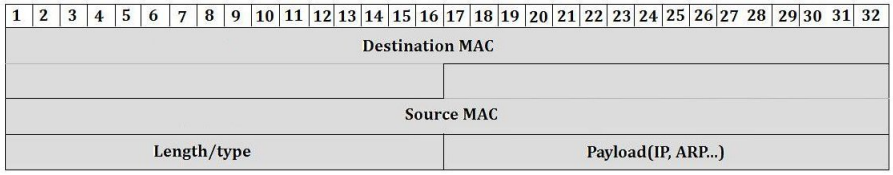

Figure 6.2: Type 0: data packet format [14]

As it can be seen there are two MAC headers tunneled through the narrow TETRA link per flame packet and the OpenVPN by itself contributes an average of 37 bytes per packet for BF-CBC and SHA1 encryption algorithm on the TETRA IP header. For an IP packet from the clients, there are two

IP headers. One is the IP header of the tunnel (TETRA IP) and the other is the IP header of the packet coming from the clients. Considering both IP headers of UDP/IP, there will be a total of 133 byte header per packet ignoring the padding bits. The traffic overhead of the header depends on the payload size. Let's consider an Ethernet type II frame tunneled over the narrowband for analysis purpose. The total packet size is 1583 byte which includes the Ethernet frame (1518 bytes) and the OpenVPN with the TETRA UDP/IP header (37+28 bytes). The header traffic overhead is given in equation 8.1. In actual case the TETRA IP datagram size has to be smaller (around 250 byts) and fragmentation of the Ethernet frame will be required.

$$Overhead_{header} = \frac{133 * 100\%}{1583} = 8.4\% \tag{6.1}$$

2. Flame control message traffic overheads: The control messages in Flame1 are the PU and the NA. The PU message is broadcast by each FIGO node for meshing. When a single VN broadcasts a PU message in the adhoc network, that node and the other VN nodes which have received the PU message will also broadcast it through the OpenVPN. For a connected mesh of K VN nodes in the adhoc network, at least K^2 PU messages will be broadcast through the narrowband link to the BON at every PU time. This is when a PU from the originator is forwarded to the BON only once by all the VNs in the adhoc network but this is not always true as a VN may forward the same PU more than once to the BON. The BON will also broadcast K(K+1) PU messages to the VNs and this is done on the downlink of the TETRA air interface. The packet size of the PU depends on the number of bridged nodes. If we assume there are on average 5 clients (bridged nodes) on each VN, the total packet size of the PU that passes through the narrowband will be 119 bytes including the OpenVPN overhead and 7 byte padding. The traffic overhead is given in equation 8.2.

$$\begin{aligned} Overhead_{PU} &= \frac{R_{pu} * 8 * P_{pu} * K^2 * 100\%}{R_T} \\ &= \frac{0.2 * 8 * 119 * K^2 * 100\%}{1500} \end{aligned} \tag{6.2}$$

Where R_{pu} is the Path Update rate, P_{pu} is the packet size of the PU, R_T is the average throughput of the TETRA link for UDP/IP packet excluding

the IP header (1500 bits/sec) [7] and K is the number of VNs in the adhoc network. In the current Flame the PU is broadcast every 5 seconds($R_{pu} = 0.2$) and the traffic overhead for 3 VNs in the adhoc network is **114.24%**. This shows that the TETRA PDCH can't even support only the PU messages of 3 VNs with 5 clients bridged under each VN.

The other control message of the Flame1 protocol is the NA and it is a unicast message sent to the neighbor nodes for link quality measurement. The packet size of the NA that passes through the narrowband link including the OpenVPN header and 6 byte padding is 71 bytes and it sent every second in the current flame. Every VN node in the adhoc will send a NA to the BON and the traffic overhead is given below.

$$Overhead_{NA} = \frac{R_{na} * 8 * P_{na} * K * 100\%}{R_T} = \frac{1 * 8 * 71 * K * 100\%}{1500} \quad (6.3)$$

Where R_{na} is the NA rate and P_{na} is the packet size of NA. For 2 VNs in the adhoc and and NA sent every second (in the current FIGO), the NA traffic overhead takes 75.7% of the TETRA PDCH capacity.

3. ARP traffic overhead: As Flame is a layer 2.5 protocol, the FIGO network is considered as a single subnet and there will be ARP broadcasting over the narrowband TETRA link. The packet size of an ARP for IPv4 on Ethernet MAC frames is 28 bytes and the total packet size of the ARP that will pass through the OpenVPN is 119 bytes including the Flame headers, padding bytes and OpenVPN overhead. The ARP traffic overhead is similar to the PU traffic overhead because an ARP request from a single client bridged to a VN will be broadcast by all the VN nodes in the adhoc network to the BON. So, the traffic overhead of ARP will depend on many factors. Let's first see how ARP works.

When a node (client) has an IP packet to be sent to another client in the same subnet, it first checks its ARP cache table for the destination client's MAC address. Then, if the MAC address is available in the cache and is unexpired, it will use that MAC address to send the IP packet to the destination client. If the destination MAC is not available on the ARP cache table or the cache table has already expired (5 minutes in wired Ethernet), it will broadcast an ARP request to the network and the destination node will reply when it receives the ARP request. The traffic overhead of ARP will depend on how often an IP packet is generated by each client, to which client the IP

packet is to be sent, the number of clients, number of VNs and the number of services which require ARP protocol. So, a mathematical model will be required to measure the traffic overhead of ARP in the FIGO network. For simplicity lets assume that a client sends one ARP request every 5 minutes in the FIGO network, the ARP traffic load for this assumption is given below. Where K is the number of VNs, N is the number of clients in the adhoc network, P_{arp} is the packet size of the ARP and R_{arp} is the rate of ARP request generated by each client.

$$\begin{aligned} Overhead_{ARP} &= \frac{R_{arp} * 8 * P_{arp} * K * N * 100\%}{R_T} \\ &= \frac{8 * 119 * K * N * 100\%}{300 * 1500} \end{aligned} \tag{6.4}$$

For four VNs (K=4) with 5 clients bridged on each VN (N=20) in the ad-hoc network, the ARP traffic overhead will consume 16.92% of the TETRA link capacity.

Generally the PU and NA of the Flame1 protocol generates a lot of traffic overheads that can easily overload the PDCH of the TETRA.

6.3 TETRA PDCH performance analysis for Flame2

The analysis of a TETRA PDCH for Flame1 above shows that only the PU or the NA control message can overload the TETRA link. When many VNs try to use the link, it will be overloaded with just few VN nodes on the adhoc network and it is clear that Flame1 can not be supported on a TETRA link if there are more than two VNs in the adhoc network.

Flame2 threats the narrow band link between the BON and a GW differently. It is better than Flame1 as it uses hierarchical routing. In Flame2 a VN will be registered at the BON and it acts as a default GW to the other nodes which are registered under it. The GW maintains a list of nodes which are reachable through it and this list is sent to the BON. This list is called a Node Base (NB) and it is periodically updated to the BON. The update can be done by sending the whole list of nodes periodically or using other efficient ways.

In this subsection, the performance of the TETRA link for Flame2 is analyzed to see if it can be supported on PDCH of TETRA and only the continuously sent control messages were considered because they contribute a lot of traffic overheads. The analysis is done using a differential update, which is an efficient way of updating mechanism proposed in Flame2 and a comparative analysis is done for an update by sending the whole list of nodes periodically. A single GW using a PDCH on a TETRA link at a time was considered for the analysis.

The main control messages generated by the Flame2 protocol which are sent through the narrowband link to the BON are given below.

1. ***L1 registration request:*** This control message occurs once when a VN decides to become a GW and it is considered that it contributes less traffic overhead in a stable FIGO network. Therefore the traffic overhead generated by this message is ignored .
2. ***L1 monitoring:*** This is used for monitoring the quality of the link. The NA control message is proposed in the Flame2 document for this purpose. The traffic overhead generated by the L1 monitoring is also analyzed.
3. ***L1 registration update:*** This control message is used to inform the BON about the nodes registered under a GW and it occurs continuously. Thus the traffic overhead of this control signal is analyzed.
4. ***De-registration and L1 selection:*** The traffic overhead contributed by these two control messages is also similar to the L1 registration control message and the traffic overhead depends on how often the GW changes its L1 tunnel to the BON. So, the traffic overhead generated by these two messages is ignored.

A. Traffic overhead of the L1 monitoring message:

The link quality measurement in Flame2 is similar to that of Flame1. Every VN on the adhoc network will try to measure the quality of its direct link to the BON and this generates a lot of traffic overheads over the TETRA link. The NA control message is used for link quality measurement and the traffic overhead is given in equation 6.5 where R_{na} is NA rate, K is the number of VNs and P_{na} is the packet size of NA over the OpenVPN. The L1 monitoring traffic overhead doesn't depend on the number of clients on the network.

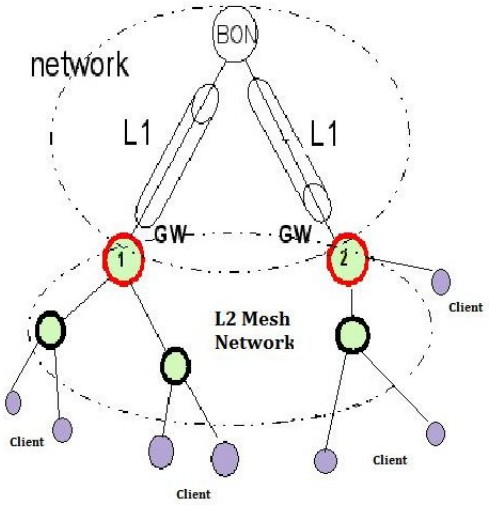

Figure 6.3: Flame2 architecture [12]

$$Overhead_{NA} = \frac{R_{na} * 8 * P_{na} * K * 100\%}{R_T} = \frac{1 * 8 * 71 * K * 100\%}{1500} \quad (6.5)$$

For K=2 and R=1 (NA sent every second), the traffic overhead is **75.7%** and for a single VN in the adhoc network, the traffic overhead is **37.85%**. Link monitoring generates a lot of traffic overhead to the TETRA link and it is not recommended to be sent through the narrowband link. A passive way of measuring the link quality like buffer status or RSSI are recommended for the new minimal Flame protocol proposed in Chapter 7.

B. Traffic overhead of the L1 registration update:

The L1 registration update contributes a lot of traffic overhead on the narrowband link if it is done by sending the whole list of nodes' addresses registered at a GW. A differential update using checksum for table inconsistency was proposed in the Flame2 document to optimize the traffic overhead of the L1 registration update [12]. In this analysis a differential update is used and the table consistency check traffic overhead is not considered. A full analysis of the table consistency is given for the proposed minimal Flame protocol in Chapter 7.

In the differential update proposed in Flame2, a GW first registers at the BON and sends the whole list of nodes (VNs and clients) which are reachable

through it to the BON. After that, it regularly updates the BON with a small keep-alive message if there is no change on its NB or it sends an Add message when a new node is registered at the GW or a Remove message when a node is no more reachable through it. For analysis purpose, it is assumed that on average 10% of the nodes registered at the GW are changing their default gateways every 60 seconds. Many nodes may be added or removed at one time but there will also be a time when no nodes are added or removed. So, this is a reasonable assumption if we consider an average of 10% of the nodes registered at GW are changing position or de-registered from the GW every minute.

The message size of the L1 registration update packet formats are specified in the Flame2 document [12]. The L1 registration message is going to be a new Flame message and the Keep-alive, Remove and Add message types will be specified on the message subtype of the Flame2 frame format. For analysis purpose, the Flame1 header is used with 1 byte added for the message subtype field.

I. Keep-alive message: The keep-alive message is a unicast message sent over the narrowband link and the sender of this keep-alive message can be identified from the external MAC. The packet size of the keep-alive message is 71 bytes including the OpenVPN overhead and padding bytes. The 37 byte OpenVPN overhead is included here because the measured average throughput of the TETRA link (1500 bits/second) didn't consider this OpenVPN header overhead. The packet format of the keep-alive message is given in figure 6.4

Figure 6.4: Keep-alive message format

The traffic overhead generated by the keep-alive message on the TETRA link is given in equation 6.6. Where P_{ka} is packet size of the keep-alive and R_{ka} is the rate at which the keep-alive is sent (update time).

$$Overhead_{KA} = \frac{R_{ka} * 8 * P_{ka} * 100\%}{R_T} = \frac{8 * R_{ka} * 71 * 100\%}{1500} \tag{6.6}$$

II. Add and Remove messages: The Add and Remove messages are

unicast messages. The packet format of Flame2 is used for analysis purpose. All the mandatory fields given for the L1-registration message on the Flame2 document are not required for the Add or Remove message. Only the length field is used and the padding bytes are ignored here. L is the number of nodes added or removed at a time. The message format of Add or Remove message is given in figure 6.5

Figure 6.5: Add or Remove message format

The traffic overhead generated by the Add and Remove messages is given in equation 6.7. It is assumed that on average 10% of the nodes are either added or removed every 60 seconds and K is the number of the nodes registered at the GW and R_{ar} is the rate at which nodes are added or removed(the assumption). 37 bytes of the OpenVPN overhead is also included in the formula given in equation 6.7.

$$\begin{aligned} Overhead_{AR} &= \frac{8 * R_{ar} * (66 + (0.1 * K) * 6) * 100\%}{R_T} \\ &= \frac{8 * (66 + (0.1 * K) * 6) * 100\%}{60 * 1500} \end{aligned} \tag{6.7}$$

The graph given in figure 6.6 shows the percentage of traffic overhead generated by the keep-alive, Add and Remove messages. This graph doesn't include the L1 monitoring traffic overhead. An update time of 5, 10, and 15 seconds was considered. Static means when there is no change on the nodes connected to the GW, that is, only keep-alive message is sent to the BON.

