

Project no.: IST-FP6-STREP- 26979
Project full title: Highly dependable ip-based networks and services
Project Acronym: HIDENETS
Deliverable no.: D3.1.2 - Annexes
Title of the deliverable: Report on resilient topologies and routing – final version - Annexes

Contractual Date of Delivery to the CEC:	30 th June 2008	
Actual Date of Delivery to the CEC:	27 th June 2008	
Organisation name of lead contractor for this deliverable	Telenor	
Author(s):	Inge-Einar Svinnset (editor), Marius Clemetsen, Geir Egeland, Audun Fossellie Hansen, Tom Lippmann, Yaoda Liu, Erling V. Matthiesen, Anders Nickelsen, Jens Myrup Pedersen, Thibault Renier.	
Participant(s):		
Work package contributing to the deliverable:	3	
Nature:	R	
Version:	1.0	
Total number of pages:	69	
Start date of project:	1 st Jan. 2006	Duration: 36 month

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Keyword list: resilience, ad-hoc network, routing, multi-channel, multi-radio, multi-homing, multi-topology, multi-path.

Version	Date	Comments
0.1	14.03.2008	First draft, first contribution to the different sections.
0.2	21.04.2008	Updated version
1.0	07.05.2008	Version for internal review

Table of Contents

ACRONYMS.....	4
A RELATED COMMUNICATION TECHNOLOGIES.....	9
A.1 RELATED LINK LAYER TECHNOLOGIES	9
A.2 RELATED NETWORK LAYER TECHNOLOGIES	22
A.3 RELATED HIGHER LAYER TECHNOLOGIES	24
B OVERVIEW OF RESILIENCE SCHEMES.....	28
B.1 RESILIENCE SCHEMES AT L2.....	28
B.2 RESILIENCE SCHEMES AT L3.....	29
B.3 RESILIENCE SCHEMES AT L4.....	33
B.4 RESILIENCE SCHEMES AT L5.....	34
B.5 CROSS-LAYER CONSIDERATIONS.....	37
C APPLICATIONS AND REQUIREMENTS.....	39
D THE FAULT HIERARCHY - A FRAMEWORK FOR FAILURE MANAGEMENT	42
D.1 MOTIVATION	42
D.2 HIERARCHY AND LAYERING	42
D.3 THE FAULT HIERARCHY.....	44
D.4 FAILURE-DETECTION.....	47
E COMPARISON OF 802.11B, 802.11A AND 802.11G.....	49
E.1 SCOPE OF THIS SECTION	49
E.2 OVERVIEW OF THE LEGACY 802.11 DSSS PHY.....	49
E.3 OVERVIEW OF 802.11B.....	50
E.4 OVERVIEW OF 802.11A.....	52
E.5 OVERVIEW OF 802.11G.....	53
E.6 RECOMMENDATION FOR THE HIDENETS PROJECT.....	54
F ALLOCATION OF PACKET DATA CHANNELS AND CHANNEL RATES IN GPRS.....	54
G NODE SOFTWARE ARCHITECTURE	55
G.1 INFRASTRUCTURE MOBILITY SUPPORT – CLIENT PART	55
G.2 IN-STACK MONITORING & ERROR DETECTION	56
G.3 PERFORMANCE MONITORING	57
G.4 GPRS/UMTS RADIO RESOURCE MANAGEMENT.....	58
G.5 TRANSPORT LAYER FUNCTIONS	58
G.6 NAMING SERVICE	59
G.7 RESOURCE/SERVICE DISCOVERY	59
G.8 SESSION CONTROL.....	59
H ABC SIMULATION PARAMETERS.....	59
I MULTI-CHANNEL MULTI-RADIO STATE-OF-THE-ART	61
I.1 MAC PROTOCOLS	61
I.2 MULTI-CHANNEL ROUTING.....	63
I.3 OTHER ISSUES	64
REFERENCES	65

Acronyms

3GPP	3rd Generation Partnership Project
ACK	Acknowledgement
AES-CCMP	Advanced Encryption Standard with the Counter mode CBC-MAC Protocol
AHN	Ad-Hoc Node
AHNet	Ad-Hoc Network
AHP	Ad-Hoc Path
AIFS	Arbitration Inter-Frame Space
AIFSN	Arbitration Inter-Frame Space Number
AODV	Ad-hoc On-Demand Distance Vector
AODV-RM	AODV with Radio Metric
AP	Access Point
APSD	Automatic Power Save Delivery
AR	Access Router
ARP	Address Resolution Protocol
ARQ	Automatic Repeat reQuest
AS	Authentication Server
ASAP	Aggregate Server Access Protocol
ATIM	Asynchronous Traffic Indication Map
BGP	Border Gateway Protocol
BS	Base Station
BSC	Base Station Controller (GSM)
BSS	Basic Service Set (Infrastructure WLAN mode)
C2C	Car-to-Car
C2I	Car-to-Infrastructure
CAC	Connection Admission Control
CALM	Continuous Air interface for Long and Medium distance (ISO TC204 WG16 draft standard on ITS)
CCA	Clear Channel Assessment
CCF	Common Channel Framework
CCH	Control CHannel
CCK	Complimentary Code Keying
CCK-QPSK	CCK with QPSK, see CCK and QPSK
CDS	Connected Dominating Set
CE	Circuit Elimination
CLD	Cross-layer design
CRC	Cyclic Redundancy Check
CS	Coding Scheme (CS-1 etc in GPRS)

CSCF	Call State Control Functionality
CSMA	Carrier Sense Multiple Access
CSMA-CA	CSMA Collision Avoidance
CTS	Clear To Send
CTX	Clear To transmit on another channel
DBPSK	Differential Binary Phase Shift Keying
DFS	Dynamic Frequency Selection
DiffServ	Differentiated Services
DCF	Distributed Coordination Function
DPSK	Differential Phase Shift Keying
DQPSK	Differential Quadrature Phase Shift Keying
DS	Distribution System
DSRC	Dedicated Short Range Communication
DSSS	Direct Sequence Spread Spectrum
DTIM	Delivery Traffic Indication Map
EAP	Extensible Authentication Protocol
EDCA	Enhanced Distributed Channel Access (IEEE 802.11e)
EDGE	Enhanced Data rates for Global Evolution
ENRP	Endpoint Name Resolution Protocol
ESS	Extended Service Set
FCS	Frame Check. Sequence (typically used in the context of IEEE 802.11 WLAN)
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FISHEYE	Fisheye State Routing
FN	Fixed Node
FW-GW	Fixed-Wireless Ad-hoc GW
GEOCAST	Geographic Addressing and Routing
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for the Mobile Communications
GW	Gateway
HR/DSSS	High Rate DSSS
HSS	Home Subscriber Server
HWMP	Hybrid Wireless Mesh Protocol
IBSS	Independent Basic Service Set (WLAN mode)
IE	Information Element
IEEE	Institute of Electrical and Electronics Engineers

IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IS-IS	Intermediate System to Intermediate System
ITS	Intelligent Transportation System
L1	Layer 1 or the "Physical layer"
L2	Layer 2 or the "Link layer"
L3	Layer 3 or the "Network Layer"
L4	Layer 4 or the "Transport Layer".
LAN	Local Area Network
LANMAR	Landmark Routing
LAR	Location-Aided Routing
LLC	Logical Link Layer
LSP	Label Switched Path
MAC	Media Access Control
MANET	Mobile Ad-hoc NETwork
MAP	Mesh Access Point
MCS	Modulation Coding Scheme (EDGE)
MDA	Mesh Deterministic Access
MIB	Management Information Base
MIP	Mobile IP
M-MIP	Multi-homed MIP
MN	Mobile Node
MP	Mesh Point
MPLS	Multi Protocol Label Switching
MPP	Mesh Portal Point
MPR	Multi-Point Relay (used in proactive ad-hoc routing protocols, such as OLSR)
MT	Mobile Terminal
MTR	Multi-Topology Routing
NACK	Negative Acknowledgement
NAV	Network Allocation Vector
NE	Neighbour Elimination
NeMo	Network Mobility
NS	Name Server
NS	Node Set
OBU	On Board Unit
OFDM	Orthogonal Frequency Division Multiplexing
OLSR	Optimized Link State Routing Protocol

OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PBCC	Packet Binary Convolutional Coding
PCF	Point Coordination Function
PDCH	Packet Data Channel (GSM)
PDP	Packet Data Protocol
PE	Pool Element
PER	Packet Error Rate
PHY	Physical Layer (typically used in the context of IEEE 802.11 WLAN)
PLCP	Physical Layer Convergence Protocol (sub-layer of PHY)
PMD	Physical Media Dependent (sub-layer of PHY)
PPDU	Packet Data Unit
PU	Pool User
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RA-OLSR	Radio-Aware OLSR
RLC	Radio Link Control
RPI	Received Power Indication
RREP	Route Reply (used in reactive ad-hoc routing protocols, such as AODV)
RREQ	Route Request (used in reactive ad-hoc routing protocols, such as AODV)
RSerPool	Reliable Server Pooling
RSU	Road Side Unit
RTS	Request To Send
RTX	Request To transmit on another channel
SCH	Services Channel
SCTP	Stream Control Transmission Protocol
SDU	Service Data Unit
SIFS	Short Interframe Space
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SNDCP	SubNetwork Dependent Convergence Protocol
SNR	Signal-to-Noise Ratio
SSID	Service Set Identifier
STA	Station
TBF	Temporary Block Flow (GPRS)
TBR	Tree Based Routing
TBTT	Target Beacon Transmission Time
TCP	Transmission Control Protocol

TDMA	Time Division Multiple Access
TFI	Temporary Flow Identifier (GPRS)
TG	Task Group
TIM	Traffic Indication Map
TKIP	Temporal Key Integrity Protocol
ToS	Theft-of-service
TPC	Transmit Power Control
TRX	Transmitter Receiver Unit
TSF	Timing Synchronization Function
UMTS	Universal Mobile Telecommunications System
VANET	Vehicular Ad-Hoc Network
WAVE	Wireless Access in Vehicular Environments (IEEE draft standards 802.11p & 1609)
WEP	Wired Equivalent Privacy
WG	Working Group
WLAN	Wireless LAN
WME	Wireless Multimedia Extension
WMM	Wi-Fi Multi-Media
WOSPF	Wireless OSPF
WP	Work Package (referring to HIDENETS WPs)
WPA	Wi-Fi Protected Access (which uses TKIP)
WPA2	Wi-Fi Protected Access version 2 (which uses AES-CCMP)
WT	Wireless Terminal
WW-GW	Wireless-Wireless Ad-Hoc GW
ZRP	Zone Routing Protocol

A Related Communication Technologies

In this chapter we describe some communication technologies that are of particular importance to the HIDENETS projects. These are technologies that form a basis for our work. The majority of the functionality might be reused in the HIDENETS project. However, HIDENETS might propose changes or additions to these technologies in order to enhance resilience

A.1 Related link layer technologies

A.1.1 Introduction

In the following we provide information about link layer technologies that are considered of particular interest to the HIDENETS project. In the first three of the following sections 3.1.2 - 3.1.4, we consider technologies related to the IEEE 802.11 standard for WLAN, while in section 3.1.5 we provide information about GPRS and UMTS.

802.11 is probably the most important link layer technology to the HIDENETS project. The IEEE 802.11 medium access control (MAC) comprises the mandatory Distributed Coordination Function (DCF) as a contention-based access scheme, and the optional Point Coordination Function (PCF) as a centrally controlled polling scheme. However, PCF is hardly implemented in any products, and DCF represents the commonly used MAC mechanism of 802.11 and the one that will be considered in the HIDENETS project.

DCF adopts carrier sense multiple access (“listen-before-talk”) with collision avoidance (CSMA/CA) and uses binary exponential backoff. A station not only goes into backoff upon collision. It also carries out a “post-backoff” after having transmitted a packet, to allow other stations to access the channel before it transmits the next packet.

The details of how the MAC layer of 802.11 works are assumed well-known. Thus, it is beyond the scope of this deliverable to provide a detailed description of the IEEE 802.11 MAC. Instead, in the following sections, we describe the amendments to the 802.11 standard that are considered particularly relevant to the HIDENETS project. These amendments comprise solutions for multi-channel operation (802.11h), for mesh networking (802.11s) and for the use of WLAN in vehicular environments (802.11p).

The physical layer (PHY) extensions of 802.11 are important, but not in the main focus of the HIDENETS project. The most commonly used PHY extensions are described in Appendix A, which gives a comparison of 802.11b, 802.11a and 802.11g.

A.1.2 Overview of 802.11h for frequency selection to support multi-channel operation in HIDENETS

A.1.2.1 Scope of this section

The purpose of this section is to provide an overview of the work carried out by the IEEE 802.11 TGh group on an 802.11h amendment to the 802.11 standard. It will focus on the mechanisms that are anticipated to be particularly important to the HIDENETS project

The work on 802.11h amendment was ratified in July 2003.

A.1.2.2 Introduction

In 1999 the 5 GHz band was regulated for the 802.11a OFDM. For use of this unlicensed band, the regulations posed some requirements to give preference to military radar operations in the spectrum (especially with respect to European radar operations).

802.11h addresses these requirements, but can also be used for other purposes. For example, 802.11h can be used to detect activities in the spectrum other than those relating to radar, such as 802.11a activities and OFDM activities in general. Furthermore, 802.11p and 802.11s plan to base their multi-channel operations on 802.11h.

802.11h provides the following features:

- Dynamic Frequency Selection (DFS)
 - Measurements of radio channels to discover radio activity, including those relating to 802.11a, OFDM and radar. They can be carried out independently, or measurements can be coordinated between two nodes, for example between an AP and a STA.
 - Multi-channel operation in terms of a coordinated movement from one channel to another, e.g. in order to avoid radio interference.
- Transmit Power Control (TPC):
 - This mechanism can be used to minimize the overall power output of the system.

A.1.2.3 Additions to the legacy IEEE 802.11 protocol

First, 802.11h specifies new Information Elements (IEs) to be used in various management frames.

- Power Constraint
- Power Capability
- TPC Request
- TPC Report
- Supported Channels
- Channel Switch Announcement
- Measurement Request
- Measurement Report
- Quiet
- IBSS DFS
- Extended-Rate PHY Information
- Extended Supported Rates

Second, 802.11h introduces a new management frame type, namely the Action Frame. It is used for request/response exchanges between nodes. In fact, today its usage extends beyond only 802.11h and a number of other 802.11 amendments reuse the Action Frame for other purposes (e.g. 802.11e uses it for exchange of traffic specifications). The Action Details subfield of this frame contains an IE describing the specific action required to be carried out. Some of the IEs mentioned above, including the TPC Request, TPC Report, Measurement Request and Measurement Report and Channel Switch Announcement and IEs, are carried in action frames. Others are carried in discovery frames (Beacon/Probe) or in association frames.

A.1.2.4 TPC (Transmit Power Control)

The Power Constraint IE is sent by the AP and carried in Beacon and Probe Response frames. It is used by the STA to calculate the maximum radio frequency transmit power measured in dB for the current operating frequency channel. It contains a “Local Power Constraint” value that informs the STA about how much the max transmit level is below the maximum transmit level in that country (measured in dB).

If the Beacon/Probe contains a Power Constraint IE, the STA must respond by reporting its minimum and maximum transmit power range it supports, using the Power Capability IE. This IE is carried by a (Re-Association Request) frame. If the STA's power adjustment range is outside the requirements of the AP, the STA's association request might be rejected.

The AP might also send a TPC Request (i.e. a TPC Request IE carried in an Action Frame) to a STA. In this way, the AP requests the STA to report the transmit power level and the link margin (in dB and calculated using free space loss theory) of the PHY. The STA includes the report in a TPC Report IE, which is carried by an Action frame back to the AP.

A.1.2.5 DFS (Dynamic Frequency Selection)

During association, the STA provides a list of supported channels to the AP (Figure 1). The information is included in a Supported Channels IE carried by the association request frame. The AP then approves or rejects the association request based on channel information.

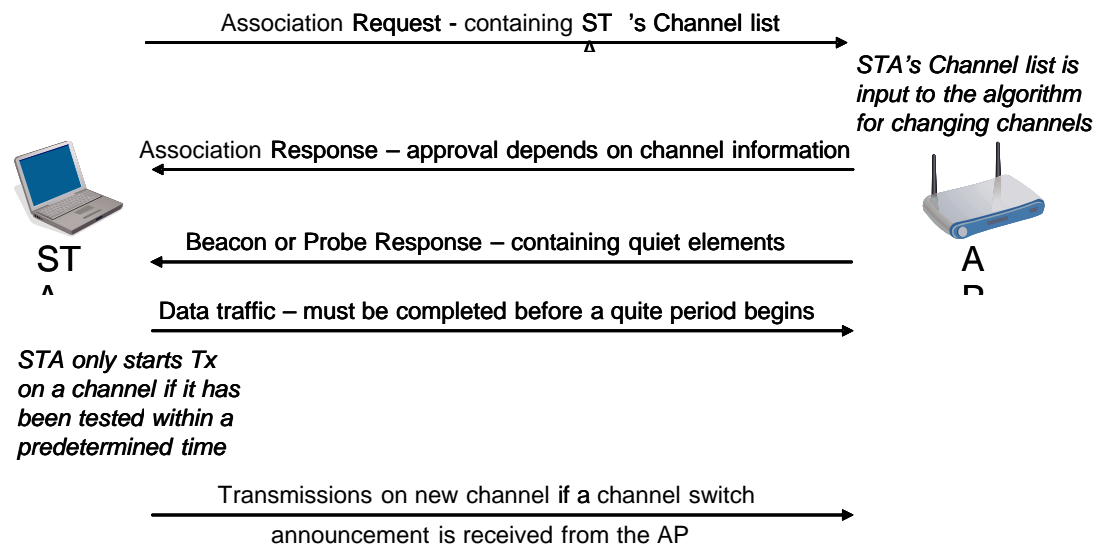


Figure 1: The Dynamic Frequency Selection (DFS) of IEEE 802.11h is used to avoid co-channel interference. It first allows the access point (AP) to permit or deny support for a channel.

Whenever an AP schedules a quiet interval by means of a Quiet IE in order to test for radar, stations may not transmit on that channel during the quiet interval (Figure 1).

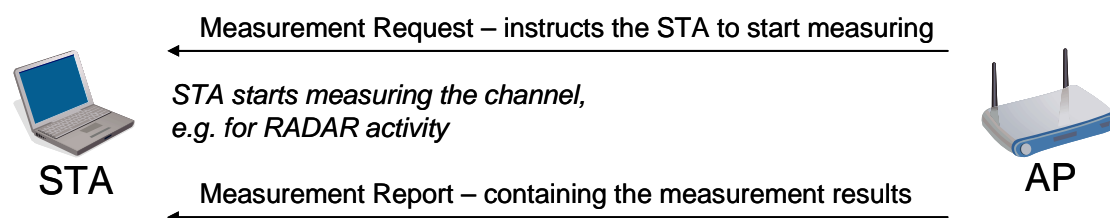


Figure 2: IEEE 802.11h allows the access point (AP) to collect measurements from an associated station (STA)

Later, the AP might request measurements from a STA via a Measurement Request IE in an Action frame (Figure 2). The STA carries out the measurements and responds by sending a Measurement Report back to the AP.

The AP might request a number of different kinds of measurements, and the requested measurement type is indicated in the Measurement Request IE. 802.11h specifies the following types of measurements: basic request, Clear-Channel Assessment (CCA) request and the Received Power Indication (RPI) histogram request. The Measurement Request IE also specifies which channel to measure, the start time (relative to the time given by the TSF of 802.11), and the requested duration of the measurements.

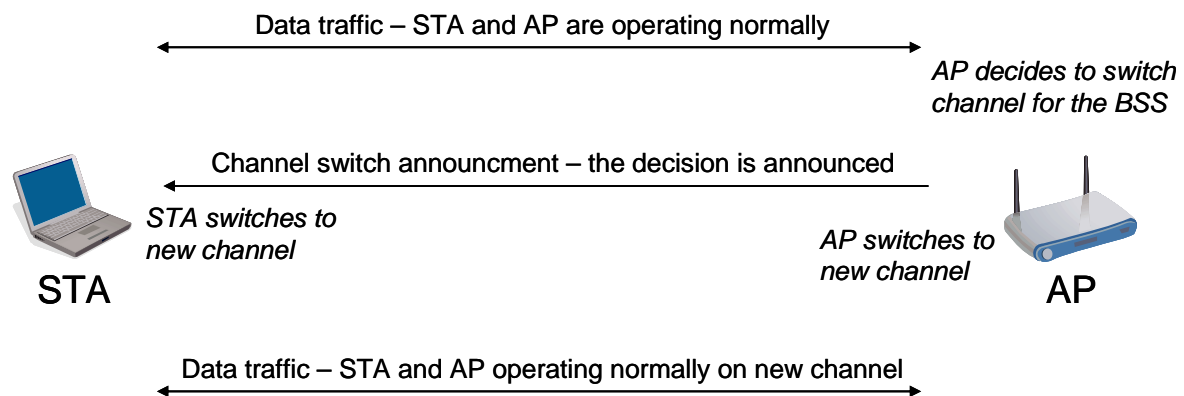


Figure 3: IEEE 802.11h allows the access point (AP) direct the station (STA) to switch to another channel.

Finally, the AP may utilize a Channel-Switch Announcement Action request to inform STAs when they should switch to a specified channel. The STA subsequently responds by switching to the new channel (Figure 3).

A.1.3 Overview of 802.11s for Wi-Fi Mesh Networking

A.1.3.1 Scope of this section

The purpose of this section is to provide an overview of the work undertaken by the IEEE 802.11 TGs group on an 802.11s amendment to the 802.11 standard. It will focus on the mechanisms that are anticipated to be particularly important to the HIDENETS project

The work on 802.11s is currently (middle of 2006) still in progress. Recently two competing proposals, SEE-Mesh and Wi-Mesh, have been merged into one joint proposal, and will form a basis for further work towards a final specification. In the following, the current status of this proposal will be presented.

A.1.3.2 Introduction

The objective of 802.11s is to work out an extension of 802.11 that will allow for multi-hop mesh networking.

The technology is based on the following building blocks:

- Topology
- Discovery
- Security
- Interworking
- Path Selection and Forwarding
- Beaconing, Synchronization, Powersave
- Separate MAC Enhancements

These features will be described in subsequent sections.

A.1.3.3 Topology

A.1.3.3.1 **Background: Legacy 802.11**

The basic 802.11 standard defines only two basic topologies:

- One topology is the infrastructure Basic Service Set (BSS) is managed by an access point, and stations (STAs) associates with the AP. Furthermore, many BSS may be combined to form a logical Extended Service Set (ESS), identified by a specific SSID, and with inter-AP communication through a proprietary distribution system (DS).
- The other topology is the Independent Basic Service Set (IBSS) where all stations (STAs) are equal peers, and there is no AP to take a centralized control over the network.

Legacy 802.11 defines no wireless multi-hop topologies. This is where 802.11s comes into play.

A.1.3.3.2 **Enhancements made by 802.11s**

The mesh network consists of nodes with different roles: The Mesh Point (MP) is a node that establishes links with other MPs, does routing and forwarding, and is a full participant in the mesh network. A Mesh AP (MAP) is an MP that also operates as a STA that is not a part of the mesh network. A Mesh Portal point (MPP) is a point where frames enter and exit a mesh network.

