Developing an Implementation Framework for the Future Internet using the Y-Comm Architecture, SDN and NFV

Glenford Mapp
School of Science and Technology
Middlesex University,
Hendon, London NW4 4BT
Email: g.mapp@mdx.ac.uk

Fragkiskos Sardis
Department of Informatics
Kings College London
London, WC2R 2LS, UK
Email:fragkiskos.sardis@kcl.ac.uk

Jon Crowcroft
Computer Laboratory
William Gates Building
JJ-Thomson Avenue
Cambridge, UK
Email:jon.crowcroft@cl.cam.ac.uk

***Abstract*—The Future Internet will provide seamless connectivity via heterogeneous networks. The Y-Comm Architecture is a reference model that has been developed to build future mobile systems for heterogeneous environments. However, the emergence of Software Defined Networking and Network Functional Virtualization will allow the implementation of advanced mobile architectures such as Y-Comm to be prototyped and explored in more detail. This paper proposes an implementation model for the Y-Comm architecture based on these mechanisms. A key component is the design of the Core Endpoint which connects various peripheral wireless networks to the core network. This paper also proposes the development of a Network Management Control Protocol which allows the management routines running in the Cloud to control the underlying networking infrastructure. The system being proposed is flexible and modular and will allow current and future wireless technologies to be seamlessly integrated into the overall system.**

***Keywords*—Future Internet, Software Defined Networking, Y-Comm Framework, Quality-of-Service, Cloud Computing, Core and Access Networks**

I. Introduction

The Internet of the Future will place new demands on common networks. There will be support for ubiquitous communication via heterogeneous networking. Users will expect networks to support access to data using the 4As paradigm: anytime, anywhere, anything, anyhow. In order to facilitate these demands, network evolution is taking place which divides the Future Internet into core and peripheral networks. These two components are taking two different evolutionary paths: the core of the network will be dominated by the deployment of fast optical networks while peripheral networks will support a number of different access technologies including WiFi, LTE, 5G and WiMax.

In addition, the need to support a diverse set of applications with different Quality-of-Service (QoS) requirements means that support for QoS needs to be incorporated into future networks. In this new environment, QoS will also include security requirements and the system should be able to balance between the security and QoS concerns. Y-Comm [1] is an architecture that has been designed to build heterogeneous mobile networks. It attempts to integrate communications, mobility, QoS and security into a single platform. It divides the Future Internet into two frameworks: the Core Framework and the Peripheral Framework.

However, the emergence of Software Defined Networking (SDN) [2] means that it is now possible to build mechanisms that can implement the Core Framework in Y-Comm leading to a workable prototype of the Y-Comm architecture. This paper contributes towards this goal by first exploring the design of an Implementation Framework for Y-Comm, it then defines the operational structure of the Core Endpoint, a key part of the Y-Comm design. Finally, the functionality of a Network Management Control Protocol (NMCP) which links the network management routines running in the Cloud using NFV to the underlying networking infrastructure at the Core Endpoint is discussed.

The rest of the paper is structured as follows: Section 2 looks at related work while Section 3 explores the concept of network evolution leading to the Implementation model for Y-Comm. Section 4 details the operational structure of a Core Endpoint. Section 5 looks at how connections are made and Section 6 examines how handover is specified. The paper concludes in Section 7.

II. Related Work

Other network architectures for mobile systems such as Hokey [3], Ambient Networks [4] and Mobile Ethernet [5] have also been explored. HoKey looked at issues of secure handover in heterogeneous networks while Ambient Networks concentrated on supporting seamless connectivity in diverse networks. Mobile Ethernet adopted the Core/Peripheral structure like Y-Comm but assumed an Ethernet-type Core. A comparison of these systems indicate that Y-Comm offers the most functionality and flexibility [6] while integrating various key mechanisms [7], [8].

III. Network Evolution

The Internet is currently evolving. Instead of large, global, but individually-managed networks, a core network is being deployed which is fast and getting faster with peripheral wireless networks situated at its edges. A Core Endpoint is an entity which is at the edge of the core network and is used to connect different types of wireless systems. A mobile node, which is now usually a smart phone, can have several interfaces including WiFi, LTE and WiMax. Vertical handover techniques between different interfaces are used to maintain the connection between the mobile node and the core network via the Core Endpoint as the user moves around. The Y-Comm architecture is an architecture for mobile computing. The architecture specified in [1].

