ETSI GR NGP 003 V1.1.1 (2017-03)

NGP Next Generation Protocol;
Packet Routing Technologies

Disclaimer

The present document has been produced and approved by the Next Generation Protocols (NGP) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference

DGR/NGP-003

Keywords

flexilink, M2CNP, Next Generation Protocol,
RINA

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Next Generation Protocols (NGP).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

Three technologies are described in the present document.

RINA embodies a theory which is informally known as the Inter-Process Communication (IPC) model. It is structured around a single type of layer - called Distributed IPC Facility or DIF - that repeats as many times as needed. In RINA, all layers are distributed applications that provide the same service (communication flows between distributed applications) and have the same internal structure, divided into data transfer (delimiting, addressing, sequencing, relaying, multiplexing, lifetime termination, error check, encryption), data transfer control (flow and retransmission control), and layer management (enrolment, routing, flow allocation, namespace management, resource allocation, security management).

Flexilink is designed for implementation in 21st century digital systems, in which packet forwarding is implemented in hardware and memory is much more plentiful than when Internet Protocol was developed. The information needed to route packets is carried separately from the packets themselves; this reduces the size of the packet header by an order of magnitude, simplifies the forwarding hardware, and allows different addressing mechanisms to be used without changing the packet format. It supports ultra-low latency live streams; these are needed for some of the new services that 5G is to support, and also provide a better service for audio and video. They can also be used for file transfer, eliminating the need for the kind of empirical throughput testing that is a feature of TCP.

M2CNP envisages a packet based routed protocol architecture with the ability to embed protocol control messaging to provide basic protocol management functions for: security, context-awareness, transmission management, and mobility. Applications and/or services, running at access network connected devices, communicate using IPC interfaces towards the M2CNP communications network. Devices may be connected via one or more access technologies at a time and are capable of mobility from one access point or Temporary Access Points Group (for multiple access) to another. The M2CNP network consists of M2CNP Packet Processing Entities, which are M2CNP routing entities that are selectively enabled with various protocol management capabilities of M2CNP and may be deployed in terms of scope in a similar manner to CE, PE, P scope routers as commonly understood in the legacy IP world.

Introduction

ETSI ISG NGP is tasked with reviewing networking technologies, architectures, and protocols for the next generation of communication systems.

The present document describes some technologies of which ISG NGP is aware, which could be evaluated against the requirements listed in ETSI GS NGP 001 [i.1] (Scenarios) and 3GPP TR 23.799 [i.3].

Ideally, ISG NGP would issue a Call for Technology and wait for responses before drafting the present document. However, new radio interfaces are now being developed for 5G, and if something other than TCP/IP is to be used with them the developers of the radio technology need to have an indication, early in the process, of the kind of shape the new protocols might have. Therefore, a first version of the present document is being produced covering technologies that have been researched by the current members of ISP NGP. It is intended that further versions will be produced, containing additional architectures.

1 Scope

The present document describes packet routing technologies that might be used in 5G radio networks and in the core of future mobile networks, and would also be suitable for use in the Internet. The description of each technology includes:

- overview of routing approach;
- key fields in user plane packets;
- procedures for setting up routes, etc.;
- support for mobility;
- support for security;
- addressing, including scalability issues.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS NGP 001 (V1.1.1): "Next Generation Protocol (NGP); Scenario Definitions".

[i.2] ISO/IEC 62379-5-2:2014: "Common control interface for networked digital audio and video products -- Part 5-2: Transmission over networks -- Signalling".

[i.3] 3GPP TR 23.799: "Study on Architecture for Next Generation System".

[i.4] J. Day: "Patterns in Network Architecture: A return to fundamentals". Prentice Hall, 2008.

[i.5] J. Day, I. Matta, and K. Mattar. 2008: "Networking is IPC: a guiding principle to a better Internet". In Proceedings of the 2008 ACM CoNEXT Conference (CoNEXT '08).

[i.6] J. Day, E. Grasa: "About layers: more or less".

NOTE: PSOC Tutorial, available online at http://pouzinsociety.org.

[i.7] J. Day: "How naming, addressing and routing are supposed to work".

NOTE: PSOC Tutorial, available online at http://pouzinsociety.org.

[i.8] G. Gursun, I. Matta, and K. Mattar: "On the Performance and Robustness of Managing Reliable Transport Connections". In Proceedings of the 8th International Workshop on Protocols for Future, Large-Scale and Diverse Network Transports (PFLDNeT), Lancester, PA, November 2010.

[i.9] Boddapati, G.; Day, J.; Matta, I.; Chitkushev, L.: "Assessing the security of a clean-slate Internet architecture". Network Protocols (ICNP), 2012 20th IEEE International Conference on.

[i.10] V. Maffione, F. Salvestrini, E. Grasa, L. Bergesio, M. Tarzan: "A Software Development Kit to exploit RINA programmability". IEEE ICC 2016, Next Generation Networking and Internet Symposium.

[i.11] J. Day, E. Grasa: "Mobility made simple".

NOTE: PSOC Tutorial, available online at http://pouzinsociety.org.

[i.12] J. Day, E. Trouva, E. Grasa, P. Phelan, M.P. de Leon, S. Bunch, I. Matta, L.T. Chitkushev, L. Pouzin: "Bounding the router table size in an ISP network using RINA". Network of the Future (NOF), 2011.

[i.13] V. Ishakian, J. Akinwumi, F. Esposito, and I. Matta: "On supporting mobility and multihoming in recursive internet architectures". Comput. Commun. 35, 13 (July 2012), 1561-1573.

[i.14] J. Small: "Threat analysis of Recursive InterNetwork Architecture Distributed IPC Facilities". BU Technical Report, 2011.

[i.15] E. Grasa, O. Rysavy, O. Lichtner, H. Asgari, J. Day, L. Chitkushev: "From protecting protocols to protecting layers: designing, implementing and experimenting with security policies in RINA". IEEE ICC 2016, Communications and Information Systems Security Symposium.

[i.16] J. Small: "Patterns in Network Security: An analysis of architectural complexity in securing Recursive Inter-Network Architecture Networks". Master Thesis, 2012.

[i.17] S. León, J. Perelló, D. Careglio, E. Grasa, D. Lopez, Pedro A. Aranda: "Benefits of Programmable Topological Routing Policies in RINA-enabled Large-scale Datacentres". IEEE Globecom 2016, NGN Symposium, December 2016.

[i.18] P. Teymoori, M. Welzl, S. Gjessing, E. Grasa, R. Riggio, K. Rausch, D. Siracussa: "Congestion control in the Recursive Internetwork Architecture (RINA)". IEEE ICC 2016, Next Generation Networking and Internet Symposium.

[i.19] R. Watson: "Timer-based mechanism in reliable transport protocol connection management". Computer Networks, 5:47–56, 1981.

[i.20] AES67: "AES standard for audio applications of networks - High-performance streaming audio-over-IP interoperability"; Audio Engineering Society, New York, NY., US.

[i.21] EBU Doc Tech 3326: "Audio contribution over IP Requirements for Interoperability"; European Broadcasting Union, Geneva, CH.

[i.22] SMPTE 2022: "Digital video on IP networks"; Society of Motion Picture and Television Engineers, White Plains, NY., US.

[i.23] ISO/IEC 7498-1: "Information technology - Open Systems Interconnection - Basic Reference Model: Part 1: The Basic Model". International Standards Organization, Geneva, CH.

[i.24] C. Ge et al.: "QoE-Driven DASH Video Caching and Adaptation at 5G Mobile Edge", in Proceedings of the 3rd ACM Conference on Information-Centric Networking (ACM-ICN '16), pp. 237-242, ACM, Kyoto, Japan, 2016.

[i.25] P. Qian et al.: "Enabling Context-aware HTTP with Mobile Edge Hint", to be published in proceedings of the 14th Annual IEEE Consumer Communications & Networking Conference (January 2017, Las Vegas, USA).

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

cluster: M2CNP aware logical grouping of one or more access points, PPEs and logical protocol control entities

endpoint: service or application or PPE that wishes to communicate with other M2CNP endpoints

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AA	Access Agent
ACIP	Audio Contribution over IP
ACM	Association for Computing Machinery
ACRA	Adaptive Code based RACH Access
ADDR	ADDRess
AEP	Access EndPoint (M2CNP)
AP	Access Point ID (M2CNP)
API	Application Program Interface
AP-ID	Access Point IDentifier
ARP	Address Resolution Protocol
ASN.1	Abstract Syntax Notation 1
ATM	Asynchronous Transfer Mode
AV	Audio-Video
BBC	British Broadcasting Corporation
CC	Cluster Controller (M2CNP)
CCNC	Consumer Communications & Networking Conference
CCRD	Cluster Content Routing Database
CDAP	Common Distributed Application Protocol
CE	Customer Equipment
CEPID	Connection EndPoint IDentifier
Cl-ID	Cluster ID (M2CNP)
CM	Cluster Member (M2CNP)
CRC	Cyclic Redundancy Check
DAF	Distributed Application Facility
DASH	Dynamic Adaptive Streaming over Http
DIF	Distributed IPC Facility
DNS	Directory Name Server
DST	DeSTination
DTCP	Data Transfer Control Protocol
DTP	Data Transfer Protocol
EBU	European Broadcasting Union
ECN	Explicit Congestion Notification
EFCP	Error and Flow Control Protocol
EN	Enabling
EP	EndPoint
ETE	End To End
FAI	Flow Allocator-Instance
FDC	Flat Distributed Cloud
FMC	Fixed Mobile Convergence
FPGA	Field-Programmable Gate Array
GPS	Global Positioning System
HD	High Definition [television]
HTTP	HyperText Transfer Protocol
IE	Information Element
IEC	International Electrotechnical Commission

IETF	Internet Engineering Task Force
IMPL	IMPLemented
IMS	IP Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IPC	Inter-Process Communication
IPCP	Inter-Process Communication Process
ISO	International Organization for Standardization
ISP	Internet Service Provider
IT	Information Technology
LISP	Locator-Identifier Separation Protocol
LV	Length and Value
M2CNP	Multi-Access, Mobility Aware and Context Aware Networking protocol
MAC	Media Access Control
MBB	Mobile BroadBand
MDP	MetaData Protocol
MEC	Mobile Edge Computing
MH	Mobile Host
MPLS	Multi-Protocol Label Switching
MS	Message Set
MTU	Maximum Transmission Unit
NAS	Non-Access Stratum
NEP	Network EndPoint (M2CNP)
NFV	Network Function Virtualisation
NGP	Next Generation Protocols
NM-DMS	Network Management - Distributed Management System
NR	New Radio
NR-eNB	New Radio eNB (base station of 3GPP New Radio RAN)
NS	Network Service
NSLD	Network Subscriber Location Database
NSM	Name Space Manager
OTS	Off The Shelf
OTT	Over The Top
PC	Personal Computer
PDCP	Packet Data Convergence Protocol
PDU	Protocol Data Unit
PE	Provider Equipment
PLMN	Public Land Mobile Network
PPE	Packet Processing Entities (M2CNP)
PTP	Precision Time Protocol
QoE	Quality of Experience
QoS	Quality of Service
RA	Resource Allocator
RACH	Random Access CHannel
RAN	Radio Access Network
RIB	Resource Information Base
RINA	Recursive InterNetwork Architecture
RLC	Radio Link Control
RMT	Relaying and Multiplexing Task
RRC	Radio Resource Control
RTP	Real Time Protocol
SDN	Software-Defined Network
SDO	Standards Development Organization
SDP	Session Description Protocol
SDU	Service Data Unit
SIM	SIMulation
SIP	Session Initiation Protocol
SMPTE	Society of Motion Picture and Television Engineers
SoC	System on Chip
SRC	SouRCe
TAPG	Temporary Access Points Group
TCP	Transmission Control Protocol

TLV	Tag Length Value
UDP	Unacknowledged Datagram Protocol
UE	User Equipment
UP	User Plane
VAL	VALidation
VERS	VERSion
VPN	Virtual Private Network
WDM	Wavelength Division Multiplexing

4 RINA (Recursive InterNetwork Architecture)

4.1 Overview

4.1.0 Introduction to RINA

The Recursive InterNetwork Architecture (RINA) is a computer network architecture that unifies distributed computing and telecommunications. RINA's fundamental principle is that computer networking is just Inter-Process Communication or IPC. RINA reconstructs the overall structure of the Internet, forming a model that comprises a single repeating layer, the DIF (Distributed IPC Facility), which is the minimal set of components required to allow distributed IPC between application processes. RINA inherently supports mobility, multi-homing, and Quality of Service without the need for extra mechanisms, provides a secure and programmable environment, motivates for a more competitive marketplace, and allows for a seamless adoption.

RINA is the result of an effort that tries to work out the general principles in computer networking that apply to everything. RINA is the specific architecture, implementation, testing platform and ultimately deployment of the theory. This theory is informally known as the Inter-Process Communication "IPC model" ([i.4] and [i.5]) although it also deals with concepts and results that are generic for any distributed application and not just for networking. RINA is structured around a single type of layer - called Distributed IPC Facility or DIF - that repeats as many times as needed by the network designer (figure 4.1). In RINA all layers are distributed applications that provide the same service (communication flows between distributed applications) and have the same internal structure. The instantiation of a layer in a computing system is an application process called IPC Process (IPCP). All IPCPs have the same functions, divided into data transfer (delimiting, addressing, sequencing, relaying, multiplexing, lifetime termination, error check, encryption), data transfer control (flow and retransmission control), and layer management (enrolment, routing, flow allocation, namespace management, resource allocation, security management). The functions of an IPCP are programmable via policies, so that each DIF can adapt to its operational environment and to different application requirements.

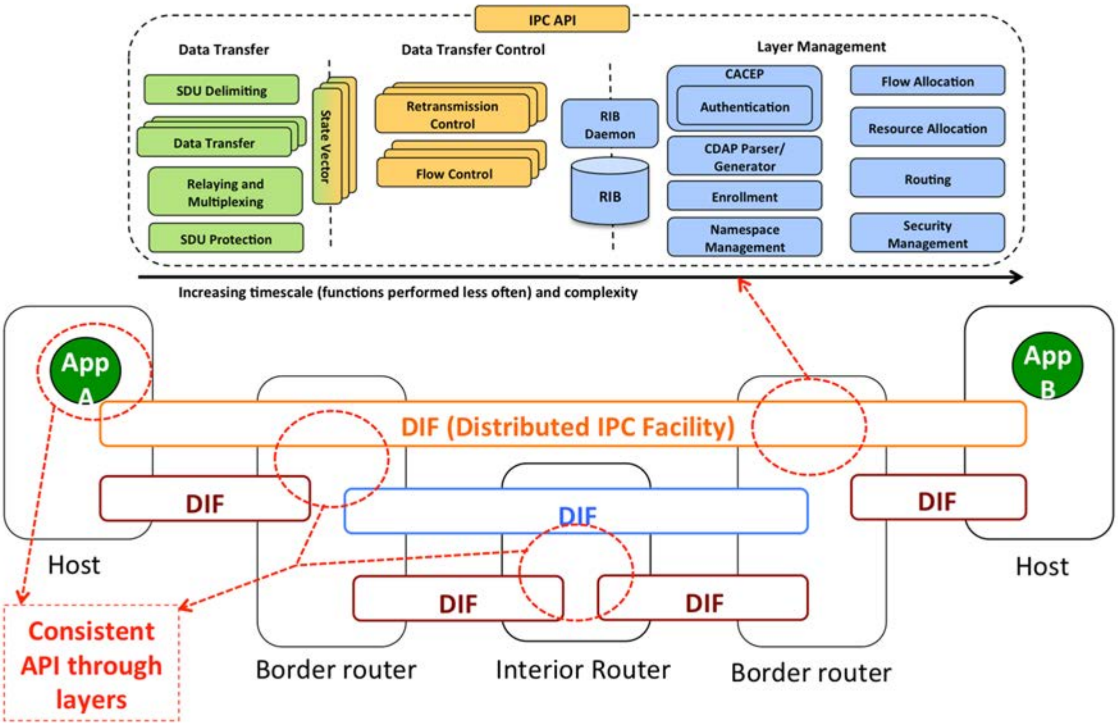


Figure 4.1: Illustration of the RINA structure: DIFs and internal organization of IPC Processes (IPCPs)

4.1.1 The DIF service definition

The DIF service definition provides the abstract description of an API as seen by an Application Process using a DIF (specific APIs are system-dependant and may take into account local constraints; in some cases there may not be an API at all, but an equivalent way to have equivalent interactions). The Application Process might be an IPC Process, reflecting the recursive nature of RINA (a DIF can be used by any distributed application, including other DIFs). All DIFs provide the same service, called flows. A flow is the instantiation of a communication service between two or more application process instances. The DIF API allows application to operate upon flows using the following four operations:

- *Allocate*: Allows an application to request a flow to a destination application, providing its application name and the desired characteristics of the flow (statistical bounds on loss and delay, in-order-delivery, minimum capacity, etc.). If the flow allocation is successful, the DIF returns a port-id, which is a local handle to the flow.