A.1.3.4 Discovery

A.1.3.4.1 **Background: Legacy 802.11**

802.11 defines two ways for a STA to discover APs (in BSSs) or other STAs (in IBSSs): The STA either scans passively for Beacons or actively for Probe Responses by first transmitting a Probe request. The STA will normally scan on different frequency channels. The Beacons and Probe Responses contain Information Elements (IE) with information about the BSS or IBSS that the originator of the packets belongs to.

With 802.11s STAs use basically the same method to discover the mesh network.

A.1.3.4.2 **Enhancements made by 802.11s**

Nodes discover each other with Beacons (or Probe Response frames), and a specific Information Element (IE) is defined for this purpose. The mesh network can span multiple channels.

A node joins the mesh network by associating securely with a neighbour that is an MP in the network. The network has a given Mesh ID, which is similar to the SSID used for ESS operation of legacy 802.11.

A.1.3.5 Security

A.1.3.5.1 **Background: 802.11i provides security to legacy 802.11**

The basic 802.11 standard uses WEP for security. However, it is well accepted that WEP is inherently insecure, and the more recent 802.11i standard from 2004 is the de-facto standard for current and future security for legacy 802.11 networks.

802.11i uses TKIP (WPA) or AES-CCMP (WPA2) for confidentiality and message integrity protection of data frames. Furthermore, a slightly modified version of 802.11x is used to carry authentication in EAP (Extensible Authentication Protocol [95]) messages over the wireless medium. This is used in both BSSs and IBSSs, however, the details of the authentication process differs slightly.

In a BSS the STA operates as supplicant and the AP as authenticator, while the authentication server (AS) might be a centralized (radius) server. The EAP method produces a master session key that is used to derive pairwise keys between the STA and the AP to protect unicast traffic. The AP is it also to derive a group key used for broadcast/multicast traffic that it transmits. After the authentication, the pairwise keys are distributed and synchronized through a 4-way handshake. The AP use the same 4-way handshake to also send the initial group keys to the STA. Since STAs may enter and leave the network frequently, the group keys might need to be changed often, and the AP might do this by a 2-way handshake that is designed specifically for group key renewal.

In an IBSS, on the contrary, a STA takes the role of both a supplicant and an authenticator, and it might also take the role of an AS. Each pair of STAs in the IBSS needs to form a pairwise key for the unicast traffic they are exchanging. Furthermore, each STA also need to form a group key for all multicast/broadcast traffic it is transmitting, and to convey this key to all neighbours. In order to keep the same Authenticator Key State-Machine for both BSS and IBSS, the initial key establishment uses the 4-way handshake also in IBSS. Since each of the two STAs that authenticate with each other, need to convey a group key to the other STA, a 4-way handshake is done in each direction (i.e. there are two 4-way handshake between any pair of nodes). However, since there is only need for one pairwise key between each pair of STAs, it is the 4-handshake initiated by the STA with the highest MAC address that produces the pairwise key to be used for the unicast traffic exchanged between them.

While 802.11i only provides protection of data traffic, the new 802.11w standard is underway in order to also protect management frames.

As we will see in the following, 802.11s reuses 802.11i (and 802.11w) for security. It is mainly the IBSS mode of operation of 802.11i (secure IBSS) that will be reused by 802.11s.

A.1.3.5.2 Enhancements made by 802.11s

The scope of security is link-security while end-to-end security is out of scope (e.g. one could use IPSec or something else). 802.11s allow for association and authentication between neighbouring MPs and MAPs. It is based on the ad-hoc security model of 802.11i, where each MP acts as supplicant and authenticator for each of its neighbours and the 4-way handshake of 802.11i / 802.11x for session key distribution is used.

For encryption and integrity protection of the transmitted data frames, 802.11i is also used. The pairwise keys are used for unicast traffic between two neighbours. When transmitting broadcast and multicast traffic, the node uses its session group key.

In addition to taking advantage of 802.11i for protecting data frames, 802.11s will use 802.11w in order to protect control and management frames. This is important, since for example routing messages will not be transmitted as data frames (as it normally would if the routing was IP-based, such as it is with the routing protocols developed in the MANET WG of IETF). Instead, routing will be carried by management frames, and these need to be protected.

A.1.3.6 Interworking

A.1.3.6.1 Background: Legacy 802.11

The basic 802.11 standard has no need for an interworking mechanism, since all communication is single-hop. The AP often works as a switch in a switched Ethernet, and ARP is used to determine to which other node in the BSS to deliver an IP packet.

For 802.11s, on the contrary, interworking is an issue that needs to be addressed by the specification.

A.1.3.6.2 Enhancements made by 802.11s

With 802.11s, the MP first need to determine if a frame it shall forward is destined for a node inside the mesh network or outside. If the destination is inside, the frame is forwarded along the path established by the routing protocol. (This kind of Path Selection and forwarding will be discussed in the next section). If the destination on the contrary is outside the mesh network, the MP must unicast the packet to one of the Mesh Portals.

802.11s will also define the use of 802.1D in the mesh portals (MPPs) for bridging a mesh network with a wired LAN.

A.1.3.7 Path Selection and forwarding

A.1.3.7.1 Background: Legacy 802.11

The basic 802.11 standard has no need for path selection (i.e. routing) and forwarding mechanisms, since all communication is single-hop. Needless to say, this is obviously not the case for 802.11s.

A.1.3.7.2 Enhancements made by 802.11s

Each mesh network uses a specific routing protocol and a specific routing metric. A node that wants to join the mesh network acquires this information already during the neighbour discovery, i.e. this information is discovered along with the Mesh ID before associating with the mesh network.

The default routing protocol is AODV with Radio Metric for on demand service. AODV is based on flooding RREQs to discover the reverse route to a destination, and the RREP returned by the destination to form the forward route.

Furthermore, tree based routing (TBR) can be used in conjunction with AODV for proactive services. An example is that a Mesh Portal can use TBR to announce routes to the rest of the mesh network.

Both AODV and TBR use distance vectors, and together they are referred to the Hybrid Wireless Mesh Protocol (HWMP). While HWMP is a mandatory protocol, there are optional protocols, too. One example is Radio Aware OLSR (RA-OLSR).

In fact, various radio metrics can be used in combination with the routing protocol to make the routing more robust against link failures. Examples include the airtime metrics and the weighted radio aware and load aware (WRALA) metric. The first metric reflects the cost of the channel, the path and the packet error rate. The latter take the PHY and MAC protocol overhead, the frame size, bit rate, link load and error rate into account.

In terms of forwarding, 802.11s uses the 4-address format of legacy 802.11. The implication of this is that a MAP that receives a 3-address format frame from one of the associated STAs, converts the frame to the 4-address format before injecting the frame into the mesh network.

The frame also contains additional fields, such as the QoS field of 802.11e. 802.11s introduces a new field, the time-to-live subfield, where the time-to-live value is decremented by each MP that forwards the frame. The intention of the field is to avoid problems with infinite loops in the network.

A.1.3.8 Beaconing, Synchronization and Powersave

A.1.3.8.1 Background: Legacy 802.11 and APSD of 802.11e

The Timing Synchronization Function (TSF) of 802.11 nodes is particularly important to allow for power save schemes. A STA that is in power save mode may turn off and go to "sleep" and only wake up at given intervals to receive packets from the AP or another node. This needs to be coordinated, and keeping synchronized time is necessary.

In a BSS the AP is responsible for maintaining the TSF. At regular time intervals (i.e. at the Target Beacon Transmission Time – TBTT) it transmits a beacon, which contains the time of the AP. The other STAs adjust their local TSF to the clock of the AP and keep synchronized in this way. The beacon also contains information about the beacon interval, so the other STAs will be able to predict the upcoming TBTTs. Thus, if a STA misses a beacon (e.g. due to the fact that the wireless medium is inherently unreliable), it will not lose synchronization, and it will probably be able to receive the next transmitted beacon. To allow scanning STAs also to keep synchronized with APs they discover, the time of the AP is also included in Probe Responses.

In an IBSS the TSF must be maintained in a distributed manner, and all STAs are responsible for transmitting beacons containing their time information. The principle is to let it be regulated in the long run by the fastest running clock. Thus, upon reception of a beacon from another STA, the recipient updates the local TSF time if the time in the beacon is larger than the local time. The received beacon also indicates the moment for the next TBTT. When the TBTT is reached, all STAs are responsible for transmitting a beacon. However, to avoid a situation with a flood of beacons, exponential back-off algorithm is used, and a backoff countdown (using twice the normal minimum contention window) must be undertaken before the STA can

transmit its beacon. If it receives a beacon from another STA during backoff, its own attempt to transmit a beacon is aborted. In this way, only one beacon is transmitted on each TBTT. However, if several beacons are transmitted, this does not undermine the functionality of the distributed TSF algorithm.

In a BSS, the STA can go into power save by letting the AP buffer packets for it while it sleeps. The STA may wake up only on every n'th TBTT to receive a beacon. The beacon contains a traffic indication map (TIM) that tells whether or not there is a packet waiting for the STA in the buffer of the AP. If there is a packet waiting, the STA transmits a PS-poll frame. If the returned data frame has the "more data" bit on, the STA might continue polling the AP. For multicast/broadcast frames, the AP uses a Delivery TIM (DTIM).

In a IBSS, any STA may buffer packets for another STA. An ATIM-interval follows the beacon transmission during which no STAs are permitted to sleep. During the ATIM-interval a STA that has buffered a frame unicasts an announcement traffic indication message (ATIM) to the destination of the frame. This makes the recipient to stay awake until the conclusion of the next ATIM interval.

To address the requirements of e.g. voice traffic with frequent wake-up intervals, the 802.11e standard introduced automatic power save delivery (APSD). It allows a STA to set up a "schedule" for delivery of frames, based on a repeating pattern of a specified number of beacon intervals. When APSD has been enabled, the AP will buffer the APSD station's frames for the number of beacon intervals specified in the APSD setup. The time offset within the beacon interval can be specified, allowing a number of stations to wake up at different times during one beacon interval to receive their traffic. Automatic Power Save Delivery is a more efficient power management method than legacy 802.11 Power Save Polling. Most newer 802.11 STAs already support a similar power management mechanism to APSD.

A.1.3.9 Enhancements made by 802.11s

Synchronization is an optional feature, and it is only supported if at least one MP in the mesh network requires it. If synchronization is not used, the MAP will still need to maintain synchronisation for the STAs it is serving as an AP. If synchronization is used in the mesh network, the IBSS synchronization method of legacy 802.11 is adopted by the MPs.

If synchronization is used, the same beaconing method as in an IBSS can also be used. Without synchronization, a separate beacon collision avoidance algorithm might be used.

In 802.11s, MPs that support power save may go to sleep. While support of neighbours sleep/wake cycles is optional, coordination of sleep/wake cycles over multiple hops is not supported.

802.11s specifies the use of different mechanisms for power save:

- The same ATIM/DTIM mechanism as specified for legacy 802.11 IBSS
- A similar solution as the APSD mechanism specified for 802.11e BSS :
 - Periodic APSD. Periodic sleep/wake schedules are coordinated between a pair of neighbours. This is anticipated used for QoS traffic, such as VoIP.
 - Aperiodic APSD: An MP in power save state uses this method only with neighbour that are always awake. The power saving MP wakes up at any time it wishes, and sends a message to the "always-awake" neighbour to indicate that it has woken up.

A.1.3.10 Other MAC enhancements for 802.11s

A.1.3.10.1 The need for further MAC enhancements

802.11 is not particularly well suited for ad-hoc network in terms of performance. Multi-hop communication is demanding, and additional mechanisms are required to increase the efficiency of multihop communication.

802.11s addresses this by requiring additional mechanisms for:

- traffic differentiation to favour the performance of some traffic classes in a demanding environment;
- multi-hop congestion control to increase overall multi-hop performance;
- communication on multiple channel to increase the potential bandwidth of the network
- QoS for periodic flows, similar to the QoS features provided by the 802.11e standard.

The corresponding MAC enhancements proposed by 802.11s are described in the following.

A.1.3.10.2 Enhancements made by 802.11s

The MAC enhancement that is probably the most important for the HIDENETS project is the proposed use of 802.11e EDCA for media access. This allows for possibilities for QoS and differentiation, which are issues quite relevant for HIDENETS.

It is also proposed a congestion control mechanism to be used within the mesh network. The purpose is to increase efficiency in a network with heterogeneous link capacities along the path of a flow. Each node monitors the local channel utilization and notifies the previous hop neighbours (i.e. signalling by unicast request and response packets) or its entire neighbourhood (i.e. signalling by a broadcast announcement packet) about the observed congestion. The packets trigger the nodes that receive them to adjust its traffic generation rate accordingly. Note however, that rate control is on a per-Access-Category basis. This means that different QoS traffic classes will be affected differently by the congestion control mechanism so that best effort traffic might need to slow down while for example voice traffic can continue without being affected.

802.11s will also provide a Common Channel Framework (CCF) that accommodates MAC operation of multiple channels, where one of the channels can be a "common channel". CCF supports a channel switching mechanism where RTX/CTX messages are exchanged on a common channel, and an MP pair switches to a destination channel and then switches back. Generally, the RTX is used to suggest a channel to switch to while CTX is used to accept or decline the suggested channel. Groups of MPs may also use CCF to switch to a negotiated channel. To support multi-channel operation, the interfaces need to support 802.11h to take advantage of the Dynamic Frequency Selection (DFS) features defined here.

Mesh Deterministic Access (MDA) is an optional mechanism to improve QoS of periodic flows. A handshake between a transmitter and a receiver sets up the deterministic access opportunities, referred to as the MDAOPs.

A.1.3.11 Recommendation

802.11s seem to be very relevant for HIDENETS WP 3 and it is therefore recommended that the project continue surveying the development within the TGs Task Group of the IEEE 802.11 Working Group.

A.1.4 IEEE 802.11p and related standardization activities

A.1.4.1 Introduction

This section summarizes the standardization activities around the world on car to car communication systems. We start with a list of relevant standardization activities. As examples to show the tendency in C2C communication systems (especially in the physical and medium access control layer), we introduce in more details two specific draft standards, i.e. IEEE 802.11p and related standards (IEEE 1609.4 mainly). We select the IEEE 802.11p and IEEE 1609.4 as our main basis for further research. These two standards are developed in US, but we expect that similar settings can/will be deployed in Europe because there is no counterpart of these two standards in Europe. The reason for looking into these standards is that the C2C consortium is considering these technologies as the most likely ad-hoc link layer techniques, and they are also considered by the major European C2C projects (GST [93], CVIS [65] etc).

We are aware of the following C2C communication related standardization activities

- WAVE: Wireless Access in Vehicular Environments
 - IEEE 802.11p: Wireless LAN Medium Access Control (MAC) and physical layer (PHY) specifications. This standardization activity aims at defining the lower layers of the communications stack
 - IEEE 1609.1: WAVE Resource Manager
 - IEEE 1609.2: 5.9 GHz Intelligent Transportation System (ITS) Radio Service Security
 - This standardization activity aims at defining 5.9 GHz DSRC Security (formerly IEEE 1556)
 - IEEE 1609.3: WAVE Networking Services Provides description and management of the DSRC Protocol Stack
 - This standardization activity aims at providing description and management of the DSRC Protocol Stack
 - IEEE 1609.4: WAVE Multi-Channel Operation
 - This standardization activity aims at providing DSRC frequency band coordination and management
- CALM: Continuous Air interface for Long and Medium range [62] [63]
 - Draft ISO standard (ISO TC204/WG16) which is an umbrella-standard for ITS. It incorporates the IEEE WAVE standard, but also technologies like GPRS/UMTS, infrared and DSRC. The recently started European CVIS project [65] builds on the CALM architecture,
- ETSI TG 37: The task group of ETSI responsible for liaising with other groups active in the field of intelligent transport system, for example CEN TC 278 and ISO TC 204, to provide ETSI deliverables as appropriate, in particular for protocol specifications and conformance testing. It also acts as a focus within ETSI for ITS activities, and will liaise with ERM RM on spectrum issues.
- ITU Y.1541: Recommendation Y.1541 [90] is a specification of standardized QoS classes for IP networks. Recommendation Y.1541 suggest to group services into classes defined according to the desired QoS requirements.

A.1.4.2 Physical layer of C2C communication system

IEEE 802.11p specifies the physical layer of the WAVE system for C2C communication. Here we start with the requirement of the WAVE system on Packet Error Rate. With those requirements in mind, we introduce the design strategy behind the WAVE system.

The WAVE systems are required to support long range operation (up to **1000m**), high speed vehicles (up to **283Km/h**), the extreme multi-path environment, the need of multiple overlapping ad-hoc networks to operate with extremely high quality of service, and the nature of the applications to be supported.

More specifically, the WAVE devices should be able to fulfil the following requirement:

- The road-side unit should be able to transfer messages to and from vehicles each travelling at speeds up to 140 km/h with a Packet Error Rate (PER) of less than 10% for PSDU lengths of 1000 bytes
- The road-side unit should be able to transfer messages to and from vehicles at speeds up to a minimum of 200 km/h with a PER of less than 10 % for PSDU lengths of 64 bytes.
- For vehicle-to-vehicle communications, DSRC devices should be able to transfer messages at relative speeds of up to a minimum of 283 km/h with a PER of less than 10 % for PSDU lengths of 64 bytes.

A.1.4.2.1 Frequency band

In IEEE 802.11p, the frequency band is divided into 7 logical channels for differentiated services. Different type of application may operate in different channels, e.g. non-safety critical application may not operate in the dedicated control channel (CCH).

A logical channel is typically a 10MHz frequency band, but channels can optionally be combined to form a 20 MHz channel. One channel is a dedicated control channel (CCH), the others are service channels (SCH). The control channel will only be used for WAVE safety messages and channel assignment overhead.

In the US, 7 channels have been assigned for ITS. In Figure 4, the US frequency plan is shown, where channel 178 is the control channel and the other six are service channels. Of the service channels, 172 and 184 have been reserved for safety-related data.

In Europe, it is most likely the 7 channel structure in IEEE 802.11p will be adopted. However, the frequency bands may be different. [ETSI/ERM](#) in collaboration with [CEPT](#) provides solutions for the available [frequency spectrum](#). In addition, ETSI/ERM TG 29 specifies devices for short range communication in close collaboration with [CEN/TC 278](#). These standards all assume the 5,8 GHz as the frequency band for Intelligent Transport System. Other frequency spectrum, i.e. in the area of 60 GHz, has been reserved and will be standardized for car-to-car communications.

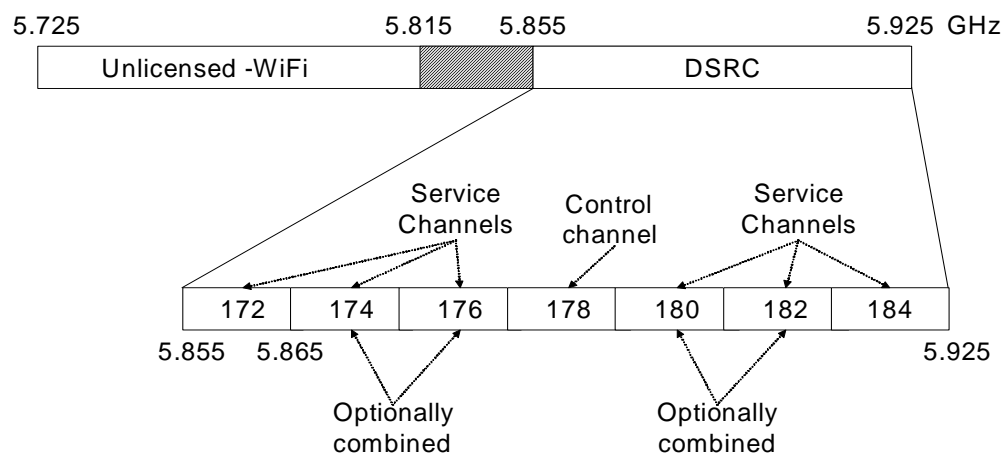


Figure 4: IEEE 802.11p Frequency band overview

A.1.4.2.2 WAVE transmitter power limit

To prioritize important applications and devices, different maximum transmitter power is allowed for different applications, devices, and in different frequency channels. With this specification, packets from public safety application are allowed to be transmitted with higher power, but private applications and vehicle safety application are given the same priority.

A.1.4.3 Medium access control layer of C2C communication system

IEEE 802.11p and IEEE 1609.4 define jointly the medium access control layer of the C2C communication system.

A.1.4.3.1 General architecture

Figure 5 is the general architecture of MAC in a DSRC device. Basically it is an extended version of IEEE 802.11a and IEEE 802.11e to enable multi-channel operation and application differentiation.

In IEEE 802.11e, there are two modes: EDCA and HCCA. Although it is not clearly mentioned EDCA will be used as the only option for WAVE operation, EDCA seems to be the choice.

Several extensions have been introduced to the WAVE system to fulfil the requirements. First, as management application is given the highest priority to all other applications including public safety applications, a separate queue is therefore maintained as shown in the left part of the figure. Whenever there are packets from management application, no other packets can be sent to the PHY layer for channel contention.

Second, more than one set of queues will be maintained at each WAVE device, one for the data flow at the control channel, and the other for the data flow at the service channels. The procedure of each set of queue follows the EDCA as defined in IEEE 802.11e. The priority in wireless channel contention is given to higher classes via congestion window and frame space (AIFS as shown in the table).

In CCH, RTS and CTS are not allowed. One of the reasons could be that packet transmitted in CCH are expected to be of small size, hence RTS&CTS are not cost efficient, only increasing the communication overhead, and hence the collision probability and interference level.

A.1.4.3.2 Multi-channel operation

The multi-channel operation depends greatly on the underlying PHY layer, i.e., the number of available radio interfaces. Multi-radio system is easier to deal with as an obvious solution is to keep one radio for the control channel, and the others for service channels. However, for one radio WAVE system, a WAVE device has to switch between control and service channels. An on-board-unit (OBU) should switch back to the control channel if no packet addressed to this OBU is received within 100ms because it is required to visit the control channel for some time every 100ms.

SCH and CCH intervals are stored in the WME MIB of the provider RSU. The channel selector utilizes a channel switching mechanism modified from IEEE 802.11h by using the channel time intervals. For an infrastructure WAVE BSS, the channel switch announcement is transmitted by using WAVE beacon frames. For an independent WAVE BSS (described as an “ad-hoc” network in IEEE 1609.3), the channel switch delay is transmitted by using WAVE announcement action frames. The WAVE beacon and announcement action frames are described in IEEE 802.11p.

After the channel switch occurs, the prioritized access activities on the previous channel are suspended, and the prioritized access activities on the current channel are started or resumed if they were suspended. The prioritized access activities on one channel are described in IEEE 802.11e. The channel switch function shall also ensure that no packet is delivered to the PHY layer just before the channel switch, so that packets shall not be transmitted on the incorrect channel.