For the sake comparison, the traffic overhead generated when updating is done by periodically sending the whole list of address connected to the GW and the differential update is calculated and the graph given in figure 6.7 shows the comparison of the traffic overhead of the two ways of updating mechanisms. From the graph, it is clear that sending the whole list periodically generates a lot of traffic overheads and it may not even be supported by the TETRA link as the number of nodes connected to the GW increases.

From the analysis given above we can see that Flame2 doesn't generate a lot of overheads if differential updating mechanism is used and link quality

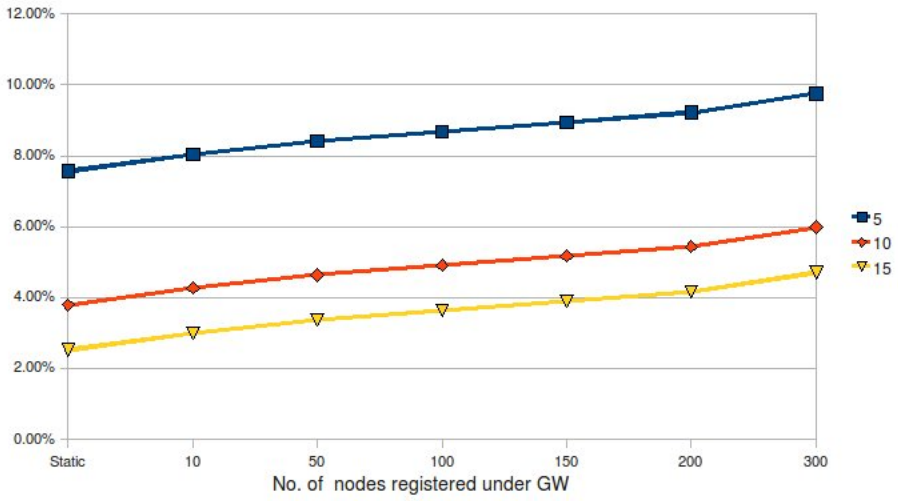

Figure 6.6: Traffic overhead of the L1 registration messages

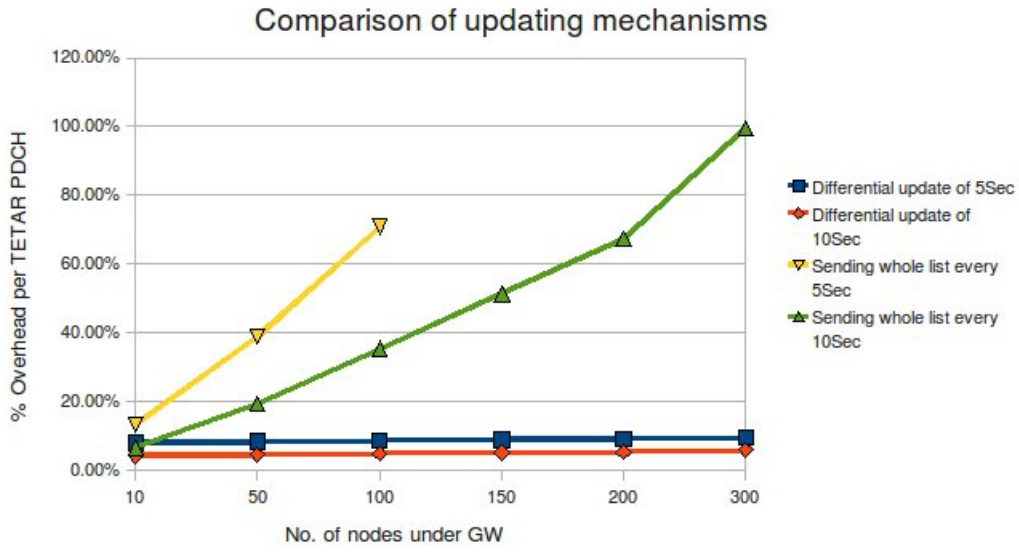

Figure 6.7: Traffic overhead comparison of the two updating mechanisms

measurement is done in a passive way. So, Flame2 can be used as basis for integrating TETRA with wireless mesh networks (in FIGO).

C. Other overheads on the narrow band L1 link:

Other than the Flame2 control messages, there are other traffic overheads that need to be considered on the narrowband TETRA PDCH link between the GW and the BON. ARP and header overheads are considered in this section.

1. ARP traffic overhead: The traffic overhead generated by ARP in Flame2 is different from that of Flame1 because in Flame2 only the GWs forwards ARP request to the BON. That is, when an ARP request is generated by a client bridged to a VN, only the GW under which the client or its cluster-head is registered at will forward the ARP request to the BON. It is clear that the ARP traffic overhead generated in Flame2 is much less than that of Flame1 but its analysis will also require some mathematical model because it depends on how often the clients are generating an IP packet that require ARP protocol and to which client, the ARP cache refreshing time, and the number of clients. The ARP cache is refreshed every 5 minutes for wired Ethernet but this may not be convenient for wireless mesh networks. As the ARP request is broadcast from the BON to all the reachable GWs, the ARP traffic overhead in Flame2 will also affect the down link of the TETRA channel more than the up link channel for high number of GWs in the FIGO network.

For a simple analysis let's assume a client in FIGO network generates one ARP request every 5 minutes. The ARP packet is 119 bytes including the OpenVPN overhead, Flame header and padding bytes. The traffic overhead of ARP in Flame2 doesn't depend on the number of non-GW VNs in the adhoc network and it is given in equation 6.8 for a single GW accessing the TETRA link.

$$Overhead_{arp} = \frac{R_{arp} * 8 * P_{arp} * N * 100\%}{R_T} = \frac{8 * 119 * N * 100\%}{300 * 1500} \quad (6.8)$$

Where N is the number of clients in the adhoc network, P_{arp} is the packet size of the ARP and R_{arp} is the rate of ARP request generated by each client.

The traffic overheads generated by each of the Flame2 control messages (L1 registration update, ARP and L1 monitoring) on a single TETRA PDCH is

given in figure 6.8. This analysis is for a single GW accessing the TETRA link and there is only single VN in the adhoc network that is, the GW is the only VN in the adhoc network and the L1 monitoring traffic overhead is 37.85% for NA sent every second. The keep-alive message is also only from the single GW. Different number of clients under the GW were assumed and an update time of 10 seconds is also considered.

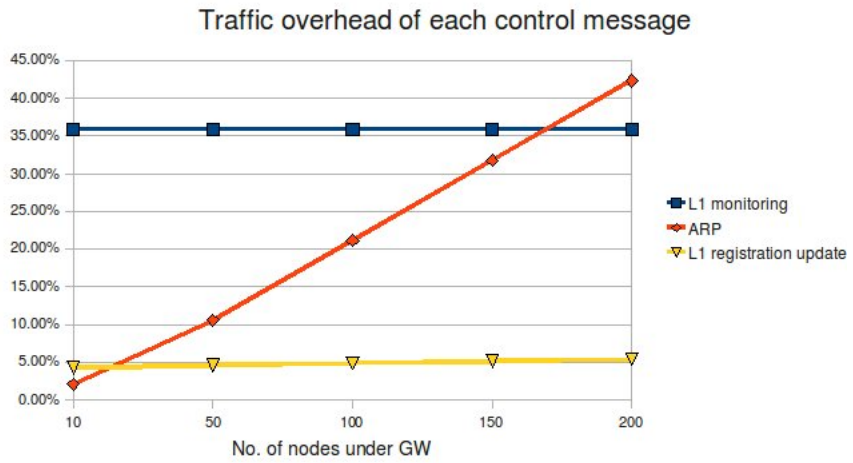

Figure 6.8: Each of the control traffic overheads in Flame2

The total control traffic overhead, that is the sum of all the traffic overheads generated by all the control messages (keep-alive, Add and Remove messages, L1 monitoring, and ARP) for 5 sec, 10 sec and 15 sec updating time is also given in figure 6.9. Most of the traffic overhead is generated by the L1 monitoring control message.

2. Header Traffic overhead: The other overhead is when an IP packet is sent through the narrowband link, the total header is big and it contributes a lot of traffic overhead to the narrowband link. This is because for a single IP packet, there is MAC header, Flame header and IP header as described in the Flame1 protocol. The traffic overhead generated by the header depends on the size the payload. For analysis purpose, an Ethernet type II frame size is considered and the traffic overhead generated by the headers is given in equation 6.9

$$Overhead_{header} = \frac{134 * 100\%}{1583} = 8.5\% \tag{6.9}$$

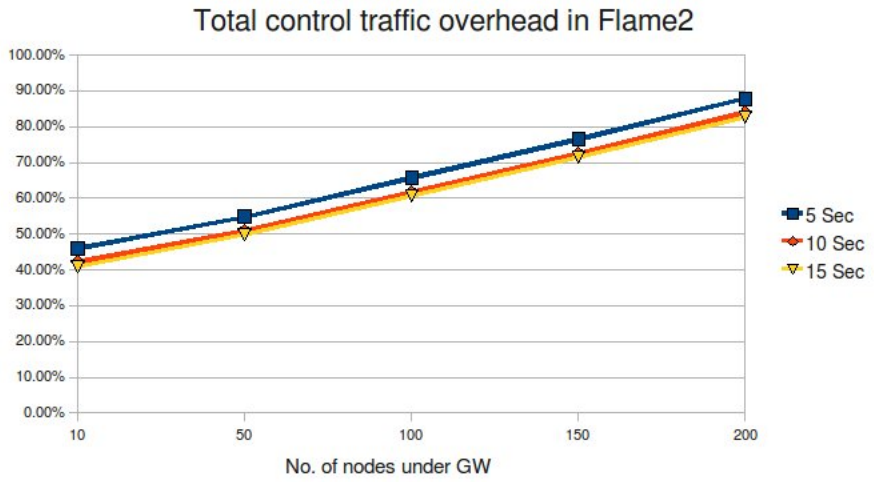

Figure 6.9: Total control traffic overhead in Flame2

The header overhead in Flame2 is similar to the Flame1 and only one byte is included for the subtype field in the Flame2 header. 8.5% of the data coming from the clients is header and this is too much for the narrowband link to the BON.

6.4 Problem definition

From the analysis of TETRA PDCH for Flame protocols in the above sections, it has been shown that the Flame protocol control messages are the most dominant traffic overheads generators in the network. Furthermore, TETRA is a narrowband wireless communications system and it can't accommodate a lot of traffic overheads. Generally, the bottleneck in integrating any narrowband cellular network like TETRA with any wireless mesh networks is the narrowband link between the adhoc nodes (VNs in FIGO) and the control center (like BON in FIGO). There are two main problems identified in this study. These problems are given below:

1. The protocol control messages are generating a lot of traffic overheads over the narrowband link to the BON. The predominating control messages are generated by the routing protocol messages (like PU and

NA in Flame) and the ARP protocol. The architecture designed to integrate TETRA with wireless mesh networks needs to consider the overheads generated by these control messages. Two general ways of limiting the control message traffic overhead which are used in the architectural design in Chapter 7 and 8 are briefly given below.

- Limiting control signals that pass through the narrowband link by introducing Layer2 or layer3 gateways similar to the principle adopted in Flame2.
- Decreasing the rate or leaving out some of the control messages at the expenses of link quality. The link quality measurement in Flame introduces a lot of traffic overheads and it is left out in the architecture design in chapter 7. Link quality measurement has to be done in a passive way like using buffer status or RSSI measurements.

2. The excessive header overhead tunneled through the narrowband link also consumes a lot of bandwidth specially when small data packets are sent over the TETRA link. Different mechanisms of avoiding the excessive overheads have been described in chapter 7.

Chapter 7

The minimal Flame Architecture

This is a new architecture and protocol proposed to integrate TETRA with the FIGO network. The TETRA link is used to interconnect the adhoc network with the BON on the infrastructure. The basis of this new protocol and architecture is Flame2. All the optimizations considered in this new architecture are to minimize the traffic overheads that pass through the narrowband link (L1) to the BON. Flame2 is considered for any of the communications on the adhoc mesh network with some additional functionalities like registering nodes at the GW with their MAC address and IP address.

The analysis for Flame protocol in chapter 6 shows that there are a lot of traffic overheads which need to be optimized for the narrowband links to the BON. These traffic overheads are generated mainly due to the control overheads and header overheads. In this chapter different mechanisms have been considered to reduce those traffic overheads. These mechanisms minimize the traffic overhead either by limiting the control message overheads and/or reducing the header overheads. Eight mechanisms have been identified in this architecture and the details of these mechanisms are explained in the following sections (7.1-7.8).

7.1 Limiting ARP/RARP broadcasting on L1

The address resolution protocol (ARP) generates a lot of traffic overhead in Flame2 specially on the downlink of TETRA when it is broadcast by the BON to all the gateways connected to it. To avoid this problem let's allow nodes to be registered at the GWs and the BON by their MAC address and IP address. The last two bytes of the IP address uniquely identify a node in the current FIGO network IP addressing scheme. So, this two byte of IP address can be used for identifying a node at the BON. When a GW first sends the list of addresses to the BON, an 8 byte (MAC and the last part of the IP) information per each node is required but this is not going to be used for the update messages (Node Remove or Node Add messages). These overheads can be further optimized by using the last two bytes of the IP address for updating purpose. The BON will have an address table with MAC address and IP address of all the nodes from every GW connected to it. Every GW will also have the same table for all the nodes registered under it as it shown in figure 7.1.