- *Write*: Writes an SDU (Service Data Unit) to the flow identified by a port-id. The application writes a full SDU of bytes in a single transaction. The integrity of the SDU is maintained by the DIF, which will try to deliver the whole SDU to the receiving application instance(s). Applications may accept delivery of incomplete or partial SDUs (this can be specified via the flow allocation request). The DIF may block the application or return error-on-write if the DIF's flow control or congestion management functions indicate to do so.

- *Read*: Reads an SDU from the flow identified by a port-id.

- *Deallocate*: Causes the DIF to terminate the flow and free all the resources associated to it.

4.1.2 The nature of layers (DIFs)

In contrast with traditional network architectures in which layers have been defined as units of modularity, in RINA layers (DIFs) are distributed resource allocators [i.6]. It is not that layers perform different functions; they all perform the same functions at different scopes. They are doing these functions for the different ranges of the environments the network is targeted at (a single link, a backbone network, an access network, an internet, a VPN, etc.). The scope of each layer is configured to handle a given range of bandwidth, QoS, and scale: a classic case of divide and conquer. Layers manage resources over a given range. The policies of each layer will be selected to optimize that range, bringing programmability to every relevant function within the layer [i.10]. How many layers are needed? It depends on the range of bandwidth, QoS, and scale: simple networks have two layers, simple internetworks, 3; more complex networks may have more. This is a network design question, not an architecture question.

4.1.3 Internals of a DIF: only two protocols required

One of the key RINA design principles has been to maximize invariance and minimize discontinuities. In other words, extract as much commonality as possible without creating special cases. Applying the concept from operating systems of separating mechanism and policy, first to the data transfer protocols and then to the layer management machinery (usually referred to as the control plane), it turns out that only two protocols are required within a layer [i.4]:

- A single data transport protocol that supports multiple policies and allows for different concrete syntaxes (length of fields in the protocol PDUs). This protocol is called EFCP - the Error and Flow Control Protocol - and is further explained in clause 4.2.

- A common application protocol that operates on remote objects used by all the layer management functions. This protocol is called CDAP - the Common Distributed Application Protocol - and is further explained in clause 4.3.

Separation of mechanism and policy also provided new insights about the structure of those functions within the layer, depicted in figure 4.1. The primary components of an IPC Process are shown in figure 4.2 and can be divided into three categories:

a) Data Transfer, decoupled through a state vector from;

b) Data Transfer Control, decoupled through a Resource Information Base from;

c) Layer Management.

These three loci of processing are characterized by decreasing cycle time and increasing computational complexity (simpler functions execute more often than complex ones):

- *SDU Delimiting*. The integrity of the SDU written to the flow is preserved by the DIF via a delimiting function. Delimiting also adapts the SDU to the maximum PDU size. To do so, delimiting comprises the mechanisms of fragmentation, reassembly, concatenation and separation.

- *EFCP, the Error and Flow Control Protocol*. This protocol is based on Richard Watson's work [i.19] and separates mechanism and policy. There is one instance of the protocol state for each flow originating or terminating at this IPC Process. The protocol naturally cleaves into Data Transfer (sequencing, lost and duplicate detection, identification of parallel connections), which updates a state vector; and Data Transfer Control, consisting of retransmission control (ack) and flow control.

- *RMT, the Relaying and Multiplexing Task*. It makes forwarding decisions on incoming PDUs and multiplexes multiple flows of outgoing PDUs onto one or more (N-1) flows. There is one RMT per IPC Process.

- *SDU Protection*. It does integrity/error detection, e.g. CRC, encryption, compression, etc. Potentially there can be a different SDU Protection policy for each (N-1) flow.

The state of the IPC Process is modelled as a set of objects stored in the Resource Information Base (RIB) and accessed via the RIB Daemon. The RIB imposes a schema over the objects modelling the IPCP state, defining what CDAP operations are available on each object and what will be their effects. The RIB Daemon provides all the layer management functions (enrolment, namespace management, flow allocation, resource allocation, security coordination, etc.) with the means to interact with the RIBs of peer IPCPs. Coordination within the layer uses the Common Distributed Application Protocol (CDAP). More detail on layer management functions and operation is provided in clause 4.3.

4.1.4 Naming and addressing

Figure 4.2 illustrates the main entities that are named in RINA. Applications are assigned location-independent names that identify the whole distributed application (a DAF or a DIF name, since a DIF is also a distributed application), a subset of the distributed application members or individual members (specific application process instances). Application names are unique within the application namespace (several, non-overlapping application namespaces may exist). When applications request a flow allocation to a DIF, they provide the destination application name as one of the arguments. If the flow allocation succeeds, the application is given back a port-id, which is a local identifier for the flow.

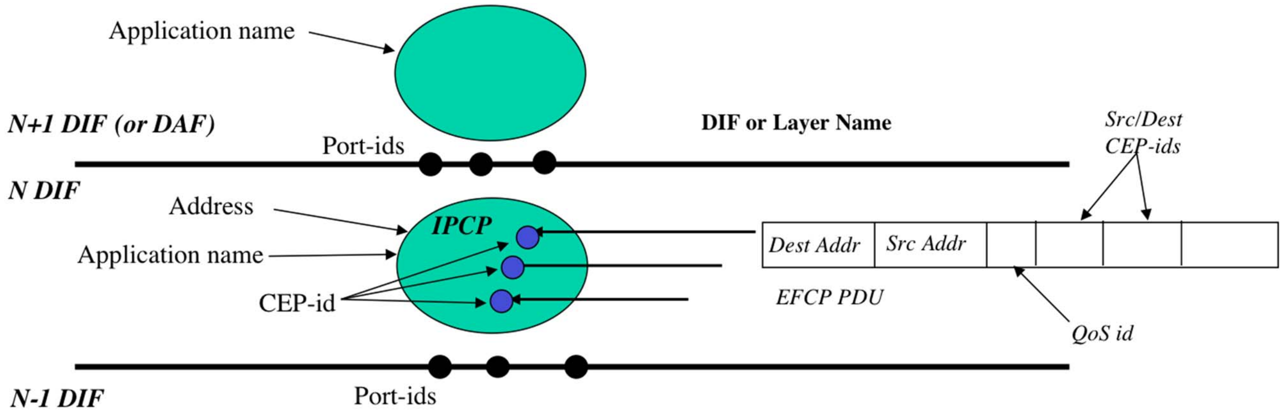


Figure 4.2: Names and addresses in RINA

IPC Processes are also application processes, therefore they have application names. However, the scope of the application namespace may be much larger than the number of IPCPs within a layer; and application names are not designed to facilitate routing within a specific layer. Therefore it is useful to assign the IPCP a synonym, called an address, which is a location-dependent but route-independent name that facilitates locating the IPCP within the DIF [i.7]. IPCPs can be assigned multiple addresses. Addresses are unique within a DIF; each DIF maintains its own address namespace. IPCPs exchange traffic with lower-level DIFs via port-ids, the same way that general-purpose applications do.

Each flow provided by a DIF is internally implemented by the means of one EFCP connection at a time. Each EFCP connection is identified by a pair of source and destination connection-endpoint ids (cep-ids), which identify the source and destination instances of the EFCP protocol machines processing the PDUs for that connection. Port-ids and cep-ids are tied together via a local binding that can change during the flow's lifetime. Decoupling port-ids from cep-ids has important security implications, as it will be explained in clause 4.5. QoS-ids identify the QoS-cube (see clause 4.1.5) to which the PDUs of the EFCP connection belong. All PDUs belonging to the same QoS cube will receive the same treatment within the DIF.

4.1.5 Consistent QoS model across layers

As shown in figure 4.1, in RINA all layers provide the same service API to their users. This API allows users of a layer to request a flow to a destination application with certain characteristics such as bounds on loss and delay, minimum capacity or in-order delivery of data. Therefore layers can pass performance requirements to each other in a technology-agnostic way, without the intervention of an external entity such as the Management System.

DIFs are designed to cover certain ranges of the performance space. A QoS cube is an abstraction of a set of policies that allow the DIF to deliver an IPC service within a certain range of the performance space (e.g. data loss, delay, jitter). Each DIF supports one or more QoS cubes, whose policies (data transfer, resource allocation, scheduling) are designed to ensure the promised performance in the operational environment of the DIF. When an application requests a flow to a DIF, the IPC Process that receives the request checks the performance requirements for that flow and tries to map it to one of the QoS cubes supported by the DIF. If there is a match, the IPCP creates a new EFCP instance for the flow, configuring it with the policies specified by the QoS cube. Each QoS-cube has a unique id within the DIF. All EFCP packets of a flow belonging to a QoS cube are marked with the qos-id of that QoS-cube, so that all intermediate IPCPs between the source and destination can identify the flows belonging to the different QoS classes and schedule them accordingly.

4.1.6 Consistent security model across layers

The distribution of security functions within the DIF and across DIFs is shown in figure 4.3. In RINA the granularity of protection is a layer, not its individual protocols, which allows for a more simple and comprehensive security model. Users of a DIF need to have little trust in the DIF they are using: only that the DIF will attempt to deliver Service Data Units (SDUs) to some process. Applications using a DIF are ultimately responsible for ensuring the confidentiality and integrity of the SDUs they pass to the DIF. Therefore, proper SDU protection mechanisms (such as encryption) have to be put in place. When a new IPCP wants to join a DIF it first needs to allocate a flow to another IPCP that is already a DIF member via an N-1 DIF which both processes share in common. Here access control is used to determine if the requesting application is allowed to talk to the requested application. If the flow to the existing member is accepted, the next step is to go through an authentication phase, the strength of which can range from no authentication to cryptographic schemes. In case of a successful authentication the DIF member will decide whether the new IPCP is admitted to the DIF, executing a specific access control policy.

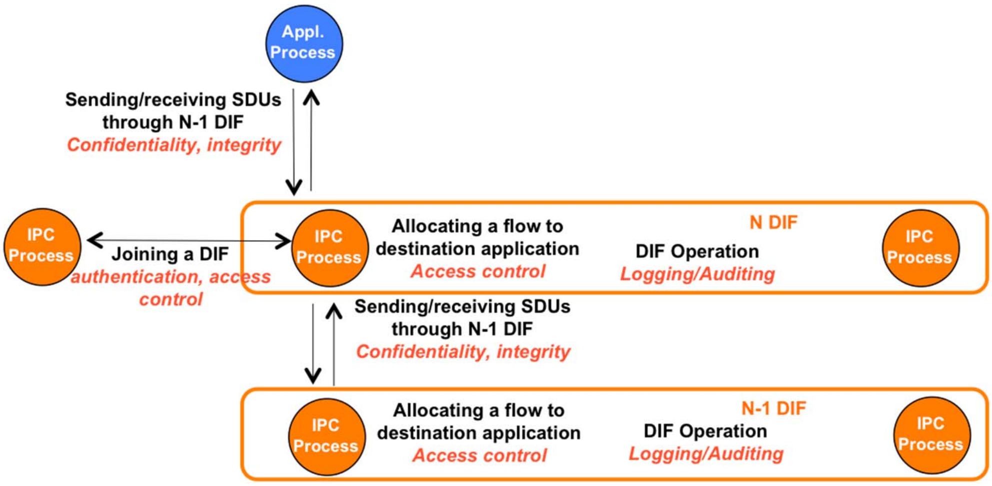

Appl. Process
Sending/receiving SDUs through N-1 DIF
Confidentiality, integrity
IPC Process
Joining a DIF
authentication, access control
IPC Process
Allocating a flow to destination application
Access control
N DIF
DIF Operation
Logging/Auditing
IPC Process
Sending/receiving SDUs through N-1 DIF
Confidentiality, integrity
IPC Process
Allocating a flow to destination application
Access control
N-1 DIF
DIF Operation
Logging/Auditing
IPC Process

Figure 4.3: Distribution of security functions within a DIF and across DIFs

4.1.7 Network Management

The Network Management distributed application (Network Management DAF or NM-DMS) is in charge of managing a collection of systems and its IPC Processes belonging to a network. NM-DMSs are distributed applications, like DIFs, therefore a collection of application processes co-operating to manage a network. Therefore NM-DMSs leverage the common distributed application machinery (RIB to model state, CDAP as a common application protocol) and the IPC services provided by DIFs to perform its task. The DAF model can be applied to network management to represent the whole range from distributed (autonomic) to centralized (traditional).

In the traditional centralized network management architecture, depicted in figure 4.4, an NM-DMS would be a heterogeneous DAF consisting of one or more application processes providing management functions (fault management, configuration management, performance management, etc.) with other DAPs providing telemetry and local management of systems (the Management Agents). Management Agents have direct access to the IPCPs in the system they manage, via local procedures. It is possible for there to be multiple MAs responsible for different DIFs in the same processing system. For example, one might create DIFs as VPNs and allow them to be managed by their "owners"; or one could imagine different DIFs belonging to different providers at the border between two providers, etc.

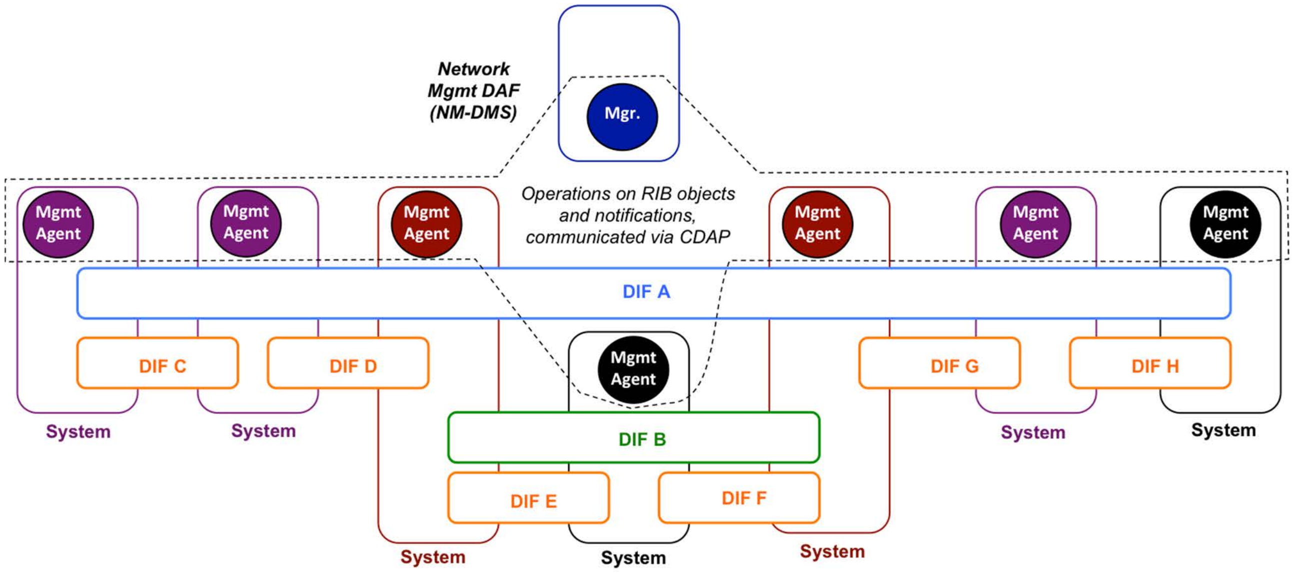


Figure 4.4: NMS-DAF in a traditional centralized management configuration

4.2 Data transfer: protocol, functions and procedures

4.2.0 General

EFCP, the Error and Flow Control Protocol, is the single data transfer protocol of a DIF. In order to allow for its adaptation to different operating environments, EFCP supports multiple policies and multiple specific syntaxes. To do so, the EFCP specification defines the hooks where the different policies can be plugged in - also describing the behaviour of default policies - as well as the abstract EFCP syntax (PDU types and its fields, without describing its encoding). EFCP leverages the results published by Richard Watson (and later implemented in the delta-t protocol). Watson proved that bounding three timers is a necessary and sufficient condition for reliable transport connection management; in other words: SYNs and FINs are unnecessary. This not only simplifies the protocol implementation, but it also makes it more reliable against harsh network environments [i.8] or transport-level attacks [i.9].

EFCP has two parts: DTP (Data Transfer Protocol), which deals with the mechanisms tightly coupled to data transfer PDUs (such as addressing or sequencing) and DTCP (Data Transfer Control Protocol), which deals with the loosely bound mechanisms such as flow control or retransmission control. DTP and DTCP are fairly independent and operate with their own PDUs, being just loosely coupled via a state vector.

4.2.1 DTP PDU Format

Figure 4.5 illustrates the abstract syntax of EFCP DTP PDUs. Note that the length of address, qos-id, cep-id, length and sequence number fields depends on the DIF environment. For example, no source or destination address fields are required for DIFs in point-to-point links.

VERS	DST ADDR	SRC ADDR	QoS ID	DST CEPID	SRC CEPID	PDU Type	Flags	Length	Seq. Number	User data
1-byte	addr-bytes	addr-bytes	qos-bytes	cepid-bytes	cepid-bytes	1-byte	1-byte	length-bytes	seqnum-bytes	n-bytes

Figure 4.5: Abstract syntax of DTP PDUs

- *Version*: EFCP version.
- *Src/destination address*: Addresses of the IPC Processes that host the endpoints of this EFCP connection.
- *QoS-id*: Id of the QoS cube where this EFCP connection belongs.
- *Src/destination cep-ids*: The identifiers of the EFCP instances that are the endpoints of this EFCP connection.