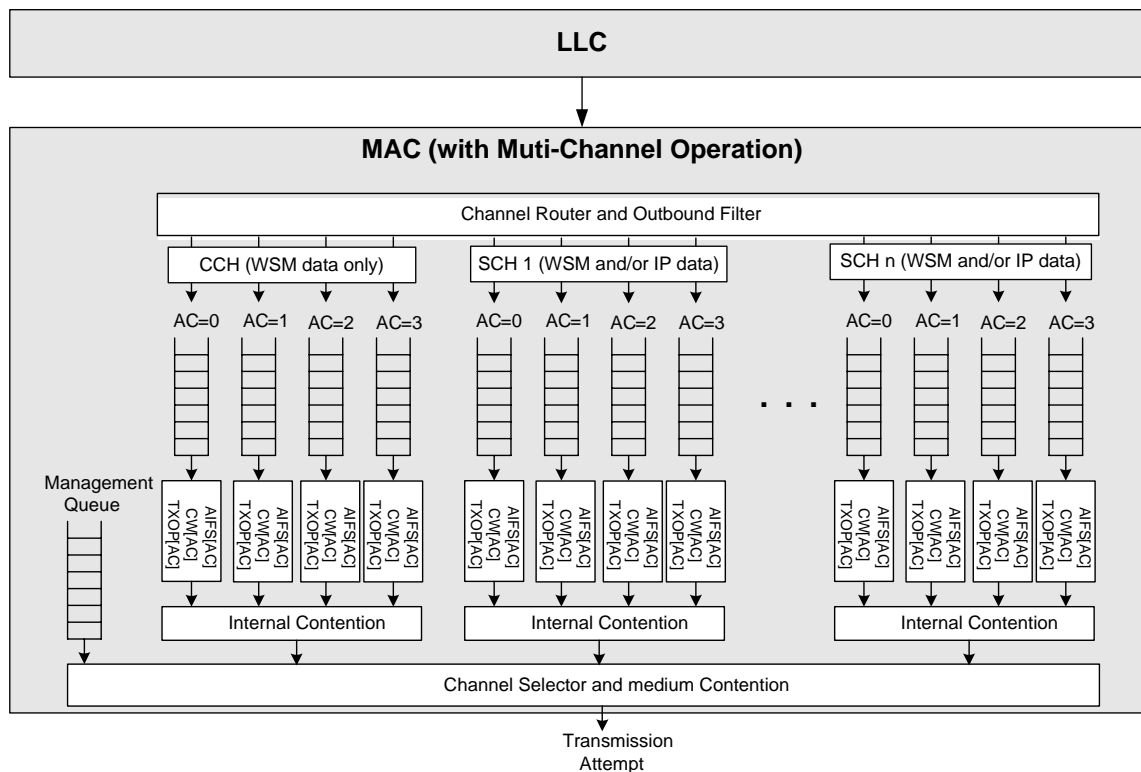


Figure 5: General architecture of MAC (with Multi-Channel Operation)

A.1.5 GPRS / UMTS

UMTS/GSM/GPRS networks, as opposed to WLANs, exercise strict, centralised control over the radio resources. The goal is to ensure the QoS agreed upon during session setup and reject sessions when resources can no longer be guaranteed. For such purposes various Connection Admission Control schemes are used in addition to load control, handover control and power control. GSM/GPRS and UMTS are built on quite different radio technologies, and as such the resource allocation regimes over the air interfaces are also rather different.

By adding GPRS functionality to the current GSM network, the operators can give subscribers wireless access to the Internet. The basic idea of GPRS is to provide a packet-switched bearer service to the GSM network.

On the physical layer GSM and GPRS use a combination of FDMA and TDMA for multiple access. The GSM radio resources are divided into channels or carriers of 200 kHz width. A transceiver (TRX) is needed in each GSM cell for each 200 kHz frequency channel in use and such frequency channel carries eight TDMA channels by being divided into eight timeslots.

The operator can allocate the time slots for either voice or data. This can be done dynamically depending on the traffic mix or certain timeslots may be reserved for e.g. data. Timeslots allocated for data are called packet data channels (PDCH). It will then be a maximum of 8 PDCHs in each TDMA frame. For GPRS the data rate for each PDCH varies between 9.05 kbps and 21.4 kbps, depending on the coding scheme used.

A Mobile Terminal (MT) transmitting or receiving IP packets is allocated a PDCH for this purpose. The IP packets are fragmented into radio blocks that are transmitted in four consecutive timeslots within one PDCH.

Some further details around allocation of Packet Data Channels in GPRS can be found in Appendix F.

UMTS does not have a frame structure like GSM so resource allocation is a more complex matter. The resources that are divided between different terminals and applications are in fact spreading codes and power. Each user can at any moment request a spreading code of variable length with a corresponding transmission power, depending on his negotiated transmission rate. The challenge is to do this in an efficient way while assuring the performance requirements of the applications.

3GPP defines four different QoS classes [27]

- conversational class
- streaming class
- interactive class
- background class

These classes can also be supported by EDGE.

The main distinction between these classes is the requirement on delay. The most delay sensitive traffic should use the conversational class while the least delay sensitive traffic should use the background class.

The support for these traffic classes in the radio access network is a matter of vendor implementations and will not be addressed in this deliverable.

A.2 Related Network Layer Technologies

Communication in the HIDENETS scenario is supposed to rely on IP and its related protocols as standardized within the Internet Engineering Task Force (IETF [94]). On the IP layer, packets will then be forwarded based on the information present in either the IPv4 or the IPv6 header. Both headers include information to support service differentiation. Normal IP operation is connectionless and is based on hop-by-hop forwarding which means that each node take independent decisions on where to forward a packet. This limits the routing and forwarding flexibility to shortest path routing to avoid forwarding loops.

As IP operates in a connectionless manner, the most used routing schemes in fixed networks are based on link state routing, where each node gets a full overview of the entire topology. After a change in the topology a full re-convergence is initiated. This process is quite time-consuming and highly dependable services may suffer.

The IP routing used in fixed networks is not always useful in wireless ad-hoc networks, due to for instance scalability issues and limited routing flexibility. For this reason adjusted and new routing approaches have been developed for wireless ad-hoc networks. This will be presented in the following.

A.1.6 Ad-hoc routing

A mobile ad-hoc network is characterized by nodes that operate as both hosts and routers. Since nodes are not necessarily next neighbours, a routing protocol is deployed to ensure connectivity in the ad-hoc network. In general, routing protocols are either based on the distance vector or the link state approach. With the link state approach, each node normally broadcasts a list of its neighbours to all other routers in the network, and each single router can construct a map of the global network topology. With the distance vector approach, on the contrary, the routers only exchange routing information between their next hop neighbours. From some or all of its neighbours, a router receives a list of the neighbour's computed distance some or all other nodes in the network. Based on the lists from its neighbours, a router is able to construct its own list of the shortest distances to each node.

A large number of routing protocols have been proposed for ad-hoc networks. One may distinguish between reactive (on-demand) and proactive (table-driven) routing protocols. A reactive protocol tries to find a route on demand, i.e. a router tries to find a route to another node only when it needs to send a packet to that node and has no fresh route to the node already. A proactive protocol, on the contrary, will continuously keep the routing tables of all nodes in the network up to date, regardless of whether there is traffic in the network or not.

Reactive routing protocols are preferred when nodes are highly mobile; when only a subset of nodes are communicating at any one time; and when communication sessions last for relatively long times. Pro-active routing protocols, on the contrary, are preferred for lower levels of mobility; and when communication is random and sporadic. With reactive routing there is no routing overhead for nodes that don't communicate.

One of the biggest problems with routing in ad-hoc networks is that it requires broadcast of routing messages. Broadcast is a very costly operation in ad-hoc networking, since there is a high redundancy (i.e. many broadcasts will cover the same geographical area), contention occurs between nodes that are

forwarding the same broadcast message at about the same time, and there are many collisions between the forwarded broadcast message. This is referred to as the broadcast storm problem. The reactive solution to this problem is to issue a broadcast only when it is absolutely necessary, i.e. only when a router needs a route for a packet it is about to send. The proactive approach, on the contrary, accepts that periodic broadcasts are necessary to maintain a routing table at each node at any time. Instead, the proactive solution is to construct network structures (e.g. connected domination sets, minimum hop spanning trees or multi-point relays) that reduce the damaging effects of the broadcast storm problem.

The most popular reactive routing protocol is AODV (Ad-hoc On-demand Distance Vector [24]), while the most popular proactive protocol is OLSR (Optimized Link State Routing Protocol [21]). In addition, W-OSPF [22], [23] is being developed. This is a wireless interface to the popular wired proactive protocol OSPF.

OLSR, W-OSPF and AODV, are the routing protocols of highest interest in the HIDDENETS project. In other words, both proactive and reactive protocols will be considered, since both have their pros and cons. One advantage of the proactive approach is that most routing protocols in wired networks are proactive. Their operation is well understood, and it might be easier to import concepts and mechanisms from the wired domain to the ad-hoc domain if the routing protocols are similar. One example is the resilience mechanisms developed for wired networks: When using a proactive routing protocol, such as OLSR, in the ad-hoc domain, it might be possible to reuse some of the resilience mechanisms developed for wired networks. This represents a particularly interesting research challenge that should be addressed by the HIDDENETS project. The objective should be to improve the resilience of OLSR (or W-OSPF) to such an extent that the resilience is comparable to that of AODV in mobile environments. It might even be possible to outperform AODV in terms of resilience. Needless to say, the HIDDENETS project should also find ways to improve the resilience of AODV, although reusing mechanisms already developed for wired networks might be difficult.

We note that a number of hierarchical and geographical approaches have been proposed, in addition to flat routing protocols, such as AODV and OLSR. The intention of the hierarchical and geographic approaches is to give higher scalability of very large ad-hoc networks. Examples of hierarchical routing protocols include fisheye state routing (FISHEYE), landmark routing (LANMAR), the zone routing protocol (ZRP). Examples of geographical routing protocols include location-aided routing (LAR), and so forth [99-102].

A short description of OLSR and AODV is given in the following.

The AODV routing protocol is probably the most widely studied and popular proposals today [24]. It allows source nodes to discover routes to a destination IP-address on demand: When a source router desires a route to a destination IP-address for which it does not already have a route, it issues a route request (RREQ) packet. The packet is broadcasted by controlled flooding throughout the network, and sets up a return route to the source. If a router receiving the RREQ is either the destination or has a valid route to the destination IP-address, it unicasts a Route Reply (RREP) back to the source along the reverse route. The RREP sets up a forward route. Thus, the pair of RREQ and RREP messages set up a bi-directional unicast route between source and destination. Once the source router receives the RREP, it may begin to forward data packets to the destination.

On-demand ad-hoc networks can scale to a larger number of nodes, if the range of the flooding is restricted to a limited number of hops. By setting appropriate values in the TTL field of the IP header of an RREQ, the hop-limit can be controlled. AODV allows for expanded ring search in which the originator of RREQs may search for a route to the destination more than one time, each time increasing the hop limit, until the route is found or until the originator gives up.

Different reactive routing protocols have different strategies to deal with route maintenance and route repair. Most protocols, including AODV, let routes that are inactive eventually time out. If a link break occurs while the route is active, AODV implements an algorithm to repair the route. The router upstream to the link breakage will send an error message upstream towards the source.

Different protocols also have different ways to manage routing state information. AODV, for example, stores state information in the network. Routers that receive RREQs set up the return routes in the route tables as backwards pointers to the source router. Similarly, RREPs that are propagated back to the source along the reverse route leave forward pointers to the destination in the route tables. (It should be noted that this approach is not always used for reactive routing protocols. The Dynamic Source Routing (DSR) protocol, for

example, does not rely on routing state in the network for the forwarding of packets. Instead, the sender of a packet encodes the route explicitly into the packet i.e. it source-routes the packet.

OLSR, on the contrary, is a proactive protocol that link-state routing protocol that periodically advertises the links in the network, and introduces optimizations of the classical link state algorithm tailored to the requirements of a mobile wireless LAN [21].

To optimize the broadcasts of routing messages, OLSR uses multipoint relays (MPRs), which are selected nodes that forward broadcast messages during the flooding process. Thus, the message overhead of the flooding process is reduced substantially compared to a classical link state algorithm. The protocol is particularly suitable for large and dense networks as the technique of MPRs works well in this context.

The MPR selection is at the core of OLSR, and it works as follows: Each OLSR node in the network becomes aware its one-hop and two-hop neighbours by periodically exchanging HELLO messages with its one-hop neighbours. It then operates as an "MPR Selector" and selects its MPR nodes among its neighbouring nodes as the minimum set of one-hop neighbours that allow reaching every two-hop neighbour throughout the nodes in the MPR set.

The link state messages are called "TC messages" according to the terminology of OLSR. OLSR minimizes the number of control messages flooded in the network by letting only MPR generate TC messages (and only MPRs to forwarding them). TC messages are used to advertise the links between MPRs and MPR Selectors. The shortest path algorithm uses those links to construct paths for every MPR Selector throughout the network. (The basic OLSR protocol provides shortest path routes in terms of number of hops.)

Another optimization provided by OLSR is that of partial link state information: An MPR node may choose to report only links between itself and its MPR selectors, allowing partial link state information to be distributed in the network. This is the minimal set of links that any MPR must advertise in order to support path computation. However, additional links may be advertised as well. This is controlled by the TC_REDUNDANCY parameter.

A.3 Related higher layer technologies

A.1.7 Transport layer protocols

Two protocols are mainly used in the IP upper-layer: TCP, which provides a reliable and controlled transport service, and UDP, which provides a lightweight but unreliable transport service. The majority of Internet applications use TCP. But UDP is more appropriate for applications that do not require every message to get delivered. Many programs will use at the same time a separate TCP connection as well as UDP sending: The most important information is sent along the reliable TCP connection, while the main data stream is sent via UDP.

A.1.7.1 UDP

UDP is defined in RFC 768 and clarified in RFC 1122. This protocol does not provide reliable data transmission. So, there is no guarantee that the UDP segment will even arrive at its destination, there is no acknowledgment mechanism. UDP messages are sent and then forgotten immediately. Thus, it is meant to provide a low-overhead transport. This is why UDP is very effective to transmit a lot of information quickly. Real-time audio or video applications are examples of use of UDP thanks to low-overhead structure and connectionless feature (no handshake, no acknowledgments).

UDP is a connectionless protocol. This means that it does not use handshake to establish an end-to-end connection before transmitting data. In contrast, connection-oriented protocols exchange control information with the remote system to verify that it is ready to receive data before sending it.

If an application needs to use broadcasts or multicasts in order to send data to multiple hosts simultaneously, then this application will have to use UDP to do so. That application does gain some benefits from doing so: It would take far too long for the sender to establish individual TCP connections with every other system on the network. UDP's connectionless service allows the sender to simply send the data to all of the devices simultaneously.

A.1.7.2 TCP

TCP is defined in RFC 793 and clarified in RFC 1122 and RFC 2581. TCP is probably the most important transport protocol in use on the Internet today. It provides a reliable data transmission and maintains a virtual connection between devices or services, because it is a connection-oriented protocol. It means that a TCP endpoint has to establish a connection before sending data.

The procedures to establish connections use the synchronize (SYN) control flag and involves an exchange of three messages which is called a three-way handshake. Once a virtual circuit has been established, the applications in use can begin exchanging data with each other. However, it is important to note that applications do not exchange data directly. TCP stores the data that it receives from upper-layer into a local send buffer. Periodically, a chunk of data will get sent to the destination system. The recipient software will then store this data into a receive buffer, where it will be eventually passed to the destination application when the message is completed.

TCP is responsible for data recovery in the event that packets are received out of sequence, lost, or otherwise corrupted during delivery. It accomplishes this recovery by providing a sequence number to each sent packet. The lower network layer treats every packet like a separate unit. Therefore, it is possible for packets to be sent along completely different routes, even though they are all part of the same message. To ensure that data has been received correctly, TCP requires an acknowledgement (ACK) from the destination machine. If the appropriate ACK determined by its sequence number is not received within a certain time limit, the packet is retransmitted. When the network is congested, this retransmission leads to duplicate packets being sent. However, the receiving machine uses the sequence number of the packet to determine if it is a duplicate and discards it if necessary.

A.1.7.3 SCTP

SCTP is an emerging candidate at the transport layer. Stream Control Transmission Protocol (SCTP) is defined in RFC 2960. SCTP is a reliable transport protocol and its advantages over TCP and UDP are:

- The transfer of data with acknowledgement and without error neither duplication.
- The data fragmentation to conform to the Maximum Transfer Unit (MTU) size of the chosen path.
- The sequence delivery of messages within multiple streams, with an option for the order of arrival.
- The optional bundling of multiple user messages into a single SCTP packet.
- The support of multi-homing to deal with network failures.

A.1.8 Access and session control

Mobile operators want to control the access to their scarce radio resources and are pushing towards the deployment of access and session control in their systems in order to block unauthorized foreign users, prevent Theft of Services (ToS), and map users' subscription profiles to authorized QoS levels in the access network. The first standardized solution for access and session control is the IP Multimedia Subsystem (IMS) [28][29], introduced by 3GPP in Release 5 of the UMTS specifications. The IMS uses SIP [30] to provide establishment and management of IP-based multimedia sessions.

1: A frame usually consists of a specified number of bits between flag sequences and usually includes an address field, a control field http://www.its.bldrdoc.gov/projects/t1glossary2000/control_field.html

A.1.8.1 SIP

The Session Initiation Protocol (SIP) has been developed by IETF in RFC 3261: “SIP is an application-layer control protocol that can establish, modify and terminate multimedia sessions or calls”.

Initiating a session requires to determine where the user to be contacted is located at a particular moment. When the user is located, SIP delivers a description of the session that the user is invited to. The most common protocol used to describe sessions is the Session Description Protocol (SDP). SIP is used to convey the response to the session initiation (accept, reject...) and to modify the session as well. Finally, SIP can be used to terminate the sessions (i.e. hang up). SIP can also manage multi-party conferences.

SIP integrates features from two protocols: the client-server design and use of Uniform Resource Locators (URLs) from HTTP (Hyper-Text Transfer Protocol) which is used for web browsing, and the text-encoding scheme and header style (To, From, Date, Subject) from SMTP (Simple Mail Transfer Protocol) which is used for email. SIP supports the requirements for establishing and terminating sessions thanks to five features:

- User location: localization of the end system to be used in the communication.
- User capabilities: determination of the media parameters involved in the communication.
- User availability: determination of the willingness of the called party to accept the call.
- Call set up: establishment of communication parameters at the called and calling party, the result of a successful call set up is the “ringing”.
- Call handling: managing call transfer and termination.

A.1.8.2 IMS

In the IMS, the SIP signalling is processed by entities called Call State Control Functionality (CSCF) servers. Fig.1 shows the IMS architecture, which consists of three different types of CSCF servers plus an additional supporting database [29][28].

- HSS (Home Subscriber Server) is the integrated database that consists of a Location Server, which stores information on the location of users, and a profile database, which stores security and service profile information for subscribed users.
- P-CSCF (Proxy CSCF) is the server initially contacted by the SIP devices. All SIP requests are sent from the client to a P-CSCF first. The P-CSCF is usually associated to a Policy Decision Function (PDF) that interacts with the access router to apply operator’s access control policies to each bearer in the access network.
- I-CSCF (Interrogation CSCF) acts as first contact point for other IMS networks; it also selects an appropriate S-CSCF, with the help of the HSS, during the SIP registration.
- S-CSCF (Serving CSCF) is mainly responsible for managing each user’s profile and the call states. It performs service control and provides interfaces to application servers.

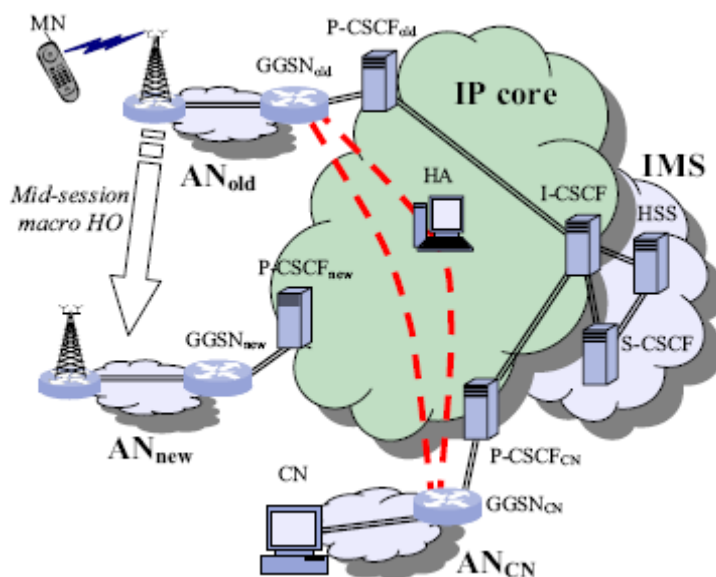


Figure 6: IMS platform architecture

Note that even though the IMS was designed with standardized interfaces that make it a network-independent platform, the current version of its specifications still does not support mid-session macro handover. In other words, whenever a user changes its global IP address, i.e. whenever a user moves to another access network, the ongoing session(s) has to be terminated and the long standard SIP-based IMS session setup procedures have to be performed once more at the new access network.

B Overview of resilience schemes

Error resilience is a key issue in HIDENETS due to the dependability-critical applications involved in car-to-car communication scenarios. Important example applications are transportation assistance and ambulance-to-hospital multimedia conversation. Here we provide a general overview of potential techniques which could be used for resilience enhancement at the different layers of the OSI protocol stack. Details about HIDENETS relevant schemes in ad-hoc domain and infrastructure domain can be found in later chapters.

B.1 Resilience Schemes at L2

Error control scheme at layer 2 aim at detecting and correcting erroneous bits in a received frame, caused by the error prone wireless channel so as to minimize the impact of channel errors on the upper layer functionalities. Error control schemes at L2 include the following error detection schemes:

- Parity Bit Check
- Checksum Check
- Cyclic Redundancy Check

And the following error correction schemes:

- Forward Error Correction (FEC)
- Automatic Repeat reQuest

Parity bit check is perhaps the simplest form of error detection technique. With this technique, the sender adds single or multiple bits into the data bit sequence. The simplest forms of parity bit check are even/odd parity scheme, in which case the receiver checks the number of 1s in the received bit sequence. However, these techniques can only detect if some odd number of bit errors have occurred. Two-dimensional parity scheme is a more advanced error detection scheme providing the following features. First, the receiver can not only detect but also locate and correct a single error bit. Second, it can also detect any combination of two errors in a packet.

With Checksum techniques, a sender sums the k-bit integers derived from the d-bit data, and use the resulting sum as the error detection bits. The receiver calculates the checksum over the received data and checks whether it matches the checksum carried in the received packet.

Cyclic Redundancy Check (CRC) is a widely used error detection technique in today's computer networking. Consider a d-bit data chunk, D, the sender appends r additional bits, R, to D such that D+R is exactly divisible by a pre-agreed pattern, G, using modulo 2 arithmetic. Upon the receipt of D+R, the receiver divides D+R by G. If the remainder is nonzero, the receiver knows that an error has occurred. CRC can detect burst errors of length less than r+1 bits and any odd number of bit errors. Moreover, a burst error of length greater than r+1 bits can be detected with probability $1-0.5^r$.