GW Address		
No	MAC address	Last part of IP
1	6 bytes	2 bytes
2	——-	——
3	——-	——

The BON now can reply to any ARP request coming from the GWs. So, This avoids broadcasting of ARP request to all the gateways connected to the BON. This is very important as it will reduce a lot of ARP flooding traffic loads. A GW shouldn't also forward ARP request to the BON if the ARP request is to a node registered under it. When a GW receives an ARP request, it checks if it is to a node registered under it, if so, it should only broadcast in the adhoc network not to the BON.

7.2 ARP catching at the GWs

It is also possible to avoid any ARP request to pass through the narrowband by letting the BON send the whole address table to every gateway and if a GW has all the tables, it can reply to any ARP request coming from all clients registered under it but this kind of optimization has another problem. As the number of nodes and gateways increases, the starting delay can be

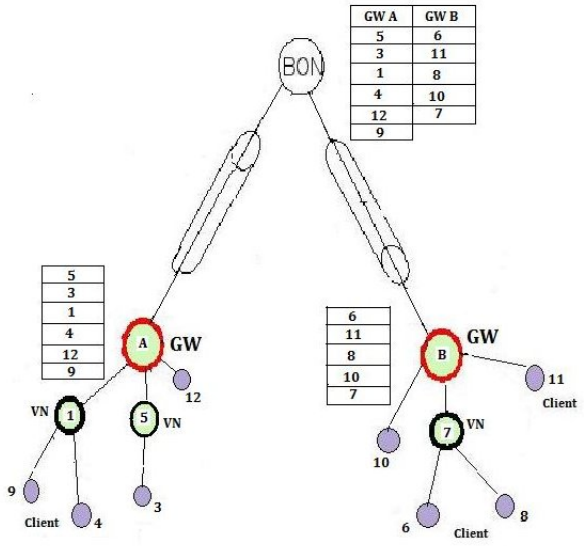

Figure 7.1: Table address at the BON and GWs

very high because the GWs require to have all the address tables before any two clients connected under different gateways start communicating. The starting time for K GWs and N nodes connected under each GW is given below. The new minimal flame header proposed in section 7.6 is used and all the GWs sharing the same TETRA PDCH is assumed.

$$Starting - time = \frac{8 * (12 + 2 * N)K^2}{R_T} sec = \frac{8 * (12 + 2 * N)K^2}{1500} sec \quad (7.1)$$

Sending the the whole table from the BON to the GWs will affect the performance of the network not only by increasing the starting delay it is also required to maintain the consistency of the tables at the BON and the GWs. So, it is not advisable to use this optimization mechanism. But we can optimize the Flooding of the same ARP request over the narrowband link from different nodes connected to the GW by allowing the GW to cache ARP reply from the BON. When the GW sees an ARP reply from the BON it will cache it on its ARP cache table and if there is the same ARP request from other nodes connected to the GW, the GW will check its ARP cache table and it can reply to this type of ARP request if the timer of the cache table is not expired.

7.3 Using a shortened MAC address on L1

The normal MAC address is a 6 byte size but any node in FIGO can uniquely be identified by the last 2 bytes of the IP address. So, the GW and the BON can use the 2 bytes instead of the normal MAC address for any MAC frame over the narrowband link and the data has to be re-framed again into normal MAC frame when it is sent to the destination node (client). The shorten MAC is not only used for sending data frames but also for updating purposes like Node Add and Node Remove. A GW knows the shorten MAC addresses of the nodes registered under it but it doesn't know the shorten MAC address of the other nodes under different GWs. So, there are two types of data frames in the minimal Flame protocol. These are data frame for IP packet and data frame for any other protocol. The data type will be identified on the message type field of the minimal Flame header given in section 7.6. The details of the minimal Flame protocol is given in that section.

7.3.1 Data frame for IP packet

When an IP packet comes to the GW from any of the nodes registered under it and the destination IP address is in the FIGO domain, the GW will use the last two bytes of the destination and source IP addresses as a shortened MAC addresses and it will make a data frame using these short MAC addresses. The BON has every information because it has all the address tables and it has to re-frame the packet before forwarding it to the destination node. The BON will use the shorten MAC address of the destination node and the normal 6 byte MAC address of the source node when the destination node is reachable through another GW because the GW with the destination node/client connected to doesn't have the address table of the source node and the GW will re-frame the packet into the normal MAC frame before it forwards it to the destination node/client. By using a shortened MAC address, 8 bytes of header overhead can be eliminated on every packet from the a GW to the BON.

7.3.2 Data frame for any other protocol

This data frame format is for any other layer3 protocol and if the IP address is out of the FIGO domain. When the GW receives this kind of layer2 frame

from the clients connected to it, it has to use the shorten MAC address of the source node and the normal 6 byte MAC of the destination node because the destination node's shorten MAC is unknown for non-IP packets or if the destination's IP address is out of the FIGO domain. The BON has every address table and it has to re-frame the data again so that the destination GW will be able to make the normal frame. The BON will use the shorten MAC address of the destination node and the normal 6 byte MAC of the source node because the destination GW doesn't have the address table for the source node. The BON can also re-frame the packet into normal frames if it has to be forwarded out of the FIGO domain.

7.4 IP header compression

The TETRA network provides an IP address to subscribers and any data or protocol can be tunnel through it. The Motorola TETRA terminal supports V.Jacobson TCP/IP header compression but it has been identified in the literature that TCP/IP is not efficient to be used in TETRA packet data communications. UDP/IP is recommended to be used for TETRA packet data communications and it is also commonly used for tunneling. It seems IP header compression is impossible for the TETRA IP header unless a different TETRA terminal is used. The only option is to compress the internal IP address. Two ways of IP header compression are given below.

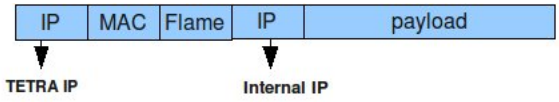

Figure 7.2: Flame Date frame on the IP tunnel of TETRA

7.4.1 Standard IP header compression

The internal IP header can be compressed if un encrypted header IP packet is sent from the clients and different IP header compression mechanism can be used to compress it. The Internet Engineering Task Force (IETF) has standardized IP header compression protocols. These are, RFC 1144 (VJ, CTCP), RFC 2507 (IPHC), RFC 2508 (CRTP), and RFC 3095 (ROHC).

The RFC 3095 (ROHC) seems to be an efficient and all rounded IP header

compression mechanism but it will require further study on how complex it is and how to implemented on the VNs. In this architecture only the GWs are assumed to undertake the IP header compression and decompression but it can be done on any of the FIGO nodes preferably at the end VNs which have bridged clients to avoid header overheads on the adhoc network.

7.4.2 Own IPv4 header compression

The standard IP header compressions are very efficient but a little bit complex because they require to exchange a lot of state information between the compressor and decompressor. Most of them are not also available as free source. So, our own stateless IP header compressor was proposed . For one thing it is not required to send the source and destination IP address as it is available on the layer2 data frame from the GW to the BON. The last two bytes of the IP address can be driven from the shorten MAC address and the remaining two bytes can be derived from the subnet prefix (X.Y.A.R/16). X.Y is fixed in the current FIGO IP addressing (private address). It is also possible to further reduce the other fields of the IP header.

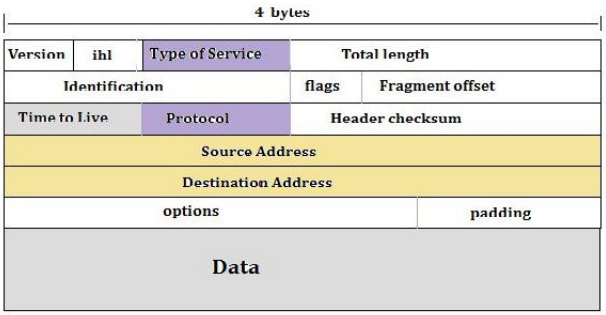

Figure 7.3: IPv4 header fields [Internet]

Most of the fields of the IP header are required by a router but the FIGO network is under the same subnet and some of the fields may not be required. The most important field is the protocol field. For the rest of them either they can be derived from the other fields or left as they are not that much important. So, Except the protocol field most of the other fields can either be left or re-derived from other information on the layer2 frame. It is assumed that IP packet fragmentation is allowed only on the TETRA IP (external IP). If fragmentation is allowed on the internal IP, the three fields (Identification, flags and fragment Offset) will be required but this is

not going to be the case because the services that can be supported over TETRA shouldn't have large IP packets. The transport protocols (UDP or TCP) headers are difficult to compress using stateless header compressor. The checksum and length part of the UDP header can be left as they can be derived from the layer2 frame but most of the TCP header fields are difficult to compress.

7.5 Leaving out the external MAC and OpenVPN

Normally the flame frame is carried as the payload of a MAC frame and there is an external MAC address which is tunneled through the narrowband link on the current FIGO. In this architecture, a GW will be registered at the BON by its tunnel ID and MAC address. The BON has to use this tunnel ID to identify from which GW a message is coming. So, the IP tunnel is used to carry the Flame frame (without the external MAC). This can't be done over a VPN as it requires a normal layer 2 frame or layer 3 packet. But in our case the Flame frame will be sent as the payload of the UDP/IP of the TETRA. OpenVPN by itself contributes a lot of header overhead to the narrowband TETRA link and it is left out in this architecture. From this optimization 14 byte MAC header overhead and 37 byte OpenVPN overhead can be eliminated.

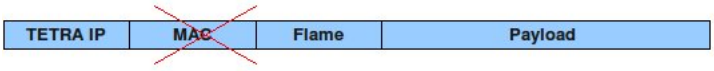

Figure 7.4: Flame frame format without the external MAC

7.6 Using minimal Flame Header

The current Flame header is a 12 byte in size but it can be optimized for narrowband links. The 4 byte IV in Flame is used for the encryption purpose and it is assumed that TETRA is a secure private network but there is a part on the link between the TETRA gateway and the BON which is a public network (Internet). There should be a security measure between the TETRA gateway and the BON otherwise there has to be other form of end-to-end encryption on the application data of the two communicating

nodes. The other importance of leaving out the encryption and decryption at the GWs and the BON is, no padding bits are required which significantly increase the packet size of the small control messages. In addition, there will also be less processing delay which would have added a lot of complexity to GWs and the BON. Note that, the BON and the GWs are expected to undertake complex tasks due to the proposed processes in the minimal Flame protocol. So, the 4 byte IV part of the Flame can be left out from the old Flame frame format in figure 6.1. The New Flame header will look like the one given in figure 7.5.

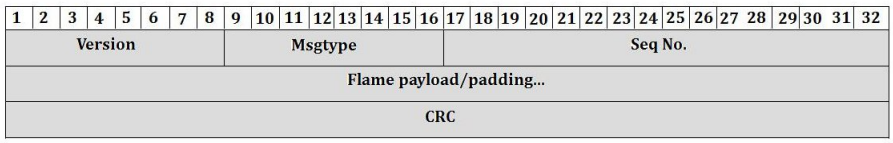

Figure 7.5: New minimal Flame header

There will be different types of payloads for the new minimal flame protocol. The payload types of this protocol are given below.

Data: This is a layer 2 data that has come from the clients or other peripherals connected to FIGO nodes. There will be two types of data frames. The data type will be identified on the message type (Msgtype) of the minimal Flame header field.

A. Data frame for IP packet: This date type is identified by the shorten MAC address. The frame format is given figuer 7.6.

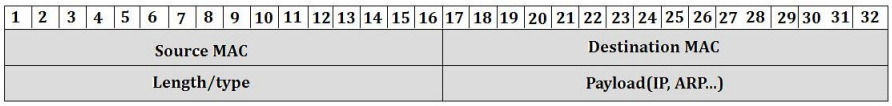

Figure 7.6: Payload format for IP packets

B. Data frame for any other protocol (including Encrypted IP header)

-The data frame from the GW to the BON will have a frame format given in figure 7.7.

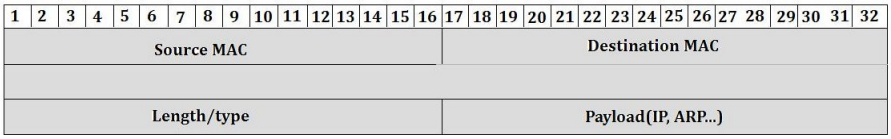

Figure 7.7: Data payload format for non-IP from the GWs

-The date frame from the BON to the GWs will have a frame format given in figure 7.8.

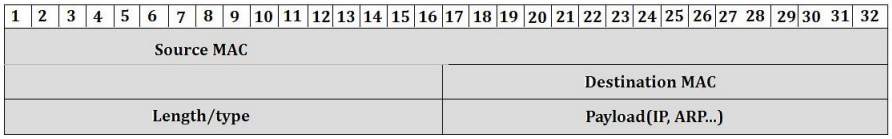

Figure 7.8: Data payload format for non-IP from the BON

L1 register: This message is required when a VN is first registering as a GW at the BON. It has to use the Flame2 protocol because the BON won't identify what type of tunnel is going to be used by that GW. This message is not payload of the new minimal Flame header. The GW will be registered at the BON by its MAC address, and Tunnel ID/type (TETRA IP address). When the GW is registered at the BON, it will inform the BON that it will use the minimal Flame protocol through the specified tunnel. The packet format will depend on the L1 register packet of Flame2 which is under development. This message is going to be used to integrate the normal Flame2 protocol for wideband L1 links and the minimal Flame protocol for narrow band links.