A. Towards an Implementation Model using Software Defined Networking and Network Function Virtualizaation

Various papers have shown how the Core Framework can be mapped onto current core network technologies such as 3G [9], UMTS [10] and LTE [11]. However, new technologies such as SDN, NFV and Cloud systems make it possible to implement the key layers of the Core Framework directly. Thus the use of these mechanisms will allow Y-Comm to be implemented using a Network-as-a-Service (NaaS) [12] paradigm. Thus it is possible to design a set of co-ordinated mechanisms as an Implementation Framework for Y-Comm as shown in Figure 1

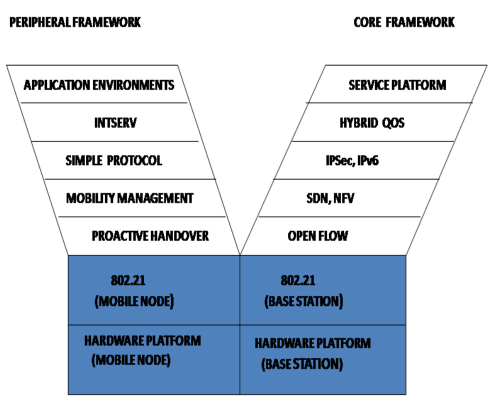

Fig. 1: The New Implementation Model for YComm

The Peripheral Implementation Model include:

- **IEEE 802.21 mechanism**: This layer, Layer 2, uses the IEEE 802.21 mechanisms [13] to control the different wireless interfaces. Higher layers can send commands to bring up or bring down links and network interfaces signal events via L2 triggers about the availability of individual networks.
- **Proactive Handover**: This attempts to do proactive handovers using Time Before Vertical Handover (TBVH) and Network Dwell Time (NDT) parameters based on data from the mobility management layer.
- **Mobility Management**: There are several mechanisms that can be used for monitoring the mobility of mobile nodes. This is based on triggers from network interfaces as well as GPS readings and parameters.
- **Simple Protocol**: The Simple Protocol (SP) [14] is a new transport protocol between the Mobile Node and the Base Station. SP also uses ECHO packets to determine the bandwidth and latency of individual connections. This allows applications on the mobile node to react to changes in the Quality-of-Service due to core network changes or vertical handovers.
- **IntServ**: This is a version of IntServ which allows the mobile node to specify its QoS requirements on a per-flow basis.
- **The Application environment**: This allows applications to use an enhanced IntServ specification to specify its Quality-of-Service requirements on individual connections.

The Core Implementation Model include:

- **Access Point/Base Station Controllers running 802.21**: This allows the Base Station and Access Routers to be controlled using an enhanced IEEE 802.21 protocol.
- **OpenFlow**: This is used for reconfigure the access network routers depending on the mobility of the user.
- **Software Defined Networking (SDN) and Network Functional Virtualization (NFV)**: This is used to implement the mechanisms and services for the Network Management Layer and is explored in detail below.
- **IPSec, IPv6**: The TCP/IP Suite is maintained but IPSec [15] is used to ensure that secure tunnels can be set up between Core Endpoints.
- **Hybrid QoS**: This is used to integrate IntServ and DiffServ mechanisms. The mapping between IntServ and DiffServ parameters is done at Core Endpoints.
- **Service Platform Layer**: This implements a Mobile Services platform in which servers can be moved closer to mobile users.

IV. Building a Core Endpoint

The key to realizing a prototype Y-Comm testbed is the design and development of a real Core Endpoint based on the Implementation model described above. The management functions run in the Cloud uisng NFV. They communicate with each other using a lightweight Remote Procedure Call (RPC) running over SP.The functions include:

- **Core Routers**: These routers attach the Core Endpoint to the core network. They are used to route data between the core and peripheral networks.
- **Routing Engine**: This is used to manage routes between the Core Endpoints and mobile nodes as well as routes between Core Endpoints.
- **Network Topology Manager**: This service maintains the topology of the network in the local area. This service handles all changes to network infrastructure including changes of points of attachment of mobile

nodes due to handover, the installation of new routers and switches as well as link and network failures.