- *PDU type*: Code indicating the type of PDU (in this case it is a DTP PDU).
- *Flags*: Indicate conditions that can affect the processing of the PDU and can change from one PDU to another.
- *Length*: The total length of the PDU in bytes.
- *Sequence number*: Sequence number of the PDU.
- *User data*: Contains one or more SDU fragments and/or one or more complete SDUs.

4.2.2 DTCP PDU Formats

Depending on the policies associated to a particular EFCP connection, the DTCP instance may be configured to perform flow and/or retransmission control functions. While the EFCP specification defines 10 operation codes defined for DTCP, in reality there are only three PDU types:

i) Ack/Nack/Flow;

ii) Selective Ack/Nack/Flow; and

iii) Control Ack. Each of these control PDUs carries addresses, a connection-id, a sequence number, and retransmission control and/or flow control information, etc.

The opcodes indicate which fields in the PDU are valid. Required fields for these PDUs can be extended by defining policies.

4.2.3 Overview of data-transfer procedures

A high-level overview of the data-transfer procedures is provided by figure 4.6. Note that this is an example scenario showing logical functions, in any case is suggesting a particular implementation strategy. In this example scenario, the DIF N provides a flow identified by port-id 1 between applications A and B. When application A writes an SDU to the port, invoking the DIF API, the SDU is processed by the delimiting function of IPCP I1, which will create one or more EFCP user-data fields from the SDU, according to the delimiting policy. EFCP user-data fields are delivered to the EFCP instance 23 - which is currently bound to port-id 1 - which creates one or more EFCP data transfer PDUs and hands them over to the Relaying and Multiplexing Task (RMT). The RMT checks the forwarding function (another policy), which returns the port-ids of one or more N-1 flows through which the PDU needs to be forwarded to reach the next hop (in this case the IPCP with address 80). In general there will be one or more queues in front of each N-1 port, and a scheduling policy will sort out outgoing PDUs for transmission according to different criteria. Once the N-1 port through which the PDU will be forwarded is known, the associated SDU protection policy can be applied to the PDU (or it can be applied when EFCP creates the PDU if there is a common SDU protection policy for all N-1 ports).

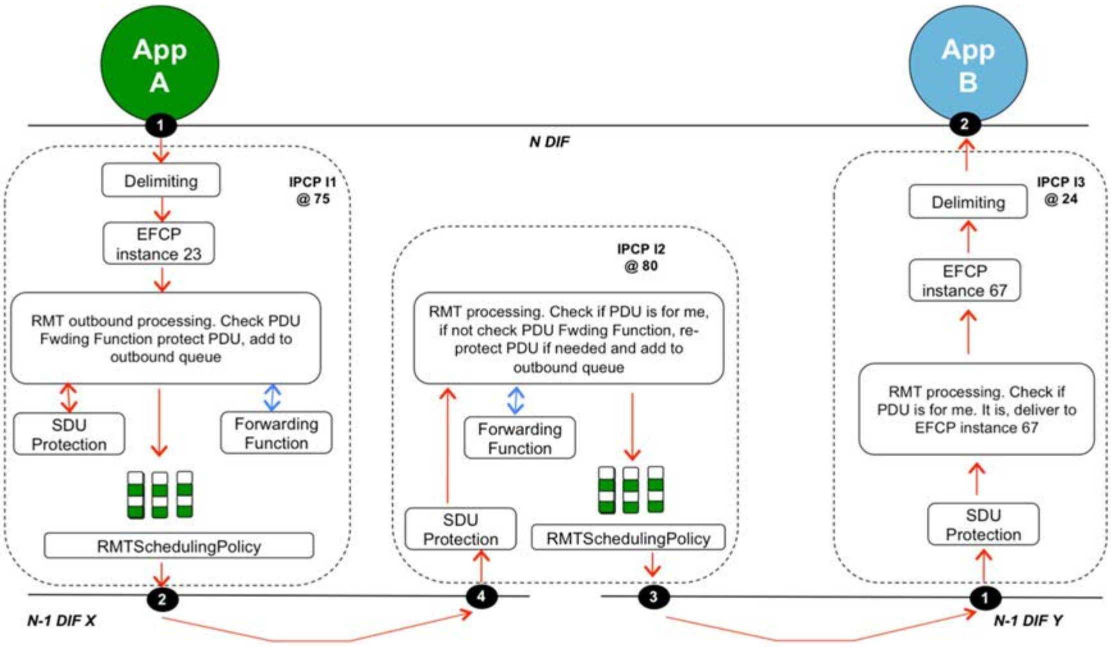

App A
App B
1
2
N DIF
Delimiting
IPCP I1 @ 75
EFCP instance 23
RMT outbound processing. Check PDU Fwding Function protect PDU, add to outbound queue
SDU Protection
Forwarding Function
RMTSchedulingPolicy
IPCP I2 @ 80
RMT processing. Check if PDU is for me, if not check PDU Fwding Function, re-protect PDU if needed and add to outbound queue
Forwarding Function
SDU Protection
RMTSchedulingPolicy
IPCP I3 @ 24
Delimiting
EFCP instance 67
RMT processing. Check if PDU is for me. It is, deliver to EFCP instance 67
SDU Protection
2
4
3
1
N-1 DIF X
N-1 DIF Y

Figure 4.6: Example of data transfer procedures

Eventually IPCP I2 reads the PDU from N-1 port 4. It removes SDU protection required to process the PDU header, and the RMT decides if it is the final destination of the PDU (depending on the DIF environment; for example, checking the destination address field in this example). In this case the IPCP is not the final destination, so the RMT checks the forwarding function, which returns one or more N-1 ports through which the PDU will be forwarded. The RMT re-applies protection if needed (SDU protection policy may be different), and handles the PDU for transmission to the scheduling policy, which eventually writes the PDU to the N-1 port.

Finally the PDU reaches IPCP I3 through N-1 port 3. SDU protection is removed, the RMT checks if it is the final destination of the PDU and, since in this case it is, it delivers the RMT to the destination EFCP instance (EFCP instance 87 in the example) for further processing. The EFCP instance updates its internal state and may generate zero or more control PDUs. EFCP recovers the PDU's user data field, and works with the delimiting function according to the configured policies in order to recover full SDUs. Finally SDUs are read from port 2 by application B.

4.3 Layer management: protocol, functions and procedures

4.3.1 Common layer management machinery

The different layer management functions of an IPC Process leverage a common machinery to exchange information with their peers. All the IPC Process externally visible state is modelled as objects that follow a logical schema called RIB, Resource Information Base. The RIB specification defines the object naming, relationships between objects (inheritance, containment, etc.), the object attributes, and the CDAP operations that can be applied on them. Access to the RIB is mediated by the RIB Daemon. The RIB Daemon of an IPCP exchanges CDAP PDUs with RIB Daemons of neighbour IPCPs. These PDUs communicate remote operations on objects. When a layer management task wants to communicate an action to a peer (e.g. a routing update), it requests the RIB Daemon to perform an action on one or more objects of one or more neighbour IPCPs. The RIB Daemon generates the required CDAP PDUs and sends them over the required N-1 flows to communicate the action to its neighbours. When the RIB Daemon receives a CDAP PDU, it decodes it, analyses what objects are involved and notifies the relevant layer management functions (who have previously subscribed to objects of their interest).

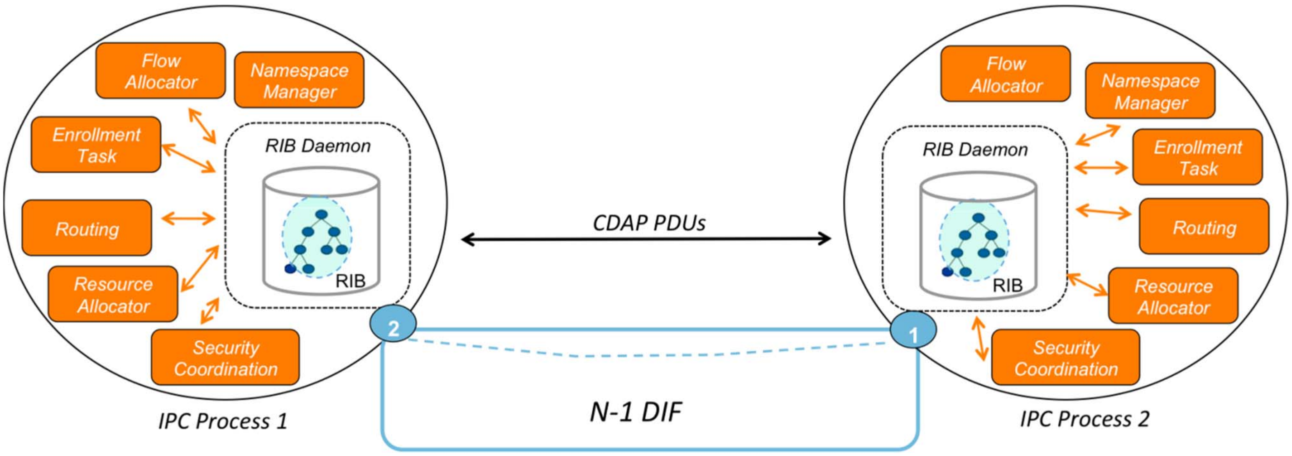


Figure 4.7: Common layer management machinery: RIB, RIB Daemon and CDAP

The whole process is illustrated in figure 4.7. This design allows the layer management tasks to just focus in the functions they provide and delegate the routine tasks of generating and parsing protocol PDUs to the RIB Daemon (in fact, layer management tasks are not even aware of CDAP). If required, new layer management functions could be added without the need to define new protocols. Moreover, the RIB Daemon can coordinate and optimize the generation of protocol PDUs from different layer management tasks; thus minimizing the layer management traffic between peer IPCPs. The CDAP specification defines an abstract syntax that describes the different types of CDAP PDUs and their fields. Multiple concrete encodings can be supported (it is just a DIF policy), such as the various ASN.1 encodings, Google Protocol Buffers, etc.

Two peer IPCPs cannot exchange any information until an association has been established between them. This association is called application connection in RINA terms. During the application connection establishment phase, the IPCPs exchange naming information, optionally authenticate each other, and agree on the abstract and concrete syntaxes of CDAP/RIB to be used in the connection, as well as in the version of the RIB. This version information is important, as RIB model upgrades may not be uniformly applied to the entire network at once. Therefore it needs to be possible to allow multiple versions of the RIB to be used, to allow for incremental upgrades.

4.3.2 Layer management functions: enrolment

Enrolment is the procedure by which an IPCP joins an existing DIF and is initialized with enough information to become a fully operational DIF member. Enrolment occurs after an IPC-Process establishes an application connection with another IPCP, which is a member of a DIF. Once the application connection is established this enrolment procedure may proceed. The specific enrolment procedure is a policy of each DIF, but in general it involves operations similar to the ones described in the next paragraph.

The Member IPCP reads the New Member IPCP's address. If null or expired, it assigns a new address; otherwise, assumes the New Member was very recently a member. The New Member then reads the information it does not have taking into account how "new" it is. These parameters characterize the operation of this DIF and might include parameters such as max PDU size, various time-out ranges, ranges of policies, etc. Once complete, the New Member is now a member and this triggers a normal RIB update (to get the latest up to date information on routing, directory, resource allocation, etc.).

4.3.3 Layer management functions: namespace management

Managing a name space in a distributed environment requires coordination to ensure that the names remain unambiguous and can be resolved efficiently. The Name Space Manager (NSM) embedded in the DIF is responsible for mapping application names to IPC Process addresses - the latter being the name space managed by the DIF NSM. Specific ways of achieving this mapping are policy and will vary from DIF to DIF. For small, distributed environments, this management may be fairly decentralized and name resolution may be achieved by exhaustive search. Once found the location of the information that resolved the name may be cached locally in order to shorten future searches. It is easy to see how, as the distributed environment grows, these caches would be further organized, often using hints in the name itself, such as hierarchical assignment, to shorten search times. For larger environments, distributed databases may be organized with full or partial replication and naming conventions, i.e. topological structure, and search rules to shorten the search, requiring more management of the name space.

The two main functions of the DIF NSM are to assign valid addresses to IPC Processes for its operation within the DIF and to resolve in which IPC Process a specific application is registered. In other words, the NSM maintains a mapping between external application names and IPC Process addresses where there is the potential for a binding within the same processing system. Therefore enrolment, application registration and flow allocation require the services of the NSM.

4.3.4 Layer management functions: flow allocation

The Flow Allocator is responsible for creating and managing an instance of IPC, i.e. a flow. The IPC-API communicates requests from the application to the DIF. An Allocate-Request causes an instance of the Flow Allocator to be created. The Flow Allocator-Instance (FAI) determines what policies will be utilized to provide the characteristics requested in the Allocate. It is important that how these characteristics are communicated by the application is decoupled from the selection of policies. This gives the DIF important flexibility in using different policies, but also allows new policies to be incorporated. The FAI creates the EFCP instance for the requested flow before sending the CDAP Create Flow Request to find the destination application and determine whether the requestor has access to it.

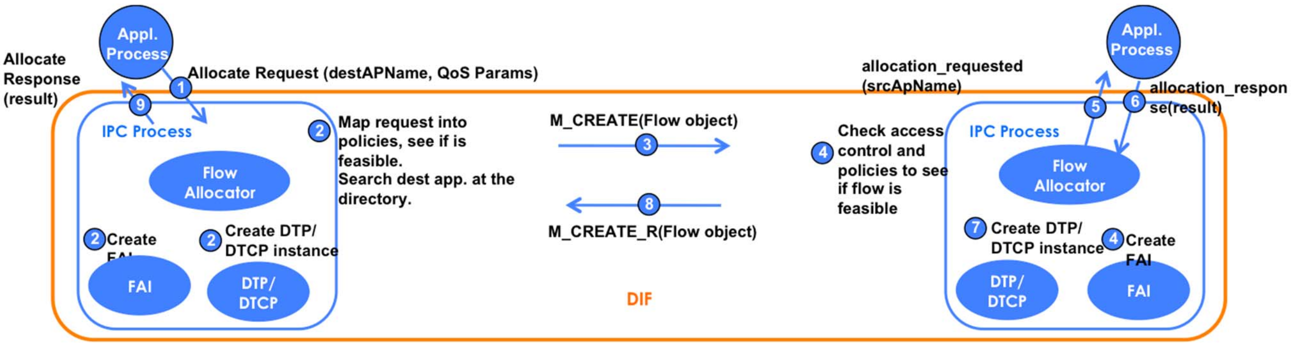

Allocate Response (result)
Appl. Process
Allocate Request (destAPName, QoS Params)
1
9
IPC Process
2 Map request into policies, see if is feasible. Search dest app. at the directory.
Flow Allocator
2 Create FAI
2 Create DTP/ DTCP instance
FAI
DTP/ DTCP
M_CREATE(Flow object)
3
8
M_CREATE_R(Flow object)
DIF
allocation_requested (srcApName)
4 Check access control and policies to see if flow is feasible
Appl. Process
5
6
allocation_response(result)
IPC Process
Flow Allocator
7 Create DTP/ DTCP instance
4 Create FAI
DTP/ DTCP
FAI

Figure 4.8: Illustration of the flow allocation procedure

A create request is sent with the source and destination application names, quality of service information, and policy choices, as well as the necessary access control information. Using the NSM component, the FAI finds the IPCP in the DIF that resides on the processing system that has access to the requested application. This exchange accomplishes three functions:

- Following the search rules using the Name Space Management function to find the address of an IPC-Process with access to the destination application.
- Determining whether the requesting application process has access to the requested application process and whether or not the destination IPC-Process can support the requested communication.
- Instantiating the requested application process, if necessary, and allocating a FAI and port-id in the destination IPCP.

The create response will return an indication of success or failure. If successful, destination address and connection-id information will also be returned along with suggested policy choices. This gives the IPC-Processes sufficient information to then bind the port-ids to an EFCP-instance, i.e. a connection, so that data transfer may proceed.

4.3.5 Layer management functions: resource allocation

The Resource Allocator (RA) gathers the core intelligence of the IPC Process. It monitors the operation of the IPC Process and makes adjustments to its operation to keep it within the specified operational range. The degree to which the operation of the RA is distributed and performed in collaboration with the other RAs in members of the DIF, and the degree to which the RA merely collects and communicates information to a Network Management System (NM-DMS) which determines the response, is a matter of DIF design and research. The former case can be termed autonomic, while the latter case is more the traditional network management approach. Both approaches have their use cases and application areas. There are basically three sets of information available to the IPC Process to make its decisions:

- The traffic characteristics of traffic arriving from the user of the DIF, i.e. the application or (N+1)-DIF.
- The traffic characteristics of the traffic arriving and being sent on the (N-1)-flows.

- Information from other members of the DIF on what they are observing (this latter category could be restricted to just nearest neighbours, or some other subset such as all two or three hop neighbours, or could be all the members of the DIF).

The first two categories would generally be measures that are easily derived from observing traffic: bandwidth, delay, jitter, damaged PDUs, etc. The shared data might consider internal status of other IPC Processes such as queue length, buffer utilization, and others. The Resource Allocator has several "levers" and "dials" that it can change to affect how traffic is handled:

- *Creation/Deletion of QoS Classes*. Requests for flow allocations specify the QoS-cube the traffic requires, which is mapped to a QoS-class. The RA may create or delete QoS-classes in response to changing conditions.