Forward Error Correction (FEC) employs error correcting codes at the sender side so that the receiver can correct error bits (due to channel imperfections). One way of doing FEC is to add redundant parity bit at the sender side, and these parity bits are then used by the receiver to detect and correct errors. FEC's performance in throughput and goodput (see **Fejl! Henvisningskilde ikke fundet.**) highly depends on the amount of redundant bits and the physical/precoded error rate. The more redundant bits appended to the original data, the more error bits in a single packet can be corrected, but at the same time the lower throughput is expected. **Hence the knowledge of bit error rate is very important for the proper configuration of FEC.** More specifically, if more redundant bits than required are appended, a lower throughput and goodput than the optimal will be observed. If less redundant bits than required are appended, a higher throughput but lower goodput than the optimal will be observed. With FEC, constant throughput and bounded delay can be assured, but the post decoding error rate can rapidly increase with increasing channel error rate. With a static FEC scheme, to cope with variable channel error rate, a powerful long code is required, which complicate the implementation of coder-decoder pair and impose a high communication overhead.

Automatic Repeat Request (ARQ) is used to request that a packet received with error(s) be retransmitted at Layer 2. This is only used when an error has been detected in the received packet by error detection schemes that can not be corrected by the error correction scheme (or no correction scheme at all) at the received side. Compared to FEC, ARQ is simple and achieves reasonable throughput when the channel error rate is not very high. However, ARQ quite often leads to longer delay due to the round-trip delay of NAK and retransmission of the message. Motivated by this observation, Hybrid ARQ, the combination of FEC and ARQ, has been developed.

There are two types of hybrid ARQ schemes, type-I and type-II. Type-I hybrid ARQ includes parity bits for both error detection and error correction in every transmitted packet, using either a single code for both error correction and detection or different codes for error correction and detection. If the number of erroneous bits in a received packet is within the error correction capability, the errors are corrected. If an uncorrectable error packet is detected, the receiver discards the received packets and requests a retransmission. The transmitter retransmits the same packet. When the retransmitted packet is received, error detection and correction is conducted using the retransmitted packet only. This process continues until the packet is successfully received or the maximum allowed retransmission attempts have been reached. Type-II hybrid ARQ is different from type-I hybrid ARQ in that all received packets (even if uncorrectable) are saved and used in the error detection and correction procedure on the next retransmitted packet.

The selection of error control schemes at Layer 2 depends on a number of factors, e.g.:

- channel condition,
- application requirements

Different implementations of IEEE 802.11 use different algorithms for rate adjustment. One algorithm is to reduce the transmission rate to that of a more robust modulation after two successive occasions of frame loss. If 802.11b is used, the node might for example reduce the nominal data rate from 11Mbps to 5,5Mbps, then to 2Mbps and finally to 1Mbps. If rate reduction does not succeed (or if it is not implemented), the node might try to re-associate with another access point (or if this is not possible it might notify the network layer of a "Link down").

Active and passive scanning are ongoing management activities of the MAC layer, although the scanning algorithms are not detailed by the 802.11 standard. The node might monitor the signal-to-noise ratio (SNR) and the signal quality and use this information when making decisions on which access point (or "base station") to associate with. If the signal quality from the access point with which a node is associated deteriorates or if the node does not receive some subsequent beacon frames, this might trigger the node to associate with another access point within the same extended service set (ESS). This kind of re-association might also occur if the rate adjustments fails (or is not implemented) as mentioned above.

If two access points use the same SSID, the node might expect that the two access points belong to the same ESS. The node might then change access point, and assume that connectivity after the handover is made possible by means of a distribution system (DS) that both access points are connected to. In this case, the node might change access point without changing default router or IP address, and the handover is transparent to the network layer.

B.2 Resilience Schemes at L3

As opposed to the error resilience techniques at Layer 2, the resilience techniques at Layer 3 mainly deal with relatively long-term error bursts due to link breakage caused by network topology changes. Depending on the type of routing, different resilience techniques should be considered. For unicasting, fast rerouting (route salvage) and multi-path routing are two example techniques. Fast rerouting addresses the problem of finding an alternative route for a broken route. Multiple routes in multi-path routing can be used at the same time to increase the reliability when some of the routes are highly prone to error. For broadcasting, the problem lies more on achieving the right balance between reliability and efficiency as the simplest form of layer 3 broadcasting, flooding, is reliable in itself but not cost-efficient. Moreover, differentiated resilience is important to satisfy different requirement on reliability of various applications.

The chosen resilience schemes will also depend on the routing approach and the forwarding approach offered by the network and its nodes. A network can either support connection-oriented or connectionless

forwarding. The former meaning that a path can be signalled from the source to the destination, and that the packets will follow this path. Connectionless means that no path is signalled and that each node determines where to send a packet on a hop-by-hop basis. Whether the network runs a proactive or reactive routing protocol will also influence on what resilience scheme that can be used. In a proactive routing scheme, all nodes know the topology in the network, while in a reactive scheme a node learns where to forward packets after having requested a route.

Other dimensions are the level of resilience required, the frequency of topology changes, and the density of mobile nodes. If very high reliability is required, one may for instance obtain this by increasing the frequency of routing-related messages for faster detection and correction of broken links (but then resulting in a reduction in the available bandwidth and more aggressive use of battery power).

B.2.1 Multi-path routing

Multi-path routing is a general term for techniques that maintain or utilize multiple paths between a source and destination pair. Multi-path routing can be used for the following purposes:

- To minimize the latency caused by re-routing after route breakage.
- To increase the end-to-end throughput by utilizing multiple disjoint paths at the same time
- To enhance resilience to security attacks based on eavesdropping.

Multi-path routing introduces several issues:

- Construction (including maintenance) of multiple paths,
- Selection of paths for a specific purpose (like for instance to maximize reliability, minimize hop count, or minimize delay)
- Adaptive usage of multiple paths.

B.2.2 Fast re-routing

Fast re-routing in general is a quick reaction to a broken route by switching the data traffic to another (potentially) active route. Depending on the type of underlying routing scheme, i.e., static or dynamic, techniques for re-routing are different.

For static routing, re-routing is normally done by pre-configuration. For example, if node A finds the next hop B is not reachable any longer, it can send the data traffic via another node C as specified in the configuration. The reason that a simple preconfiguration may work lies in the assumption that the chances of multiple failure happening at the same time is very low. Since static routing is normally used in networks with very low chance of network dynamics, the chances of multiple node/link failure happening at the same time is considered to be very small. In this case, a broken route can be considered as a single node/link failure, which can be detected by nodes that is along this path and neighbouring to the failure point.

For dynamic routing, a broken route can be either repaired by the node detecting the broken route or replaced by another known route (either at the point of detection or at the source of the data flow). The topic of replacing a broken route will be covered by multi-path routing, hence not described here. How a broken route can be repaired mainly depends on what information is available at each host, i.e. is it a link state routing or distance vector routing? It is much easier to repair the broken link with link state routing as normally with link state routing each host maintains a link list, which can be easily used to find a path to the next hop, second next hop, or even the destination in the original path.

B.2.3 Change of access router

Sometimes the node might not have alternative access point that belongs to the same ESS within range. If the networking layer is notified that the link is down, it might try to change to an access point not belonging to the same ESS. It will then be attached to a new access router (AR) and might also have to change IP address, and perhaps carry out mobility tasks, such as a mobile IP re-registration. These are routing tasks, although routing at terminals in infrastructure mode is very limited, and hardly includes more than replacing the default route, undertaking some ARP updates or involving Mobile IP. This functionality is therefore addressed by an orange box labelled "Rerouting or change of AR".

B.2.4 Efficient and reliable broadcast

Broadcasting is a functionality that delivers a message from a single source to all hosts in the network. However, it can also be used potentially to deliver important and reliability-desired message to a single destination as having more forwarding nodes eventually increase the resilience of delivery. Specifically broadcasting can be used for unicasting purposes in the following cases: When the delay required to discover a route to the destination(s) is too long for a message with specific requirement on delay, or when a reliable UDP transmission is required but the communication condition, i.e., network dynamics and channel condition is not good enough to assure successful unicast message delivery.

Concerning broadcasting in HIDENETS scenario, two major issues are foreseen; efficiency (in communication cost) and reliability. Efficiency is important in the ad-hoc domain of the HIDENETS scenarios due to the limited resources. Reliability is a key aspect of HIDENETS project, the error prone characteristics of wireless channels and network dynamics make the reliability requirement more challenging.

B.2.5 Multi-homing at layer 3

Multi-homed Mobile IP (*M-MIP*) [31] increases the performance and dependability of mobile hosts in a network where Mobile IP (MIP) is used for mobility management [32]. This technique is an example of an implementation of multi-homing that is transparent to IP routing protocols.

The multi-homing is implemented as an extension to MIP, by associating multiple care-of-addresses with a mobile host. Simultaneous connections can be managed and also a mobile host can associate with multiple access points. If you want to connect to multiple access points using a single interface card, it is necessary to work in ad-hoc mode (IBSS), hence M-MIP is sometimes seen as Layer2+Layer3 solution; with multiple interfaces it is also possible to use infrastructure BSS mode. The M-MIP protocol enhances the throughput, provides a more reliable connection and allows for fast re-associations.

Another mechanism developed in the Shim6 Working Group of the IETF introduces the concept of Hash Based Addresses (HBA) [82]. With HBA, a multi-homed host incorporates a hash over all IPv6 addresses it is assigned on various interfaces into the address used for the initiation of a communication session with another party. Since the hash is transferred upon initiation of the session, there is a mechanism where the multi-homed host can switch to another of the assigned IP addresses in a secure way. In addition, Hash Based Addresses introduces a shim layer between the number of underlying addresses used, and the initial address (a.k.a. the Upper Layer ID - ULID) which is seen by the upper layers. A shim layer is a 'layer' in between, i.e. in this case a 'layer' between L4 and L3 to conceal the layering violation built into the use of the IP addresses in the transport identifier (which cause us so much trouble both in terms of mobility and multihoming).

A similar shim layer is introduced for the Host Identity Protocol (HIP) developed in the HIP WG of the IETF [39]. The Host Identity Protocol (HIP) is based on the observation that the transport layer protocols (such as TCP and UDP) use the IP-addresses as part of their transport-layer identifiers. This cross-layer violation is the origin of many problems associated with mobility and multi-homing. Instead of this cross-layer violation, HIP proposes to separate the end-point identifier and locator roles of IP addresses. It introduces a new Host Identity (HI) name space, based on public keys, where the public keys might be self generated. Thus, the public key generates the Host Identity, which is used to identify an end point.

Finally, the Mobike WG was set out to develop a mechanism where a host can change IP addresses in a secure way [40]. Their starting point was from IPsec and IKE. The main idea is to add features to IKEv2 to

update the IKEv2 SA and IPsec SA endpoint addresses without need of the rekeying the SA. In this way support roaming, mobility, and multi-homing will be enabled. IKEv2 needs an extension to support multiple IP addresses tied to one IKEv2 SA and IPsec SA. Furthermore, some way to authenticate multiple IP addresses and the change between IP addresses is also needed.

B.2.6 Protection and restoration

The purpose of a recovery scheme is to recover an amount of traffic on a path from the misbehaviour caused by a failure in the network in question. The ordinary path is termed the working path, and traffic is being switched over to the recovery path in case of failure. A path provided with recovery mechanisms is termed a recoverable path. If the recovery path is capable of replacing the working path without service degradation, it is called an equivalent recovery path. A limited recovery path, on the other hand, may exhibit certain constraints when compared with an equivalent recovery path, such as limited bandwidth or lack of guarantee for packet ordering. A network (or a path) provided with recovery mechanisms is said to be resilient.

The cardinality between recovery paths and working paths, the path mapping, may vary from "one for one" (1:1), via "one for many" (1: n) to "many for many" (m : n). The recovery paths may be used to transport low-priority traffic, given that this traffic can be deleted without further notice when a failure occurs.

There is the "one plus one" (1+1) mapping as well, where traffic is carried on both the working and the recovery path. Switching takes place at the receive end only (single-ended scheme) when a failure occurs, rendering these schemes with the potentially shortest recovery time. Single-ended schemes may not carry low-priority traffic when there are no failures on the path.

With protection we mean that the recovery path or path segments are created prior to the detection of a failure on the working path. With restoration (or re-routing) we mean that the recovery path or path segments are created dynamically after the detection of a failure on the working path. Such mechanisms can be introduced at different layers, e.g. optical layer protection, MPLS protection and restoration, IP restoration. In the context of HIDDENETS we will mainly talk about layer 3 protection and restoration.

In case more layers are involved a requirement would be that recovery mechanisms at one layer should be transparent to a recovery mechanism at a higher layer, so that flapping is not introduced. MPLS introduces priorities and pre-emption so that one LSP can pre-empt a lower priority LSP in case of failure. This opens up for using the recovery path for lower priority traffic in normal situations.

B.2.7 Differentiated resilience

Network resilience can be defined as the ability of a system/network to adapt to changes like for instance node and link failures and traffic pattern changes. In contrast to redundancy, where backup systems are installed, resilience mechanisms aim to recover from failures or even to resist being affected by them. Proper resilience mechanisms will enhance network performance and availability and thus the fault tolerance of the system.

Resilience is included in the QoS terms as defined by [1]. It may be viewed as orthogonal to other QoS requirements like delay, packet loss, and throughput. Traffic streams may have strict requirements to loss or delay, but loose requirements to resilience. Thus a voice call, which is often prioritized during normal operation because of its strict delay requirements, may be closed down after a failover to make room for a web session with higher requirements to resilience.

The basic idea behind a differentiated resilience approach [3] is that not all traffic requires full recovery guarantees in cases of failure and the requirements for recovery response times will also differ. From this fact different resilience classes can be defined based on the specific requirements of services and user groups. This has also obvious implications for dimensioning, since we do not need to dimension capacity for full recovery of all traffic in case of failure. Another important aspect is that unused recovery bandwidth should be available for low priority elastic traffic. This will increase network utilization and make the network more cost efficient.

Differentiation with respect to resilience basically depends on two factors

- i. Requirement for recovery time.
- ii. Bandwidth guarantee with respect to recovery.

We distinguish between full recovery (i.e. equivalent bandwidth) and partial recovery (i.e. lower bandwidth) or no bandwidth guarantee (best-effort recovery).

In [3] four resilience classes are introduced with recovery time requirement a) below 100 ms, b) between 100 ms and 1s, c) between 1s and 10s, d) no requirement. This requirement will be fulfilled by using appropriate recovery mechanisms; i.e. protection (dedicated/shared, local/global) or restoration (local/global, re-routing), while in [5] three levels regarding recovery time were used as a basis for defining resilience classes (combining c) and d)).

A recovery time less than 100 ms will imply using 1+1 protection or some sort of dedicated (1:1) or shared local protection. A recovery time between 100 ms and 1 s will imply using dedicated (1:1) or shared global protection. Finally, recovery times above 1 s imply using restoration or IP re-routing. Nevertheless, note that research is ongoing to enable IP re-routing in sub second intervals.

Resilience classes and the relation to QoS requirements will be further described in section D.4.

B.3 Resilience Schemes at L4

Resilience at Layer 4 deals with the insurance that endpoints stay connected by looking at the end-to-end characteristics of the communications. While common timeout techniques implemented at Layer 4 usually deal with issues such as network congestion, SCTP brings a new perspective by addressing network failures thanks to its multi-homing feature.

B.3.1 Multi-homing at layer 4

The Stream Control Transmission Protocol (SCTP) [33] is a transport protocol, originally designed for the transport of telephony (SS7) protocols over IP, aiming at duplicating the reliability aspects of these protocols. SCTP combines the strengths of TCP (i.e. congestion control, error detection and retransmission) with additional features like multi-homing and multi-streaming.

One of the features of SCTP is multi-homing support. This is accomplished by allowing multiple IP addresses to be used for both endpoints in the association between these two endpoints. So there is a single SCTP association, while both nodes have (at least) two addresses, of which one is the primary address. The secondary address can be used in case of failure, for explicit requests from upper layers or to retransmit lost packets. This allows utilization of redundancy in the network and provides resilience against link failures, which makes SCTP useful in situations that require high availability.

Detection of loss and duplication of data chunks is enabled by numbering all data chunks in the sender with the so-called Transport Sequence Number (TSN). The acknowledgements sent from the receiver to the sender are based on these sequence numbers: each received SCTP packet is acknowledged by sending a Selective Acknowledgement (SACK) which reports all gaps. The SACK is contained in a specific control chunk. Whenever the sender receives four consecutive SACKs reporting the same data chunk missing, this data chunk is immediately retransmitted (fast retransmit). Most up-to-date operating systems already support a similar optional extension to TCP. Retransmissions are timer-controlled. The timer duration is derived from continuous measurements of the round trip delay. Whenever such a retransmission timer expires, (and congestion control allows transmissions) all non-acknowledged data chunks are retransmitted and the timer is started again doubling its initial duration (like in TCP).

Another interesting feature of SCTP is the support of heartbeat messages to monitor the reachability of far-end transport addresses. An SCTP instance monitors all transmission paths to the other endpoint of the SCTP association. To this end, HEARTBEAT chunks are sent over all paths which are currently not used for the transmission of data chunks. Each HEARTBEAT chunk has to be acknowledged by a HEARTBEAT-ACK chunk.

The number of events where heartbeats were not acknowledged within a certain time, or retransmission events occurred is counted on a per association basis, and if a certain limit is exceeded (the value of which may be configurable), the peer endpoint is considered unreachable, and the association will be terminated.

B.3.2 Retransmissions at Layer 4 – Connection-oriented transport protocols

Retransmission mechanisms for the reliable transport layer protocols follow the same pattern: this is handled by setting a timer when sending data and if the data is not acknowledged when the timer expires, a retransmission occurs. The default value for the first retransmission timer (Tr.) is usually 1-1.5 second, and the timer value exponentially increases after each retransmission. The retransmissions stop (and the data is considered definitely lost) when the timer reaches 26.Tr. (i.e. 6 transmissions).

We give the specifics for SCTP in particular:

Retransmission of DATA chunks occurs from either (a) timeout of the retransmission timer; or (b) receipt of SACKs indicating the DATA chunk has not been received. To reduce the potential for congestion, the rate of retransmission of DATA chunks is limited. The retransmission timeout (RTO) is adjusted based on estimates of the round trip delay and backs off exponentially as message loss increases.

In an active association with fairly constant DATA transmission, SACKs are more likely to cause retransmission than the timeout. To reduce the chance of an unnecessary retransmission, a 4 SACK rule is used, so that retransmission only occurs on receipt of the 4th SACK that indicates that the chunk is missing. This is intended to avoid retransmits due to normal occurrences such as packets received out of sequence.

B.3.3 Reliable multicast transport

For IP multicast, different reliability mechanisms must be used from those for unicast when large groups of receivers should be supported. This is because the standard acknowledgement-based retransmission mechanism (like that implemented by TCP) would result in a lot of acknowledgements at the sender side.

There are two main groups of solutions: Forward Error Correction (FEC) and Negative-Acknowledgement (NACK). The FEC based solutions send redundant information to allow for repair at the receiver side when packets are lost. The NACK based solutions resend packets on request from receivers, but as only information regarding missing packets are communicated towards the sender, the amount of receiver feedback is reduced as compared to traditional acknowledgement-based retransmission schemes.

FEC and NACK based methods may be used in isolation, or combined. Note that for some FEC mechanisms it is possible for one repair packet to repair different lost packets within a given set (block) of packets. An example of a combined solution would be to use this kind of error correction together with a NACK based solution. Then repair packets are only sent when requested, and one packet may be used by several receivers having lost different packets within a block.

Both NACK and FEC based methods may be implemented end-to-end or with router support. In the latter case routers may perform NACK suppression by aggregating NACK messages from downstream routers/receivers to reduce the number of NACK messages towards the sender. Routers may also help reduce the amount of FEC packets by only sending them on links with receivers that have requested them.

The standardization of FEC and NACK based reliable multicast is carried out in the Reliable Multicast Transport (RMT) group of the IETF.

B.4 Resilience Schemes at L5

SIP manages IP communications with the notion of session between endpoints. The session is meant to provide means to negotiate and control the communication aspects such as the application used (i.e. describe the type of media), the codecs required, etc. In itself, SIP does not offer much resilience capabilities but rely on other layers to minimize the impact of its failures. It is crucial to minimize SIP failures because they impact the user experience too. When initiating a phone call (e.g. VoIP) or changing some parameter of an ongoing session, SIP mechanisms are invoked: if the SIP layer is failed, the phone call cannot be initiated or the parameters of the ongoing session cannot be changed, even though the session application (running on top of SIP) is operational.

RSerPool is one example of resilient architecture, requiring the deployment of peer servers for redundancy-based resilience. This type of solution (and others, such as the cluster-based architecture) is mainly aimed at increasing the dependability of a system –it also aims at increasing the system’s capacity (which in turns also favours the system’s availability).

B.4.1 Retransmissions at Layer 5 - SIP

SIP uses, on a per-request basis, the traditional timer technique to detect end-to-end failures and to consequently trigger retransmissions. If SIP messages are carried over UDP, the client retransmits the requests after a predefined time interval, and doubles after each retransmission (until the SIP request is finally acknowledged, or dropped after the maximum number of retransmissions) for congestion-control purpose. The SIP timer is an estimate of the round-trip time and its default value is 500ms but it is recommended to be larger in case of high latency access links. The retransmissions cease upon reception of a 183 provisional response or a 200OK final response, or after a maximum of 7 transmissions of the request. When sending the next request, the SIP resets the timer to the predefined time interval.

For reliable transport protocols such as TCP, the layer responsible for the retransmissions depends on their respective timer value. E.g., it is recommended to set the timer to a large value at the SIP layer if transport layer retransmissions are desired over application layer retransmissions.

B.4.2 Distributed Redundancy at Layer 5 - the RSerPool Framework

The RSerPool concept [34] is simple and relies on redundancy to be deployed anywhere in an IP network, even in different sub-networks. Hosts that implement the same service are called pool elements (PE) and form a so-called pool, which is identified by a unique pool handle (i.e. a pool identifier). The users of a server pool are referred to as pool users (PU). Another entity, called name server (NS) or ENRP server, is in charge of monitoring the pool, keeping track of the PEs’ status, and to help the PUs know which PEs the requests can be sent to. Figure 7 depicts the RSerPool architecture.

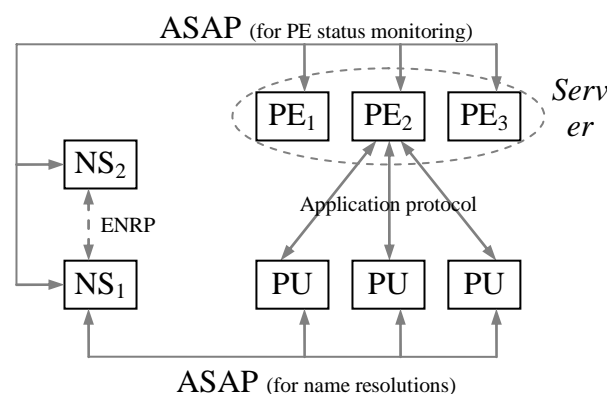


Figure 7: RSerPool architecture for one pool server. NS1 is the default name server for the server pool; NS2 can be used as a backup NS for this pool and as the default NS for another pool

The functionality of RSerPool is based on two novel protocols: Endpoint Name Resolution Protocol (ENRP) [35] and Aggregate Server Access Protocol (ASAP) [36]. ASAP provides the name resolution functionality, i.e. the translation of a pool handle, sent by a PU, into a set of PEs’ transport addresses (IP addresses and port numbers), and adds a suggestion for a server selection policy. Then, the PU can keep in a cache the information obtained from the NS and use it for sending future requests. The second RSerPool protocol is ENRP. Name servers use this protocol mainly to disseminate the status of their PEs among their peers to make sure that the information is consistent and up-to-date in every pool since a PE can belong to more than one pool.