Node register: This message is for registering nodes or clients for the first time at the BON. It is going to be used for sending the address table from the GWs. The BON will maintain this table for routing information. This message is not used for a node changing its default gateway without changing its IP address. There are two types node register messages. Single node register and multiple nodes register and they will be identified on the Msgtype field of the minimal Flame header. The packet formats of the node register messages are given in figure 7.9 for single node registration and 7.10 for multiple node registration.

Node Add: This message is used by the GW for nodes which have been registered at the BON but have moved to a different GW without changing

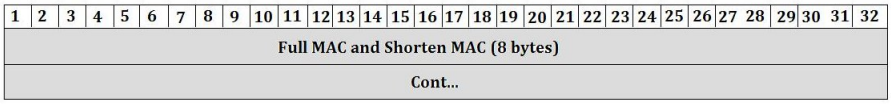

Figure 7.9: Single Node Register packet

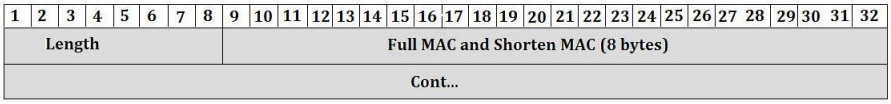

Figure 7.10: Multiple Node Register packet

their IP address or for those node which have been disconnected from a GW for a moment. There are two types of Node Add packet formats. These are single Node Add and multiple Nodes Add. The packet format of the Node Add message is given in figures 7.11 and 7.12 for single and multiple node addition respectively.

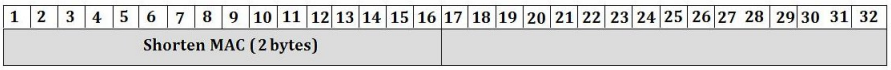

Figure 7.11: Single Node Add packet

Node Remove: This is used by the GW when a registered node is no more reachable by the GW. It can also be used by the BON to inform a GW that a node is registered on another GW and it has to be removed from the old GW's address table. There has to be some kind of threshold when to send a Node Remove message to avoid the problem of nodes popping up and down because of the changing wireless environment. The packet format is the same with the Node Add message given above.

Hash message: This message is used to check the consistency of the address table at the BON and GWs. It can be sent by the BON or GW. The MAC address of the nodes on the address table is arranged in ascending or descending order and hashed using a hash function. MD5 or CRC is proposed on the Flame2 document but there could be other types hashing functions with variable input. The size of the hash value has to be big enough to avoid any hash collision and hash collision is also dependent on the type of hashing function. A four byte hash value is assumed to be enough

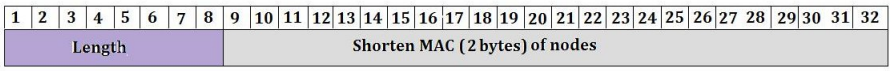

Figure 7.12: Multiple Node Add packet

for the table consistency check. The packet format of the hash message is given in figure 7.13

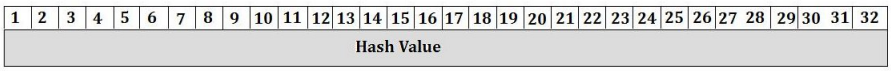

Figure 7.13: Hash packet payload

ACK or NACK: These messages are used to reply for the control messages given in the minimal Flame protocol. The ACK or NACK messages should carry a copy the sequence number of the control message received and it can just copy it from the minimal Flame header of the received control message.

L1 de-registration: This message can also be sent by a GW or BON when a GW is no more required to act as a GW.

7.7 Leaving out the L1 monitor and keep-alive messages

The link quality measurement control message is left out in the new minimal Flame protocol because it generates a lot of traffic overheads and the link quality has to be measured in a passive way. The keep-alive messages is also left out in this new protocol because the GWs can use the TETRA card to know the availability of the TETRA air interface link without any generation of control messages. The fixed line from the TETRA gateway to the BON is assumed to be a stable link. In this case, the BON wont be able to know whether a GW is available or not on time but it can use the other control messages to do that. For example, when the GW dies, other nodes which were registered under that GW will try to register at the BON through other available GWs and the BON will send Node Remove message to the old GW. If the old GW is not replying, the BON may decide that the GW is disconnected. This mechanism doesn't consume a lot of bandwidth but it may take a buffer or memory space at the BON. The other indicator

is the hash message because it is sent to the BON regularly at lower rate. When a GW is not sending hash message to the BON, the BON will assume that the GW is disconnected.

7.8 Data compression

It has been tried to optimize the narrowband link by compressing the headers and avoiding the excessive control signals. The other remaining option is compressing the whole data using lossless data compression mechanism but this mechanism won't be always efficient if the data has already been compressed. Attempts to compress data that has been already compressed will usually result in an expansion, as will attempts to compress all but the most trivially encrypted data does. So, there has to be a way to identify whether the data is already compressed or encrypted. This shows that it is required to inspect what kind of application data is coming from the clients. Compressing data coming from a client is not a wise idea because it can already be an encrypted data as we have assumed that there has to be a kind of end-to-end encryption and often data is compressed before encryption for security purpose.

The main point is how to identify whether a given application data is compressed or encrypted. If it is possible to identify that, one of the data compression standards available in the literature can be used. There should also be a way to identify whether the data is compressed in the FIGO system or not because the end VN has to decompress it. So, it is a little bit complex to implement data compression in the FIGO system but there is still a possibility to do compression for an application data which is neither compressed nor encrypted by the clients.

7.9 Analysis of the minimal Flame protocol

The analysis of the minimal protocol for TETRA PDCH considers all the optimizations given above except the data compression. The most resource consuming traffic loads are those that are regularly sent through the narrowband link. The traffic overheads of the GW registration and node registration at the BON are ignored because they happen infrequently.

1. Traffic overhead of the Node Add and Remove messages: A

modified differential updating from a GW to the BON without the keep-alive message was assumed. A GW sends a Node Add when a new node is registered at the GW or Node Remove message when a node is not reachable through the GW. For the analysis purpose, It is assumed that on average 10% of the nodes registered at the GW are changing their default gateways (added or removed) every 60 seconds. Many nodes may be added or removed at some time but there will also be a time when no nodes are added or removed. So, the assumption is that on average 10% of the nodes change their GW every minute and it is a reasonable assumption. The 10% assumption is also in accord with the traffic overhead analysis for Flame2 protocol described in Chapter 6.

The packet size of the Node Add and Remove message depends on the number of nodes added or removed at a time (10% of the nodes under the GW). The packet format is given in figure 7.14. The multiple Add and Remove message is considered for analysis purpose but the packet size will be smaller when a single node is added or removed at a time.

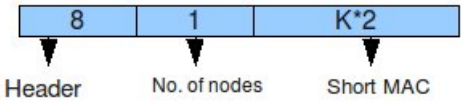

Figure 7.14: multiple Node Add or Remove message size

The traffic overhead contributed by these control messages is given in equation 7.2.

$$\begin{aligned} Overhead_{AR} &= \frac{R_{ar} * 8 * (9 + (0.1 * K) * 2)100\%}{R_T} \\ &= \frac{8 * (9 + (0.1 * K) * 2)100\%}{60 * 1500} \end{aligned} \tag{7.2}$$

Where K is the number of nodes registered at the GW and R_{ar} is the rate at which Node add or remove is sent (every 60 seconds assumed)

2. Traffic overhead of table inconsistency: The GW will send a hash message at a lower rate to the BON. This will contribute a traffic overhead. How often should the hash message be sent will depend on the probability of occurrence of table inconsistency at the BON and the GW. Table inconsistencies may occur through a power glitch at the BON and this is so

infrequent. The other possibility of table inconsistency is when a bit error of the table update messages (Node Reg, Node Add and Remove) is able to pass to the BON undetected. Channel bit errors may occur on the wireless or wired part of the narrowband link. The bit errors on the wireless part will be taken care by the LLC and MAC layers of TETRA . TETRA uses two types links for data communication. Basic link is used for voice data and advanced link is used for packet data communications. Advanced link has a reasonable error detection and correction mechanisms. The wired part is similar to the Internet and the error detection and correction mechanisms of this part will depend on each interface of the routers encountered on the path. The error which has occurred on the wireless or wired part of the link to the BON will further be checked by the CRC of the minimal Flame protocol. So, the table inconsistencies occurrence will depend on the probability of bit error on the path and the probability it passes all the error correction and detection mechanisms on the path undetected. It seems measurement is required to decide on how often the hash message should be sent because the table inconsistency depends on many factors. This hash message can also be used as a keep-alive message so that the BON will know the availability of GW. The hash message is a 12 byte long size and the traffic overhead generated by this message is given in equation 7.3. Where R_{hm} is the rate at which the hash message is sent

$$Overhead_{hash} = \frac{R_{hm} * 8 * 12 * 100\%}{R_T} = \frac{R_{hm} 8 * 12 * 100\%}{1500} \tag{7.3}$$

If a hash is sent every 30 seconds, the traffic overhead is 0.21% of the TETRA capacity. This 30 second interval is just to show how much traffic overhead is generated by the hash message. In practice it expected to be at very low rate.

The worst case is when table inconsistency occurs because it is expensive to send the whole table to the BON. This traffic overhead also depends on the number of nodes registered at the GW. For the analysis purpose lets assume table inconsistency occurs every 5 minutes (practically it has to be at a very low rate). The traffic overhead of the table inconsistency is given in equation 7.4 and the graph is given in figure 7.15. Where R_{ti} is the rate at which table inconsistency occurs(every 5 minutes assumed) and K is the number of nodes registered under GW. The traffic overhead linearly increases with the number of nodes registered at the GW.

$$Overhead_{inconsy} = \frac{R_{ti} * 8 * (9 + K * 8)100\%}{R_T} = \frac{8 * (9 + K * 8)100\%}{300 * 1500} \quad (7.4)$$

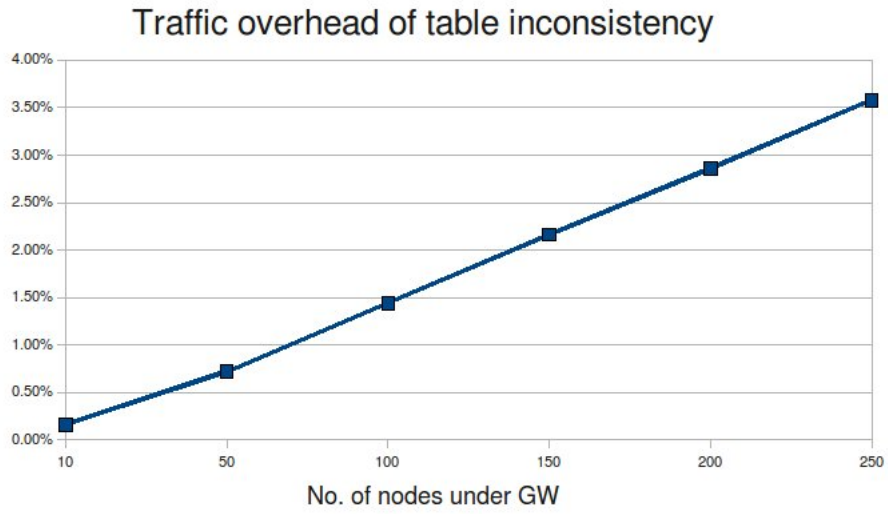

Figure 7.15: Traffic overload of the table inconsistency

3. Traffic overhead of ARP: The traffic overhead of the ARP is optimized by the BON and the GWs in the minimal Flame protocol. The GWs limit forwarding of ARP through the narrowband by caching ARP replies from the BON and they don't broadcast an ARP request to the BON if the request is to a client registered under them. The BON won't also broadcast ARP requests to the GWs as it can reply for any ARP request.

The GW is caching ARP so that it can reply to similar ARP requests from the other clients and it is not forwarding ARP request destined to a client registered under it. So, the traffic overhead of the ARP in the minimal Flame protocol can reasonably be assumed as if there is a single client under the GW for few number of devices with services that require ARP. The ARP traffic overhead is given in equation 7.5 where R_{arp} is the rate at which ARP requests are generated by the clients (every 5 minutes in wired Ethernet) and P_{arp} is the ARP packet size including the Flame header.

$$Overhead_{ARP} = \frac{R_{arp} * 8 * P_{arp}100\%}{R_T} = \frac{8 * 42 * 100\%}{300 * 1500} = 0.08\% \quad (7.5)$$

The overall control traffic overhead of the minimal flame protocol on a single TETRA PDCH for a single GW using the channel is given in figure 7.16. It considers all the control traffic overheads addressed above. A hash message sent every 30 seconds and table inconsistency occurring every 5 minutes was considered for this analysis. The analysis shows for different number of nodes registered at the GW (10 to 200) and most of the traffic overhead is contributed by the table inconsistency checks.