- *Data Transfer QoS Sets*. When an Allocate requests certain QoS parameters, these are translated into a QoS-class that in turn is translated into a set of data transfer policies. The RA may modify the set of data transfer policies for particular QoS classes. For example, one could imagine a different set of policies for the same QoS-class under different load conditions.

- *Modifying Data Transfer Policy Parameters*. It is assumed that some data transfer policies may allow certain parameters to be modified without actually changing the policy in force. A trivial example might be changing the retransmission control policy from acknowledging every second PDU to acknowledging every third PDU.

- *Creation/Deletion of RMT Queues*. Data Transfer flows are mapped to Relaying and Multiplexing queues for sending to the (N-1)-DIF. The RA can control these queues as well as which QoS classes are mapped to which queues. (The decision does not have to be exclusively based on QoS-class, but may also depend on the addresses or current load, etc.)

- *Modify RMT Queue Servicing*. The RA can change the discipline used for servicing the RMT queues.

- *Creation/Deletion of (N-1)-flows*. The RA is responsible for managing distinct flows of different QoS-classes with the (N-1)-DIF. Since multiplexing occurs within a DIF one would not expect the (N)-QoS classes to be precisely the same as the (N-1)-QoS classes. The RA can request the creation and deletion of (N-1)-flows with nearest neighbours, depending on the traffic load offered to the IPC Process and other conditions in the DIF.

- *Forwarding Table Generator Output*. The RA takes input from other aspects of layer management to generate the forwarding table. This is commonly thought of as the output of "routing". It may well be here, but approaches to generating the forwarding table not based on graph theory are also supported.

4.3.6 Layer management functions: routing

A major input to the Resource Allocator is Routing. Routing performs the analysis of the information maintained by the RIB to provide connectivity input to the creation of a forwarding table. To support flows with different QoS will in current terminology require using different metrics to optimize the routing. However, this needs to be done while balancing the conflicting requirements for resources. Current approaches can be used but new approaches to routing will be required to take full advantage of this environment. The choice of routing algorithms in a particular DIF is a matter of policy.

4.3.7 Layer management functions: security coordination

Security coordination is the IPC Process component responsible for implementing a consistent security profile for the IPC Process, coordinating all the security-related functions (authentication, access control, confidentiality, integrity) and also executing some of them (auditing, credential management). The sophistication of this layer management function is a matter of policy.

4.4 Support for mobility

Mobility is nothing more than multi-homing where the points of attachment change a bit more frequently. By assigning addresses to IPCPs (instead of Points of Attachments to N-1 DIFs), RINA solves multi-homing without the need of any special protocols [i.7]. If an N-1 port of an IPCP goes down, the IPCP will issue a routing update that will be propagated through the DIF. PDUs en route to the IPCP will just be routed along a different path, they do not have to be dropped because they are addressed to the IPCP, not to one of its "interfaces" (N-1 ports).

With this built-in multi-homing support, mobility can also be easily supported without the need of extra protocols, just taking into account the following considerations [i.11].

- *Addresses locate the Mobile Host*. IPCPs are assigned location-dependent addresses that allow them to be located within the graph of the DIF they belong to. If the Mobile Host (MH) that contains the IPCP moves too far the address will no longer be aggregateable causing an increase in router table size and potentially less efficient routing. This means that its address within the layer needs to change to keep router table size manageable and efficient. This procedure is not complicated because addresses are just synonyms of the IPCP name, therefore dynamic renumbering in a DIF is not an issue [i.11].

- *Responsiveness to location change*. In traditional static network architectures, where there are basically two scopes: the very small point-to-point data link layer and the whole world of the network layer, routing updates may take far too long to be responsive enough for certain mobility environments. But in RINA, by creating more layers of the same rank or layers of greater rank, the size of routing tables and the time to update them can be bounded by design [i.12]. Lower layers where points of attachment change frequently would have small scopes that could be updated quickly; higher layers would have greater scopes where points of attachment changed less frequently and updates would take longer but still a fraction of the time that a MH moving across it would change points of attachment, and so on. Since all layers are equal, adding extra layers involves little overhead in terms of PDU header size and protocol processing delay [i.11].

- *No loss of data*. The same flows are in place through the lifetime of the application connection. There are no tunnels to set up and tear down. Data would only be lost if there was no physical communication with the network.

- *Manageability*. The application names never change. The mapping of application name to (N)-address and of (N)-addresses to (N-1)-addresses is ensured by engineering the scope of the layers to ensure the update time is small compared to the rate of change of (N-1)-addresses.

- *Scalability*. The repeating structure can be used to handle any density of MHs and virtually any rate of change. The number of layers can be different in different parts of the network depending on the requirements; allowing this approach to scale indefinitely. The only constraints are imposed by physics. In fact, this is probably the only approach that does scale to the densities that are expected.

An average-case cost analysis to compare the multi-homing/mobility support of RINA, against that of other approaches such as LISP and Mobile-IP is performed in [i.13].

4.5 Support for security

RINA relies on several design principles that make RINA networks inherently more secure than current Internet networks, also with much less security overhead [i.15]:

- *Securing layers instead of protocols*. In RINA the fundamental object to be protected is the layer (DIF) instead of its individual protocols. Clause 4.1.6 showed the different security interactions between layers and within a layer, discussing the placement of the different security functions. Small et al. perform a threat analysis of RINA at the architecture level in [i.14], concluding that when proper authentication, SDU protection and access control policies are put in place, a DIF is a securable container (a structure used to hold or transport data that can be made not subject to threat).

- *Scope as a native construct, controlled connectivity by default*. The current Internet model is based on having connectivity with everyone by default (due to the global scope of the Internet layer). However, multiple means of controlling the scope of the connectivity are used in a network to have more control over who can connect to whom, such as: VLANs, firewalls, VPNs, ACLs, etc. In contrast, RINA supports a controlled connectivity model, in which each layer is sized according to its connectivity requirements: scope is a native construct.

- *Separation of mechanism from policy*. A security policy specified for a particular DIF can be re-used right away in any DIF, since all DIFs have the same structure. This principle minimizes the number of required security mechanisms in the network, while still allowing customization at every DIF. In [i.16], Small performs an initial comparison between RINA and the current Internet, concluding that RINA networks can deliver on security requirements with less complexity and overhead.

- *Decoupling of port allocation from synchronization*. TCP overloads the port-id to be both a local handle (transport port number) and a connection-endpoint-id. The lack of application names overloads transport ports with application naming semantics, causing the application to listen to static well-known ports. In contrast in RINA port-allocation and synchronization are separate functions. The port allocation procedure is triggered by an application requesting a flow to a destination application. The source IPCP dynamically assigns a local port-id to the flow and creates an instance of EFCP that takes care of the feedback synchronization aspects. This EFCP instance is identified by a dynamically generated cep-id that is mapped to the port-id via a local binding. A similar procedure happens at the destination IPCP. Moreover, the state of ports and connections is managed with different approaches: port state is explicitly created and removed by applications (hard-state), whereas connection state is created and removed following a timer-based approach (soft-state). Bodapatti et. al [i.9] show how RINA leverages this design to achieve a greater resiliency than TCP/IP to transport-level attacks such as port scanning, connection opening or data transfer.

- *Use of a complete naming and addressing architecture*. The lack of application names in TCP/IP causes IP layers to expose addresses to applications. This disclosure of information facilitates spoofing of IP addresses and, in combination with the use of monitoring tools such as traceroute or ping allows attackers on end hosts to learn about the addresses of potential targets in a layer. In RINA, applications use names to request flows to each other, which then are internally mapped by the DIF to the addresses of IPCPs within that DIF. DIF addresses are not divulged outside of the DIF.

4.6 Addressing and scalability

Clause 4.1.4 has already introduced the core naming and addressing principles in RINA. All applications have names that uniquely identify them within a certain application namespace. Application names are location-independent, so that applications can preserve their identity regardless of their point(s) of attachment to the network.

A DIF is a distributed application, with IPCPs being application processes. Each IPCP has an application name, but it can also have one or more synonyms to facilitate the location of the IPCP within the layer. These synonyms (called addresses) are location-dependent, that is, indicate the location of the IPCP with respect to an abstraction of the connectivity graph of the layer. Addresses are also route-independent, so that the IPCP address does not force a particular binding of the IPCP with an N-1 point of attachment (since it complicates multi-homing and mobility). Within each layer application names are mapped to IPCP addresses via a directory function, which maintains a distributed mapping between the names of applications registered to the DIF and the addresses of the IPCPs these applications are attached to.

Given its repeating nature, where each DIF has its own private internal addresses, and with the existence of policies that constrain the membership size of each DIF, RINA can achieve much better address scalability compared to that of the current Internet [i.12]. In addition, RINA comes with a complete naming and addressing schema. The narrower scope of topology changes and late binding from node address to N-1 point of attachment makes the RINA architecture much more scalable in terms of routing overhead. The choice to use location-dependent addresses contributes to the scalability within a single DIF [i.17].

4.7 Interworking and migration

RINA can be deployed incrementally where it has the right incentives (benefits outweigh deployment cost), and interoperate with other networking technologies. Figure 4.9 depicts a number of potential RINA adoption scenarios.

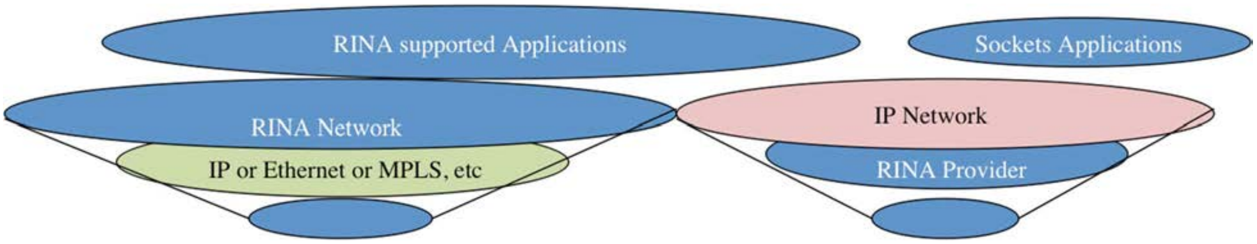


Figure 4.9: RINA interworking with other networking technologies

RINA can be used:

- *As an underlay technology*. As an alternative to MPLS in service provider networks, or as a substrate to create highly scalable and optimized datacentre network fabrics. It can transport IP, Ethernet or any other networking protocol.
- *As an overlay technology*. To create VPNs over IP, Ethernet, MPLS, etc. segmenting the underlying layer(s) for multiple users/tenants and providing enhanced application support (location-independent application names, flow service tailored to application needs, etc.).
- *Interoperating with legacy technologies via gateways or protocol translation*. For example to allow applications in a RINA-enabled datacentre to be reachable by non-RINA-enabled clients through the Internet.

RINA can also support legacy applications coded to the sockets API via a sockets-emulator layer (faux sockets API). This allows current applications to run without modification over RINA networks, although these applications will be limited by the semantics of the sockets API.

4.8 Assessment against NGP key issues

Table 4.1 summarizes how RINA addresses the issues identified in table 1 of the Scenarios document [i.1].

Table 4.1: Summary of RINA against key issues

Issue ID	Issue Name	Issue Description
01	Addressing	RINA provides a complete naming and addressing architecture, with location-independent application names and location-dependent but route-independent node names (i.e. addresses). Within a DIF, addresses reflect the location of an IPCP within the DIF's connectivity graph, thus facilitating routing. Addresses are just synonyms of the IPCP name; therefore having multiple addresses or changing them during the IPCP's lifetime is not an issue. Addresses are never exposed outside a DIF: each DIF has its own address format (some DIFs do not even need addresses). The namespace manager maintains a distributed mapping of application names to IPCP addresses within a DIF.
02	Security	Several design decisions in the RINA architecture contribute towards more secure networks with less overhead and cost: protecting layers instead of protocols; having scope as a native construct; separating mechanism from policy; decoupling port allocation from synchronization; and a complete naming and addressing scheme.
03	Mobility	Mobility is just dynamic multi-homing with more frequent (and expected) failures. The built-in multi-homing support of RINA without the need for special protocols reduces the mobility problem to routing updates and changing IPCP addresses from time to time. The number and size of layers in the different parts of a network can be designed to support any Mobile Host density and virtually any rate of change.
04	Multi-Access Support. (including FMC)	Native support for multi-homing, the expressiveness of the DIF API, and separation of mechanism from policy within a DIF allow RINA to support this scenario very efficiently. Each IPCP in a DIF can be programmed with adequate forwarding policies to forward the traffic through the required N-1 DIFs based on different criteria (for load-balancing, fail-over, etc.). Users can express their preferences for their flows via the IPC API, and the resource allocator can balance these preferences by applying the forwarding policies that best serve users' needs.
05	Context Awareness	Users can request the characteristics for their flows via the DIF API (bounds on loss and delay, in-order delivery of data, etc.). These requests influence the choice of policies for that particular flow. IPCPs in a DIF can leverage the common layer management machinery (RIB, CDAP) to exchange any context information that may be deemed useful for network operation (without the need to define any new protocols). This information can influence the selection of the DIF policies at any given time (i.e. IPCPs may decide to change a resource allocation policy as a result of the context information being exchanged). This information can be also exported to the network management system, which can make global network optimizations based on the information from a number of DIFs.

Issue ID	Issue Name	Issue Description
06	Performance (including Content Enablement)	The policies of each DIF can be adapted to the operational environment of the DIF and the requirements of the applications it supports. EFCP policies can be optimized for different applications: different congestion controllers can be easily plugged into EFCP, without redesigning the protocol; and different scheduling policies can be applied to multiplex the traffic of flows belonging to different QoS cubes. Explicit congestion notification and congestion management at each DIF makes response to congestion faster and more predictable [i.18]. Layers can signal their requirements to each other via the DIF API; therefore, N-1 DIFs can provide the level of service required by the DIFs they support.
07	Network Virtualisation	The structure of RINA inherently supports network virtualisation without the need of extra additions. DIFs can recurse as many times as needed by the network designers. Moreover, RINA's approach towards virtualisation and programmability is more efficient than current practice: since it is the same block that repeats all the time, with well defined, programmable policy points [i.10] the overhead can be reduced and implementation strategies be more effective.
08	Energy Efficiency	One of the more effective ways to reduce the energy consumption of current networks is to make them simpler: with fewer devices and with less processing per device. RINA addresses the first point by reducing the "network functions" and "types of network devices" to the minimum required. Location-dependent but route-independent addresses can drastically minimize the size of routing/forwarding tables [i.17], thus minimizing the fast memory usage in routers (which highly impacts energy consumption). Having a single protocol repeating across many layers also drastically simplifies protocol processing, which should provide gains in energy consumption.
09	IoT support	RINA is an architecture, not a building. Buildings designed to the RINA architecture can be optimized for all kinds of networks, including resource-constrained environments like IoT. Length of EFCP fields can be minimized (some of them may not be even needed depending on the environment: addresses, cep-ids, qos-id). Policies can be simplified to the bare minimum required, achieving a good trade-off between required functionality and resource usage. Very minimal RINA implementations can fit into low-power devices. Built-in support for multi-homing and mobility without extra mechanisms can also benefit IoT environments.
10	e-Commerce	COMMENT: Is not this just mapping to security and performance requirements? (already addressed by rows 2 and 6?)

5 Flexilink

5.1 Overview

5.1.1 Background

Flexilink grew out of a project to implement audio networking over ATM at the BBC, including implementing switches tailored to the needs of live audio and using Ethernet as an ATM physical layer (standardized by the Audio Engineering Society as AES51). It retains the useful features of ATM, and particularly of the way in which it was implemented in the BBC project to support live media, while jettisoning the unsuccessful aspects.

It also has ancestry in a ring network technology that was used in industrial applications and in PC networking towards the end of the 20th century. As with other technologies which were created at that time, such as IP, network elements were implemented on a platform which consisted of a computer with a small amount of peripheral circuitry. Flexilink is designed for newer kinds of platform, such as FPGAs and SoCs.

Whereas many packet networks are designed only to carry data between computers, Flexilink is also designed to meet the rather different requirements of live digital audio and video, which form an increasing proportion of the traffic on today's networks, and other traffics with low-latency requirements that are proposed for 5G.

A prototype implementation exists and is being used in research at Birmingham City University and elsewhere. The main components of the technology are in the public domain and are not believed to be subject to IPR claims.

5.1.2 Main differences from IP

In Flexilink networks, the information needed to route packets is carried separately from the packets themselves. This reduces the size of the packet header by an order of magnitude, and simplifies the process of forwarding the packets in switches.

Most importantly, it allows different addressing mechanisms to be used without changing the packet format, and supports mobility without needing artificial devices such as IP-in-IP tunnels.

There is a separate service for constant-bit-rate traffic such as audio and video, offering very low latency as standard. This service can also be used for file transfer, where it eliminates the need for the kind of empirical throughput testing that is a feature of TCP (see clause 5.1.5).