State-sharing

RFC3237 requires that the name servers should not resolve a pool handle to a transport layer address of a PE that is not in operation. Thus, name servers share information about the current status of all the pools they monitor. This allows other name servers to act as backups when PUs' home name server fails and always keep the name service available.

Note that the requirements for high availability and scalability defined in RSerPool do not imply requirements on shared state. ASAP may provide hooks to assist an application in building a mechanism to share state (e.g. a so-called cookie mechanism), but ASAP in itself will not share any state between pool elements.

Failure-detection

All ASAP messages exchanged between an NS and a PE must use SCTP as transport protocol, and all ASAP messages exchanged between an NS and a PU must use either SCTP or TCP as transport protocol. Therefore, as the transport protocol between RSerPool entities is reliable (SCTP or TCP), transport layer supervision is provided, ensuring, at least, node and link failure detection.

ASAP has monitoring capability to test the reachability of PEs. When detecting a failure at the ASAP layer, the ASAP endpoint should report the unavailability of the specified PE by sending an `ENDPOINT_UNREACHABLE` message to its home NS. When the unavailability of a PE is detected at another layer, it should be reported to the ASAP layer via the Transport Failure Primitive.

Each PE is supervised by one specific name server, called the home NS. Home name servers specifically "audit" their PEs by periodically sending unicast `ENDPOINT_KEEP_ALIVE` messages at the ASAP layer. The NS sends this message to the PE as a "health" check. E.g., in the case when the transport level heartbeat mechanism is insufficient, the ASAP layer mechanism increases the probing frequency. The goal is to determine PEs' health status in a more timely fashion. The `ENDPOINT_KEEP_ALIVE_ACK` message is sent by the PE to the NS as an acknowledgment to the `ENDPOINT_KEEP_ALIVE` message.

Using ASAP keep-alive messages adds to the accuracy of SCTP failure-detection. While SCTP-level heartbeats monitor the end-to-end connectivity between the two SCTP stacks, ASAP keep-alive messages monitor the end-to-end liveness of the ASAP layer, i.e. one layer above SCTP. This level of failure-detection implies that failures at the application layer at the PE cannot be detected, unless the application also implements a failure-detection mechanism on its own.

Failover

SCTP makes use of its multi-homing capability to provide network failover. When it detects a failure on the primary path, it switches the future communications over the next available active path between the two endpoints (changing the network interfaces, which are connected to different networks when possible). If it is found unreachable, before notifying the sender of the failure, ASAP can automatically select another PE in that pool and attempt to deliver the message to that PE. In other words, ASAP is capable of transparent failover amongst application instances in a server pool.

When a PE fails, the failover is triggered only when the application layer discovers that an error has stopped the communication. Then, the PU initiates a failover by requesting another name translation at the NS, to get an up-to-date suggestion for an active PE, or by using the server status information in its cache and immediately try another server in the list. Using the cache, the fail-over can be done as soon as the failure is detected, but with some probability that an unavailable server is selected; a repeated name resolution on the other hand slows down the failover, but increases the chance to request the service to an active server.

B.5 Cross-layer considerations

B.5.1 Summary of resilience schemes

In the previous sections, we have presented an overview of resilient schemes that may contribute to the end-to-end dependability in HIDENETS scenarios. WP2 and WP3 have together started to draw a detailed picture of the software architecture of an (ad-hoc) node (chapter **Fejl! Henvisningskilde ikke fundet.**), introducing blocks at different layers. Also, the integration of resilient solutions at different layers should not make the software less stable or robust because of interaction effects (e.g. common access/write of a variable by blocks at different layers). Therefore, a thorough analysis of the cross-layer design should be added to the block interaction discussion. This analysis shall focus on cross-layer design techniques and on potentially conflicting interactions between blocks.

Table 1 summarizes the different resilience schemes considered in HIDENETS applicable at the communication level, i.e. resilience schemes at the node level can be found in WP2 work and is not included in this table.

L6-L7	c.f. WP2		
L5	SIP retransmissions		
L4	SCTP multi-homing	TCP-SCTP retransmissions	
L3	M-MIP, mobike, HIP, MULTI6 multi-homing	Differentiated resilience	<ul style="list-style-type: none"> • Fast re-routing • Multi-path routing • Efficient & reliable broadcast
L2	Error control	Multi-channel MAC	

Table 1: Summary of resilience schemes at communication level

B.5.2 Cross-layer design (CLD) background

Generally speaking, CLD refers to protocol design done by actively exploiting the dependence between protocol layers to obtain performance gains. This is unlike layering, where the protocols at the different layers are designed independently.

There are many CLD proposals in the literature. Here, we are specifically interested in how the layers are coupled, which can be done the following basic ways:

- Creation of new interfaces

Some CLDs require creation of new interfaces between the layers. The new interfaces are used for information sharing between the layers at runtime. We further divide this category into three subcategories depending on the direction of information flow along the new interfaces:

- Upward: From lower layer(s) to a higher layer
- Downward: From higher layer(s) to a lower layer
- Back and forth: Iterative flow between two layers

- Merging of adjacent layers

Another way to do CLD is to design two or more adjacent layers together such that the service provided by the new superlayer is the union of the services provided by the constituent layers. This does not require any new interfaces to be created in the stack. Architecturally speaking, the superlayer can be interfaced with the rest of the stack using the interfaces that already exist in the original architecture.

- Design coupling without new interfaces

Another category of CLD involves coupling two or more layers at design time without creating any extra interfaces for information sharing at runtime. E.g., one layer is taken as a reference layer and one or more other layers are designed accordingly. While no new interfaces are created, the architectural cost here is that it may not be possible to replace one layer (especially the reference layer) without making corresponding changes to another layer.

- Vertical calibration across layers

The final category is what is called vertical calibration across layers, i.e. adjusting parameters that span across layers. Basically, the performance seen at the level of the application is a function of the parameters at all the layers below it. Hence, it is conceivable that joint tuning of parameters can help achieve better performance than with individual settings.

Vertical calibration can be done in a static manner, which means setting parameters across the layers at design time with the optimization of some metric in mind. It can also be done dynamically at runtime, which emulates a flexible protocol stack that responds to variations in the channel, traffic, and overall network conditions. Static vertical calibration does not create significant consideration for implementations since the parameters can be adjusted once at design time and left untouched thereafter. Dynamic vertical calibration, on the other hand, requires mechanisms to retrieve and update the values of the parameters being optimized from the different layers. This may incur significant cost in terms of overheads, and also impose strict requirements on the parameter retrieval and update process to make sure that the knowledge of state of the stack is current and accurate.

While those CLD proposals are conceptual, the following introduces concrete ways to implement them in a node/system.

- Direct communication between layers

This is applicable when runtime information sharing between layers is necessary (e.g., in CLDs that rely on new interfaces or in dynamic vertical calibrations), which requires that the variables at one layer are visible to the other layers at runtime. There are many ways in which the layers can communicate with one another. For instance, protocol headers may be used to allow flow of information between layers. Alternatively, extra interlayer information could be treated as internal packets. [37] presents another similar proposal, so-called cross-layer signalling shortcuts (CLASS). CLASS allows any two layers to communicate directly with each other. These proposals are adapted to cases where only a few cross-layer information exchanges are to be implemented. However, when variables and internal states from different layers are to be shared that way, a number of implementation issues relating to managing shared memory spaces between layers usually need to be resolved.

- A shared database across the layers

In this case, a common database can be accessed by all layers. The common database is like a new layer, providing information storage/retrieval service to all layers. This approach is particularly well suited to vertical calibrations across layers. An optimization program can interface with the different layers at once through the shared database. Similarly, new interfaces between the layers can also be realized through the shared database. The main challenge here is the design of the interactions between the different layers and the shared database (e.g. consensus problem).

- Completely new abstractions

The third set of proposals present completely new abstractions. Consider, for example, the proposal in [38], which presents a new way to organize the protocols: in heaps, not in stacks as done by layering. Such novel organizations of protocols are appealing as they allow rich interactions between the building blocks of the protocols. Hence, potentially they offer great flexibility, both during design as well as at runtime. However, they change the very way protocols have been organized, and hence may require completely new system-level implementations.

C Applications and requirements

For some scenarios, like the differentiated resilience scenario (see section **Fejl! Henvisningskilde ikke fundet.**), it is necessary to sort applications according to requirements. The requirements are related both to in-transfer performance (packet loss, packet delay, delay variation, throughput etc), to resilience in terms of recovery times when failures appear and to the quality of the back-up path.

Differentiation of QoS (in-transfer performance) means that different classes of traffic receive different treatment with respect to queue management and scheduling. The idea is to reflect the fact that requirements with respect to e.g. delay and packet loss may be rather different, e.g. voice requires low delay but may tolerate some loss and can therefore use a priority queue with a small buffer, while some data application may tolerate much more delay but are loss sensitive and can therefore use a low priority queue with a long buffer. Each packet is marked (e.g. DiffServ marking) according to which service class it belongs and this marking is used to map the packets to the correct queue at the router interfaces.

3GPP defines four different QoS classes [17],[27] as described in section [reference to Annex]. Another approach is by ITU-T in recommendation Y.1541 [90] with essentially 6 QoS classes (class 0 – class 5) and opening up for some provisional classes. This is an ongoing discussion in standardisation fora and is a balance between the need to differentiate between applications with rather different requirements in terms of QoS on one hand, and avoiding too much complexity into the system on the other hand.

Our approach is not so much different. As a first approach we will try to map the applications into the four classes described below. Some minor points can be added. First of all we use the term ‘real-time’ instead of ‘conversational’ (ref 3GPP) to stress that not all applications in this class are interactive. Secondly we use the term ‘near real-time’ instead of ‘streaming’ not to restrict ourselves to streaming applications in this class.

With respect to in-transfer performance we therefore distinguish between the following four QoS-classes:

- i. Real-time
- ii. Near real-time
- iii. Interactive data
- iv. Background data

The real-time class is intended for conversational traffic like voice and video conference, but also for one-way applications with stringent requirement on delay and delay variation. Traffic in this class shall be prioritized in nodes to minimize queuing delay.

The near-real time class has more relaxed delay requirements and are typically meant for streaming type of applications; i.e. the requirement is more on delay variation and throughput than on delay.

The interactive data class is meant for elastic data traffic with a high requirement on packet loss and delay (round-trip-time) within certain bounds to preserve the interactivity. A typical example is web-browsing.

The last class, background data, is meant for data that is not expected within a certain time, but the loss requirement is still high. Typical examples could be email and back-up type of applications. Also typical peer-to-peer traffic, responsible for a non-negligible part of today’s Internet traffic, should preferably be mapped to this class.

We will not detail requirements of services in this deliverable. Some requirements are given in [4]. Typically the requirements will depend on the use, i.e. emergency use, business use or regular use. In Table 2 we have tried to map some applications to QoS-class depending on emergency, business or regular use.

QoS classes	Emergency use	Business use	Regular use
Real-time <ul style="list-style-type: none"> • Preserve time relation (variation) between packets • Stringent and low delay 	Emergency voice call, Video conference	Business voice call, video conference	Regular voice call, video conference, online gaming
Near real-time <ul style="list-style-type: none"> • Preserve time relation (variation) between packets (relaxed) • Throughput 	Emergency audio and video streaming, streaming data	Business video/TV/radio (audio and video streaming), streaming data (positioning)	Entertainment video/TV/radio (audio and video streaming)
Interactive data <ul style="list-style-type: none"> • Preserve payload content and responsiveness 	Interactive data like online notification to hospital	Web, office doc download	Web
Background <ul style="list-style-type: none"> • Preserve payload content • Data not expected within a certain time 	Non-interactive data communication and messaging like emergency vehicle warning	Non-interactive data communication and messaging like office, email, telemetry	Non-interactive data communication and messaging like email, ...

Table 2: Application matrix

In HIDENETS we have propose to distinguish between five resilience classes:

- i. Emergency class
- ii. High resilience class
- iii. Medium high resilience class
- iv. Medium low resilience class
- v. Low resilience class

These are the Quality of Resilience (QoR) classes and will be further described below. Not all of these were used in the studies in HIDENETS, but they are still described here for completeness.

In the following we only discuss full recovery versus no bandwidth guarantee. In the last case bandwidth will be allocated according to some (not strict) priorities, or indirectly via the use of scheduling mechanisms (DiffServ), so that one class may experience worse throughput degradation than others.

Another factor that is not discussed here is the potentially degraded QoS of the recovery path, i.e. the quality of the transmission after recovery. Other aspects to discuss in a differentiation scenario, but not discussed now, are availability, retainability and accessibility performance (ref [97]). (Retainability: The ability of a service, once obtained, to continue to be provided under given conditions for a requested duration. Accessibility: The probability that the user of a service after a request receives an acknowledgement within specified conditions.)

In this subsection the QoS classes defined above are investigated with respect to resilience requirements resulting in the preliminary proposal for QoR classes (Table 3). As indicated above, the resilience requirements taken into account here are recovery time and bandwidth (i.e. full recovery or no bandwidth guarantee).

Applications for emergency use will be mapped to emergency resilience class irrespective of in-transfer performance requirements. This class has the highest requirement on recovery time and will also require full recovery (equivalent bandwidth). It requires very fast set-up and high level of reliability. In order to avoid quality degradation it should not be statistically multiplexed with other classes (only this class cannot be preempted). The fast set-up requirement implies that 1+1 or some kind of local protection mechanism (1:1) will be used (see section **Fejl! Henvisningskilde ikke fundet.** of).

For the other applications we first discuss the need for the different QoS classes:

Class 1: Real Time

This class has a high requirement on recovery time and dedicated or shared protection (1:1, 1:n) with full recovery requirement should be chosen. If shared protection the requirement will only be guaranteed in single failure cases. This has implications for the availability figures but not the resilience class as defined below, so these applications belongs to a high resilience class.

In future we may see applications with more relaxed requirement to recovery time and bandwidth (adaptive applications), and such application may map to a medium high resilience class, although bandwidth can be renegotiated (Table 3).

Class 2: Near real-time

This class does not have so strict delay requirements and recovery time requirements. This means that we may use more relaxed mechanisms; i.e. shared protection and global recovery mechanisms with full recovery. This will then map to a medium high resilience class.

Assuming some application in future may reconnect automatically and adapt to lower available bandwidth, these applications may use a medium low resilience class.

Class 3: Interactive data

For this class restoration or IP-rerouting is probably good enough; i.e. a relaxed requirement for recovery time. But TCP or application layer timeouts should be met. This means that in some cases (at least referring to convergence times in today's IP networks) rerouting will not perform well enough, and global recovery could be used. For elastic traffic we can tolerate lower throughput in failure situations. In total we than end up with a medium low or low resilience class (Table 3).

Class 4: Background

For this traffic IP rerouting 'always' possible. But, due to DiffServ low priority, throughput will probably be more affected than for class 4. Such differentiation is not distinguished in the proposed scheme, and we end up with the low resilience class.

The proposed resilience classes are shown in Table 3.

Resilience class	Recovery time τ	Bandwidth guarantee	Mechanism
Emergency	$\tau < 100$ ms	Yes	1+1 protection / 1:1 local protection
High	$\tau < 100$ ms	Yes	Dedicated 1:1 or shared 1:n local protection
Medium high	100 ms $< \tau < 1$ s	Yes	Shared 1:n global protection
Medium low	100 ms $< \tau < 1$ s	No	Shared 1:n global protection
Low	$\tau > 1$ s	No	IP rerouting

Table 3: Resilience classes

D The Fault Hierarchy - A Framework for Failure Management

D.1 Motivation

In terms of the HIDENETS project and the work undertaken in WP3, resilience is about managing failures and implementing mechanisms that make the system resilient and robust despite possible occurrences of failures. In this context we distinguish between a failure that is the loss of ability to function as intended and an error that is a detected deviation from the correct service state or agreed specification. An error is caused by a fault or due to outside interference. The error may propagate to a failure. On the other hand a fault is the adjudged or hypothesized cause of an error or a defect that gives rise to an error. So in short a fault is what is built into the system, an error is an incorrect state and a failure is the observable phenomenon. From a communication point of view it seems most natural to contribute to failure management by analyzing possible faults, errors and failures and their consequences.

D.2 Hierarchy and Layering

The concept of a fault hierarchy derives from the concept of network layering. Thus, in this subsection we will briefly describe layering in a broader context, and explain how faults and failure management fit into the layered communication model.

Historically, networking technology has followed two separate paths of development: On the one hand, the telecom industry has tailor-made networks to provide particular end-user services. For example, the plain-old telephone networks were tailor-made to provide telephony services. Thus, the technology was vertically integrated and optimized for the end-user service, with often unclear boundaries between the networking layers of the OSI-model.

On the other hand, the computer-network industry provided generic connectivity services. For example, Ethernet was developed for local link layer connectivity, IP and routing for global networking connectivity, TCP and UDP for end-to-end connectivity and so forth. The communication is layered, and each layer provides generic services to the layer above. The interactions between layers and the assumptions that a layer makes about the layer below, are minimized. The technology is modular, and through proper standardization different off-the-shelf products can be put together to form relatively inexpensive networks. Indeed, in principle a nearly unlimited number of different end-user services can be run on top of these layered networks.

Like most of the industry today, the HIDENETS project has not the resources or ambitions to re-invent networking. Thus, the project will instead follow the layered communication model that dominates today, and improve this model in terms of enhanced resilience. In some cases, however, the HIDENETS project might propose cross-layer optimizations that might help improving resilience. Nevertheless, in the HIDENETS project it is natural to start analyses from the perspective of layered communication. This is also the case for analyses of faults and failure management.

With a layering model, each layer implements mechanisms that handle faults and errors that occur within the layer. An error which is treated and contained within the layer is not observable from outside, so that layer does not fail. Errors that are not fixed inside the layer could propagate as failures till the layer boundaries. A failure (observed at the interface of a lower level service) is an external fault for the layer above, so we will refer it as a failure that the layer above has to treat. In many cases, the failure in a lower layer sooner or later reaches a higher layer in the networking stack where it can be fixed, and hidden for the remaining upper layers.

First, we note that different faults may occur in different layers. It is therefore natural to introduce the term "fault hierarchy" as a framework for fault analyses. The fault hierarchy illustrates in which layer(s) different types of faults may occur. It also illustrates the interaction between faults at different layers. For example, buffer overflow at one layer might translate into a packet loss at the layer above; a bit error at one layer might translate into a lost packet at the layer above; or network contention/congestion at one layer might translate into increased packet delay at a higher layer.

Finally, we note that there is a high degree of resilience and a large number of resilience mechanisms already implemented in today's layered networking model. Most routing protocols of IP (including the exterior

routing protocol of IP, BGP, which binds the Internet together) were designed with robustness in mind. If one border gateway or router is removed, the routing protocols adapt and - if possible - find alternative routes around the failed router within relatively short time.

Although IP routing is a good example of resilient design, it is not the only one. Indeed, as mentioned above each layer might try to fix errors occurring in the layer itself, some of which are caused by failures in the layer below.

D.3 The Fault Hierarchy

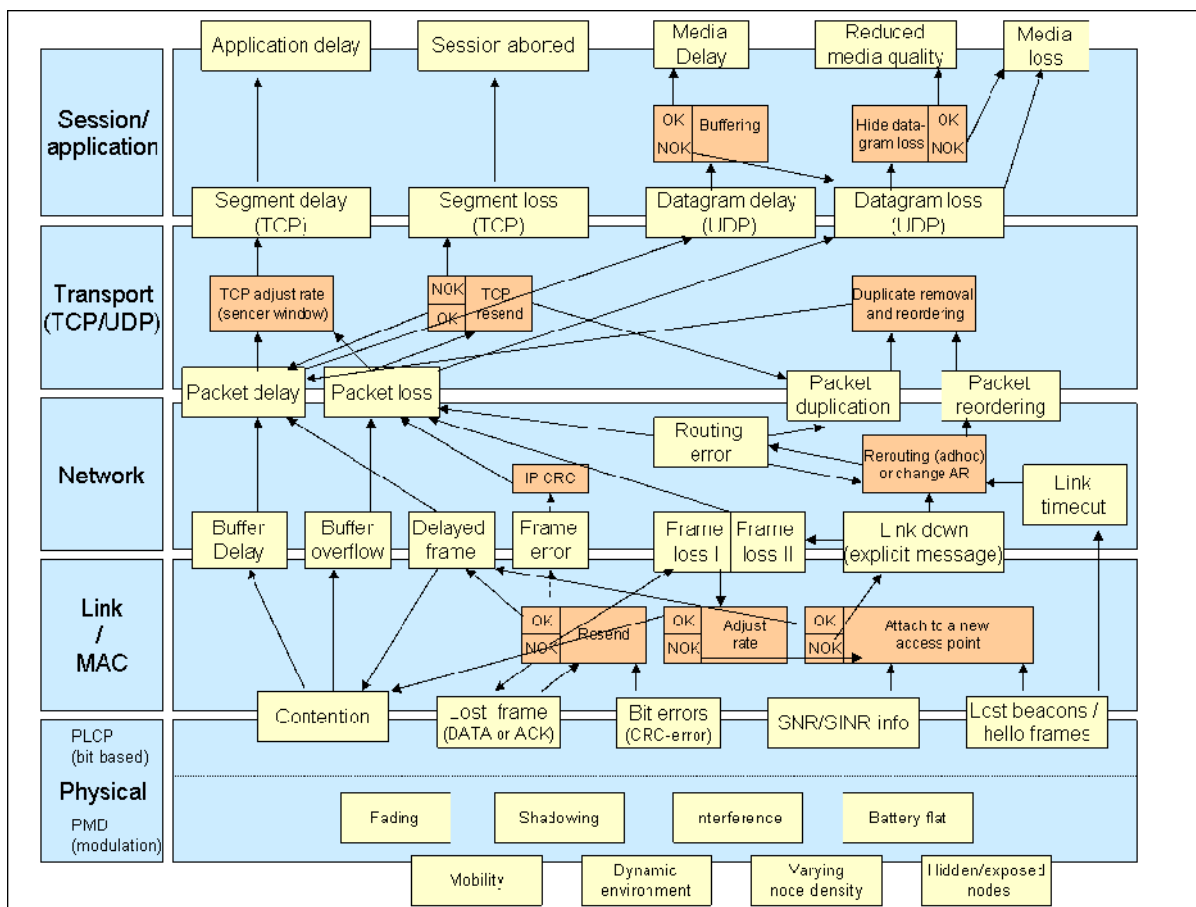


Figure 8: The Fault Hierarchy to be used as a framework for fault analyses in WP3 of the HIDENETS project matrix.

Figure 8 gives an outline of a fault hierarchy that will be used as a framework in the HIDENETS project. The faults are described by the yellow boxes, while the built-in resilience mechanisms are illustrated with orange boxes. ("OK" refers to the case where the resilience mechanism is able to fix the error, while Not OK - NOK - refers to the opposite.) The boxes are placed relative to the different layers, i.e. the physical layer, link layer, network layer, transport layer and session/application layer. The main focus here is on faults caused by failures in the layer below, and these are placed on the boundaries between two layers. The boxes will be explained in the following subsections.