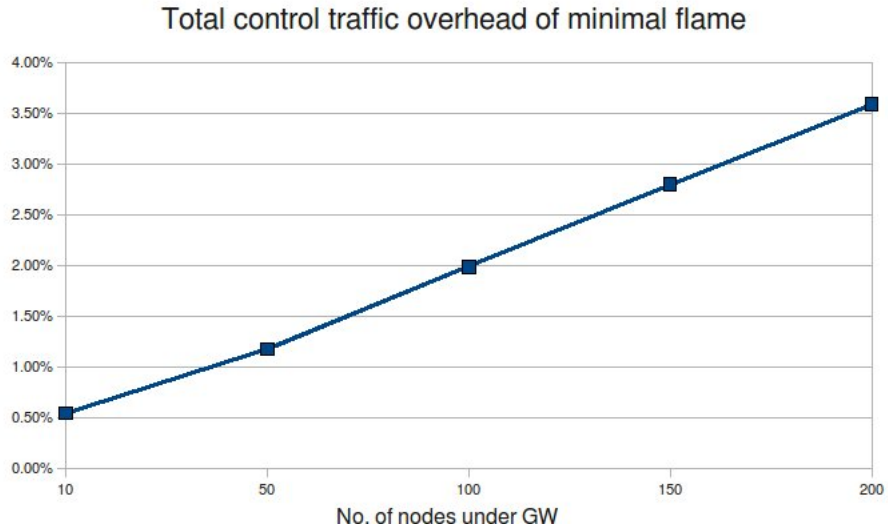

Figure 7.16: Total control traffic overhead of minimal flame protocol

From the graph given in figure 7.16, we can see that the total traffic overhead of the new minimal flame protocol is less than 4.0% of the 1.5Kbps TETRA link capacity for less than 200 nodes registered under the GW.

4. Traffic overhead of the header: The other traffic overhead considered over the narrowband is the overhead caused by the headers for data coming from the clients. There are two types of data frames from the GW to the BON defined above. The frame format given in figure 7.17 is for an IP data coming from any client at the GW. A UDP/IP packet sent from the clients is considered and the own IP header compression proposed is also used.

In this case, the overhead depends on the payload size but let's consider an Ethernet type II frame of length 1518 bytes. So the traffic overhead is given below and it is reduced to 1.6% in the minimal protocol from 8.4% in Flame2:

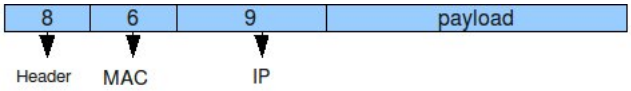

Figure 7.17: Data frame for UDP/IP packet

$$Overhead_{header} = \frac{24 * 100\%}{1518} = 1.6 \tag{7.6}$$

The graph given in figure 7.18 shows the comparison of the control traffic overhead generated by the minimal Flame protocol and Flame2 protocol. In this comparison the traffic overhead of Flame2 protocol using differential updating mechanism of an update time of 15 seconds was used.

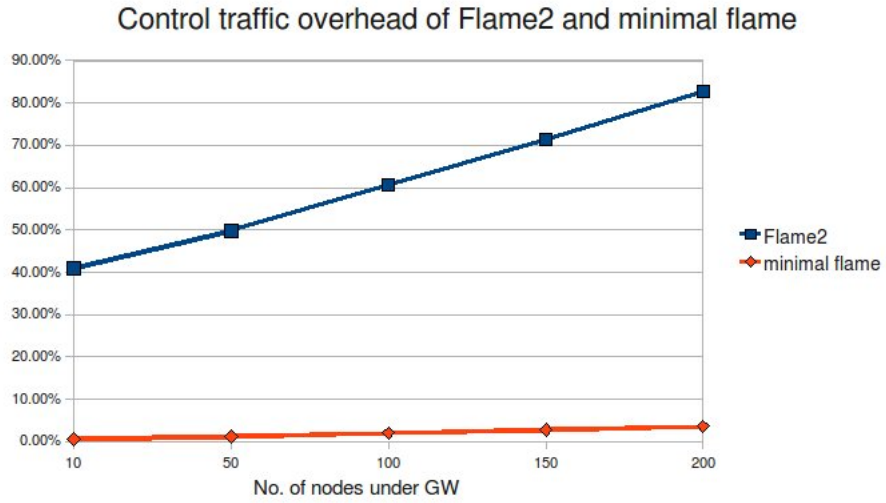

Figure 7.18: Control traffic overhead comparison of Flame2 and minimal Flame

From the graph above we can see that the new minimal flame protocol generates far less traffic overhead than Flame2. The analysis considers a single GW using the TETRA link at a time. From the analysis, we can conclude that it is possible to use narrowband links in the FIGO network with differential updating mechanism and it is also possible to further optimize the header and other control overheads by using the proposed minimal Flame protocol.

7.10 Limiting number of GWs on a single PDCH

Commonly a single PDCH is available in a TETRA cell but it can be configured in such a way that multiple PDCH will be available for packet data subscribers. Multiple PDCH are not common in a TETRA cells because higher priority is given to voice communications and most of the resources are allotted to voice communications unless there is a special case where packet communication is required.

When more than one GWs are accessing on a single PDCH in TETRA, there has to be a way to coordinate the usage of the channel. This is because as the number of GWs increases the total control traffic overhead over the narrowband link is the sum of the total control traffic overheads of each of the GWs accessing the channel.

The other problem of multiple GWs accessing on a single PDCH in TETRA is the channel usage efficiency of the random access mechanism used in TETRA. A packet data subscriber in TETRA encounters two random accessing processes before it starts transmitting data through the TETRA link. First it needs to contend in a slotted aloha mechanism on the Main Control CHannel (MCCH) to get access for the PDCH. This random access is done with all the terminals (voice and data terminals) on the main carrier. After the subscriber gets access to the designated PDCH, it can start sending its pending packet data messages. In the example shown in figure 7.19, a packet data message is divided into five segments because it requires five time slots to be transmitted on the TETRA air interface (IP datagram size of 136-164 bytes, see table 6.1). The first segment is sent with random access (slotted aloha) and is therefore subject to potential collisions, which will require this segment to be later re-transmitted. The successful transmission of the first segment will indicate to the system that the subscriber requires 4 additional time slots to complete its transmission [19]. Hence, the system will later on allocate the requested time slots. So, there is reservation for the rest of the segments.

After the subscriber is hooked onto the PDCH, it may stay there after it finishes sending its pending message or it has to contend again on the main carrier for another message. This depends on the ready timer of SNDCP protocol. For analysis purpose lets consider the ready time is long enough and the subscribers stay on the PDCH. So, lets ignore the random access delay and bandwidth usage efficiency on the MCCH of a main carrier.

From the experiment done for a UPD/IP packet data on TETRA PDCH, the efficient message size is a packet that fits in 4-7 time slots of the TETRA air interface [7]. So, if we take a message size of 5 time slots (segments), the first segment will experience a slotted aloha random access and the rest will have a reserved access as depicted in figure 7.19.

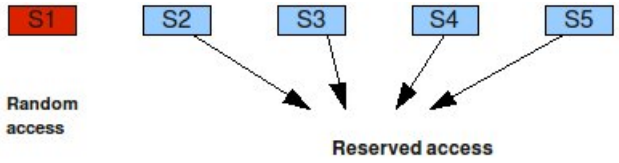

Figure 7.19: Slots with random and reserved access

The efficiency of channel utilization in slotted Aloha is given in equation 7.7. N is the number of nodes accessing the channel and P is the probability of message generation by each of the nodes.

$$S = N * P * (1 - P)^{(N-1)} \qquad where \qquad P = \frac{1}{N} \tag{7.7}$$

The table below shows the efficiency of channel utilization calculated for different nodes accessing the channel in a slotted aloha channel access mechanism.

Channel utilization efficiency in slotted aloha

number of nodes(N)	Efficiency(S)
2	50.00%
3	44.44%
4	42.19%
5	40.96%
6	40.19%
infinity	37.00%

If we calculate the efficiency for 4 GWs accessing the TETRA PDCH, for the first segment of the message the channel efficiency is 42.19% and for the remaining 4 slots is 100% (reserved). The channel utilization efficiency is given in equation 7.8.

$$Channel\ utilization\ efficiency = \frac{(0.4219 + 4) * 100\%}{5} = 88.4\% \tag{7.8}$$

So 11.6 % of the TETRA capacity will be wasted for contention in random accessing mechanism. The graph given in figure 7.20 shows the channel utilization efficiency for different number of GWs access the TETRA PDCH.

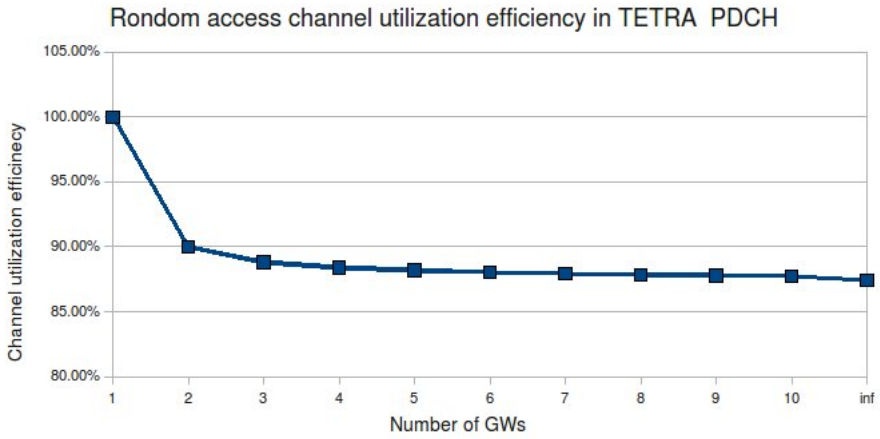

Figure 7.20: Channel utilization efficiency for different No. of GWs

From the graph given in figure 7.20, we can see that allowing many GWs to access the PDCH in TETRA increases traffic overheads or consumes extra bandwidth but using only one GW also has its own drawback. One of the main drawbacks is single point failure. If there is only one GW and all the nodes are registered through it, any communications will be stopped and re-registration of all the nodes will be required when the old GW dies or it is disconnected from the BON and a new GW is registered at the BON.

Considering the bandwidth wasted for contention as one kind of traffic overhead and adding to the control traffic overheads generated by the GWs, the total traffic overhead is calculated for different number of GWs accessing on a single TETRA PDCH at a time. For analysis purpose 50 nodes registered under each GW was considered. What we can see from the graph given in figure 7.21 is that we have to pay a lot of traffic overhead penalty to increase the robustness of the network.

The graph given in figure 7.22 shows the total traffic overhead of the minimal Flame protocol for different number of GWs and different number of total nodes registered under each GW. We can see from the graph below that for the same number of nodes in the network only increasing the GWs from one to two will add around 12% overhead and if the number of GWs is increased

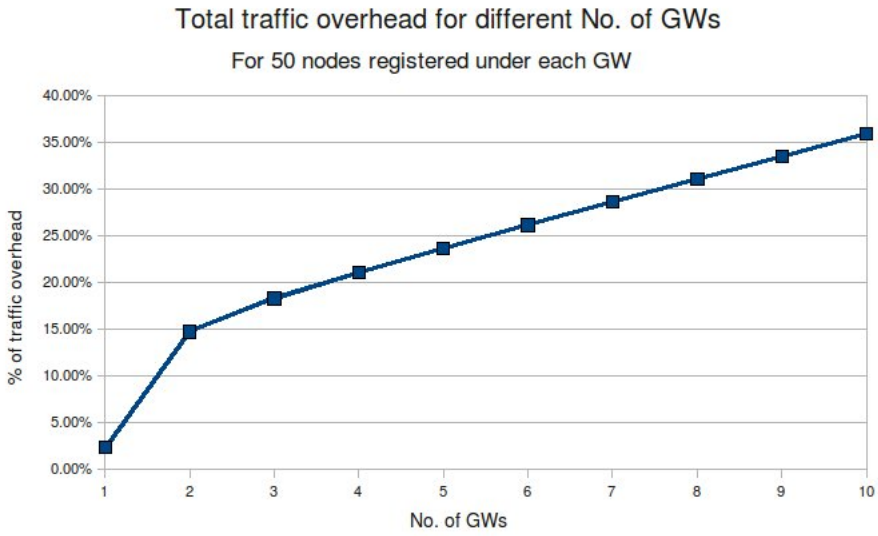

Figure 7.21: Traffic overhead for different number of GWs

to 10, the total traffic overhead will increase by more than 30% as given ion graph 7.22.

The solution to this kind of problem is to control the number of GWs in a TETRA cell. The BON can control the number of GWs in a TETRA cell and this can be done if the GWs register at the BON by the TETRA cell ID so that the BON will know how many GWs are accessing PDCH of a TETRA cell at a time. This mechanism only controls the number of GWs in a cell but it is difficult to know how many TETRA PDCH are available in a cell.