5.1.3 Flows, and separation of control and forwarding

Each packet is part of a "flow", and the packet header only needs to contain a locally-significant identification of the flow to which it belongs, along with packet-specific information such as the length. Information about the flow is conveyed by control plane messages, which can carry much more explicit detail than can reasonably be included in packet headers, and do not need to be constrained to use a particular form of addressing. The rationale behind this is as follows:

- In practice, most IP packets are part of a TCP or RTP session.
- The memory needed to store information about each flow is no longer a scarce resource; memories are typically about 1 million times as big as they were when the technology on which IP is based was developed.
- The service experienced by a packet in an IP network often depends on the flow to which the packet is identified as belonging; this identification is typically assembled from 13 bytes of information in five separate fields in the header.
- Separation into control and data (or forwarding) planes is already a feature of mobile networks, and of SDN, and also of the internal structure of most network switches.

Thus Flexilink flows make explicit, and easier to implement and to control, a feature that is in practice also present in IP networks. The control plane is assumed to be implemented in software running on CPUs, while the data plane is assumed to be implemented in hardware (or "logic"; FPGA for prototyping, SoC for production). This division of labour allows switches to be more energy-efficient than if they had to include a processor fast enough to examine every packet.

In the control plane, each flow has an identifier which (unlike an ATM call reference) is the same everywhere in the network. This makes it easy to detect routing loops, rendering protocols such as Spanning Tree unnecessary.

Applications are able to participate in control plane negotiation, for instance to specify the bandwidth required for a media stream and maybe offer a choice of trade-offs between quality and bandwidth in the case of compressed media. Control plane messages also carry metadata between the endpoints, including information on coding formats; thus they carry the information which in IP networks is carried by SIP and SDP.

5.1.4 Services

Flexilink provides two main services, referred to as AV and IT. Each has its own packet format, and on point-to-point wired links the two are multiplexed together, with the AV service having priority and IT packets able to use all the bytes that are not occupied by AV packets. The multiplexing is done in a way that avoids any requirement for fragmentation headers.

The AV service is designed to offer the lowest possible latency for continuous media such as audio, video, and control loops (including "tactile internet"). Point-to-point links are formatted into "allocation periods", and each AV flow is assigned one or more packet "slots" in each period. To keep the latency to a minimum, the allocation periods of all links are phase-aligned; a very simple mechanism for this has been found to be effective.

Routing of AV flows is thus TDM-like, with a fixed latency for each flow and no need to examine the packets in order to route them. (This will potentially be useful for switching in the optical domain.) The service is similar to that from cross-point audio and video routers, and flows can be multicast by simply setting multiple outputs to "take" from the same input. Because each flow is assigned its own slots, it cannot be affected by traffic on other flows, and no further policing or traffic shaping is necessary.

The IT service is intended for the kind of bursty, unpredictable traffic that occurs with communication between computing processes, such as when downloading web pages. The maximum packet size is similar to Ethernet, but the minimum payload size is just one octet and no gap is required between packets. The packet header is similar in size to an MPLS label. In the prototype implementation, the service is "best-effort", with only one priority level, and only one output queue per port, because flows requiring QoS are assumed to use the AV service. However, it would be possible to support multiple traffic classes, resource reservation, traffic shaping, etc., in the control plane messages and routing tables if required; note that this does not require any additional information in the packet header. Similarly, the current implementation does not support multicasting of IT packets but that could be added if required, as could Explicit Congestion Notification (ECN).

The IT service may be connection-oriented, in which case it is one-to-one (and flows are usually connected in pairs to form a bidirectional path), or connectionless, in which case it is many-to-one; see clause 5.3.3 for more detail.

An IT packet can be carried over Ethernet or UDP by encapsulating it in an updated version of the format specified in AES51, and legacy packet formats can be carried over the IT service by adding an IT packet header to the front: see clause 5.7.

There are two other services: asynchronous and signalling.

The asynchronous service carries AV packets over other network technologies (see clause 5.7.2). The service they experience will depend on the facilities provided by the host network, and where packets pass from the asynchronous service to the AV service there is additional buffering which will add to the latency.

The signalling service carries control plane messages in IT packets. Signalling packets get a higher priority than the IT service, so an overload of IT packets cannot prevent control plane messages getting through. In the prototype implementation, control plane messages only pass between adjacent network elements, so they do not go through the IT packet forwarding mechanism; also, an overload of control messages from one neighbour cannot prevent control messages from other neighbours getting through. Protection for control messages between non-adjacent units (e.g. to or from an SDN controller) is for further study.

5.1.5 Choice of service for data transport

The IT service has similar characteristics to IP networks, i.e. transit time will increase if there is congestion, and packets may be lost if buffers in intermediate nodes overflow. It is therefore the appropriate service for transport protocols such as TCP to use.

However, a data transport protocol could also be designed to run over the AV service. The fixed bandwidth of an AV flow (which can be set to the maximum that the two endpoints and the intermediate network links can support) removes the need to adjust transmission rates based on the time it takes for acknowledgements to arrive. Packets will not be lost due to buffer overflow, so it is not necessary for the recipient to send frequent acknowledgements; in the case of a file transfer, notification of missing or corrupt packets or confirmation at the end of the transfer that the whole file has been correctly received should suffice, and these messages can be carried in the control plane, without needing to set up a reverse flow in the user plane. A file could also be multicast to a potentially large number of recipients. The benefits of this version would be most noticeable for large transfers; the information in the control plane messages would allow the network to route the flow over high-capacity services such as WDM if appropriate.

5.2 Packet formats

5.2.1 General

The data plane carries packets which consist of a header and a payload. Each packet belongs to a "flow", which defines the action to be taken at each point on its journey.

The payload is an octet string which is carried to the endpoint(s) without being inspected or altered, except that it may be possible for occasional bit errors or loss of an entire packet to occur due to faults in equipment or corruption of transmitted signals. Also, IT packets may be lost because of buffer overflow. Mitigation of errors (e.g. by forward error correction or by detection and retransmission) is assumed to be implemented in the end systems. Which measures are appropriate will depend on the application; for instance: in a data file transfer, accuracy is more important than timeliness, whereas timeliness is more important than accuracy for live audio and video. Control plane messages can report an estimate of the reliability of equipment and transmission media along the route, and hence the likelihood of errors.

The header codes the payload length and, for IT packets only, a local identification of the flow.

The routing mechanisms for the two kinds of flow are different, and in the prototype implementation follow different paths through the logic, although they share the links between network elements.

5.2.2 AV packets

The AV service was originally envisaged as supporting slots (and packets) of any size up to about 4K octets, with the slot size being in steps of 2 or 4 octets so that small packets (such as single audio samples) could be carried efficiently and large packets could be carried without fragmentation. This was partly to get away from one of the drawbacks of ATM, that its fixed-size cells were too large for some kinds of traffic and too small for others. However, it was found to have a number of disadvantages compared to fixed-size slots able to hold one packet each.

The maximum size of an AV packet needs to be small so that a flow's slot allocations can be more evenly spread, to avoid blocking off large areas in the allocation period (see clause 5.3.4.1), which would increase the latency experienced by other flows and hence also require a bigger packet buffer in switches. On the other hand, slots need to be at least as big as the internal data paths in a switch, which need to be quite wide to provide the throughput needed in today's networks. In the prototype implementation, after some experimentation a slot size supporting up to 63 octets of payload was chosen. This is distressingly close to the size of an ATM cell, but unlike ATM the unused bytes at the end of a slot are not wasted but are used for IT packets. Also note that an MPEG2 Transport Stream packet fits neatly into three AV packets. The header is a single octet formatted as:

1 bit:	odd parity
1 bit:	flag *f* (see below)
6 bits:	*n*, the number of payload octets

A null packet, or a slot that does not contain a packet, has $f = 1$ and $n = 0$. The flag *f* is not used for routing, but is available for use by endsystems; if it is used to guide reassembly of longer messages, it should be set to 0 in the last fragment of a message and 1 in others, so that adding or dropping null packets will have no effect.

Ideally this format would be used throughout the system. Note that the header does not change when the packet is forwarded.

5.2.3 IT packets

An IT packet header carries two pieces of information: the payload length and a "flow label", which is a locally-significant handle on the flow. In the prototype implementation, each of these is coded as a 13-bit value and a 3-bit CRC. It would be possible to also include a "congestion experienced" flag. The flow label, being local to the link on which the packet is transmitted, is changed each time the packet is forwarded, and in practice is the address of the entry in the recipient's routing table. The payload length is not changed, except that if a packet is forwarded between links that have different header formats the entire header may need to be replaced.

The MTU in the prototype implementation is 1 788 octets. A global minimum MTU needs to be specified, at least enough to carry 1 280-octet IPv6 packets, or maybe large enough to tunnel 2 000-octet Ethernet envelope frames without fragmentation.

5.3 Control plane procedures

5.3.1 Message format

Control plane messages are specified in ISO/IEC 62379-5-2 [i.2], and use a tag-length-value format. This makes it easy for recipients, even those implemented with small microcontrollers (such as some IoT devices), to parse the message and extract the information they need, while also making the encoding fully extensible. It is also more space-efficient than text-based coding formats.

5.3.2 Identifiers

5.3.2.1 Equipment identifiers

Each physical network element (switch or end equipment) has a 64-bit globally unique identifier or "unit id". This can be an EUI-64; EUI-64s always have 00 in the least significant 2 bits of the first octet, and additional forms with nonzero code points in those two bits are also defined, for instance for temporary identifiers that can be issued to end equipment that does not have its own, and for identifiers based on Private Enterprise Numbers instead of OUIs.

5.3.2.2 Call, route, and flow identifiers

Each flow (see clause 5.1.3) has a 128-bit globally-unique identifier partitioned into:

64 bits: unit id of the "owner", the unit that created the flow identifier;

32 bits: call reference;

7 bits: route reference;

1 bit: direction (1 = towards owner, 0 = away from owner);

24 bits: flow reference.

The first 96 bits form a "call identifier". The owner is responsible for ensuring that the call identifier is unique.

A "call" can be composed of several flows, for instance a television programme may be composed of separate flows carrying low-resolution video, additional information to create a higher-resolution image, several different audio streams, captioning text, and metadata; some destinations might only take a subset of the flows, e.g. only the low resolution video and one of the audio streams. Flows that are part of the same call have the same call identifier and are distinguished by their flow reference and direction.

Several copies of a flow may be transmitted over different routes, where this is required for resilience. The different copies have the same flow reference and are distinguished by their route reference.

The globally-unique flow identifier makes it easy to avoid setting up routes that include loops, and to detect whether routes that are duplicated for resilience actually follow separate paths.

Flow identifiers are only used in the control plane.

5.3.2.3 Addressing

A wide variety of types of address or identifier can be used. The called party can also be identified by a service it provides or a piece of content the caller wishes to access.

See clause 5.6 for more details.

5.3.3 Setting up routes

5.3.3.1 Procedure for connection-oriented model

The FindRoute control plane message type is used to set up a flow. The procedure is initiated by a unit (network element) which will be at one end of the route; it sends a FindRoute request message to one or more neighbouring units, which in turn process it and forward it on until it reaches a unit which will be the other end of the route.

With the small switches that form the prototype implementation, each unit simply floods the request to all its neighbours, except where that would form a loop. The globally-unique flow identifier makes loops easy to detect. In a network with larger switches, additional information, either from a central resource similar to an SDN controller or distributed by a peer-to-peer protocol, should be used to limit the number of neighbours to which a request is forwarded.

There are two replies to a FindRoute request: an immediate acknowledgement, followed later by either a FindRoute response or a ClearDown request, the latter indicating that the route cannot be set up. Further messages may be exchanged after the response: "confirmation" in the same direction as the request and "completion" in the opposite direction.

Processing of the messages includes setting up the user plane flow in the routing tables. In some cases this needs to be done by the later messages in the sequence, for example so that data cannot flow until authentication procedures have been completed (see clause 5.5). In other cases it is set up by the earlier messages to reduce the latency between initiating the request and data flow beginning.

5.3.3.2 Connectionless service

The connectionless IT service uses the same packet type and data plane mechanism as the connection-oriented IT service, but the flows are in general many-to-one, i.e. packets arriving at a switch from different sources may be forwarded on the same flow. The payload therefore needs to include information that will identify the sender, including an address to which replies should be sent. One use for this service is to carry IP datagrams.

Note that although these flows are used in a similar way to MPLS Forwarding Equivalence Classes, the way they are administered means there is no need for a "time to live" field in the packet header.

A packet for a destination for which there is no existing flow is encapsulated in a FindRoute request message and uses the same procedures as connection-oriented call set-up, returning the data plane flow label to be used in subsequent packets. If it reaches a network element that already has a suitable flow for the required destination, the packet is sent on that flow.

Network elements record when the most recent packet was forwarded on each flow, and clear it down if nothing has been seen for a specified time. Note that this is very similar to route caching in IP switches.

5.3.3.3 Additional information in FindRoute messages

FindRoute messages can include a wide variety of information both for the network and for the units at the ends of the flow, including information to help in choosing the route and to tell the recipient the format of the data; see ISO/IEC 62379-5-2 [i.2] for details. They can also include charging information, which might be used for micropayments for access to content or services as well as for traditional call charging.

5.3.4 Synchronization of AV flows

5.3.4.1 Slots

Each link between network elements is formatted into "slots", and the slots are grouped into "allocation periods". Each AV flow is allocated one or more slots per allocation period. The framing on the link shows where each allocation period starts and a flow's allocation is of the same set of slots in each period.

In the prototype implementation, an allocation period contains 1 936 slots and lasts 0,99968 ms, so the allocation repeats 1 000,32 times per second. The minimum allocation for a flow is one slot per period, i.e. 1 000 packets per second plus a tolerance for the source of the data having a clock that is up to 320 ppm faster than the reference used by the source of the frame timing.

A longer period would allow finer-grain allocations but increase the size of the routing tables, e.g. doubling the number of slots per period would halve the minimum number of packets per second but double the size of each port's routing table. A system-wide baseline period needs to be specified, but individual links can implement periods that are an integer multiple or submultiple of the baseline.

The flow to which an AV packet belongs is identified by the packet's location in the allocation period.

5.3.4.2 Frame alignment

To achieve the minimum latency at a switching point, there needs to be a fixed phase relationship between incoming and outgoing allocations. This is referred to in the prototype implementation as "tight" frame alignment.

When a link first comes up, it can only carry IT flows. Negotiation via SyncInfo control plane messages establishes whether the two sides have a common reference; if not, further negotiation arranges for the subnetwork on one side of the link to take its timing from the other.

The links used to convey frame timing form a spanning tree, but there is no need for a "spanning tree protocol" to configure the network; changes only occur when links come up or go down, and any link that is on the tree remains on the tree until it goes down.

5.4 Support for mobility

Any part of a flow can be re-routed without affecting the rest of the flow. Thus if a mobile device changes its point of attachment, e.g. moves to a different cell, the flows connected to it can be switched from the old cell to the new without the systems with which it is communicating, or higher-layer processes in the device itself, needing to be aware of any change.

When a server receives an incoming call it is supplied with the flow label to be used in packets to the client; it does not need to know the client's address and is therefore unaffected if the client's location changes.

5.5 Support for security

5.5.1 Authentication

The FindRoute messages that are used to set up calls and flows can include as much or as little identification of the caller as is required. Client and server can exchange authentication information in these messages, and reject the call if necessary.

There are also facilities for the network to report whether a call comes from a trusted source.

Using domain names directly in addresses (see clause 5.6) potentially allows DNS servers to be more resilient to "spoofing".

5.5.2 Denial of service

As noted in clause 5.1.4 above, AV flows have reserved capacity with which other traffic cannot interfere.

An overload of control plane messages coming into one port of a switch cannot stop control plane messages being received on other ports, and (provided the switch serves its ports in rotation) will have only a minimal effect on the service experienced by other ports.

FindRoute messages can include a specification of the throughput expected on each IT flow. A switch could monitor the actual throughput of each flow and tear down flows that misbehave, or apply other traffic policing measures.

5.6 Addressing and scalability

There are no "address" fields in packet headers.

The CalledAddress field in FindRoute control plane messages in ISO/IEC 62379-5-2 [i.2] supports a wide variety of addressing schemes, and there are a large number of reserved code points which would allow more to be added. As well as traditional addressing schemes such as IPv4, IPv6, and E.164, which identify an interface, and the 64-bit unit identifiers for physical equipment (see clause 5.3.2.1), other means of identifying the target of the call can be supported, such as content-centric addressing, which might connect either to the system that hosts the content or to some nearer device that holds a cached copy. Domain names could be used directly, instead of needing a separate process to convert them to IP addresses.

An address can be composed of an identifier preceded by one or more locators; the locators are processed in sequence, and each identifies the context within which the next part of the address will be interpreted. Examples of locators are the address of a gateway and the identifier of equipment that hosts a service. This allows addresses that have local scope to be used in global contexts.

The flow labels in IT packet headers are local to each link, and can be tailored to the requirements of different kinds of link, for instance by using a larger field on links that are expected to carry a larger number of flows. A group of IT flows can be routed through a core network as a single flow by simply adding another header, in a similar way to an MPLS "push" operation.

5.7 Interworking and migration

5.7.1 Definitions

A "physical link" is a connection on which frames are tightly phase-aligned as described in clause 5.3.4.2.