D.3.1 Simplifications in the Fault Hierarchy

First we note that it is very difficult to provide a complete picture of the fault hierarchy with every possible fault described in detail. It is also difficult to describe every interaction between faults and between the mechanisms in a layer used to fix faults. The more the fault hierarchy is described in detail, the more complex it gets, and the less useful it is as a framework. Thus, we have attempted here to find a suitable level of detail. As seen in the Figure 8, the description is already quite complex, and describing the fault hierarchy in more detail than what is done in the figure might not serve its purpose well.

The fact that a model will not be able to cover all possible faults is easy to realize by mentioning some extreme cases of faults. For example, the hardware might be smashed in a car accident or hit by lightning. Likewise, there might be hackers that purposely try to launch DoS attacks, e.g. by transmitting bogus 802.11 management frames that dissociate all nodes attached to an access point / base station. Finally, nodes might be mal-configured in all sorts of way, leading to nearly all sorts of possible errors. The main point here is

that the potential types of faults are quite large, and too large to be included in the framework presented in this section. The first simplification is therefore to only study some of the most plausible causes of failures.

Another simplification of our model is that it might conceal the fact that faults might have different degrees of gravity. For example, the "packet delay" fault box in Figure 8 says nothing about the extent of the delay. The packet delay might be so low that it does not affect the functionality of the higher layers, and the effects of this delay might for example be removed by a streaming buffer at the application layer (Figure 8). On the other hand, the packet might be delayed to such an extent that it is mistakenly perceived as a dropped packet by the higher layers. For example, if the delay is above a certain threshold, the TCP might react by reducing its sender window and consequently the senders mean transmission rate. The same argument goes with packet loss. In some cases only a single packet is lost, which might for example not influence the quality of a voice conversation considerably. In other cases, there might be a large number of consecutive packets that are lost, in which case the consequences are much more serious.

The faults at the physical layers are not described in detail in Figure 8 and there are many reasons why. First, the physical layer is not the main focus of the HIDENETS project. These faults will normally translate into link-layer faults, where the focus of the HIDENETS project begins. Secondly, the interactions between the boxes here can be quite complex, and it might be better not to show all the interactions in the figure than showing them all.

Nevertheless, let us briefly discuss the boxes at the physical layer. The interactions between the boxes at the physical layer follow the following lines: Radio communication is inherently unreliable and unstable, both in face of node mobility, a dynamic environment (in terms of both external objects and other transmitting nodes) and varying node densities. This results in fading, interference and shadowing. Interference might also derive from the hidden terminal and exposed terminal problems. Fading, shadowing and interference might in turn lead to bit errors, loss of data frames or even loss of management frames (such as beacon frames in infrastructure mode or hello frames in ad-hoc mode). The latter failure might make a direct influence on the behaviour of the layers above. A flat battery or other hardware failures might also result in frame loss.

Finally, fading, shadowing, interference and a missing node (e.g. due to a flat battery) are effects that all influence the signal-to-noise ratio (SNR), which might again be useful information for the link layer. In fact, some routing protocols might also take advantage of the SNR. There are extensions for both AODV and OLSR that allows these protocols to be radio aware and take radio metrics into account when routing decisions are to be made. (Likewise, IEEE 802.11s are also using such extensions to AODV and OLSR for the path selection procedures for mesh networking.)

D.3.2 Faults within a Layer

Although the main focus is on faults caused by failures in the layer below, there might also be a large number of faults that occur within a layer and which are fixed locally. Some of the latter faults might influence higher layers, and some of these are illustrated in the figure. For example, a routing error occurs within the networking layer, but might result in packet loss and might consequently be perceived as a packet loss fault by the layer above. As shown in the figure, the routing error might be detected and finally result in another reroute (a.k.a. route repair).

D.3.3 Faults that are forwarded across a layer

Some of the faults are forwarded to the layer above without being fixed there, e.g. lost frame at the link layer might result in a packet loss error at the networking layer. Other examples are also found in the figure. For example, contention (/congestion) at the link layer, might translate into buffer delay or buffer overflow at the lower part of the networking layer (e.g. in the device driver or in the socket). These are perceived by the transport layer as packet delay and packet loss, respectively, and as datagram delay and datagram loss by the application or session above if UDP is being used. Finally, a lost datagram might for example result in a loss of VoIP signal (referred to as "media loss" in the figure).

Furthermore, the figure also shows that if a TCP segment is delayed, it might delay the execution of the application above, while a TCP segment loss might eventually result in a TCP reset in which the TCP session is aborted. Finally, loss of a multicast hello packet (in ad-hoc mode) or of a beacon frame (in infrastructure

mode), might result in link timeout at the networking layer, in which the networking layer assumes that the link is down.

D.3.4 Faults related to existing resilience mechanisms at the link layer

There are fault repair mechanisms already implemented in the current protocol stack. These resilience mechanisms are shown in the orange boxes of Figure 8, and the relation between these and the faults in the yellow boxes are shown by arrows. These boxes will be described in turn in this subsection. Our description will start out with the fault repair mechanisms at the link layer and then move up the layers.

As explained in section B.1 Automatic Repeat Request (ARQ) is used to request that a packet or frame received with error(s) be retransmitted at Layer 2. If such retransmission does not succeed this is perceived as another lost packet or frame, and the retransmission process is repeated. This is illustrated by the arrows to and from the yellow "lost frame" box in Figure 8. For each time a packet is retransmitted, a retry counter is incremented. When the retry counter exceeds 4 (for "long" packets sent with RTS/CTS) or 7 (for "short" packets sent without RTS/CTS), the IEEE 802.11 standard mandates that the frame is discarded. This fault is referred to as "Frame loss I" in Figure 8. This figure also illustrates the use of rate adjustment as described in B.1. The rate adjustment is carried out in the orange "Adjust rate" box in Figure 8. If rate reduction does not succeed (or if it is not implemented), the node might try to re-associate with another access point (or if this is not possible it might notify the network layer of a "Link down").

Further the node might monitor the SNR and the signal quality and use this information when making decisions on which access point (or "base station") to associate with. This decision is made in the orange "Attach to a new access point" box in Figure 8. If the signal quality from the access point with which a node is associated deteriorates or if the node does not receive some subsequent beacon frames, this might trigger the node to associate with another access point within the same extended service set (ESS). This kind of re-association might also occur if the rate adjustments fails (or is not implemented) as mentioned above.

D.3.5 Faults related to existing resilience mechanisms at the network layer

The orange box labelled "Rerouting or change of AR" addresses the cases where rerouting or change to a new access point is necessary due to a link failure. Also in the case that the node is in ad-hoc mode, a "Link down" notification might lead to changes in the routing protocol. This change (or "reroute") is also carried out in the orange "Rerouting or change AR" box. Alternatively, the reroute might also be triggered by the loss of a number of hello frames. In order to capture this type of fault, a link timeout is implemented in the routing protocol at the network layer, as illustrated in the figure.

If the node is in ad-hoc mode and is a part of an ad-hoc network, the functionality carried out in the "Rerouting or change of AR" box might be considerably more complex than the functionality carried out if the node is in infrastructure mode. Thus, for a node that is in ad-hoc mode the "Rerouting or change of AR" box could probably easily be divided into a large number of additional yellow fault-boxes and a additional orange resilience-boxes that tries to fix these additional errors within the context of the routing protocol. Thus, a separate fault hierarchy could probably be constructed for the routing protocol separately. This is however out of the scope of this deliverable.

Finally, the orange "IP CRC"-box is another resilience mechanism at the network layer. Although the cyclic redundancy check carried out at the MAC layer of 802.11 captures most packets that are corrupted by bit errors, there is a certain theoretical chance that a corrupted packet might pass the CRC-test. The minimal chance of this happening is illustrated by the broken arrow from the orange "Resend" box at the link layer to the yellow "Frame error" box above. Such corrupted frames will be captured by the CRC-check of the IP-packet carried out on the next-hop router. (Again there is a hypothetical chance that the packet might again pass the test. However, since the chances are small, we have not illustrated any higher layer mechanisms, such as the optional UDP CRC-check, that might deal with this.)

D.3.6 Faults related to existing resilience mechanisms at the transport layer

TCP controls the transmission rate by adjusting the sender window, and uses "additive increase & multiplicative decrease" (AIMD) for congestion avoidance. This mechanism is illustrated by the orange "TCP adjust rate" box in Figure 8. TCP normally clocks the transmissions by the received ACKs, and thus constantly increases its sender window additively. However, if a packet is lost or excessively delayed, TCP

assumes this is due to buffer delay and buffer overflow and that there is congestion in the network. TCP responds by reducing the sender window to one half of its original. The transmissions continue with additive increase, until the sender window again has to be reduced to half its size. A reduced sender window translates into reduced transmission delay, and delay of the entire TCP segment that is under transmission.

Lost packets might be fixed by the orange "TCP resend" box. The reason is that TCP retransmits packets that are not ACKed (or are explicitly or implicitly NACKed). If TCP succeeds, it might at least result in increased overall transfer delay of the packet that is resend. If it erroneously resends a packet, packet duplication might be the result. If TCP tries resending over and over again without succeeding, the TCP session will finally be aborted.

The receiver of TCP packets maintains a receiver window. Duplicate packets are discarded at the receiver window, and out-of-order packets are ordered within the window. This functionality is handled by the orange "Duplicate removal and reordering box" in Figure 8.

D.3.7 Faults related to existing resilience mechanisms at the session or application layer

Real-time applications might implement a buffer that accounts for varying datagram delay (i.e. jitter). However, the "Buffering" introduces additional "Media Delay". Normally, the buffer might only handle a limited amount of datagram delay, and datagrams delayed more than this limit will be perceived as "Datagram loss".

Some real-time applications, such as some video streaming and audio-streaming applications, are able to "Hide datagram loss", for example by inserting a "dummy" voice or video packet that fits well into the context. It might result in some "Reduced media quality". However, the opportunities to do this type of repair require that only a limited share of packets is lost. With a higher share of dropped datagrams - or with a set of consecutive missing datagrams - the application might not be able to cope with the fault and the result will be "Media loss".

D.4 Failure-detection

The one issue that unites almost all approaches to distributed computing is the need to know whether certain components in the system have failed or are otherwise unavailable. The ability to efficiently and accurately detect failures is a key element underlying reliable distributed computing, as improving recovery time (time to detect, to diagnose and to repair) improves reliability.

Often, distributed systems cannot distinguish between process (application), node or network failures. Failures at network level are in general related to failures of routers and gateways, or to severe degradation of the service level due to network congestion, causing minimum performance requirements to be violated. It is considered that in actual implementations, the hardware is often the least likely source of failures.

Roughly speaking, a failure detector is viewed as a distributed oracle that provides hints about failures in the system. Various methods to detect failures are used, of which failure monitors, heartbeats and polling are the most popular. Most of the failure detectors are implemented at the transport layer, using time-out mechanisms. Working at the transport layer limits the detection to node and link failures. Detecting and diagnosing application layer faults are hard to achieve.

In an asynchronous system, it is impossible to distinguish a crashed process from a very slow one. Hence, the knowledge of a node concerning another node is always subject to some uncertainty. A failure detector can only have suspicion about the state of an observed process. If the cost induced by false alarms can be tolerated, then it might be worthwhile for the fault-tolerant solution to react to wrong suspicions as this implies that the fault-tolerant solution will also react to more (i.e., most, if not all) actual failures than when applying tighter reactivity policies.

The notions of service availability and reliability are the main focus in our framework, so we must ensure that the systems detects errors or failures as soon as possible to permit the failover mechanisms to act rapidly and avoid critical delay that could endanger the ongoing communications. In most cases, the failure detector assumes a node crash and triggers the failure-resilience mechanisms to minimize the user's perception of the error/failure. Nevertheless, fault-diagnosis is sometimes necessary and in HIDENETS the definitions of availability and reliability are related to service provisioning, which is implemented at the application layer. Then, discovering a link or node failure is not enough. So the failure-detection functionality must interact

with all the levels, up to the application layer. The detection of any application error/failure is then crucial.. The trade-off between accuracy and reactivity should be carefully considered. The optimal case is when the failures are detected “early enough”, i.e. before the next request is sent to the serving entity, so that the appropriate resilient mechanism has already switched transparently to another available node/process. In other terms, accurate and proactive fault diagnosis should be implemented to ensure optimal dependability.

E Comparison of 802.11b, 802.11a and 802.11g

E.1 Scope of this section

This appendix provides an overview of 802.11b, 802.11a, and 802.11g. Since all these extends only the physical layer (PHY) of legacy 802.11, an overview of the PHY layer of legacy 802.11 is provided as an introduction to the other PHY extensions. By the same token, this section will not discuss MAC layer issues, since these are not relevant to the PHY extensions discussed here. (However, 802.11g discussed below involves also a few MAC layer issues, which will be treated here.) Knowledge of the generic 802.11 MAC layer is assumed in this section.

E.2 Overview of the legacy 802.11 DSSS PHY

E.2.1 Core functions

The 802.11 PHY provides three levels of functionality

1. The PLCP sublayer provides a vertical frame exchange between the PHY and the MAC layer
2. The PMD sublayer uses PHY-specific spread spectrum modulation to transmit bits over the media
3. The PHY provides a carrier sense indication back to the MAC to verify if there is activity on the wireless channel. This is referred to as a Clear Channel Assessment (CCA).

In the following the PLCP sublayer and the PMD sublayer will be described in further detail.

E.2.2 PLCP sublayer

The PHY Packet Data Unit (PPDU) consists of a PLCP preamble and a PLCP header that wraps up a MAC PDU. The preamble is used so that the receiver can acquire and synchronize to the incoming channel, and thus be able to correctly conduct the demodulation for the rest of the packet.

The PLCP preamble is sent at 1Mbps for all the PHY extensions in the 2.4 GHz band. This includes the legacy IEEE 802.11 DSSS PHY, the 802.11b HR/DSSS PHY and the 802.11g CCK-OFDM PHY. On the contrary, for the 802.11a OFDM PHY, which operates in the 5.0 GHz band, the preamble is sent at 6Mbps.

The legacy DSSS PHY specifies that the PLCP header is sent at 1Mbps, while the following MPDU might be sent at 1Mbps or 2Mbps. Generally, the main point with the other PHY extensions is to allow that these can be transmitted at higher rates. This requires specifications of new modulation techniques.

The first 128-bit field of the preamble (of the DSSS PHY) is specified to be a string of ones before they are scrambled. The receiver uses these 128 bits (referred to as the SYNC field of the preamble) to synchronize the receiver's carrier tracking and timing. The last 16-bit field of the preamble contains the hexadecimal word "F3A0". This is referred to as the SFD field of the preamble and marks the start of the PLCP header (which is also the start of the PPDU frame).

The PLCP header contains

- an 8-bit Signal field, which tells which kind of modulation is used in the following MPDU part of the frame;
- an 8-bit Service field reserved for further use;
- a 16-bit Length field indicating the number of micro-seconds it takes to send the MPDU; and
- a 16-bit CRC field as an error detection field over the other fields of the PLCP header. (Needless to say, the CRC is calculated prior to the scrambling of the underlying PMD sublayer) It should be noted that the PHY is not involved in error detection of the overlying MAC layer. Instead, the MAC frame has its own FCS field for this purpose.

E.2.3 PMD sublayer

All bits that the PMD receive from the PLCP sublayer for transmission are first scrambled using a 7-bit self-synchronizing polynomial. This is done to randomize transmissions of long strings of zeros and ones. This means that also the one-bits in the SYNC field of the PCLP preamble (above) are scrambled.

After having been scrambled, each bit is XORed with a specific 11-bit Barker word (or a Barker word consisting of 11 chips). Thus, if the bit-rate is 1Mbps, the chip-rate is 11Mbps.

In this way the spread-spectrum properties of 802.11 is introduced.

Barker words are generally known to have excellent correlation properties. However, they are not orthogonal as in UMTS (Wideband CDMA) or in GPS. Indeed, all STAs in the network use the same Barker word.

The DSSS PHY uses differential phase shift keying (DPSK), with counter-clockwise symbol phase rotation. DPSK does not require a clock reference for data recovery. DPSK is applied in 2 flavours: For 1Mbps transmission, Differential Binary Phase Shift Keying (DBPSK) is used, where 0-chip or 1-chip is coded by a 0-degree or 180-degree phase shift. For 2Mbps transmission, DQPSK is used instead. Here, a set of two chips is coded into a 0-degree, 90-degree, 180-degree or 270-degree phase shift.

The direct sequence introduced by the Barked word spreads the signal over a wider bandwidth and at a reduced RF power level. It results in a spectral Sin X/X shape. However, the resulting signal is filtered according to the IEEE specification, especially targeting at reducing the side-lobes of the spectral shape.

Needless to say, the bits are treated oppositely at the receiver side: The signal is first de-modulated into chips, and the bits are found by XORing with the Barker word. Finally, the PMD descrambles the bits, before they are sent further up to the PLCP sublayer.

E.2.4 Physical implementation of MAC parameters

- The Short Interframe Space (SIFS) is set to 10us;
- the slot length is (aSlot) is set to 20us; and
- the aCWmin is set to 31.

E.3 Overview of 802.11b

E.3.1 The main functionality added by the PHY extension

802.11b is a PHY extension to 802.11. This means that a node with 802.11b uses the same MAC as that used with legacy 802.11. The only difference is that it uses a different PHY. The main differences are found in the modulation techniques, simply because the main purpose of 802.11b is to increase the nominal bit-rate of the MPDU transmission.

The IEEE 802.11b PHY is currently one of the most widely deployed PHY extensions, challenged only by the more recent 802.11g PHY extension (below). It is commonly also referred as the HR/DSSS PHY, and operates in the same 2.4 GHz band as the legacy 802.11 DSSS PHY. The PLCP and PMD sublayers of the HR/DSSS PHY also operate basically in the same way as for the DSSS PHY.

The 802.11b HR/DSSS PHY has two main functions:

1. The PMD sublayer of HR/DSSS extends the PSDU data rates to 5.5 Mbps and 11Mbps, by means of an enhanced modulation method.
2. The PLCP sublayer of HR/DSSS provides a rate shift mechanism, which allows the network to fall back to 1Mbps and 2Mbps and to interoperate with the legacy DSSS PHY.

Let us first look at the changes in the PLCP sublayer, before we finally discuss the changes in the PMD sublayer.

E.3.2 Enhancement of the PLCP sublayer

The PLCP of the HR/DSSS comes in two variants, with long preamble and short preamble. With the "long preamble", the PCLP preamble and PLCP header are equal to those of the legacy DSSS PHY, the bits are scrambled with the same algorithm, and the PLCP header is sent at 1Mbps with DBPSK modulation. Thus, this variant is fully compliant with the legacy DSSS PHY described above.

The "short preamble" variant, on the contrary, uses a SYNC field of the preamble of only 56 bits. (Thus, the PLCP preamble is of totally 72 bits). It uses the same scrambling polynomial, but it is initialized with a different 7-bit seed bit pattern than that used for the long preamble. Thus, this mode of operation does not interoperate with legacy 802.11. Therefore all short preamble radios must also be able to fall back to long preamble operation in order to be IEEE 802.11 compliant. In addition to saving some overhead with a shorter preamble, the short-preamble mode of operation transmits the PLCP-header at 2Mbps with DQPSK modulation, saving additional overhead.

Since the HR/DSSS allows the MPDU to be sent at 5.5 Mbps or 11Mbps, two new values are defined for the Signal field of the PCLP header to accommodate use of these higher rates. This feature and the fact that support for long preamble operation is mandatory, form the basis for the fallback mechanism to legacy 802.11 operation. However, the fallback mechanism is also supported by the MAC layer, where MAC management messages are used to negotiate transmission rates between the access point (AP) and the station (STA).

Two bits of the Service field are used to indicate different modes of operation of the modulation. Finally the 7th bit of the Service field is borrowed to the Length field, to increase the Length field with an additional bit.

Apart from this, the PLCP sublayer is basically the same as that for legacy 802.11.

E.3.3 Enhancement of the PMD sublayer

The mandatory modulation of HR/DSSS is Complimentary Code Keying (CCK) with QPSK, also referred to as CCK-QPSK. The Packet Binary Convolutional Coding (PBCC) is an optional coding scheme that will not be discussed here.

With the high rate operation the 11-chip Barker code is replaced by 64 different complimentary codes, each consisting of 8 chips. 8 bits of the incoming bit stream forms a symbol, and the symbol rate is 1.375 MSps, resulting in a bit rate of 11 Mbps.

The last 6 bits of the symbol is used to select one of the 64 different complimentary codes. The remaining 2 bits of the symbol are used to QPSK the symbol. This means that there is a phase shift between each 8-chip symbol, that encodes for 2-bits. In addition, the chips within the symbol are also modulated with QPSK.

The chipping rate is kept at 11Mbps. Thus, each complimentary code is transmitted at 1,375 MSps, yielding 6 bits of information, while 2 bits of information comes from the modulation. The resulting bit rate is 11 Mbps. The modulation has the same bandwidth as legacy 802.11. The spreading rate remains constant, while only the data rate changes.

The 5,5 Mbps is obtained by replacing the set of 64 complimentary codes, with a set of 4 different complimentary codes. Thus, the code selection codes for 2 bits, while the remaining 2 bits are encoded by QPSK of the complimentary code. Therefore, each symbol now yields 4 bits instead of the 8 bits obtained with the 11Mbps operation, resulting in half the bit rate.

E.3.4 Physical implementation of MAC parameters

- The Short Interframe Space (SIFS) is set to 10us;
- the slot length is (aSlot) is set to 20us; and
- the aCWmin is set to 31.

E.4 Overview of 802.11a

E.4.1 The main functionality added by the PHY extension

While the other PHY extensions operates in the 2,4 GHz ISM band, the 802.11a PHY extension operates in the 5,0 GHz band. However, they all operate with the same MAC.

802.11a provides data rates ranging from 6Mbps to 54 Mbps. In doing so, it uses Orthogonal Frequency Division Multiplexing (OFDM), and the PHY extension is referred to as OFDM PHY. The specification is similar to the OFDM PHY specification of the ETSI HyperLAN II. Not surprisingly, the result is a specification that is quite different from the base 802.11 specification and from the 802.11b PHY amendment.

E.4.2 The PLCP sublayer

The PLCP-preambles of the OFDM PHY consists of 12 symbols; 10 short symbols followed by 2 long symbols. The short symbols takes 0,8 us, and are used to get a coarse synchronization. The long symbols take 4 us, and are used for fine-tuning. The resulting preamble takes totally 16 us.

The PLCP header is 24 bits, and consists of a 24-bit Signal Field and a 16-bit Service Field.

The Signal Field consists of:

- a 4-bit rate field, to encode the modulation used for the Data Field, which is the latter part of the frame;
- a 1-bit Reserved field, reserved for future use;
- a 12-bit Length field to encode the number of bytes in the PSDU
- a parity bit;
- a 6-bit symbol Tail field.

The PLCP preamble and the Signal Field are transmitted and modulated at the lowest bit rate, i.e. at 6Mbps. The Service Field of the PLCP header, on the contrary, is defined as being part of the Data Field, which may be transmitted at a different bit rate.

The Data Field contains the Service Field, the PSDU, 6 tail-bits, and some padding bits at the end of the frame.