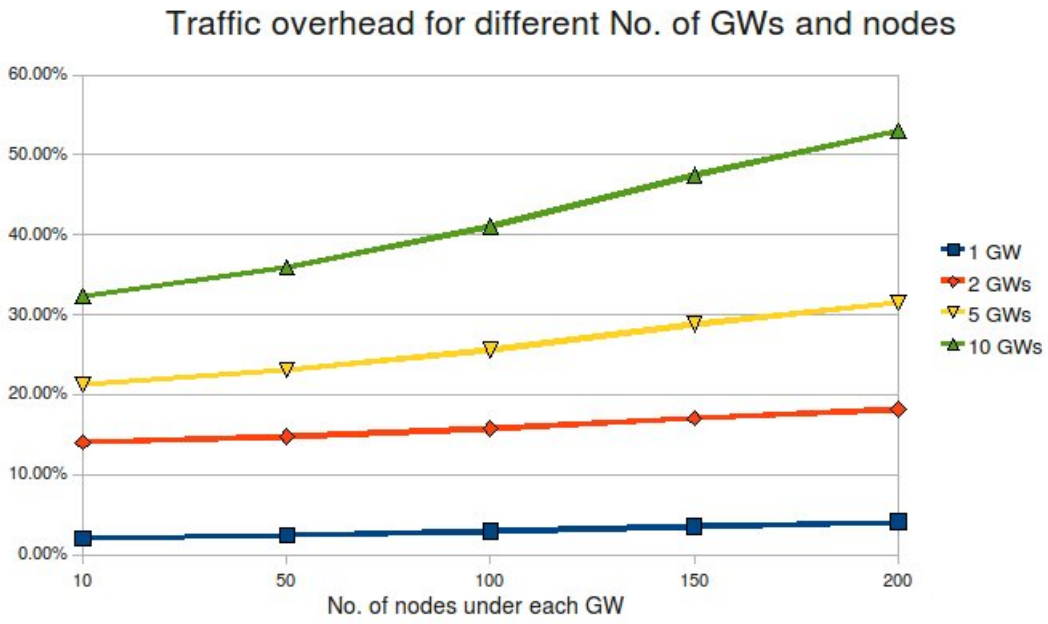

Figure 7.22: traffic overhead for different nodes under each GW

Chapter 8

Architecture based on MIPv4

The minimal Flame architecture proposed in Chapter 7 is based on Layer2 gateways that are used to isolate the local control messages of the adhoc network. The other possibility is to deploy Layer3 gateways (routers) so that any local meshing and other control messages will be limited within the adhoc network. But this architecture has another problem. Since the gateways are routers, clients in the adhoc network will be identified by their IP address. Clients are mobile nodes and they may move from one gateway to another. So, there will be IP mobility. There are two protocol that handle IP mobility. These are Mobile IPv4 (MIPv4) and Mobile IPv6 (MIPv6). MIPv6 has better features than MIPv4 but it has bigger header. MIPv6 can be used for broadband link but not advisable to be used for narrowband channels like TETRA. So, the architectural design proposed in this chapter for the TETRA-FIGO integration is based on MIPv4.

8.1 Introduction to Mobile IPv4

Mobile IP is an IETF standard communications protocol that is designed to allow mobile device users to move from one network to another while maintaining a permanent IP address [15]. In mobile IP, all required functionalities for processing and managing mobility information are embedded in well-defined entities, the Home Agent (HA), the Foreign Agent (FA), and

the Mobile Node (MN). A Correspondent Node (CN) is any node that is communicating with a mobile node while the MN is in a foreign network. The current Mobile IPv4 protocol is completely transparent to the transport and higher layers [20].

The Mobile IP protocol allows the MNs to retain their IP address regardless of their point of attachment to the network. This can be fulfilled by allowing the MN to use two IP addresses. The first one, called home address, is static and is mainly used to identify higher layer connections, e.g., TCP. The second IP address that can be used by a MN is the care-of address. While the mobile is roaming among different networks, the care-of address changes. The reason of this is that the care-of address has to identify the mobile's new point of attachment with respect to the network topology.

The HA uses the mobile's care-of address to tunnel IP packets from a CN to the foreign network and the packet is decapsulated and delivered to the MN. Due to the fact that the packet arrives at the MN, being addressed to its home address, it will be processed properly by the upper protocol layers, e.g., TCP [20]. The Mobile IP packet flow forms a triangular route as the CN sends the packet to the HA, then the HA will tunnel to the FA and the MN can directly send an IP packet to the CN as given in the figure 8.1.

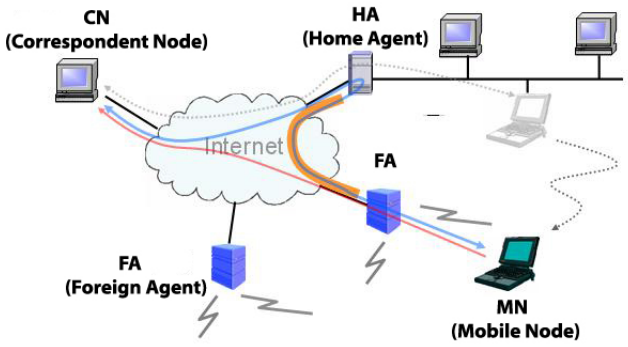

Figure 8.1: Triangular routing in MIPv4 [Internet]

There are mechanisms or procedures by which MIPv4 works. These mechanisms which are used in the architecture design are briefly given below.

Discovering the care-of address: The care-of address discovery procedure in Mobile IP is based on the ICMP (Internet Control Message Protocol) router advertisement standard protocol [20]. These router advertisements which contains care-of addresses are known as agent advertisements and are

broadcast at regular intervals by HAs and FAs. However, if a mobile needs to get a care-of address instantaneously, it can broadcast or multicast a solicitation to the FA or HA. The care-of address can be at the FA or it can also be collocated at the MN.

Registering the care-of address: After the MN gets a care-of address it has to inform the HA and this is accomplished by using a registration procedure. The MN sends a registration request to the HA. After accepting a registration request, the HA begins to associate the home address of the MN with the care-of address for a pre-specified time duration, called registration lifetime. The group that contains the home address, care-of address, and registration lifetime is called a binding for the MN. This binding is updated by the MN at regular intervals by sending a registration request to the HA.

Tunneling to the care-of address: The HA tunnels IP packets to the care-of address on the FA or the collocated on MN by using encapsulation mechanisms.

Proxy and gratuitous Address Resolution Protocol (ARP): When the MN is located in its home network, the other nodes use ARP cache entries for this MN. While a mobile node is registered on a foreign network, its HA uses proxy ARP [15] to reply to ARP requests it receives that seek the mobile node's link-layer address or it uses gratuitous ARP to update the ARP caches of nodes on the home network. This causes such nodes to associate the link-layer address of the HA with the MN's home (IP) address. The intercepted packets by the HA are then tunneled to the care of address.

Route Optimization in Mobile IP: Mobile IP protocol is extended to allow a more efficient routing procedures, such that IP packets can be routed from a correspondent host to a MN without going to the HA first [20]. This is done by sending binding update message to the correspondent node from the HA. The route optimization protocol uses binding warning, binding request, binding update, and binding acknowledgement control messages in oreder to properly operate [20].

After a brief explanation of the basics of Mobile IP, lets see how it can be used to integrate the TETRA network with wireless mesh networks. Two architectures are proposed in this section and these architectures are compared based on the traffic overhead, complexity and routing optimization.

The two proposed architectures are named as Architecture-1 and Architecture-2. Not many modification are applied in Architecture-1. But some changes to the VNs, the GWs and the BON are introduced in Architecture-2. Cross-

layer routing is assumed on the VNs and GWs in Architecture-2. The details are given in section 8.3.

8.2 Architecture -1

This architecture is when the HAs and FAs are located at the gateways on the adhoc network. A gateway acts as a home router to some of the clients. A client may select its home network when it first joins the FIGO network. Once it gets its home address, it can use this address as its permanent home IP address when it moves to a different gateway.

When a client moves to a different GW, it gets a care-of address from the FA of the new GW and it registers its care-of address at its HA. Any communication on the local network is on layer2 (it can be Flame1) and when a client (CN) on its home network wants to send an IP packet to a different client at a foreign network, it sends IP packet to the destination client's HA. The HA of the destination client inspects the IP packet and if the destination client is moved to a different gateway, it tunnels the packet using the care-of address of the client. Figure 8.2 shows the path of an IP packet sent from a CN in its home GW-3 (sub-network3) to a client/MN whose home network is sub-network1 but it has moved to sub-network2 (GW-2). This is inefficient because an IP packet from the CN traverses the narrowband link twice before it reaches the destination client. To improve this inefficient routing, route optimization can be used but an IP packet for the first time from the correspond node to a MN on a FA has to follow the path given in figure 8.2. Mobility binding messages between the HA and the CN will also add another control traffic overheads.

The worst case occurs when both the interacting clients are in different foreign GWs because a packet sent from any of the two communicating clients will traverse the narrowband link twice before it reaches the destination node. Apart from this, this architecture has another problem. The GWs in the adhoc network will be required to be meshed in layer3 otherwise every communication will be through the BON even when they can reach each other through other VNs in the adhoc network. So, every VN has to be a router (Layer3) in the adhoc network for meshing the GWs and mobile IP on a wireless mesh network meshed on layer 3 wouldn't be efficient because every router hides its layer2 communication and it is difficult to know which client is connected to which router when there is high rate of change

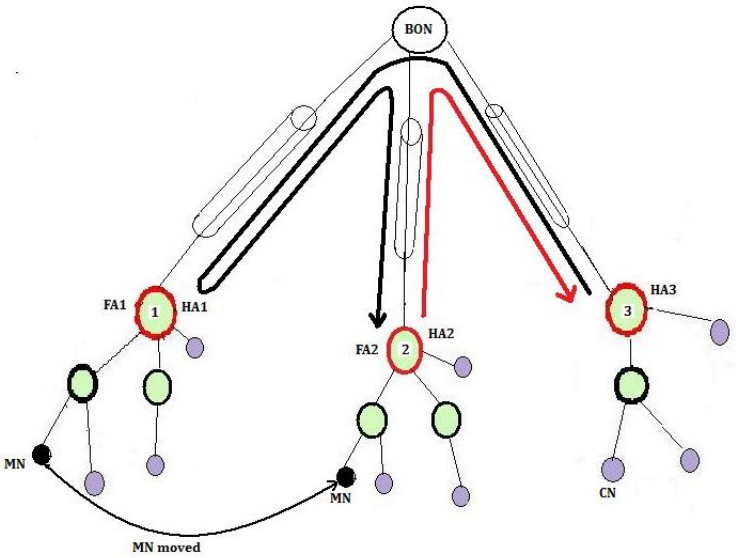

Figure 8.2: Architecture-1 when the HA and FA are on the gateway routes

of topology due to mobility.

Having seen the drawbacks of architecture-1 lets analyze the traffic overheads generated due to the control messages that pass through the narrowband link and the excessive header overheads that are present per packet. The traffic overheads of architecture-1 that are sent through the narrowband link are given as follows:

A. Routing protocol traffic overheads: The BON and the GWs are routers and they need to exchange routing information. The GWs can be meshed among each other in layer3 for Architecture1. OLSR is a common adhoc routing protocol and it can be used in the adhoc network for meshing the GWs. Only the traffic overheads over the narrowband link are seriously addressed in this project work. OLSR should not be used between the GWs and the BON because it generates a lot of control traffic overheads to the narrowband link. If we assume an OSPF routing protocol between the BON and the GWs, the routing information which are frequently exchanged among the routers are the Link State Update (LSU) and Hello messages. The LSU is sent at least every 30 minutes and packet size varies depending on the number of routers but it contributes less traffic overhead as it is sent in a long time interval. The Hello message is a 48 byte packet and it is

sent every 10 seconds in local CISCO routers but it is configurable. So, the traffic overhead of the Hello message of a GW to the BON on the TETRA link is given in equation 8.1. Where R_H is the rate of Hello message (every 10 seconds) and P_H is the packet size of the Hello message.

$$Overhead_{HELLO} = \frac{R_H * 8 * P_H 100\%}{R_T} = \frac{8 * 48 * 100\%}{10 * 1500} = 2.56\% \qquad (8.1)$$

The Hello message from the BON is broadcast to all the gateway routers on the downlink of the TETRA PDCH and the traffic overhead on the TETRA down link will depend on the number of gateways accessing through a single TETRA link.

B. Mobile IP protocol traffic overheads: The Mobile IP protocol by itself generates traffic overheads but this traffic overheads depend on the mobility of clients. The control messages generated by Mobile IP protocol is the mobility binding message and it is described below:

Mobility binding message: A client uses a registration message to register its care-of address at its home agent using the UDP/IP protocol. The registration message has an extension for authentication. If we assume HMAC-MD5 authentication protocol, the total packet size of the registration message including the authenticator field is 74 bytes. A manually distributed (out-of band) security key is assumed to avoid traffic overheads.

A client in a foreign network sends binding update messages to the HA at regular intervals before the binding expires. The binding update message is the same as the registration message in Mobile IPv4 and the updating rate is configurable and it has to be determined from the mobility of clients in the network and the capacity of the narrowband links. The traffic overhead due to the mobility binding between the clients on a foreign network and their HA of a single GW accessing through the TETRA link is given in equation 8.2. Where R_{mb} is the rate of binding update, P_{mb} is the packet size of the mobility binding message, K is the number of visiting nods on a foreign GW. For 10 nodes on a visiting network and 5 minutes binding update rate, the traffic overhead is 1.3%. In actual case the binding update rate should be very low.

$$Overhead_{binding} = \frac{8 * R_{mb} * P_{mb} * K * 100\%}{R_T} = \frac{8 * R_{mb} * 74 * K * 100\%}{1500} \quad (8.2)$$

The HA also sends a binding update to the correspondent node for route optimizations. The binding warning, binding request, and binding acknowledgement are also control messages used for routing optimizations. This kind of traffic overhead depends on the number of clients communicating with a correspondent node while they are on a foreign network and the duration of the session communication.

C. Header overheads: The other overhead in Mobile IP is the header because of the duplicate IP headers in the IP-in-IP encapsulation and the external TETRA IP. This overhead depends on the size of the payload of the IP packet. As the TETRA link is a narrowband link, a delay sensitive packet has to be small enough in order not to experience a lot of delay. Efficient and lossless IP header compressors can be used to reduce this kind of overheads but if we assume without any IP header compressor, there will be 76 bytes header per packet for a UDP/IP.