An "island" consists of network elements that are connected to each other by physical links.

A "virtual link" is implemented by tunnelling flows across other networking technologies.

A "gateway" is a connection to a network that uses a different technology.

5.7.2 Connecting islands via other technologies

IT packets are carried over virtual links by encapsulating them either directly in Ethernet or in UDP as shown in figure 5.1. Similar formats could be used with other services such as MPLS.

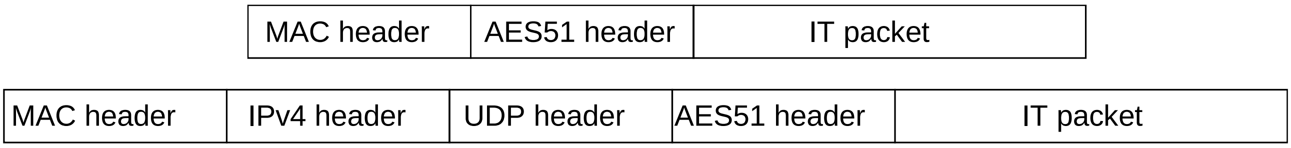


Figure 5.1: Encapsulation of IT packets

AV packets are carried by encapsulating them in an IT packet which is carried over a virtual link; at the receiving end there is a de-jitter buffer for each AV flow, and the label in the IT packet header shows to which flow the AV packets belong. A null AV packet is generated if the de-jitter buffer is empty when a packet is required for onwards transmission.

"Loose" frame alignment, whereby one island takes its frame timing from the other, is used to ensure that packets do not accumulate over time in the de-jitter buffer.

This provides a migration path whereby applications can be developed using the new technology before it is ubiquitous. The service experienced by AV flows will gradually improve as virtual links are replaced with tightly-aligned physical links.

5.7.3 Tunnelling other technologies across islands

In the current implementation, if terminal equipment which uses IP is connected to an Ethernet port, and elsewhere in the island there is a port connected to an IP network, a tunnel is set up in which each Ethernet packet is carried as the payload of an IT packet. This has the advantage of simplicity and works well in situations where all traffic between the terminal equipment and IP destinations can go through a single gateway. The terminal equipment can also set up a virtual link, through which it can access destinations (such as management agents) within the Flexilink network.

For more complex situations, where different IP addresses need to be routed through different gateways, the connectionless service (see clause 5.3.3.2) should be used to carry the IP datagrams, with the Ethernet MAC headers being stripped on entry and added on exit; this requires ARP to be implemented at the entry and exit points.

Tunnelling of MPLS has not yet been investigated, but should be straightforward.

This provides a migration path whereby parts of an existing network can be replaced with the new technology without affecting applications developed for the previous technology.

5.7.4 Other gateway functions

Gateways could also provide translation further up the stack, for instance by connecting RTP-based standards such as AES67 [i.20], EBU ACIP [i.21], and SMPTE 2022 [i.22] to AV flows, with interworking between SIP/SDP and ISO/IEC 62379-5-2 [i.2] signalling, and between PTP and the timing information in the AES51 headers.

5.8 Assessment against NGP key issues

An assessment of Flexilink against the criteria listed in annex A is given in table 5.1.

Table 5.1: Assessment of Flexilink against key issues

Issue ID	Issue name	Assessment	Validation Status	Explanation
1a	Addressing scalability	EN	IMPL	Addresses do not appear in packet headers, only in control plane messages, where a wide variety of addressing schemes (and other means of identifying a remote entity) can be supported, including identification of interfaces, equipment, or content, and the use of locators to define the scope within which an identifier is to be interpreted.
1b	Addressable entities	EN	IMPL	As 1a.
2a	Authentication	EN	BASIC	The control plane procedures for setting up flows allow the communicating parties, and network service providers, to require the exchange of authentication, authorization, location, and/or accounting information before the flow is connected in the data plane. The control plane messages also support the carriage of information on the trustworthiness of systems over which the data will be conveyed.
2b	Privacy	YES	IMPL	The control plane procedures for setting up flows support setting up a bidirectional flow between a client and a server without disclosing the client's identity or location to the server.
2c	Robustness	YES	IMPL	Any packets that are not part of a flow are ignored (see also 2a), and sources that are "jabbering" can easily be isolated. Flows passing over a link can be re-routed as soon as loss of signal is detected, whether caused by link failure or failure of the remote equipment.
3	Mobility	YES	BASIC	Any part of a flow can be re-routed without affecting the remainder of the flow. Thus a connection to a mobile device can be switched to a different cell without the device at the other end of the connection needing to be aware that it has moved.
4	Multi-access support (including FMC)	YES	IMPL	The identification of devices is independent of the routing mechanism in the user plane, so a device can freely use whatever access networks are available, including switching flows between mobile and fixed access networks.
5	Context awareness	YES	IMPL	Control plane messages can carry any indications of context that are required, supplied by the communicating entities or the network. Examples include: location of the UE (e.g. from GPS) or of its point of attachment; ability of the UE to support different media coding schemes or protocols; and available bandwidth, QoS, etc., in the network.
6a	Delay gaining access to a network	YES	IMPL	A request for access to a service or a piece of content can be routed to the nearest provider, thus supporting edge computing and local caching of content without requiring any special action by the UE. The connectionless service (see clause 5.3.3.2) supports caching of routes to frequently-used services, minimizing the time required to access them. Connection of a flow is a single transaction, whereas in an IP network additional transactions, for example DNS and ARP, are required.
6b	QoS for live media	YES	IMPL	The AV service provides guaranteed throughput with the lowest possible latency. As well as latency-critical applications, this benefits one-way communication (such as streamed media) by removing the need for congestion control, buffering, etc.
6c	Efficient use of channel	YES	IMPL	User plane packet headers are made as small as possible, carrying only the packet length and (in the case of IT packets) a locally-significant identification of the flow.
6d	Data transfer performance	YES	BASIC	The AV service can also be used for transfer of large data files, obviating the need for the congestion control mechanisms which limit the performance of TCP. A request for access to a service or a piece of content can be routed to the nearest provider without requiring any special action by the UE.

Issue ID	Issue name	Assessment	Validation Status	Explanation
7a	Centralized control	YES	BASIC	Routing decisions can be made centrally; flows have unique identifiers and their requirements are signalled explicitly in the FindRoute messages.
7b	Network Function Virtualisation	EN	BASIC	Decoupling addressing from user plane routing allows switches to partition the resources of a physical network to form a number of independent virtual networks.
8	IoT support	YES	IMPL	The AV service provides the ultra-low latency that will be required for some applications. It also provides very low jitter, reducing buffering requirements in devices that receive live streams. Packet formats and protocols are simplified, reducing the amount of processing power required in end devices.
9	Energy efficiency	YES	IMPL	Much less per-packet processing is required in switches than with systems in which user plane routing tables use globally-significant addresses and flows are identified by examining multiple fields in packet headers. This is expected to result in significantly lower power consumption for switching equipment.
10	e-Commerce	YES	BASIC	FindRoute control plane messages can include charging information, enabling micropayments for access to content.
11	Edge Computing	YES	IMPL	A request for access to a service or a piece of content can be routed to the nearest provider without requiring any special action by the UE.

6 Multi-access, Mobility-aware, Context-aware-Networking Protocol (M2CNP)

6.1 Overview

The IETF has defined the suite of IP protocols for the use of the internet over several decades, and evolved the protocol stack architecture to support the ISO 7-layer model for layers 3 and above [i.23].

However, the protocol was never designed with in-built security or mobility, and the performance of the layer 4 transmission layer protocols, particularly TCP, is not ideal for handling the heterogeneous communications paths that are used to connect to the internet today and their notable variability in performance, particularly when considering communication paths that include radio access links.

This situation is hardly surprising considering that the internet protocols were initially developed for predictable links that were at the time of design largely all fixed wire-line based interfaces. However this is not the case today where many different access technologies connect to the still fixed and largely wire-line or optical core transmission networks. Also, people now expect to operate devices over the internet with mobility and using a variety of different contexts, which was not the case when the internet started.

As such, a new internet protocol architecture approach that better accommodates end-to-end communication paths that include multiple access techniques and mobility is needed: a Multi-access Mobility-aware, Context-enabled Networking Protocol (M2CNP). Also, any such new protocol needs much improved basic security for accessing the internet.

6.2 System Architecture

6.2.1 Introduction

M2CNP envisages a packet based routed protocol architecture with the ability to embed protocol control messaging to provide basic protocol management functions for: security, context-awareness, transmission management, and mobility.

It is envisaged that M2CNP operates using enhanced routers that include the protocol management functions.

It is envisaged that M2CNP provides support for the existing fundamental protocols operating between:

i) user devices and internet servers;

ii) peer servers; and

iii) peer devices.

It is expected that M2CNP would be introduced initially to cellular based user devices and interworked to the legacy internet at the north edge of the cellular network and as the protocol architecture becomes more established then it would be rolled out into the internet so that the interworking nodes can be retired gradually as the protocol usage grows more widespread.

The M2CNP Architecture is illustrated in the following figure 6.1.

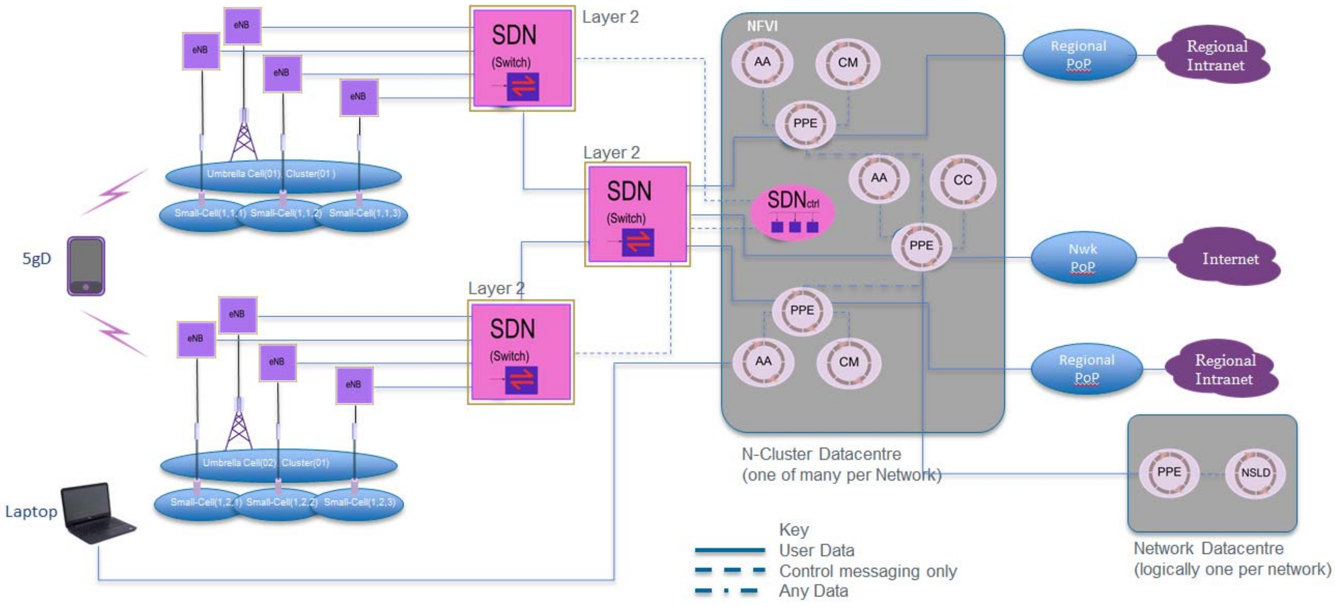


Figure 6.1: M2CNP Architecture

Applications and/or services, running at access network connected devices, communicate with each other or M2CNP or IP network side applications and services by using IPC interfaces towards the M2CNP communications network.

The access network connected devices may be connected via one or more access technologies at a time and are capable of mobility from one access point or Temporary Access Points Group, TAPG (for multiple access) to another.

The access technologies (Cellular LTE and Fixed) are illustrated with interconnecting Software Defined Networking (SDN) set of access switches that are configured as L1/2 switch tributaries towards the M2CNP network.

The M2CNP network consists of M2CNP Packet Processing Entities or PPE which are M2CNP routing entities that are selectively enabled with various protocol management capabilities of M2CNP and may be deployed in terms of scope in a similar manner to CE, PE, P scope routers as commonly understood in the legacy IP world.

The M2CNP based architecture is logically grouped into protocol management administrative "clusters" of M2CNP-aware access points, PPEs and logical protocol control entities.

Services and Applications (the M2CNP Access EndPoints AEP) that access the M2CNP network "associate" to a cluster in order to gain access to the communication services of that cluster and the other interconnected M2CNP clusters that the cluster has associative permissions to which to route.

PPEs (the M2CNP Network EndPoints NEP) that access the M2CNP network "associate" to a cluster in order to gain access to the communication services of that cluster and the other interconnected M2CNP clusters that the cluster has associative permissions to which to route.

There are key protocol management control, functional entities that may be deployed or enabled with a PPE called Access Agent (AA), Cluster Controller (CC), and Cluster Member (CM). All of these protocol management entities operate a group of Network Service (NS) procedures in order to set up basic Addressing, Authentication, and Security services via the NS(Associate) procedure in order to access the communication services of the M2CNP network and connect to other internetwork-associated M2CNP networks and gateways or access agents to other legacy IP networks.

6.2.2 M2CNP Protocol Management Entities

6.2.2.1 Access EndPoint

The Access EndPoint looks for M2CNP broadcasts from the AP, identifies the Access Point ID and Cluster-ID and then joins the M2CNP network for routing service by operating an NS[Associate] procedure towards the PPE(Cluster Controller), which always has the M2CNP address "CC".

The CC allocates the AEP to a CM or CC depending on its i) User Profile, ii) access network list reachability of connected Access networks (either by signal strength/quality assessed by the mobile as OK/Not-OK for radio access or physical connection assertion for fixed network access) and its assessment of current connected Network Profile.

As the AEP moves based on access reachability, it reports its new (broadcast-received-id's, user profile update) with an M2CNP, NS[Change-of-Address] message to the CC, which keeps track of the AEP mobility for network updates.

There is no handover within the cluster, only a move as the AEP just sends new uplink messages to the new reported Edge PPE at the CM or CC level it has been allocated by the CC with an Intra-Cluster Move(). The AEP keeps the same Cluster based M2CNP address whilst in the same cluster.

If the AEP usage/reachability changes notably (implying a need for a CC<-CM) or CM<->CC change) the AEP notifies the CC with an NS[Context-Update] message and the CC assesses and informs if a change is required with an NS[Intra-Cluster-Move] message.

If a NS[Change-of-Address] implies a change of Cluster to a new Cluster where the Old PPE(CC) is authenticated with the new PPE(CC) then the CC directs a cluster update to the AEP and forwards the NS information for the AEP to the new Associated Cluster. The New Cluster allocates the AEP a new M2CNP address in the new cluster and the new PPE(CM) allocates the AEP a CM, CC service Edge PPE which it sends back to the old cluster which commands the AEP to move with an NS[Inter-Cluster-Move] message.

The AEP informs any ongoing service and/or application peers of its change of cluster address whilst keeping its own Application or Service Address.

6.2.2.2 Access Point

Access points enabled for M2CNP network routing service provision, need to first NS[Associate] with their connected PPE(CC).

Each Access point broadcasts its M2CNP Access Point ID (AP-ID) and M2CNP Cluster-ID (Cl-ID) in an M2CNP, NS[Broadcast] message.

The Broadcast message uses the M2CNP, Cl-ID/AP-ID binding that it gets when it associates with its PPE(CC).

6.2.2.3 Packet Processing Entity

PPEs are either Edge PPEs or Network PPEs.

Edge PPEs are connected to at least one access network and are connected to at least one M2CNP network PPE.

Network PPEs are connected to one or more other PPEs to form an M2CNP cluster network and may also be connected to Edge PPEs

All Edge PPEs send periodic updates of M2CNP, NS[Broadcast] messages towards their connected access points in order to advertise M2CNP network capabilities, when their capabilities or Cluster membership changes due to network optimization.

6.2.2.4 Cluster Controller Functional Entity

The Cluster Controller is an entity that acts as the NS[Associate] point for each entity wishing to gain access to this cluster.

An Edge PPE(CC) supports AEPs that wants to gain communications access to a local cluster providing multi-access network service to the M2CNP network(s) by associating with the cluster of access points that is on the physical locale of the accessing AEP.

A Network PPE(CC) enables a NEP(PPE) that wants to gain communications access to another peer NEP(PPE) for the purposes of routing AEP messages by associating with that PPE(CC).

A PPE becomes a (CC) either because it is explicitly nominated because it has been configured to support CC service or it has implicitly asserted that it should elect to support CC capabilities as there are no other CCs in its L1/2 connected locale.

AEPs provide a base user profile to the CC in order that the PPE may decide the best PPE to provide M2CNP service to it.

6.2.2.5 Cluster Member Functional Entity

A PPE becomes a (CM) because it is explicitly nominated by the network management system or because it has implicitly asserted that there is a suitable PPE(CC) to associate with that it is connected to via its L1/2 connected locale.