E.4.3 The PMD sublayer

The two lowest bit rates are encoded with BPSK, the next two bit rates with QPSK, then follows QAM-16 and finally the two highest bit rates are encoded with QAM-64.

Each modulation method is accompanied by Forward Error Correction (FEC) using convolutional coding, typically at the rate of $\frac{1}{2}$ or of $\frac{3}{4}$ (except for the 48Mbps modulation, which uses a code rate of $\frac{2}{3}$). The possibility to vary the FEC robustness leads to two different bit rates per modulation method. In this way, the 802.11a provides the following bit rates: 6, 9, 12, 18, 24, 36, 48 and 54 Mbps.

After having scrambled the bits, the PMD introduces FEC by convolutional coding. The bits are then interleaved.

OFDM divides the channel into 52 orthogonal subcarriers; 48 data subcarriers and 4 carrier pilot subcarriers. The subcarriers are combined using an inverse Fast Fourier Transform (FFT) on the interleaved bits, symbol shaping is done and the bits are finally transmitted with PSK or QAM modulation.

E.4.4 Physical implementation of MAC parameters

- The Short Interframe Space (SIFS) is set to 16us;
- the slot length is (aSlot) is set to 9us; and
- the aCWmin is set to 15.

E.5 Overview of 802.11g

E.5.1 The main functionality added by the PHY extension

802.11g can be seen as a combination of 802.11a and 802.11b. First, it operates in the 2,4 GHz ISM band, but provides OFDM with the same modulation methods and bit rates as 802.11a. For backward compatibility, it also supports CCK.

Unlike the other PHY amendments, 802.11g also implements new MAC mechanisms that are tailored to ensure backward compatibility and inter-operability with 802.11b (and the legacy 802.11 DSSS PHY).

E.5.2 Mandatory PLCP and PMD features

The following is mandatory in 802.11g:

- support for 802.11b CCK
- support for 802.11a signalling (however, in the ISM frequency band)

Since the use of short preamble of CCK is also mandatory, 802.11g must support 3 different PLCP preamble and PLCP header formats; two for CCK operation and one for OFDM operation. Please refer to the sections on 802.11b and 802.11a for details on these formats.

E.5.3 Optional PLCP and PMD features

As a non-mandatory option, 802.11g also specifies a hybrid CCK-OFDM mode of operation. Basically, the frame starts with a PLCP preamble and PLCP header equal to that used in 802.11b. The preamble is transmitted at 1Mbps with DBPSK, and depending on whether the preamble is long or short, the PLCP header is transmitted at 1Mbps or 2Mbps, in line with the 802.11b specification.

The big difference from the 802.11b amendment is that the following PSDU is modulated with OFDM. The PSDU contains a Long SYNC field of 8 us and a Signal field of 4 us transmitted at 6Mbps. Then follows the data symbols transmitted at a rate between 6Mbps and 54Mbps. Finally comes a 6us signal extension field.

Since CCK-OFDM uses the same PLCP preamble and PLCP header as for 802.11b, interoperation with 802.11b or legacy 802.11 stations is not a problem. However, since CCK-OFDM is an optional feature (mainly because it is an inefficient way of using OFDM), other mechanisms are required to ensure smooth interoperation between 802.11g and 802.11b. 802.11g implements some MAC mechanism for this purpose.

E.5.4 802.11g MAC mechanisms to ensure interoperation with 802.11b

As mentioned earlier, a main function of the PHY is to provide a CCA indication to the MAC layer to verify if the medium is busy or not. The CCA can be partly based on energy detect (ED) and partly on carrier sense (CS). In addition, the Network Allocation Vector (NAV) provides the MAC with a virtual carrier sense mechanism.

In a mixed-mode network with both 802.11b and 802.11g devices, the CS of the 802.11b devices will hardly detect the OFDM transmissions of the 802.11g devices. A way to circumvent this problem is to use the RTS/CTS mechanism, where the RTS and CTS packets are transmitted at 1Mbps and will be decoded by the 802.11b STAs. These packets set the NAV in the 802.11b stations, and ensures that these stations are prohibited from sending during the OFDM transmission, despite that the 802.11b stations can sense the OFDM transmission. CTS-to-self, where the transmitting 802.11g STA sends a CTS to itself, is also a mechanism that can be used successfully to avoid the interoperation problem, instead of a traditional RTS/CTS exchange.

E.5.5 Physical implementation of MAC parameters

- The Short Interframe Space (SIFS) is set to 10us (or 16 us if you include the Signal Extension of 6 us);
- the slot length is (aSlot) is set to 9us; and
- the aCWmin is set to 15 in a 802.11g-only network and to 31 in mixed environment also encompassing 802.11b STAs.

E.6 Recommendation for the HIDENETS project

It seems that the differences in PHY extensions are a little out of scope to the HIDENETS project. It is recommended that the project keep its main focus on the MAC layer and above. Thus, describing the MAC layer of legacy 802.11, and the enhancements provided by the 802.11e standard is probably more appropriate as a knowledge basis for further work with the HIDENETS project. In light of recent discussions within the HIDENETS project, 802.11h (for dynamic frequency/channel selection) is also an issue that seems relevant.

F Allocation of Packet Data Channels and channel rates in GPRS

For allocation of PDCHs in GPRS Session Management first has to establish a PDP Context and as part of this negotiating QoS profile etc. The QoS profile will be signalled to the TBF module of MT. This module will establish a Temporary Block Flow (TBF) when MT wants to send a data segment (packet or a burst of packets). This is a unidirectional MAC connection identified by a Temporary Flow Identifier (TFI). To establish a TBF a Packet Channel Request is sent to the TBF module of Base Station Controller (BSC, the controller unit in the radio access network). This signalling message uses random access and contains radio priority and the number of radio blocks that shall be transmitted. BSC answers with a Packet Uplink Assignment that contains TFI and assigned PDCHs. MT is then ready to transfer the data.

The TBF is temporary and is maintained only for the duration of the data transfer, which means until there are no more blocks to be transmitted and all the transmitted blocks have been acknowledged by the receiving entity. Concurrent TBFs may be established in opposite directions.

Also downstream the allocation of PDCH is per TBF, thus we need signalling per data 'segment' between BSC and MT. A 'segment' can be a packet or a flow of packets.

In the convergence layer (SNDCP) an IP-packet coming from the network layer is first split into segments. A maximum segment size can be defined; the default value is 1503 bytes. In the logical link layer (LLC) a header (1 byte Address field and variable length Control field) and a tail (3 byte Frame Check Sequence field) are added.

Four coding schemes (CS-1, CS-2, CS-3, CS-4) can be selected for GPRS depending on the radio conditions. The difference is the level of redundancy. CS-1 and CS-2 offer good error detection and correction with lower throughput than CS-3 and CS-4. In the first phase of GPRS only these two techniques are in use. CS-3 and CS-4 provide higher throughputs but have little or no error correction capabilities. The data rates of the four coding schemes are:

- CS-1: 9.05 kbps
- CS-2: 13.4 kbps
- CS-3: 15.6 kbps
- CS-4: 21.4 kbps

In the radio link layer the LLC frame is split into segments of size either 181 bits or 268 bits according to channel coding scheme used, CS-1 or CS-2 respectively. With header and tail added we then get the RLC radio block after coding and puncturing.

If MT supports multiple simultaneous applications and QoS, it has a packet scheduler that can give priority to one application over the other. In the same manner BSC will have a packet scheduler that can give priorities according to the negotiated QoS profiles. The packet scheduler is in fact a 'radio block' scheduler, meaning that the radio blocks are queued for transmission on a given PDCH, and one radio block is transmitted at a time. The packet scheduling algorithm is not standardised and can have great impact on the perceived quality of the applications.

Radio Link Control (RLC) provides a retransmission service. In fact, RLC can be used in three different modes:

- Transparent mode: In this mode no overhead is added to higher layer data.

- Unacknowledged mode: In this mode no retransmission protocol is used and data delivery is not guaranteed.
- Acknowledged mode: This mode provides Automatic Repeat reQuest (ARQ) mechanism for flow control and error correction.

In the acknowledged mode the following procedure takes place. After receipt of a complete RLC SDU, successful receipt is acknowledged or retransmission of PDUs is requested by sending one or more status PDUs to the RLC peer. At the sending side transmitted PDUs are put in a retransmission buffer. On receipt of the status PDUs they are then either retransmitted or deleted. Retransmitted PDUs are multiplexed with newly generated PDUs from higher layers.

EDGE uses the same GSM frame structure as GPRS, but with enhanced data rates. Edge theoretical performance with different coding schemes is:

- MCS-1: 8.8 kbps
- MCS-2: 11.2 kbps
- MCS-3: 14.8 kbps
- MCS-4: 17.6 kbps
- MCS-5: 22.4 kbps
- MCS-6: 29.6 kbps
- MCS-7: 44.8 kbps
- MCS-8: 54.4 kbps
- MCS-9: 59.2 kbps

This implies 3 – 4 times higher data rates for end-users compared to GPRS.

G Node software architecture

This appendix contains the description of the services that was not included in the main volume of the deliverable. These services have not been developed further within HIDENETS and only include a general description.

G.1 Infrastructure mobility support – client part

The infrastructure mobility support aims at providing connectivity directly to the infrastructure domain for mobile client nodes, or groups of mobile nodes. It ensures that sessions are not broken when a node in infrastructure mode (as opposed to ad-hoc mode) changes IP address as a consequence of a handover from one access point to another. Candidates for mobility support in the IP network include MIP (L3), NeMo (L3), SCTP (L4), and SIP (L5). Possibly other mobility schemes may be used, like Layer 2 mobility functions for instance.

In HIDENETS, the following choices were made:

- Over the UMTS and GPRS access networks, the legacy mobility functions of these networks are used.
- When changing access network (i.e. vertical, or macro, handover), MIP support is assumed. This implies the presence of a home agent node in the infrastructure domain.

Keeping a mobile node connected to other entities in the infrastructure, even when mobile, is an important aspect for dependable service provisioning and contributes greatly to the quality of the user's experience. I.e., while moving from one access point to another, the ongoing sessions should be sustained and not broken. MIP provides transparency to the layers above it so sessions are not lost because of the change of IP address.

This service is relevant to all use cases that involve communications between cars and the infrastructure domain.

This component is intended for mobile nodes, i.e. the cars.

Handovers are not always started in case of mobility; handovers can also provide means to be 'always best connected' for improved QoS and such. Therefore, policies with clearly defined handover rules at different layers (L2, middleware...) should send triggers to the infrastructure mobility support service in the mobile node.

The infrastructure mobility support should notify the context repository manager so that other services that monitor QoS and other aspects of ongoing communications do not trigger their recovery and/or reconfiguration mechanisms during a handover phase. Once the handover has completed, these other services can start monitoring and reacting to faults.

G.2 In-stack monitoring & error detection

This service represents the monitoring and error detection carried out inside the protocol stack and makes it visible to other blocks. The information is pushed to the Network Context Manager which serves as a repository for all monitoring and measurement results. This service is always running in the HIDENETS context and gathers all types of information that can be directly, and locally, obtained from the protocols implemented in the node.

This service has not been investigated in detail in this project but we do believe that it will be important for future use and therefore several mechanisms will be required that select the available and relevant in-stack information from the protocols, that instrument the protocol stack to get access to the relevant parameters, and that allow to choose where the monitoring is implemented (either (1) directly within the protocols, which transmit the error detection information to the in-stack monitoring service that then just acts as a storage service for error detection information that serves the diagnostic manager or (2) the monitoring is done by the in-stack monitoring itself with specific mechanisms). One must also define a format for storing part of the information in the network context repository.

In the protocol stack, monitoring and error detection is performed at all layers. This includes, for example, link failure detection, buffer overflow detection at the IP layer, and packet loss detection at the transport layer. This information should be available for other blocks that need knowledge of system performance. The service provides this information by updating the Network Context Repository with the monitored values and detected errors.

This block is relevant for all use cases. The information is relevant for applications and use cases that require extensive information on the state and performance of the network and communications with other nodes. For fast and correct failure management this information is very important. Taking advantage of monitoring that is performed in the protocol stack reduces the need to monitor these parameters in separate blocks.

This block affects all entities.

At start-up time, input needed from the layers in the protocol stack in terms of which parameters should be monitored (e.g. packet delays, packet loss) and, for error detection, which associated value (e.g. timeout value at TCP layer).

At run-time, the communication adaptation manager informs the in-stack monitoring about the changes of timeout values and number of retransmissions in order to set the error detectors according to these regularly updated parameters.

The in-stack monitoring and error detection information is pushed to

- network context repository
- communication adaptation manager
- diagnostic manager

G.3 Performance monitoring

Avoiding performance degradation and possibly lost connection requires good knowledge of system performance and failures. This should not only be based on information from the protocol stack, but also other parameters of interest that may not easily be obtained only by looking at node-local information. This block implements monitoring of parameters that are not directly available from the protocol stack. This implies that the performance monitoring operates on demand, i.e. subscription by other blocks (e.g. QoS coverage manager, replication manager) for a given use-case, as opposed to the in-stack monitoring that performs regular and standard protocol monitoring, gathers the information and distributes it to the network context repository. The measurements carried out may also involve several nodes – possibly monitoring performance end-to-end.

This block handles performance monitoring carried out outside of the protocol stack. Performance monitoring is the collection of information on important system parameters that may be used to infer the performance of a system. Performance monitoring can be used to infer the performance of a link, a path in the network (like a LSP in MPLS, or the path packet travels between two nodes), a sub-graph (like the access network), overall network performance, or even higher layer information like TCP connection set-up time, information on packet reordering and application/service performance.

Performance monitoring is important for detecting failures, and also in choosing the best failover links, network paths, or higher layer components.

Performance monitoring can be carried out using passive or active measurements. With passive measurements existing traffic is monitored to derive the properties of interest. With active measurements, on the other hand, measurement traffic is injected into the system (network), and the response is measured. Some relevant parameters may already be available at a networked device. In a router, for example, there are statistics per interface on the number of packets sent and received - accessible via SNMP.

Performance monitoring can in principal be carried out on any protocol layer, but in this context we are primarily addressing monitoring for layer 1-4. Following is a list of some important parameters for protocol layers 1-4:

- Physical layer: Loss of signal, degradation in signal quality and bit errors.
- Link layer: Capacity, frame loss/retries and frame delay.
- Network layer: Throughput, packet loss, packet delay and jitter.
- Transport layer: Packet reordering and connection set-up time.

At higher layers, the result will depend on the components operating at that layer, as well as all the lower layer components involved.

This information is relevant for applications and use cases that require extensive information on the state and performance of the node, and possibly also involving several nodes. For fast and correct failure management this information is very important.

All.

Does not depend on input from any other building block. The block may use information from local MIBs (Management Information Base) as well as from local traffic monitoring and active traffic injection (with monitoring of the response).

This building block will provide input to the Network Context Repository, plus other blocks that have subscribed to information specific to the use-case(s) of interest.

G.4 GPRS/UMTS Radio Resource Management

In general, Radio Resource Management (RRM) can be said to be a set of functions used for optimal utilisation of air interface and hardware resources. Traditional mobile networks exercise strict, centralised control over the radio resource. RRM aims to guarantee the QoS agreed upon during session setup, reject sessions when resources can no longer be guaranteed, maintain the planned coverage area, and offer a high overall system capacity. The following functions are included:

- Admission control (AC) handles all new incoming traffic and checks if a new connection can be admitted to the system. AC is based on the resource and QoS requirements of the individual connections, and the available resources of the system. AC is also involved during handovers and connection modifications.
- Load control (LC) is used to avoid that the system load gets to too high with the consequence that the system breaks down. When the load exceeds a given threshold actions are taken to reduce the load.
- For GPRS/EDGE the timeslots can either be used for circuit switched traffic (voice) or packet switched traffic (streaming, data). Depending on specific vendor implementations timeslots may be reserved for circuit or packet traffic or used for both types of traffic (but not multiplexed together). Multiplexing of packet traffic in one timeslot (Packet Data Channel) may also be restricted by implementation and is a task for RRM.
- The Packet Scheduler (PS) divides the air transmission time between the active connections according to the resources that have been reserved for the individual connections.
- Handover Control (HC). When a connection is handed over from one access point to another is handled by the handover control.
- Power Control (PC) maintains the radio link quality and controls the power used.

G.5 Transport layer functions

The transport layer implements end-to-end functions between the sending and receiving host/server. In the car-to-car case, transport layer functionality is present in all cars/terminals, but for a particular connection, it is only in use in the sender and receiver and not in the intermediate nodes.

Typical transport layer functions may include connection establishment and release, reliable or unreliable data transfer, sequencing, time stamping, flow control, congestion control, and error control (detection and recovery). Connection establishment and sequence numbering enables reactive error control mechanisms, which means that the sender to keep track of which data has been correctly received or not, and if necessary ask for a retransmission. Proactive error control mechanisms add redundant information to enable correction of bit/packet errors (e.g. based on checksum). Flow control ensures that the sender does not overwhelm the sender with data, and congestion control ensures that the senders do not overload the network. Time stamping enables for instance correct play-out of media information at the receiver.

Transport protocols implement one or more of these functions, and the choice of transport protocol in effect determines the type of service that is offered to the session layer. Examples of Internet transport protocols that give quite different services are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).

A TCP connection is closely tied to the IP address of the sender and the receiver. Keeping the TCP connection is therefore difficult if the mobility solution requires a change of IP address, when switching interface if the currently used link breaks, or during multi-homing. SCTP is a relatively new transport protocol that allegedly offers higher availability and reliability than TCP or UDP by being able to support multi-homing and failover recovery. Furthermore, it handles multi-streaming which in some cases may increase the user-perceived performance, and it solves some of the limitations of TCP.

TCP performance over ad hoc wireless networks degrades with increasing path length (in terms of number of ad-hoc hops). One way to handle this is using Split-TCP, which splits a long TCP connection into a set of

short concatenated TCP connections. A subset of the intermediate ad-hoc nodes are selected as proxy nodes and act as terminating points for these short connections. A sender with Split-TCP keeps separate windows for end-to-end reliability and local congestion control to control the transmission of segments, effectively separating the TCP objectives of congestion control and reliability.

G.6 Naming service

The naming service is used to map a logical name to an IP address. It accommodates the use of URLs or other textual logical names in the network.

IP addresses, which computers use to communicate, are normally very inconvenient for human beings because they are difficult to remember. It is also inconvenient to program developers, because IP addresses might change.

By drawing parallels to the fixed Internet, the importance of a naming service is obvious: Without DNS, it is hard to imagine the wide deployment and popularity of the Internet that we witness today. The naming service is the ad hoc network's counterpart to DNS.

G.7 Resource/Service Discovery

A method is needed to find and locate resources and services available in the network. The discovery process resolves service names and descriptions to information that can be used to initiate the service, such as addresses and port numbers.

Finding services and resources can be difficult in an ad hoc network, although very useful in many scenarios.

G.8 Session control

The session control building block manages the sessions set up between applications in two or more different cars/terminals or between cars/terminals and a server.

SIP is a relevant candidate for controlling media sessions, i.e. creating, modifying and terminating sessions with two or more participants. SIP also includes functionality for presence, event notification and instant messaging services. Via SDP, SIP may determine the media capabilities of the target end points. Thus, sessions may be set up and modified to fit the highest level of media capabilities that can be supported by all end points. SIP can also be used to implement mobility management.

Other session layer functions like dialogue control, token management, and synchronization may also be relevant to the car-to-car setting depending on the use case chosen. In that case, such functions will be included at a later stage.

H ABC simulation parameters

This section lists the parameters used in the simulations, along with default values.

Simulation area	Square of 600 * 600 meters
Simulation time	1000 s
Initial period with no logging of statistics:	50 s
Propagation model	Free space model as implemented in J-Sim - see (*) below
Mobility model	Random Waypoint - as implemented in CanuMobiSim. Square of 600*600 meters, minimum speed is 2 m/s, maximum speed is 4 m/s, no idle periods, random initial positions.
Number of Access Points	9

Placement of Access Points	Grid, starting with first AP at X/Y = 100/100, and with 200 meters spacing in both X and Y direction, 3 in each row.
Access Point total capacity	Default is 10, but increased for some of the simulations
Access Point High priority capacity	Default is 5, but increased for some simulations
Number of Best Effort (BE) nodes	50
Number of Best Effort no congestion or High Priority nodes	From 0 to 45
AP unavailable period (**)	10 s
Beacon period	100 ms
Beacon timeout	320 ms (3.2 * beacon period)
Resource reservation refresh period	300 ms
Resource reservation refresh timeout at AP	960 ms (3.2 * resource reservation refresh period)
Resource reservation reply timeout at MT	1035 ms (3 * 1.15 * resource reservation refresh period) (***)
AP failure rate	0.003 (exponential distribution, but only one AP can be failing at a time)
AP minimum/maximum repair time	20 s / 50 s (uniform distribution)

(*) When approaching an AP, beacons are received by MTs at approximately 260 meters distance. When moving away from an AP, loss of beacons is detected by MT at approximately 270 meters.

(**) When an MT detects that an AP has become unavailable or receives a negative reply, the MT will not try that same AP again for this time period.

(***) The MT will change the status of the AP to unavailable if it does not receive a reply during this time period.

I Multi-Channel Multi-Radio State-of-the-Art

Multi-channel networks are an active research area. Many papers can be found on multi-channel wireless mesh networks. A selection of these papers will be discussed here to give an overview of the state of the art in multi-channel networks. In section I.1, multi-channel MAC protocols will be discussed, followed by a discussion on multi-channel routing in I.2 and finally other issues that were found will be listed in section I.3.

I.1 MAC protocols

There are many ways to divide multi-channel protocols in different categories, but it is most common to make a distinction in the number of interfaces that is required. Here, MAC protocols that use only one radio and MAC protocols that use multiple radios are discussed separately. Most MAC protocols in literature are implemented as a MAC extension layer, which allows the use of commercially available hardware (see Figure 9).

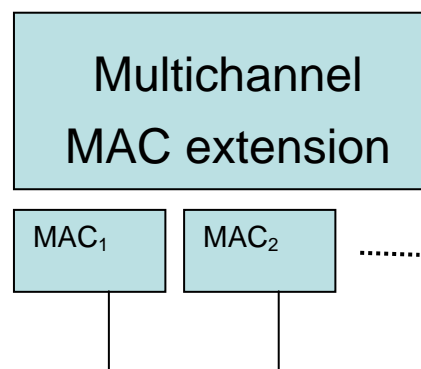


Figure 9: Multi-channel MAC extension layer

Single radio

With a single radio tight synchronization is required to ensure connectivity and allow communication between nearby nodes. Several solutions have been proposed in literature, which can be divided in two categories [44]:

Periodic rendezvous on a common channel

[41] [45] Nodes are always on the common control channel, except when transmitting data on other channels. 1st method: Sender sends RTS with possible channels and receiver chooses a channel and replies with CTS. 2nd method: Self-organizing Packets with channel information broadcasted on the common channel by every node. RTS/CTS on common channel.

[46] [48] Time synchronization in a distributed manner using beaconing as in IEEE 802.11 power saving mechanism. Channel negotiation during the ATIM window, every node is required to listen to the default channel in the ATIM window (besides being awake). All channels, including default channel, is used for data transfer outside the ATIM window.