D. Service discovery protocol overheads: These protocols allow automatic detection of devices and services offered by these devices on the network. There are many types of service discovery protocols and most of them are broadcast on the network. These protocols may flood the network and care should be taken specially on the narrowband link. DHCP is one of the service discovery protocol and it shouldn't be allowed to be broadcast on the narrowband link either by configuring static IP to the clients or implementing DHCP at the GWs. If there is a special service provided by a device and all the nodes have to know this service, the gateway routers have to take actions to limit the rate of this broadcast on the narrowband link.

8.3 Architecture-2

In this architecture, the HA is located on the infrastructure (at the BON) and the FAs are located at the GWs with some modification to the GWs and the BON. There is a single HA and all the clients are located at foreign networks. All the gateway routers on the adhoc network act as FAs and

provide care-of address for the clients. All clients are on the visiting list of the gateway routers and these GWs act as slave routers as they don't provide home address to any clients. Since all the clients are on the foreign network, they will turn off their home Address Resolution Protocol (ARP) [15] and every client assumes that all the other clients are reachable through only the BON (HA). This will cause an inefficient routing among the clients under the same gateway router (FA). Route optimization on the adhoc network is proposed below. The other problem of this architecture is the traffic overhead of the mobility binding update message between the clients and the HA because all clients are in a foreign network. The solution to this problem is addressed in the following subsection.

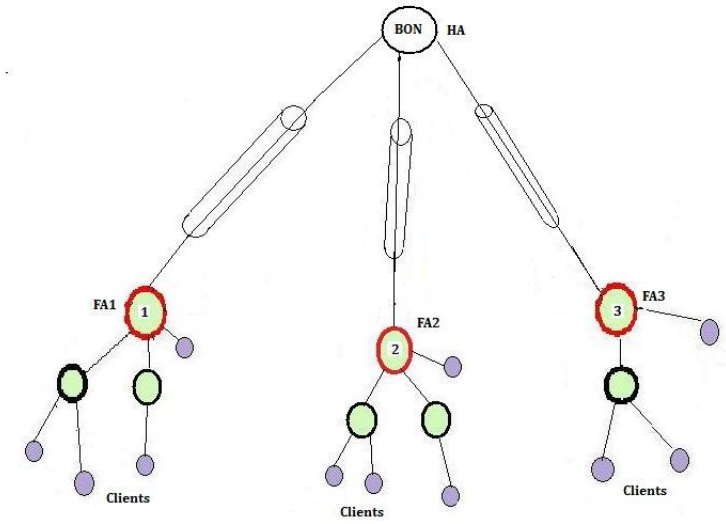

Figure 8.3: Architecture-2 when HA is at the BON

8.3.1 Mobility binding update traffic overhead

Architecture-2 solves the problem of traversing IP packets and a mobility binding message more than once on the narrowband link but it considers that all nodes are on the foreign network and all the clients will send mobility binding messages to the HA. This will increase the control traffic overhead as the number of clients increases. The traffic overhead is given in equating 8.3 where K is number of clients under GW (FA), R_{mb} is the mobility binding update rate and P_{mb} is the packet size of the mobility binding message. For

100 nodes under a foreign network and every 5 minutes binding update rate, the traffic overhead is 13.16%. In actual case the binding update rate is very low.

$$\begin{aligned} Overhead_{binding} &= \frac{8 * R_{mb} * P_{mb} * K * 100\%}{R_T} \\ &= \frac{8 * R_{mb} * 74 * K * 100\%}{1500} \end{aligned} \tag{8.3}$$

The traffic overhead of the mobility binding update will highly depend on the binding update rate (R_{mb}), that is, how often a client sends a re-registration message to the HA before the registration expires. Let's first see the effect of mobility binding message on the teachability of a client (MN) from the HA. In Mobile IP, a mobile node has to register its new care-of address at the HA when it moves to a foreign network or changes its FA. This registration is for a limited life time and the mobile node has to re-register before the registration expires if it stays on the same FA for longer time. So, what would be the effect if the registration life time is long? The only effect of this situation is that the HA wouldn't notice on time if the MN is disconnected from the network and it will forward packets destined to that MN to the FA where it was registered before. But the HA will be able to know if the client is dead or disconnected using ICMP or other means. So, it is possible to make the re-registration rate very low if the mobile node stays in the same FA which is the case in MIPv4 (the default registration life time is 36,000 seconds in MIPv4). If we ignore the re-registration traffic overhead, the mobility binding traffic overhead will only depend on the number of clients which are changing their FA at a time. The traffic overhead is given in equation 8.4 where K is the number of clients changing their FA, R_{fa} is average rate of change of FA by clients (every 60 seconds assumed) and P_{mb} is the packet size of the registration message.

$$Overhead_{binding} = \frac{8 * R_{fa} * P_{mb} * K * 100\%}{R_T} = \frac{8 * 74 * K * 100\%}{60 * 1500} \tag{8.4}$$

8.3.2 Route optimization in the adhoc network

The adhoc network consists of vehicular nodes (VNs) and any of the VNs can act as a GW if it has a TETRA card. So, selecting a GW is not a trivial problem. Flame2 may be used for solving this kind of problem. Apart from

this, there is one main problem that needs to be addressed in architecture-2. The problem is there is always inefficient routing among clients under the same FA. In architecture-2, all clients are in foreign networks and when a client wants to send an IP packet, it always sends it to the HA (single HA at the BON) because it assumes that the other clients are at the HA (at the BON). So, The FA (GW) has to be modified so that it can inspect the destination home IP address before it forwards to the HA and if the destination client's IP address is on the visiting address list, the FA should send the packet to the destination client instead of forwarding to the HA (BON). This mechanism is important because the data communication among clients under the same FA should not overload the narrowband link to the BON but it doesn't solve the inefficient routing on the local adhoc network. Every client in the adhoc network will forwarded its packet to the FA because a client in a foreign network turns off its home ARP. It can only learn the MAC address of the FA gateway from the agent advertisement [15]. So, the routing of IP packets between two clients under the same FA is not efficient because the packets have to always pass through the FA gateway. Hierarchical Mobile IP (HMIP) is a protocol of MIPv6 and it can't solve this kind of problem as it is designed to solve micro mobility but the problem here is different because every node in architecture-2 is in a foreign network. The solution to this problem is proposed as follows:

Lets assume there are different number of VNs with in the adhoc network and Flame1 is used for meshing the VNs as it was proposed in the Flame2 architecture [12]. The Path Update (PU) message is used in Flame to report about the bridged nodes (clients) in the FIGO network but it reports only the MAC address of the clients. So, the VNs don't know the home IP address of the clients in the adhoc network and routing information for the adhoc network is in layer2 until it reaches the GW (FA).

The solution proposed here is to use cross-layer routing at the end VNs where clients are connected to. In order for the VNs to use cross-layer routing, they are required to know the home IP address of the clients in the adhoc network and this can be done by broadcasting a message that contains the MAC and home IP address of a client by the FA (GW). This message can be a modified PU and it should be broadcast only once when a client is registering at the FA or on demand when the FA sees inefficient routing. Every VN in the adhoc network will be able to know which client is reachable through which VN using the Flame1 protocol and the modified PU that contains MAC and home IP of clients. Then, they can route packets on the adhoc network efficiently.

The FA should be located at the GW. The agent advertisement is broadcast by the GW and the VNs in two ways. One is for the clients and this shouldn't be in Flame frame format because the clients don't understand Flame protocol and it has to be done by all VNs and the GW. The other is for the VNs for multi-hop broadcast and it has to be done in a Flame frame format. When a client listens to an agent advertisement, it will register at the FA and HA. When a VN receives a packet from a client destined to the FA which is always the case, it first inspects the destination home IP address and if the destination node is in its routing table it will make a Flame data frame (Type1) and forward the frame to that destination node if not, it will simply forward it to the FA. Note that, the VNs don't need to inspect the IP header of a Flame1 frame forwarded from another VN. Only the end VNs inspect the IP header. The VNs can play a great role in handling micro mobility of clients because when a client moves to a different bridging VN without changing its FA, the Flame protocol will handle it. When the VNs receive agent advertisement from different GWs, they can select and broadcast only one agent advertisement to the clients bridged by them. Meshing of GWs wont be required in this architecture as the Flame will handle the meshing in the adhoc network.

Another important issue that need to be addressed in architecture-2 is when the link between the GW and the BON is not available, the clients can't register at the HA which is located at the BON and they assume they are unreachable and there will not be any communication among the clients in the adhoc network. This problem has to be solved by modifying the GW in such a way that it can reply to registration requests from clients when the link to the BON is broken or by using multi-homed agents.

8.4 Comparison of Architecture-1 and Architecture-2:

The routing protocol traffic overheads of architecture-1 are also available in architecture-2. The GWs in architecture2 shouldn't be meshed in layer3 because normally the GWs (FAs) will hide the clients connected to them and there will be inefficient layer3 routing in the adhoc network. The clients should statically be configured with a fixed home IP address to avoid the DHCP service discovery protocol overhead on the narrowband link. The mobile IP protocol traffic overhead is also the same with architecture-1 for

very slow re-registration rate (mobility binding). The mobility binding update from the HA to a CN in is not required for route optimization in architecture-2. IP-in-IP encapsulation can be eliminated in architecture-2 because there is already a TETRA IP tunnel but the BON will need to modify its routing mechanisms in such a way that it has to associate the TETRA IP address of the GW (FA) with the clients' care-off addresses so that the BON uses the TETRA IP instead of the care-of address when it tunnels an IP packet to the clients. All in all architecture-2 with all its route optimizations is better than architecture-1 to integrate TETRA with wireless mesh networks and this architecture is used to compare with minimal Flame in Section 8.5 below.

8.5 Comparison of the MIPv4 and minimal Flame

The minimal Flame architecture is better than the Flame2 in terms of control traffic overheads and header overheads because different optimization mechanisms are used in the minimal Flame protocol. This section is to compare MIPv4 and minimal flame architectures in terms of control traffic overheads, header overheads, complexity, scalability, and client modification.

Header overheads: The header overhead of the minimal Flame is 24 bytes if the proposed IP header compression is used if not it is 43 byte for a UDP/IP. The header overhead of MIPv4 for UDP/IP is 48 bytes without any header compression and with some routing modifications at the BON and the GWs.

Control message traffic overheads: The control traffic overheads of the minimal Flame are generated by the differential updating and the ARP protocol but there is no ARP traffic overhead on the narrowband link in MIPv4 architecture. Routing protocol and the mobile IP protocol traffic overheads described in section 8.2 are the control traffic overheads of MIPv4 architecture. The graph given in figure 8.4 shows the comparison of the control traffic of the two architectures with 10% of the clients under a FA changing their GW every 60 seconds assumed which was the case in the minimal Flame protocol traffic overhead analysis.

Complexity: Both the architectures require the modification of the GWs and the BON in the FIGO network but minimal Flame is much more complex than MIPv4 architecture because there is packet or frame modification in minimal Flame and it is required to handle these non standard packet

Figure 8.4: Control traffic overhead MIPv4 and minimal flame

formats or frame formats. IP header inspection is required in both architectures but in the architecture based MIPv4, the VNs are also required to inspect the IP header of the packets coming from the clients.

Scalability: The number of clients in the minimal Flame architecture depends on the number of VNs in the adhoc network. Each VN can support 255 clients. This is because of the IP address range allocated to each VN in the FIGO network and this IP address space is also considered in the header optimization. The number of VNs in the adhoc network also depends on the scalability of the Flame1 protocol. The number of clients in MIPv4 architecture only depend on the IP space of the HA. The FA can use a single IP address as care-of address for all clients register under it. So, MIPv4 architecture is much more scalable than the minimal Flame but still the scalability of Flame1 protocol used in the adhoc network part of the MIPv4 based architecture will limit the number clients that can be supported in the network.

Client modification: A client is not required to have new softwares except for security associations in minimal Flame protocol where as in MIPv4 architecture, the client is required to have Mobile IP client version in addition to the security association softwares.

Chapter 9

Application of the TETRA-FIGO integration

This chapter is to give an overview of the type of services that can be supported on the TETRA-FIGO integration from the simulations and measurements undertaken on the TETRA PDCH. The difference between the simulation set up provided in the literature and the TETRA-FIGO integration is also stated in this chapter.

9.1 Measurements and simulations over TETRA PDCH

There are measurements and simulations of IP data transmission over TETRA packet data service in the literature and it has been shown that the TETRA PDCH can't be used for interactive voice or data communication because of the bandwidth limitation but it can be used for non-interactive data communications like location information, remote data base query and transmission of sensor surveillance/security data [19].

The simulation of IP packets over TETRA PDCH considers many of the TETRA network parameters and protocols. In the simulation undertaken for small UPP/IP packets of (80, 100, 120 and 140 bytes message size) and different number of mobile stations accessing a single PDCH at a time with data message transmission rate of (1, 2, 3 msg/min) of each mobile station was considered [19]. Different scenarios of the combinations of message size,

number of users (MSs) and transmission rate was simulated. The simulators claim that the average packet delay and packet failure probability for all scenarios was an increasing function of the message size, the number of users and the message transmission rate. To take a typical example of the result, for a 1msg/min and less than 100 users, the delay remains approximately below 1,200 ms, and approaches 2,500ms as the number of users gradually increases to 300. For a 2msg/min rate the delay varies between 2,000 and 10,000ms for over 180 users, and the performance further degrades for the 3msg/min rate [19].