A CM function provides an AEP, M2CNP network communications access to its connected PPE routing services during an association with the CC if the CC decides that this CM is best able to provide it M2CNP access based on its connected APs and User and Network profile at the time the AEP makes an association with the CC.

An AEP may signal to its associated CC at any time that its access or user profile has changed and this may result in the CC directing the AEP to move its servicing PPE to another CM or CC to provide better service.

6.2.2.6 Access Agent Functional Entity

The Access Agent (AA) is an optional service optimization enabled with M2CNP routing. The AA may be deployed either at the PPE(CM) or PPE(CC) in order to provide AA service optimization for either access connection level I the M2CNP architecture.

The AA is assumed to be a trusted node, optionally certificated by the PPE(CC) to the AEP for optimized service provision during association to the cluster and following any Edge PPE update thereafter.

The AA function is intended to act as an agent to an AEP that requires services such as optimized content delivery. In this case, the cluster associated AEP requests service directly to the AA and the AA terminates the M2CNP communication path with the AEP. The AEP is now able to action the request from the AEP with its full optimization capabilities and acts on behalf of the AEP thereafter or until the AEP releases the AA.

The AA provides such services as pre-fetching, content caching acceleration and interworking between M2CNP and IP.

6.2.2.7 Network Subscriber Location Database

Each Subscriber Cluster Update is stored on a Network Subscriber Location Database (NSLD) that may be used to find out which Cluster in the network they are currently associated with.

Each Cluster Controller keeps a record of which AP or TAPG each associated AEP is connected to.

Discovery of a particular AEP is operated by the CC, providing M2CNP location listings towards the Network Location Database and logging the AEP location updates.

6.2.2.8 Cluster Content Routing Database

Content is not indexed in M2CNP by address, but by historical request updates to a Cluster Content Routing Database CCRD from AAs based on previous routing experience and explicit input from the network operators. The Cluster Content Routing Database is then able to respond to content routing queries for content requests by name and/or ID and/or content tag from a PPE or AA to gain an indexed best next hop for the requested content according to content tags defined by the network operator or content SDO.

The database is operated on a cluster basis.

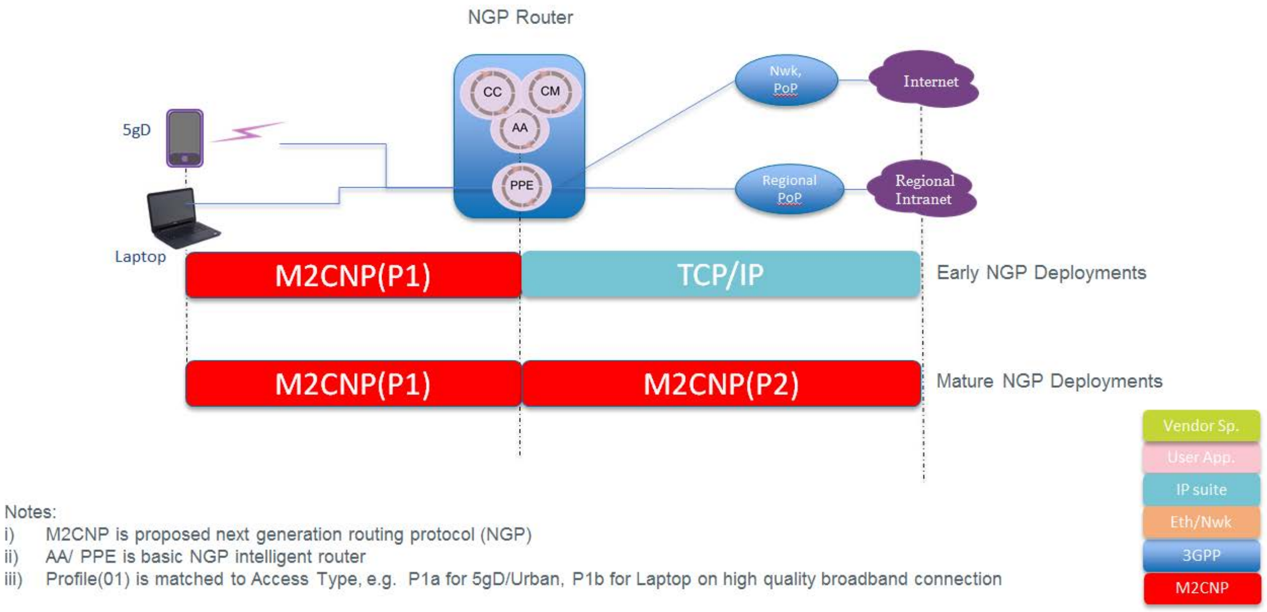
One CCRD(home) may query another CCRDD(neighbour/network) for best next hop inter-cluster routing.

6.3 Protocol Stack

The long-term target of the proposed M2CNP protocol architecture is to provide a more access aware network and internetwork solution that provides a L3/L4 replacement of what the IP architecture today delivers as illustrated in the M2CNP/IP high level protocol stack diagram of figure 6.2.

However, it is appreciated that whilst any new or evolved networking/internetworking architecture is being introduced there will be a phase of operation where the new architecture will have to interwork with the legacy architecture. As such, for the proposed M2CNP architecture it is envisaged that new evolutions of access networks would adopt a new networking/internetworking protocol architecture first and that for M2CNP this would be interworked towards the existing Internet using an AA/PPE acting as both an Agent and protocol interworking entity. Then, as the new protocol architecture is adopted in the internet the AA would purely act as an agent, when the AEP selects AA usage for content/web service edge optimization. This is as illustrated in the M2CNP/M2CNP high level protocol stack diagram of figure 6.2.

Non-AA access in the future would then natively operate only M2CNP routing ETE.



Notes:
i) M2CNP is proposed next generation routing protocol (NGP)
ii) AA/ PPE is basic NGP intelligent router
iii) Profile(01) is matched to Access Type, e.g. P1a for 5gD/Urban, P1b for Laptop on high quality broadband connection

Figure 6.2: Proposed AA Operation, Supporting Phased M2CNP Introduction

Figure 6.3 illustrates the protocol architecture for operating the Network Service, message set over M2CNP for protocol management control procedures.

In this example it is assumed that the access network is legacy LTE or an evolved New Radio (NR) 3GPP next generation Radio Access Network (RAN) and that the traditional Phy/MAC/RLC/PDCP protocols are reused or evolved for NR usage, from those used in LTE today.

It is further assumed that the PDCP protocol is evolved to support M2CNP at both the mobile device and the eNB/NR-eNB and that the eNB/NR-eNB provide ethernet or other L2 bridge forwarding towards the M2CNP access PPE.

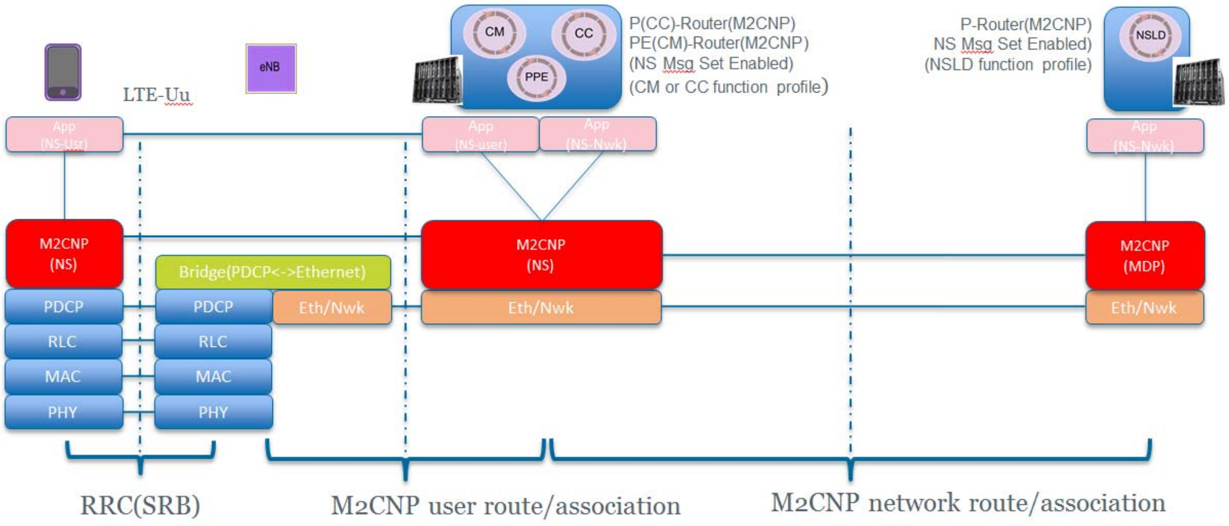


Figure 6.3: M2CNP Protocol Management Control Messaging (NS Message Set)

Figure 6.4 illustrates the protocol architecture for operating user plane packet routing over M2CNP with an AA acting as a transport gateway between M2CNP at the edge and the existing internet protocols.

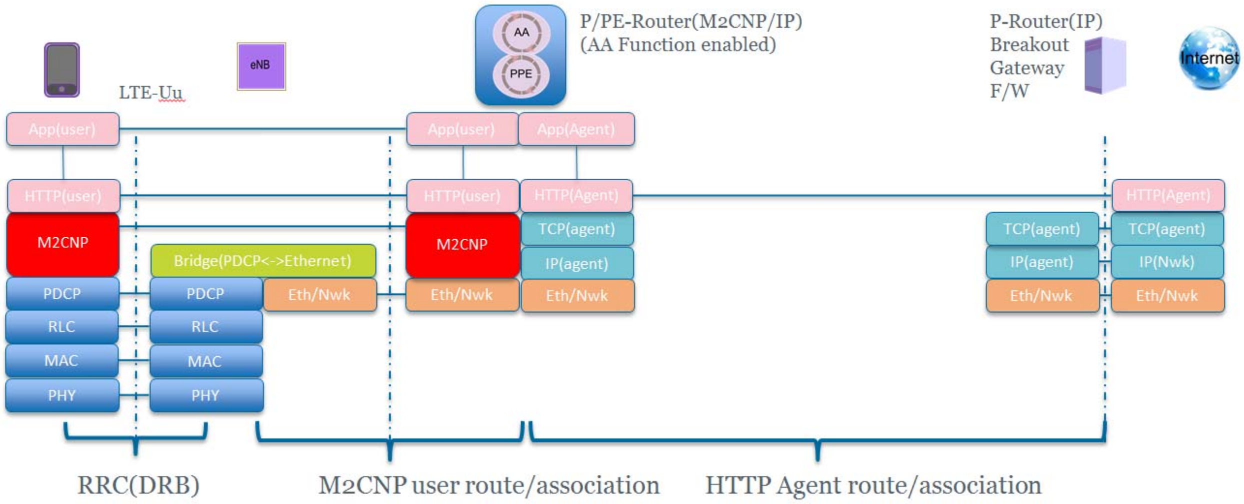


Figure 6.4: M2CNP User Plane

Figure 6.5 illustrates the in-network routing protocol architecture between routers and the control association between peer P routers and PE to P routers.

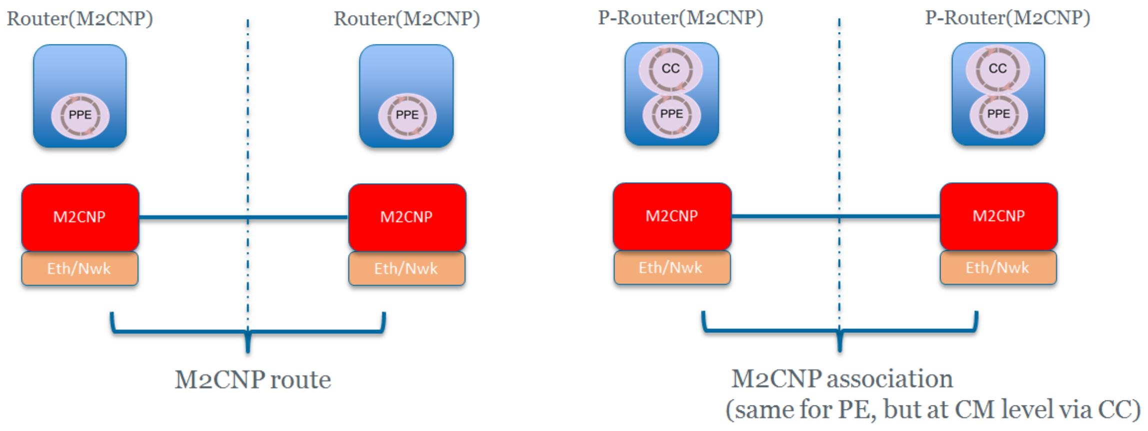


Figure 6.5: M2CNP Networking

6.4 Access Agent Functionalities and Benefits

6.4.0 Preview

In this clause, the functionalities of AA at the user plane are discussed in details. Furthermore, a high-level overview of AA's control plane functionalities is presented, where each of them is described in more details in the following clauses.

6.4.1 User Plane Functionality

6.4.1.0 General

The M2CNP architecture includes the following enhanced user plane functionalities:

i) Tailored treatment of access and core transmission parts of a network service request, which reduces latency and enhances resource utilization, hence improving user QoE.

ii) Context-driven intelligent content management: first, AA monitors and predicts context information on users, content and network. Second, the predicted context information is used to drive AA's decisions on intelligent content management. This improves the efficiency of content caching and prefetching when compared to context-less operations.

6.4.1.1 Network Service Request Handling

6.4.1.1.0 Types of service

Generally, the M2CNP architecture is capable of supporting multiple types of network services. The specific service that is being transferred over the M2CNP is identified using the 8-bit message set in the header.

In this clause, two representative examples are discussed - HTTP and voice services.

6.4.1.1.1 HTTP Request Handling

When a user initiates an HTTP request (for e.g. a webpage or a video), it is directly sent to the AA in the cluster over M2CNP.

When the AA receives the request, it checks if the requested content has been cached locally at the cluster. If yes, it serves the request immediately. If no, it acts as a reverse HTTP proxy and fetches the content from another content source, e.g. a server in the Internet or a neighbouring cluster. It then serves the user request when the content is delivered from the content source. In the meantime, it has the option of caching the content for future requests. More details on the caching operation is discussed in clause 6.4.1.2.

Note that when the AA is identifying a source to fetch the content from, instead of relying on a DNS server to resolve the request to a server (identified by IP address), it utilizes network context information (e.g. latency, congestion) to dynamically choose the most appropriate content source on-the-fly. More details on these contexts are discussed in clause 6.4.1.2.

6.4.1.1.2 Voice Request Handling

Voice requests are handled by operating M2CNP AEP discovery towards the NSLD to establish the cluster that the AEP is currently associated with and then querying the CC of the returned Cluster to find the address of the AEP/AP to route to within the cluster.

It is assumed that once the address of the M2CNP AEP is known then the calling AEP operates IMS or SIP based call setup towards them and RTP/RTCP for the user voice plane.

6.4.1.2 Context-Driven Intelligent Content Management

The AA supports the monitoring and prediction of the following context information:

- User context: this mainly refers to two aspects - users' mobility pattern, and users' preference on content consumption:
    - A user's mobility pattern can be implicitly reported during the association phase with the cluster. Afterwards, it can be updated periodically through MDP messages.
    - A user's content consumption preference is monitored by the AA as it keeps record of its historic requests and network activity. A few examples include the category of content (news, sports, etc.) that a user likes to request, the time duration of each content consumption session, etc.
- Network context: this refers to all the information that are related to the networks that the AA is connected to. Specifically, it may include the M2CNP network to the south of AA, as well as the public Internet to the north of it. The context information includes:
    - Latency: the transmission delay between the AA and each user/server.
    - Packet error: this includes events such as packet loss, out-of-order packet arrivals, as well as congestion event notifications (if ECN-like schemes are used). Such events are monitored between the AA and the users, as well as between the AA and the content servers.
    - Throughput: this refers to each user's download throughput, as well as the AA's throughput when downloading files from remote servers
- Content context: this applies to content applications such as web pages and videos, and it mainly refers to the popularity of each piece of content. The scope of the popularity can be within each cluster, or neighbouring clusters can share their content popularities (e.g. top 100 content).

The AA uses the multi-dimensional context knowledge above to perform intelligent content management operations. Specifically, it involves the following:

- Prefetching: the AA may use user context to predict the user's content demand in the near future, and prefetch the content to local storage before the user requests it, so that the user can experience low latency when actually requesting the content from AA. For example:
    - During a video streaming session, the AA may pre-fetch ~30 s worth of video ahead of the user's playback progress. It may pre-fetch more if the user's signal strength is (or is predicted to be) weak, hence benefiting more from low-latency local content access.
    - When a user requests a webpage, the common practice is that the user sends hundreds of HTTP requests, one for each (small) object in the webpage (such as scripts, icons, etc.). With the AA, the user just needs to send one request for index.html. The AA will then analyse the webpage structure and dependencies among hundreds of objects, pre-fetch all of them from a content source, push them to the user, hence saving the user from sending hundreds of requests.

- Caching: the AA can analyse the content popularity within its cluster (and nearby clusters, if possible), and choose to cache popular content when they are fetched from other servers. Besides popularity, AA's caching operation takes other context into account as well. For example, in a video streaming application, consecutive video segments are treated as a group when being considered as cache candidates.