Common hopping sequence

[49] Every node keeps switching channels in a certain order, which is called the hopping sequence. A distributed algorithm, that uses a broadcast packet containing the current hopping sequence, coordinates the channel switching when nodes wish to communicate. The number of slots is equal to the total number of channels, parity slots are used to introduce a pseudo-random factor in the hopping sequence to ensure connectivity and prevent network partitioning.

Multi radio

When multiple interfaces are available on a node, there are many ways to implement the MAC protocol, the main design choices will be presented here, specific multi-channel protocols can be found i.e. in [43] [44] [45] [50] [51] [52].

The number of interfaces

As mentioned in the introduction, to make maximum use of the available N channels, $2*N$ interfaces should be present within the interference range, because then on each channel, two nodes will be able to communicate with a radio fixed on that channel. In a sparse network, i.e. the simple case of only two nodes within communication range, each node will need N interfaces to maximize the network capacity. But when there are 2 pairs of 2 communicating nodes within range, all channels can be used with only $N/2$ interfaces per node. So when capacity is the criterion, the optimal number of interface per node depends on the node density and the traffic pattern.

But there are other reasons to change the number of interfaces:

- More interfaces makes it easier to realize a connected network, i.e. by using a common channel or by using enough radios to guarantee connectivity (2 radio with 3 channels, 3 radios with 5 channels, etc.) or by using a channel assignment algorithm.
- The per-node-throughput increases with multiple interfaces as nodes are capable of receiving and transmitting on multiple channels in parallel.
- Synchronization requirements are less strict with multiple radios
- Less channel switching delay

There are also disadvantages to using multiple radios:

- Higher power consumption
- Costs of the hardware

Static vs. Dynamic Channel Assignment

Interface assignment is the assignment of a channel to an interface. The channel assignment algorithm that performs the assignment will be discussed in section 1.4.2. But on a system level, a choice can be made between static and dynamic channel assignment.

- **Static assignment**
Each interface is assigned to a channel either permanently or for a “long” interval, where “long” is typically defined relative to the interface switching delay. Static channel assignments are especially useful when the channel switching delay of the hardware is rather large.
- **Dynamic assignment**
Each interface can frequently switch between channels, which allows effective use of more channels with less interfaces, but strong coordination between nodes is necessary.
- **Hybrid assignment**
Of course a combination of both approaches is possible, i.e. when using one static interface on a dedicated control channel and a dynamic interface for data channels. Hybrid algorithms allow simplified coordination algorithms while still having the flexibility of dynamic assignments.

Using a Common Channel

Many protocols use a common frequency channel, either with a dedicated radio or with periodic rendezvous. There are different reasons and uses for such a common channel, which are:

- **Connectivity**
With a common channel for all nodes, the network is always connected in the sense that all nodes are

always reachable on this channel (disregarding congestion).

- Control
In many protocols, a common channel is dedicated for control messages, i.e. for reservations of other channels. In most cases no data traffic is allowed on the control channel, when periodic rendezvous is used, the channel is used as a data channel outside of the reservation window.
- Broadcast support
As mentioned in the introduction, broadcast is more difficult compared to a single channel system. One way to implement broadcast support is by using a common channel for broadcast messages.

So using a common channel can have multiple advantages, but makes limited use of multi-channel, while the common channel has single channel characteristics with respect to interference and congestion.

Channel assignment

Part of the MAC protocol is the channel assignment algorithm [53] [54] [55] [56]. The goal is to assign a channel to each interface, while ensuring connectivity and maximizing capacity. This can also be described as a colouring problem, where all links in the network need to be assigned a channel.

The channel assignment strategy can be either static, dynamic or hybrid. Other design issues of the channel assignment algorithm are:

Several metrics can be used to choose a channel:

- idle channel
- signal power observed by channel
- received signal power
- channel load

Also a distinction can be made between:

- distributed algorithms
- centralized algorithms

I.2 Multi-channel routing

The benefits of using multiple channels can be further exploited on the network layer. A routing protocol that uses multiple channels can increase the performance and capacity benefits and additionally offers delay improvements. Relevant proposals in literature will be presented here.

[43] The Multi-Channel Routing algorithm (MCR) by Stigge is based on an architecture with both fixed and switchable interfaces. Broadcasts are used to inform neighbours of the fixed channel. The routing is based on a routing metric that is the weighted sum of the hop count, the diversity cost and the switching cost. Route discovery and maintenance is similar to DSR. Simulations show an increased throughput and decreased end-to-end delay compared to DSR (with an interface switching delay of 100us).

[51] [52] Kyanasur and Vaidya observe that common routing protocols can be used on a multi-channel architecture (since it transparently manages the channels and interfaces), but the channel diversity can be fully utilized by using multi-channel routing algorithms for multi-hop communications. A path metric, including the switching delay, is used in a DSR-like routing algorithm instead of the shortest-path routing that is common in DSR and AODV.

[68] Alicherry et al. optimize the throughput in multi-hop infrastructure wireless mesh network by combining the channel assignment and routing in a joint, centralized channel assignment. The algorithms that are used approximate a solution for the channel assignment problem, which is NP-hard.

[69] Raniwali et al. use a load-aware channel assignment to improve the routing that can be combined with different routing algorithms. It is designed for a multi-interface wireless networks that serve as backbones to connect to wired infrastructure. The proposed channel assignment algorithm is a centralized algorithm. In [70] a modified and distributed algorithm is used, where the load-awareness is based on local information.

I.3 Other issues

- Although there is a lot of research on multi-channel systems, there are few publications on using multi-channel in VANETs. The information that is available is related to standard development in car-to-car communication.
- Many multi-channel protocols require a certain amount of synchronization between nodes. A common assumption in VANET literature is the availability of GPS information. Besides the positional information, GPS could be used for synchronization, i.e. three-step synchronization [57]
 - o physical layer for data-packet synchronization
 - o beaconing for intra-group synchronization
 - o GPS for inter-group synchronization
- To increase scalability, topology control that adjusts dynamic transmission power can be used to reduce the contention loss in dense networks [57].
- When analyzing literature it is important to distinguish whether the solution can be implemented with commercially available IEEE 802.11 hardware or if changes in the hardware are necessary. Many protocols that use available hardware are implemented as a MAC extension layer (see Figure 9).

References

- [1] A. Autenrieth, A. Kirstädter, “*Fault Tolerance and Resilience Issues in IP-Based Networks*”, Second International Workshop on the Design of Reliable Communication Networks (DRCN2000), Munich, Germany, April 9-12, 2000.
- [2] A. Autenrieth and A. Kirstädter, “*Engineering end-to-end IP resilience using resilience-differentiated QoS*”, IEEE Communications Magazine, vol. 40, no. 1, pp. 50 - 57, Jan 2002
- [3] A. Autenrieth, “*Recovery Time Analysis of Differentiated Resilience in MPLS*,” Proc. DRCN 2003, pp. 333-340, Banff, Canada, Oct. 2003.
- [4] M. Radimirsch et al, “*Use case scenarios and preliminary reference model*”, EU FP6 IST project HIDENETS, deliverable D1.1. September 2006.
- [5] M. Düser, J. Götz, I-E. Svinnset, J. Tapolcai, “*Final report on Traffic Engineering and resilience strategies for NOBEL solutions. Part A: Overview of the Evolution and Convergence of Traffic Engineering and Resilience in future Transport Networks*”. EU FP6 IST project NOBEL, deliverable D27, part A. October 2005.
- [6] E. Gustavson & A. Jonsson, “*Always Best Connected*”, IEEE Wireless Communications, pp. 49 - 55, feb 2003.
- [7] IEEE 802.21 “*Media Independent Handover*”, DCN: 21-05-0241-00-0000, Harmonized MIH proposal draft, IEEE, mar 2005.
- [8] P. E. Engelstad, G. Egeland, S. S. Bygds, R. Geers, and T. Urnes, “*Middleware Supporting Adaptive Services in On Demand Ad Hoc Networks*”, In Proceedings of 9th International Conference on Intelligence in service delivery Networks (ICIN'2004), Bordeaux (France), Oct 2004
- [9] P. Magnusson et al, “*Radio resource management distribution in a beyond 3G multi-radio access architecture*”, In IEEE GLOBECOM '04, vol. 6, 2004, pp. 3472 - 3477
- [10] Mobile IPv6 (MIP6), <http://www.ietf.org/html.charters/mip6-charter>, (Last Visited: June 12 2005, working Group of the Internet Engineering Task Force (IETF).
- [11] R. Koodli, “*Fast Handovers for Mobile IPv6 (work-in-progress)*”, IETF Internet Draft, oct 2004
- [12] P. McCann, “*Mobile IPv6 Fast Handovers for 802.11 Networks (work-in-progress)*”, IETF Internet Draft, feb 2005
- [13] T. Cinkler, P. Demeester, and A. Jajszczyk, “*Resilience in communication networks*”. IEEE Communications Magazine, vol. 40, no. 1, pp. 30 - 32, jan 2002
- [14] *HIDENETS Node Software Architecture*. HIDENETS WP2/3 internal document
- [15] Open Broadband Access Networks (OBAN). <http://www.telenor.no/fou/prosjekter/oban/> IST FP6 project
- [16] AMBIENT Networks. <http://www.ambient-networks.org/> IST FP6 project.
- [17] 3GPP TS 23002-710: “*Network Architecture*“, V7.1.0, March 2006
- [18] J. Moy, “*OSPF Version 2*“, IETF RFC 2328 (STD 54), April 1998.
- [19] R. W. Callon, “*Use of OSI IS-IS for routing in TCP/IP and dual environments*“, IETF RFC 1195, December 1990
- [20] Y. A. Rekhter, “*Border Gateway Protocol 4 (BGP-4)*“, IETF RFC 4271, January 2006.
- [21] T. Clausen, P. Jacquet, “*Optimized Link State Routing Protocol (OLSR)*”, IETF RFC 3626, October 2003

- [22] P. Spagnolo et al., “*OSPFv2 Wireless Interface Type*”, <draft-spagnolo-manet-ospf-wireless-interface-01>, May 2004
- [23] M. Chandra, “*Extensions to OSPF to Support Mobile Ad Hoc Networking*”, Internet draft ‘draft-chandra-ospf-manet-ext-02’, October 2004.
- [24] C. Perkins, E. Belding-Royer, S. Das, “*Ad hoc On-Demand Distance Vector (AODV) Routing*”, IETF RFC 3561, July 2003
- [25] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne, “*Fast IP network recovery using multiple routing configurations*”, in Proceedings of INFOCOM, Apr. 2006
- [26] A. F. Hansen, O. Lysne, T. Cicic and S. Gjessing, “*Fast Proactive Recovery for Concurrent Failures*”, submitted for ICC 2007
- [27] 3GPP TS 23.107: “*Quality of Service (QoS) concept and architecture*”, V6.3.0, June 2005.
- [28] 3GPP TS 23.228, “*IP Multimedia Subsystem (IMS)*”, V6.5.0, March 2004
- [29] P. Kim, W. Boehm, “*Support for Real-Time Applications in Future Mobile Networks: the IMS Approach*”, in the proceedings of WPMC’03, Oct. 2003.
- [30] J. Rosenberg et al., “*SIP: Session Initiation Protocol*”, IETF RFC 3261, June 2002
- [31] <http://media.csee.ltu.se/publications/2003/ahlund03multihoming.pdf>
- [32] C. Perkins, “*Mobile IP*”, IEEE Communications Magazine, p 66-82, May 2002
- [33] http://www.cs.ou.edu/~netlab/Pub/sctp_tutorial.pdf
- [34] M. Tuexen, et al., “*Architecture for Reliable Server Pooling*”, <draft-ietf-rserpool-arch-10.txt>, July, 2005
- [35] Q. Xie, R. R. Stewart, “*Endpoint Name Resolution Protocol*”, <draft-ietf-rserpool-enrp-01.txt>, November 2001
- [36] R. R. Stewart, Q. Xie, “*Aggregate Server Access Protocol (ASAP)*”, <draft-ietf-rserpool-asap-01.txt>, November 2001
- [37] Q. Wang and M.A. Abu-Rgheff, “*Cross-Layer Signalling for Next-Generation Wireless Systems*”, In: Proc. IEEE Wireless Communication and Network Conference, New Orleans, LA, Mar. 2003
- [38] R. Braden, T. Faber and M. Handley, “*From Protocol Stack to Protocol Heap - Role-Based Architecture*,” In: Proc. Hot Topics in Net., Princeton, NJ, Mar. 2002
- [39] HIP Working Group, IETF, <http://www.ietf.org/html.charters/hip-charter.html>
- [40] Mobike Working Group, <http://www.vpnc.org/ietf-mobike/>
- [41] J. Li, Z. J. Haas, M. Sheng, “*Capacity Evaluation of Multi-Channel Multi-Hop Ad Hoc Networks*”, In: Proceedings of ICPWC 2002, New Delhi, India
- [42] P. Kyasanur, N. H. Vaidya, “*Capacity of Multi-Channel Wireless Networks: Impact of Number of Channels and Interfaces*”, Technical Report, March 2005
- [43] M. Stigge, “*Routing in Multi-Channel Multi-Interface Ad Hoc Wireless Networks*”, Projektseminar: Self-Organizing Middleware for Mobile Systems, Humboldt-Universität zu Berlin, Institut für Informatik, 2005
- [44] P. Kyasanur, J. So, C. Chreddi, N.H. Vaidya, “*Multi-Channel Mesh Networks: Challenges and Protocols*”, (invited paper) in IEEE Wireless Communications, April 2006

- [45] J. Chen and Y- D. Chen, "AMNP: Ad Hoc Multichannel Negotiation Protocol for Multihop Mobile Wireless Networks", ICC 2004 - IEEE International Conference on Communications, no. 1, June 2004 pp. 3607-3612
- [46] J. So, N. Vaidya, "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver", MobiHic'04, May 2004, Roppongi, Japan
- [47] J. Li, Z. J. Haas, M. Sheng, Y. Chen, "Performance Evaluation of Modified IEEE 802.11 MAC For Multi-Channel Multi-Hop Ad Hoc Network", 17th International Conference on Advanced Information Networking and Applications (AINA'03), March 27-29, 2003, Xi'an, China
- [48] J. So, N. H. Vaidya, "A Multi-channel MAC Protocol for Ad Hoc Wireless Networks", Technical Report, Jan. 2003
- [49] P. Bahl, R. Chandra, J. Dunagan, "SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks", In: Proc. 10th Annual Int. Conf. on Mobile Computing and Networking (MobiCom), Philadelphia, September 2004
- [50] J. Wang, Y. Fang, D. Wu, "A Power-Saving Multi-radio Multi-channel MAC Protocol for Wireless Local Area Networks" to appear in IEEE International Conference on Computer Communication (INFOCOM'06), Barcelona, April 2006
- [51] P. Kyasanur, N. H. Vaidya, "Routing and Link-layer Protocols for Multi-Channel Multi-Interface Ad Hoc Wireless Networks", Mobile Computing and Communications Review, Vol.1, No.2
- [52] P. Kyasanur, N. H. Vaidya, "Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks [WCNC]", WCNC 2005
- [53] A. P. Subramanian, R. Krishnan, S. R. Das, H. Gupta, "Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks"
- [54] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, W. Arbaugh, "Efficient Strategies for Channel Management in Wireless LANs", Computer Science Technical Report CS-TR-4729, UMIACS Technical Report UMIACS-TR-2005-36, June 2005
- [55] K. K. Leung, B. J. Kim, "Frequency Assignment for Multi-Cell IEEE 802.11 Wireless Networks", 58th vehicular technology conference; VTC 2003 Fall, Orlando, FL, October 2003, IEEE, 2003, Pages: 1422 - 1426, ISBN: 0-7803-7954-3
- [56] C-Y. Chang, P- C. Huang, C- T Chang, Y- S Chen, "Dynamic Channel Assignment and Reassignment for Exploiting Channel Reuse Opportunities in Ad Hoc Wireless Networks"
- [57] H. J. Reuterma, Y. Zang, L. Stibor and G. R. Hiertz, "Vehicular Wireless Media Network (VWMN) – A distributed broadband MAC for inter-vehicle communications", VANET '05, September 2005
- [58] L. Wischhof, A. Ebner, H. Rohling, M. Lott and R. Halfmann, "SOTIS - A Self-Organizing Traffic Information System", In: Proceedings of the 57th IEEE Vehicular Technology Conference (VTC '03 Spring), Jeju, Korea, 2003
- [59] H. Füßler, "Thoughts on a Protocol Architecture for Vehicular Ad-Hoc Networks", Presentation, 2nd Workshop on Intelligent Transportation (WIT 2005), Hamburg, Germany, March 2005
- [60] M. Torrent-Moreno, M. Killat, H. Hartenstein, "The challenges of robust inter-vehicle communications" Vehicular Technology Conference, 2005
- [61] T. K. Mak, K. P. Laberteaux and R. Sengupta, "A Multi-Channel VANET Providing Concurrent Safety and Commercial Services", California PATH program, march 2005
- [62] CALM website (<http://www.calm.hu/>)

- [63] ISO TC204/WG16 – CALM (<http://www.tc204wg16.de/>)
- [64] Lee Armstrong's DSRC site (<http://www.leearmstrong.com/DSRC/DSRCHomeset.htm>)
- [65] CVIS project site <http://www.cvisproject.org/>
- [66] IEEE 802.11p draft standard
- [67] IEEE 1609.4 draft standard
- [68] M. Alicherry, R. Bhatia, L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks", Mobicom '05, 2005
- [69] A. Raniwala, K. Gopalan, T. Chiueh, "Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks"
- [70] A. Raniwala, T. Chiueh, "IEEE 802.11-based Multi-channel Mesh Network", IEEE Infocom, March 2005
- [71] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks", in ACM Mobicom, 2004.
- [72] NEMO, <http://www.ietf.org/html.charters/nemo-charter.html>
- [73] taps.rice.edu/presentations/rice-only/9-10-robinson.ppt
- [74] AODV @ IETF, <http://moment.cs.ucsb.edu/aodv-ietf/>
- [75] OLSR, <http://www.olsr.org/>
- [76] R. Wakikawa, "Gateway management for vehicle-to-vehicle communication", presentation at V2VCOM 2006
<http://www.v2vcom.org/V2VCOM%202005/V2VCOM%20Wakikawa.pdf#search=%22gateway%20discovery%20vanet%22>
- [77] Korkmaz, Ekici and Ozguner, "A New High Throughput Internet Access Protocol for Vehicular Networks", VANET '05, 2005.
- [78] Baldessari, Festag, Matos, Santos and Aguiar, "Flexible Connectivity Management in Vehicular Communication Networks", Proceedings of 3rd International Workshop on Intelligent Transportation, March 2006
- [79] P. Ruiz, "Adaptive Gateway Discovery Mechanisms to Enhance Internet Connectivity", Ad Hoc & Sensor Wireless Networks, Vol. 1. pp. 159-177, 2005
- [80] M. Ghassemian, V. Friderikos and A. Aghvami, "On the Scalability of Internet Gateway Discovery Algorithms for Ad hoc Networks", IWWAN2005, 2005
- [81] I. van Beijnum, "A Look at Multihoming and BGP", 2002
<http://www.oreillynet.com/pub/a/network/2002/08/12/multihoming.html>
- [82] Shim6, IETF working group
<http://www.ietf.org/html.charters/shim6-charter.html>
- [83] S. Fu and M. Atiquzzaman, "SCTP: State of the Art in Research, Products, and Technical Challenges", IEEE Communications Magazine, pp. 64-76 April 2004
- [84] C. Ahlund and A. Zaslavsky, "Multihoming with Mobile IP", 2003
<http://media.csee.ltu.se/publications/2003/ahlund03multihoming.pdf>
- [85] P. Eronen, "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", IETF RFC 4555, 2006, <http://www.rfc-archive.org/getrfc.php?rfc=4555>
- [86] A summary of proposals to support multihoming for IPv6:
<http://www.potaroo.net/drafts/draft-huston-multi6-proposals-00.html>

- [87] M. Ghassemian, V. Friderikos, P. Hofmann and C. Prehofer, “*An Optimised Gateway Selection Mechanism for Wireless Ad Hoc Networks Connected to the Internet*”, IEEE VTC 2006
- [88] I. Stojmenovic, M. Seddigh, J. Zunic, “*Dominating sets and neighbour elimination based broadcasting algorithms in wireless networks*”, IEEE Transaction on Parallel and Distributed Systems, Vol. 13, No. 1, January 2002
- [89] J. Wu J and H. Li, “*On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks*”, Proc. of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Aug. 1999, 7-14
- [90] ITU-T Recommendation Y.1541: “*Network performance objectives for IP-based services*”, Geneva October 2005
- [91] A. Casimiro et al, “*Resilient architecture (preliminary version)*”, EU FP6 IST project HIDENETS, deliverable D2.1.1 December 2006
- [92] A. Bondavalli et al, “*Evaluation methodologies, techniques and tools*”, EU FP6 IST project HIDENETS, deliverable D4.1 December 2006
- [93] GST project site <http://www.gstforum.org/>
- [94] IETF home page <http://www.ietf.org/>
- [95] B. Aboba et. al., “*Extensible Authentication Protocol (EAP)*”, IETF RFC 3748, June 2004.
- [96] D. Newman, “*Benchmarking Terminology for Firewall Performance*”, IETF RFC 2647, August 1999
- [97] ITU-T Recommendation E.800: “*Quality of Service, Network Management and Network Performance*”, Geneva August 1994
- [98] B Aboba et al., “*Link-local Multicast Name Resolution (LLMNR)*”, < draft-ietf-dnsext-mdns-47.txt >, August 2006
- [99] M. Gerla et. al., “*Fisheye State Routing Protocol (FSR) for Ad Hoc Networks*”, IETF Internet Draft, June 2002.
- [100] M. Gerla et. al., “*Landmark Routing Protocol (LANMAR) for Large Scale Ad Hoc Networks*”, IETF Internet Draft, June 2002.
- [101] Z. J Haas et. al., “*The Zone Routing Protocol (ZRP) for Ad Hoc Networks*”, IETF Internet Draft, July 2002.
- [102] Y.- B. Ko and N. H. Vaidya, “*Location-aided routing in mobile ad hoc networks*”, Wireless Networks 6, 2000. pp. 307–321
- [103] P. E. Engelstad, A. Tonnesen, A. Hafslund, G. Egeland, “*Internet Connectivity for Multi-Homed Proactive Ad Hoc Networks*”, Proceedings of IEEE International Conference on Communication (ICC'2004), Paris, June 20-24, 2004.
- [104] P. E. Engelstad and G. Egeland, “*NAT-Based Internet Connectivity for On-Demand Ad Hoc Networks*”, Proceedings of Wireless On-Demand Ad Hoc Networks (WONS'2004), Madonna di Campiglio, Italy, January 19-23, 2004. (Lecture Notes on Computer Science LNCS2928, Springer 2004, pp. 342-356)
- [105] P. E. Engelstad, G. Egeland, T. V. Do, “*Investigating Race-Conditions in Multi-homed On-Demand Ad Hoc Networks*”, Proceedings of IEEE Wireless Networking and Communication Conference (WCNC'2004), Atlanta, Georgia, March 21-25, 2004
- [106] ETSI ES 282 003: “*Resource and Admission Control Sub-system (RACS); Functional Architecture*”, Release 2, September 2006.