From the simulation results it is clear that the TETRA PDCH can't handle any interactive communications because for example, a 3msg/min of 120 byte message size means 6 bytes/sec of total transmission rate for each user and this can't handle any interactive communications.

9.1.1 Differece of the simulation set up and the TETRA-FIGO integration

The simulation set up descried above is different from the TETRA-FIGO integration architecture proposed in chapter 7 and 8. The simulation assumes that all the users have direct access to the available single TETRA PDCH and there will be a lot of contention on the main control channel and assigned secondary channel for the PDCH but only the GWs have direct access to the TETRA PDCH in the TETRA-FIGO integration and there will be less contention and even no contention when a single GW is allowed to access the PDCH. However, there are more control messages that are sent through the narrowband link in the TETRA-FIGO integration than in the simulation set-up.

The other difference is that in the TETRA-FIGO integration there are two headers per packet because the GW tunnels packets from the clients/users to the BON using the TETRA IP. So, a packet with its header from the clients is being tunneled through the TETRA network. But the simulation set-up assumes every user has a TETRA card is identified by the TETRA IP addresses and there is only one header per packet (the TETRA IP). It is clear that header compression will enhance the performance of the TETRA-FIGO integration for small size IP packets data communications because there is double header per packet on the TETRA link but this integration can't still be used for interactive communications. Because the clients and some of the VNs are not required to have TETRA card in the TETRA-

FIGO integration, few TETRA cards only for the GWs will be required and thus this integration is economical.

Chapter 10

Conclusion and future work

The TETRA-FIGO integration architectures proposed in chapter 7 and 8 above have optimized the control traffic overheads that passes through the narrowband link to the BON by introducing GWs to limit the control traffic overheads in the local adhoc network and by leaving out some of the control traffic overheads in the expense of link quality measurements. The comparative analysis of the control traffic overhead and header overheads for Flame2 protocol and minimal Flame protocol shows that these overheads can significantly be minimized in the minimal Flame architecture. The architecture proposed based on MIPv4 also solves many of traffic overheads because it uses routing in the IP level (Layer3). The local routing optimization in the adhoc network of the architecture based on MIPv4 introduces some traffic overheads and can limit the number of clients and VNs that can be supported under one GW. The comparison of the minimal Flame protocol with MIPv4 shows that minimal Flame is better than the architecture based MIPv4 in terms of control traffic overheads and almost the same in terms of header overheads with header compression applied to the minimal Flame.

Generally the TETRA-FIGO integration can't support interactive voice or data and non-interactive voice communications but it can support some non-interactive data communications. The TETRA network is a private network and secure enough so that the TETRA-FIGO integration can be used for security key distribution in the FIGO network and after the nodes get the keys, they can use the other communication systems (WiFi, UMTS) for data communications.

The proposed architectures were mainly designed to integrate the two specified networks, TETRA and FIGO. Some assumptions were considered in the architecture design and if we want to adapt these architectures to other networks, care should be taken with those assumptions because the robustness of the system can be affected due to those assumption. One of the assumptions in the TETRA-FIGO integration is that TETRA is a private and robust wireless network and its security is good enough to leave out some of the security measures like the OpenVPN in the existing FIGO network but this may not be true with other wireless networks.

Simulation and testing of the TETRA-FIGO integration was not undertaken in this project work because of time and material constraints and we propose it to be done as a future work. The simulation of the TETRA-FIGO integration should consider the adhoc and the infrastructure network parts. The adhoc mesh network is composed of the VNs communicating in multi-hop mode and these VNs bridge the clients which are out of the FIGO network. Most of the communication in the adhoc network is using WiFi. The infrastructure network comprises of the air interface of the TETRA network and the fixed network, which extends the link from the TETRA gateways to the BON. The simulation results should at least come up with the maximum number of clients that can be supported on a single TETRA channel for different packet size and different number of GWs.

Bibliography

[1] The SAFECOM Program, "Statement of Requirements for Public Safety Wireless Communications and Interoperability" March 10, 2004

[2] TETRA Memorandum of Understanding (TETRA MoU)

[3] Project MESA, "Service Specification Group - Services and Applications"

[4] Project MESA, "Technical Specification Group System; System and Network Architecture"

[5] John Dunlop, Demmissie Girma, and James Irvine "Digital Mobile Communications and the TETRA System", 1999

[6] ETSI EN 300 392-2 v2.6.1. (2006). Terrestrial Trunked Radio (TETRA); "Voice plus Data (V+D); Part 2: Air Interface (AI)." May 2006.

[7] D. I. Axiotis1, D. Xenikos2 "UDP Performance Measurements over TETRA IP"

[8] ETSI EN 300 392-5 V1.3.1 "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 5: Peripheral Equipment Interface (PEI)"

[9] Tom Lippmann, Twente Institute for Wireless and Mobile Communications, "FIGO system design release- 5"

[10] Michel Lammertink, Twente Institute for Wireless and Mobile Communications, "Forwarding Layer for meshing (Flame) design report", 2008

[11] P.Stavroulakis, "Terrestrial Trunked Radio-TETRA A globla security tool", 2007

[12] J.Stoter, S. Heemstra, Twente Institute for Wireless and Mobile Communications, "Flame2 L1 routing", 2009

[13] S.Heemstra de Groot, Twente Institute for Wireless and Mobile Communications, " Flame2 Routing Architecture", 2009

[14] Michel Lammertink, Twente Institute for Wireless and Mobile Communications, "Flame implementation report"

[15] IP mobility support for MIPV4, RFC 3344

[16] ETSI EN 300 392-7 V3.1.1 "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 7: Security"

[17] Tom Lippmann, Neill, Twente Institute for Wireless and Mobile Communications, "FIGO link quality report"

[18] Dimetra release 5.1/5.2, "Packet Data Service; PD-Programmers Guide"

[19] D. I. Axiotis1, Apostolis K.Salkintzis "Packet data measurements over TETRA: network performance analysis", 2009

[20] Georgios Karagiannis, and Geert Heijenk, "Mobile IP"

[21] Public protection and disaster relief spectrum requirements, "ECC Report 102"

Appendices

.1 State diagram of the New Minimal Flame protocol

The minimal flame protocol is mainly designed to optimize the narrow link between the GW and the BON. It assumes Flame2 protocol to be used on the ad

hoc network with some additional functionalities where nodes are registered at the GW by their MAC and IP address. The current IP addressing mechanism in the FIGO network is assumed where the VNs are assigned a range of IP addresses to provide IP addresses to the clients connected to them. The subnet address space of the current FIGO network X.Y.A.R/16 is considered.

When a GW is registered at the BON, it will advertise about its TETRA link to the nodes close by. The clients and VNs will be registered or identified at the GW by their MAC and IP address. Then the GW will send this address table to the BON. The last two bytes of the IP address is used for registering and identifying clients at the BON. These last two bytes of the IP can also be used to identify clients on the adhoc network so that the VNs can use these short MAC addresses for any IP data communication in the local adhoc network but this document assumes Flame2 to be used on the local ad hoc network.

Once the address table is sent to the BON from the GWs, any data communications can take place between any of the nodes in the network with infrequent table inconsistency check. The state diagram of the VNs and the BON is given in the following subsections.

..1 State diagram of the VNs

The are two main states at the VNs and figure 1 shows the state diagram of the VNs and each state can have many sub-states under it but here only the main states are given. The action undertaken at each state are also described below.

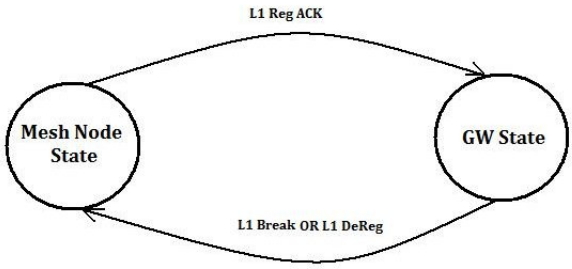

Figure 1: State diagram of the VNs

A. Mesh node state: At this state the VN is not registered at the BON as a GW and it will function as normal mesh node of the ad hoc FIGO network. It then starts to register as a GW at the BON if TETRA link is the best link by sending L1 Reg Req message to the BON. If the BON replied with L1 Reg NACK, the VN should not try to register again. After the VN is registered as a GW, the BON may send L1 De-registration message (L1 DeReg) to the GW if it decides that the VN should no more act as a GW. This can happen if no or too few nodes are registered at the GW. The GW may also send L1 DeReg to the BON for the same reason. If the link to the BON is broken at any time, the GW has to come to the mesh node state but there should be a threshold or temporary break state to tolerate the dynamic behaviour of the wireless link.

L1 Registration has to be done using the L1 registration message of Flame2 because the VN cant use the minimal Flame protocol before it registers by its tunnel ID and its MAC address at the BON. The BON has to associate the tunnel ID with the MAC address of the GW so that any message coming through that tunnel is from that specified GW. This is mainly because the external MAC address is removed in the minimal flame protocol. The new minimal flame protocol frame is sent as the payload of the UDP/IP of the TETRA network. The VN may also send the TETRA cell ID to the BON so that the BON can control the number of GWs on a cell accessing the TETRA PDCH. This is because commonly there is only a single PDCH available in a TETRA cell and many GWs shouldnt be allowed to access this single channel at a time. After it has received an L1 Reg ACK from the BON, the VN will start to act as GW and it starts advertising about its TETRA link.

B. GW state: The VN enters to this state when it receives L1 Reg ACK from the BON. Some of the action undertaken at this state are given below.

Node registration at GW: After the VN is registered at the BON by its tunnel ID (TETRA IP address) and its MAC address, the GW will advertise about its TETRA link. Then nodes will register at the GW by their MAC address and IP address. This node registration message is by using the GW Registration Re- quest of Flame2. The Node Base (NB) should contain the MAC address and IP address (at least the last two bytes of the IP). If no Node/client is registered at the GW, it may send L1 DeReg message to the BON because it is not worthy staying as a GW without any client connected to it. When there are nodes register at the GW, it will send a Node registration message to the BON.

Node registration at the BON: The GW has to send Node Register message s to the BON when there are nodes registered under it. The GW expects Node Reg ACK messages from the BON and if no ACK or Node Reg NACK is received within the specified time, it will try to send the Node Register message again until the number of tries time out. The BON also use the NACK message to deny access for specific clients/nodes connected at the GW.

Data exchange: The GW and BON exchange data packets and process buffered actions. Node Add and Node Remove messages are exchanged between the GW and the BON. The table consistency check is also done by sending a hash mes- sage at a slow rate.

Table Consistency check: There are many reasons by which the consistency of the table at the GW and the BON to be affected. One reason is when an error on the Node Reg message is not detected by the CRC and there could also be a power glitch which affects the table consistency. To avoid these kind of problems, the GW sends a hash of the its table and the BON will also check this Hash by calculating the hash of the specified address table of the GW. If the Hash calculated by the BON is not the same as the hash sent from the GW, the BON will send a hash NACK to the GW and the GW has to send the complete table to the BON. If many nodes are registered at the GW and the table is big, it can be split up into blocks to avoid sending a big table as proposed on Flame2 [12] but the TETRA link wont support many nodes and this kind of optimization is not considered in the new minimal protocol.

..2 State diagram at the BON

The state diagram at the BON is similar to the state diagram at the GW and it has to map the actions at the GW which incorporate the BON. Figure 2 shows the state diagram at the BON for each GW and similar to the state diagram of the GW, the main state are depicted on this state diagram.

A. Default State: At this state the VN is not registered as a GW at the BON. So, there is no activity for the VN at this state. When the BON receives L1 Reg message from a VN, it can reply with L1 Reg ACK or L1 Reg NACK. When the BON replies with L1 Reg ACK, it will enter to the GW Registered State and when it replies with L1 Reg NACK, it will stay at this Default State. The BON shall send L1 Reg NACK if it has decided that the VN should not be allowed to act as a GW on the TETRA link. This may happen when it is decided that a limited number of GWs are allowed to access the TETRA link at a time.

B. GW Registered State: The BON enters this state when it replies L1 Reg ACK to a VN. When the link is broken or L1 DeReg message is exchanged between the BON and the GW at this state, the BON will enter to the Default state. Some of the action undertaken at this state are given below.

Node Registration: The BON expects the GW to send Node Register messages and it acknowledges these messages or it is also possible to use the Node Reg NACK message by the BON to deny access for specific nodes connected at the GW. If the GW is not sending Node Register messages and no node is register at the BON through that GW, the BON may send L1 DeReg message to the GW so that the VN will no more act as a GW.

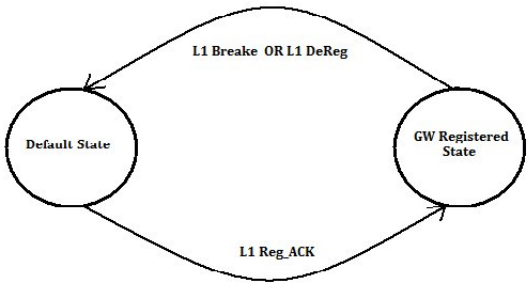

Figure 2: State diagram at the BON

Data exchange: Data frames are forwarded to the GW and the Node Add, Node Remove and hash messages are processed. This is the map of the data exchange activity at the GW.

Table consistency check: The BON will reply with hash NACK to the GW when there is table inconsistency and the GW will send the whole table to the BON. Sending the whole table is just the same as sending Node Reg message to the BON for the whole nodes under the GW. The table consistency check can be considered as one sub-state at the BON or GW. After getting the whole table and checked the consistency, the BON and the GW can exchange any data packets.