- Multiple parallel TCP connections: it is known that using multiple TCP connections in parallel to download a content item leads to better overall performance in a network environment that has packet loss. The AA can use its network context knowledge to dynamically decide the optimal number of TCP connections that should be established for each download session, hence optimizing user's download performance.

6.4.2 Control Functionality

6.4.2.1 User Association

When a new user joins a cluster, it needs to "associate" with the cluster, and the AA performs a key role in this process when content optimization is selected.

6.4.2.2 User Authentication

AA is responsible for authenticating user's identity under the following scenarios:

- Initial attachment: when a new user joins the AA's cluster, and the AEP requests AA service, the AA authenticates the user during the attachment process.

- When a user moves between different eNBs within a single cluster, the new eNB may request the cluster's AA to authenticate the user.

6.4.2.3 Address Translation

For each user, the AA maintains a binding between its 16-bit cluster address and its user identity information. The AA is responsible for the translation between them as well.

6.4.2.4 Intra-Cluster Mobility

See M2CNP Protocol Management Entities.

6.4.2.5 Inter-Cluster Mobility

See M2CNP Protocol Management Entities.

6.5 Protocol Field Structure/Addressing

The N2CNP protocol header structure is as illustrated in the following figure 6.6.

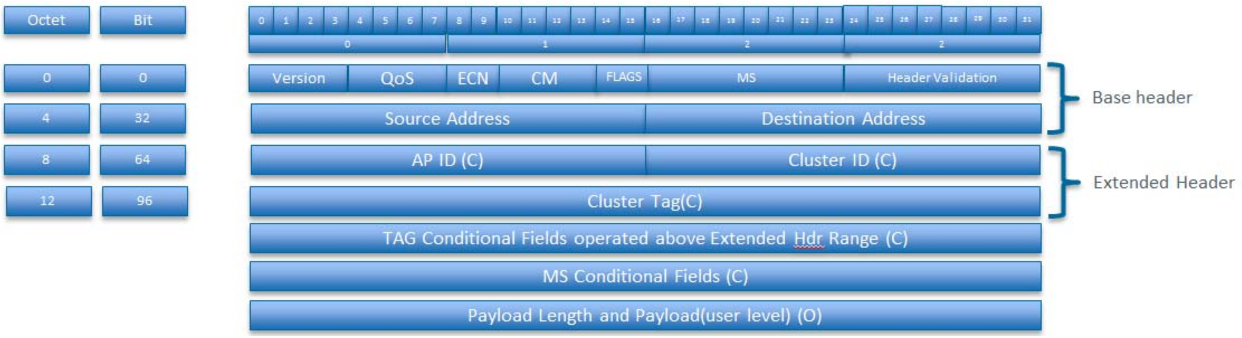


Figure 6.6: M2CNP Protocol Fields

The M2CNP protocol fields are explained in table 6.1.

Table 6.1: M2CNP Protocol Fields

Field Name	Field Abbreviation	Usage	Bits	Field Description
Version	Version	M	4	16 versions per Epoch, ~25 years.
Quality Of Service	QoS	M	4	16 x QoS profiles, pertinent to usage per version.
Explicit Congestion Notification	ECN	M	2	node to explicit node-node congestion notification.
Congestion Management Policy	CM	M	4	16 different policy options per version.
Header Flags	FLAGS	M	2	00: No extension header local intra-cluster routing, 01: Contains Extended Header, inter-cluster routing, 10 Global Routing required, 11 Spare.
Message Set	MS	M	8	MS(01) is Network Service (NS) management control for "Association" and "Mobility" handling.
Header Cyclic Redundancy Check	CRC	M	8	8-bit CRC.
Source Address	Src	M	16	M2CNP source address of application for this association allocated to user by CC entity within source cluster.
Destination Address	Dest	M	16	M2CNP destination address of application within destination cluster.
Access Point ID	AP-ID	C	16	M2CNP access point ID that uniquely identifies the AP within the Cluster.
Cluster ID	Cl-ID	C	16	M2CP Cluster ID, used to identify the address of the cluster that the user is currently associated to or trying to associate to (1 to N Access Points: Wi-Fi, Cellular, mmm-Wave or Physical port (e.g. Ethernet).
Cluster Tag	Cl-Tag	C	32	Token that is generated for the communicating app/process on associating to the network that temporarily binds the "User-ID" in the Network (e.g. IMSI or Temporary ID of app or server-app, and PLMN-ID, in sync with AuC/MS-Encryption/MS-Integrity and may be updated on Cluster Handover or AuC refresh. Included with all NS messages and when extended header is required.
…TAG Conditional Fields		C	…	Operated above Extended Header Range. These fields are all LV coded and the tag determines the Type depending on Access/Network technology connected Octet aligned.
… MS Conditional Fields		C	…	Operated above Extended Header Range. These fields are all TLV coded Octet aligned.
Payload Length		C	16	Included if there is a payload and maximum allowed is up to 16,384 octets minus header. First Octet specifies the word length to count in 8, 16, 32, 64, 256 Second Octet specifies number of words in payload.
Payload	Payload	O	…	MS Conditional Fields operated above Extended Header Range.
NOTE: Field usage: M = Mandatory [default], C = Conditional, O = Optional.				

6.6 Security

The M2CNP protocol NS message set is envisaged to operate both mandatory and optional security features during the initial NS[Association] of either a) an AEP connecting to an M2CNP access PPE or b) mutually associating PPEs.

The mandatory security features of M2CNP are:

i) Mutual Identification and Authentication of the AEP and the Access PPE (or 2 PPEs).

ii) Encryption of the NS Message Set fields.

iii) NS Message Integrity Checking.

These security features are managed using keys issued by the camped-on CC at time of association. The CC obtains these keys from the home network connected Authentication Centre (AuC) at the NSLD.

The CC maintains a log of all of the currently allocated keys on a per cluster basis in order to keep the key search and revocation list as short as possible for the scope of the Cluster

Keys are transferred from CC to CC inter-cluster mobility.

Keys will need to be refreshed on a periodic basis and potentially other critical triggers such as every N x NS procedures or M x Mbytes of transmission using security procedures over the NS message set.

The optional security features of M2CNP are used as follows:

i) ETE Payload Encryption.

If this option has been selected by an EP, it is indicated in the TAG at association, which is included in the first packet of every new peer session thereafter and then its key is used in encryption setup for packets from itself and vice versa in the other direction. The public key is shared from that EP to setup encryption to its peer.

6.7 Routing

The M2CNP protocol operates using 4,096 million temporary addresses per network (Source Address x cluster Address) which is roughly half of the ambitious 100,000 devices that ETSI estimate are needed per AP for IoT and MBB users.

Routing is operated using a scalable 3 level addressing system as shown in table 6.2.

Table 6.2: Levels of addressing

Level	Scope	Addressing Required
Locale	Within Cluster	M2CNP-Address, Base Header
Network	Within Network	(1) & Extended Header
Global	Inter - Network	(2) & Network PLMN-ID

The lowest level of addressing is included in every packet.

Network and Global level routing information additions per packet are negotiated either in Cluster or in Network or between Networks as Flow setups (Tag, indexes the |User-ID| and |Src-Nwk|, e.g. IMSI and PLMN. |User-ID| and security bindings are referenced using the Tag and [Cluster-ID] and [Src-Nwk].

When a communicating device needs to find out where its peer is, then it can do one of the following:

Explicit Address Routing	Source knows the full or partial address of the user a layer protocol needs to find a peer.
OTT Look-up	Source/Router uses an OTT registration and Database lookup for the next hop.
NSLD Query	Source application executes an NS Subscriber Query towards the NSLD.
Content Route Look-up	Source/Router uses a local app compliant content server to look up the next hop.

6.8 Message Sets

6.8.0 General

The M2CNP protocol includes a field that enables management control messages to be transported between communicating EPs.

6.8.1 Network Services

Message Set 01 is Network Services or NS, which enables a set of functionality that the M2CNP protocol uses operate protocol management between EPs using the following procedures:

Protocol Maintenance

Broadcast( )
APs broadcast: Nwk ID, Cluster ID, AP ID. Cluster-M2CNP address is (0).

Association( )
used to determine the initial CC allocated M2CNP address within cluster;

authenticate with CC/NMLD/AuC;

declare base context information in MS field extensions;

determine optimal CC, CM to camp on and direct by CC/CM address;

CM address is M2CNP(1 to 1024) range.

Context-Update( )
If the EP context changes then the EP reports this change of reachability or context to the CC using this message.

Change-of-Address( )
EP reports reception of new broadcast messages that meet a set of criteria for update to the CC.

The Change of Access address may be singular or TAPG based.

Tag-Forward( )
Forwards the Tag information about the user from an old CC to a potentially new CC.

Mobility

Intra-Cluster
On a peer EP moving from AP(a) to AP(b) within the same cluster, then the peer EP reports to the old CC a better candidate at a new CM/CC and reports Change-of-Address( ) to the CC with the updated base context info. The CC decides if a move is required and commands an Intra-Cluster-Move( ) to the EP. The revised Tag information is updated at the CC.

The Tag value is not changed.

The Tag information at the CC is revised with (AP-ID).

The AEP M2CNP address is not updated.

Also a ContextUpdate may result in the CC commanding the EP to intra-Cluster Update from CC to CM or vice versa based on change of reachability and/or context e.g. change of current type of transmission or mobility entropy required.

Inter-Cluster
Similar process to Intra-Cluster process in that the EP reports a Change-of-Address( ) to the old CC.

The old CC Tag information is forwarded to the new CC using a Tag-Forward( ).

The new CC assigns a new Tag Value for the EP towards the old CC using a Tag-Forward-Response( ).

The old CC commands an Inter-Cluster-Move( ) towards the EP.

The EP joins the new cluster with the new Tag Value.

The EP M2CNP address value is updated for the new cluster.

Higher layer sessions are re-setup/re-started (as restart is fast on 5G) < 20 ms (1 voice frame, min 10 Mbit/s at M2CNP layer).

6.9 M2CNP assessment against criteria

An assessment of M2CNP against the criteria listed in annex A is given in table 6.3.

Table 6.3: M2CNP assessment against key issues

Issue ID	Issue name	Assessment	Validation Status	Validation Results & Notes	References	Additional Notes
1a	Addressing scalability	YES	BASIC	Basic analytic assessment of addressing is estimating 40 to 80 % header reduction and largest Routing table reduction of 40 to 65 % as c.f IP.		
1b	Addressable entities	YES	BASIC	4 layer structured dynamic addressing used in association with temporal binding of static IDs (Intra-Cluster, Inter-Cluster, Operator, Con.		
2a	Authentication	YES	PoC	PoC of Flat Distributed Cloud network realized with 20 users, multiple routers and Context Aware Enrolment applications complementing Cellular Security.	See: http://www.surrey.ac.uk/5gic/flat-distributed-cloud	
2b	Privacy	YES	NONE			
2c	Robustness	EN	NONE			
3	Mobility	YES	PoC	Simulations completed and taken to PoC stage for Cluster mobility. Inter-Cluster PoC mobility in progress		
4	Multi-access support (including FMC)	YES	PoC	Multiple Access technologies demonstrated: Cellular, Wi-Fi & Fixed.		
5	Context awareness	YES	PoC	FDC implemented demonstrating context awareness in a cellular framework and used to complement mobility.		

Issue ID	Issue name	Assessment	Validation Status	Validation Results & Notes	References	Additional Notes
6a	Delay gaining access to a network	PART	PoC	Significant work done on Access delay including RRC IE caching and memoryfull signalling reduction/RACH ACRA radio enhancements proposed in association with NAS/ Enrolment extension for network level access. 30 ms access delay after RRC establishment, c.f 100 ms to 2 s for Cellular NAS Attach and 100 ms to 500 ms for Service Request.		
6b	QoS for live media	YES	NONE			
6c	Efficient use of channel	YES	NONE			
6d	Data transfer performance	YES	PoC	MEC server work demonstrates that Context Aware dynamic DASH control provides significantly enhanced Video performance for HD/4K transmission.	See references for Issue ID(11).	
7a	Centralized control	YES	NONE			
7b	Network Function Virtualisation	EN	PoC	UoS have demonstrated ETE (UE to intranet, internet, UE) NFV implementations of LTE/EPC and benchmarked TCP/IP virtualisation using OTS platforms. Typical UP latency additions ~5 ms c.f static switch implementations. Now moving to enhanced protocols to demonstrate improved Virtualisation latency.		
8	IoT support	EN	PoC	Demonstrated IoT-on-the-go for cellular using UE as gateway.		
9	Energy efficiency	EN	NONE			
10	e-Commerce	EN	NONE			

Issue ID	Issue name	Assessment	Validation Status	Validation Results & Notes	References	Additional Notes
11	Edge Computing	YES	PoC	See answer for 6d. Video enhancement enabled using MEC server adopting M2CNP techniques.	IEEE, 2016, ACM, "QoE-Driven DASH Video Caching and Adaptation at 5G Mobile Edge", C.Ge [i.24], etc. Forthcoming IEEE CCNC p January 2017, Enabling Context-aware HTTP with Mobile Edge Hint, P.Qian [i.25], etc.	

Annex A:
Assessment against NGP requirements

For each technology, the "issue detail" column should be replaced by at least three columns, the first headed "Assessment" and containing one of the following codes indicating the extent to which the issue is addressed:

- NO = The issue is present in the technology to a similar degree as in current networks.
- PART = The issue is present in the technology but to a lesser degree than in current networks.
- YES = The issue is fully resolved.
- N/A = The technology's scope does not include the parts of the system to which the issue relates.
- EN = Enabling: the technology's scope does not include the parts of the system to which the issue relates, but it includes features which help other technologies to resolve the issue.

The second column should be headed "Validation Status" and contain one of the following codes indicating the extent to which this aspect of the technology has been demonstrated to address the issue, or be blank if the code in the preceding column is "NO" or "N/A".

Additionally the assessment should include a statement of Validation Status for the technology as follows after the issue "Assessment" column, the codes for this "Validation Status" are as follows:

- NONE = Currently, purely theoretical technology proposal.
- BASIC = A paper study of the technology has been conducted with simple analysis of the associated protocol information elements and protocol behaviours against typical use cases and set of typical scenarios.
- SIM = Simulation of the technology has been completed for its protocol information elements, and protocol behaviours against typical use cases and set of typical scenarios.
- IMPL = Small scale proof-of-concept implementations have been realized for the technology to prove this particular aspect.
- PoC = As IMPL, where the system has been tested with 5-10 real users, multiple router implementations, and at least 3 different types of service.
- VAL = Large scale validation implementations have been realized for the technology to prove this particular aspect. Validation stage includes: 20+ real users, multiple router implementations, at least 3 different types of service and at least 2 different types of access technology.

Table A.1: Template for assessment of technologies against key issues

	Issue name	Issue detail
1a	Addressing scalability	The way in which the technology decides how to forward each packet should not place limitations on the size of network.
1b	Addressable entities	The technology should allow entities other than network interfaces to be identified, and support communication with them.
2a	Authentication	The technology should support verification of the identity of a communication partner, including resisting "spoofing" attacks.
2b	Privacy	The technology should allow a communication partner to remain anonymous.
2c	Robustness	The technology should resist attacks such as denial-of-service, and should be self-healing when equipment failures occur.
3	Mobility	The technology should support efficient re-routing when points of attachment change (e.g. in mobile handover).
4	Multi-access support (including FMC)	The technology should allow devices to access the network via several different sub-networks simultaneously.
5	Context awareness	The technology should support exchange of information regarding conditions that could affect decisions regarding the kind of service that it is appropriate to offer or provide.
6a	Delay gaining access to a network	The technology should minimize the time needed to register a device with an access network.
6b	QoS for live media	The technology should support low-latency (~1 ms) digital media, including bandwidth reservation and elimination of packet loss due to buffer overflow.
6c	Efficient use of channel	The technology should minimize the total number of transmitted bytes needed to convey a given piece of user data.
6d	Data transfer performance	The technology should avoid the various issues that have been identified with TCP, e.g. performance of its congestion control mechanisms on mobile networks.
7a	Centralized control	The technology should support centralized control of routing (e.g. SDN).
7b	Network Function Virtualisation	The technology should support emerging features such as Network Slicing and Self-Organizing Networks.
8	IoT support	The technology should support the connection of large numbers of devices, many of them having limited memory and processing capability and/or a requirement to minimize power consumption.
9	Energy efficiency	The technology should minimize the amount of processing required in network equipment and end-systems.
10	e-Commerce	The technology should support apportioning of economic value, including accounting facilities that allow licence fees and other service charges to be recorded, in a way that is scalable, traceable, and cost-effective.
11	Edge Computing	The technology should allow requests to access services or content to be routed automatically to local proxies or caches.

Annex B: Authors & contributors

The following people have contributed to the present document:

Rapporteur:

- John Grant, Nine Tiles

Other contributors:

- Eduard Grasa, i2CAT
- Chang Ge, University of Surrey

Annex C:
Change History

Version	Information about changes
1.1.1	First publication after approval (30 September - 2 October 2011; Prague)

History

Document history		
V1.1.1	March 2017	Publication