- **Event and Trigger Manager**: This service deals with events and triggers that happen at the lower layers of the architecture.
- **QoS Broker**: The QoS Broker is used to manage the QoS issues for Core Endpoints. This service ensures that the incoming or outgoing flows do not exceed the network capacity of the Core-Endpoint and does the mapping between IntServ flows and DiffServ traffic classes.
- **The AAAC Service**: This manages the security of the Core Endpoint: This includes all aspects to do with the use of the networking infrastructure.
- **Server Platform Services**: This is used to deploy additional mobile services at the Core Endpoint.
- **Third Party Services**: These services allow clients of mobile operators to use the local Cloud infrastructure to offer bespoke services to their clients. This may be extra computing or storage facilities.

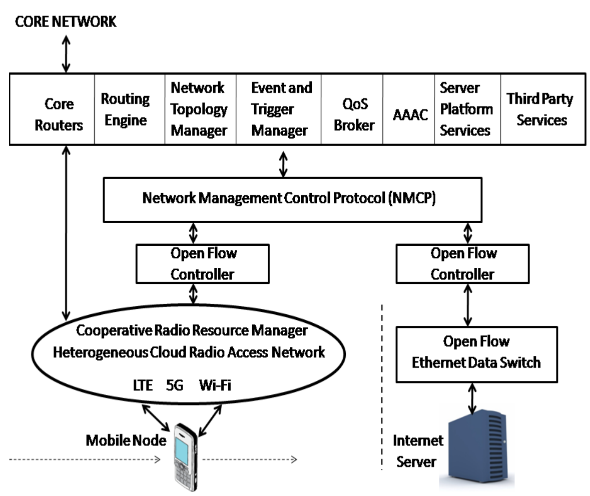

Fig. 2: Showing the Operational Structure of a Core Endpoint

The proposed system will evolve with the development of new mobile technologies as shown in Figure2. This evolution will allow the smooth management of local heterogeneous networks by the Heterogeneous Cloud Radio Access Network (H-CRAN) and the Cooperative Radio Resource Manager (CRRM). H-CRAN will be used to access and control individual networks which CRRM will be used to optimize the overall radio access environment. CRRM will also support OpenFlow and hence the upper layers of the architecture can remain unchanged. The use of NFV and SDN at the Core Endpoint will also facilitate the softwarisation of radio technologies as proposed in 5G with the deployment of Cloud-RAN [16] at the Core Endpoint.

V. The Network Management Control Protocol

The Network Management Control Protocol is used to allow the high-level network management functions and services discussed above to control and manage networking infrastructure. NMCP can be implemented by directly translating it into OpenFlow commands or by using a number of emerging Northbound APIs.

NMCP also supports various communication entities: An endpoint is a device that can send or receive data. Mobile nodes and servers are examples of endpoints. A link is a connection between two entities. A path is a connection between two endpoints. A path is made up of one or more links. A data-flow is the movement of data between two endpoints. Data-flows allow us to specify the actual data flowing along physical links. Making these flows first-class objects allows for much greater flexibility in how the network is managed.

NMCP commands are divided into 5 groups

- Link commands: there are commands to create and remove links as well as to activate and deactivate links. A link must be activated in order to forward packets.
- Path commands are used to create, modify and delete paths
- Flow commands are used to create, modify, delete as well as to merge and de-merge flows
- Parameter commands are used to get and set parameters
- Events Notification commands are used to set and delete event notifications

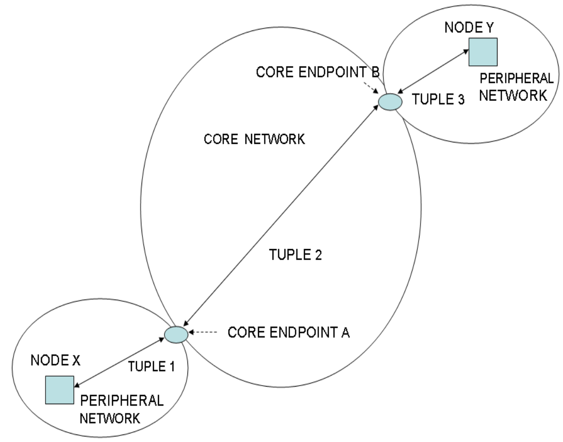

Fig. 3: Connection Setup using NMCP

VI. Connection Setup using NMCP

In this section we show an example of using NMCP. The connection setup is shown in Figure 3. A link in NMCP is represented by a tuple containing the entities that are linked together and a label referred to as a **forward connection label** or fcl. The fcl contains details of the physical characteristics such as speed of the link as well as QoS and security information related to the connection. Fcls are therefore treated as capabilities and cannot be tampered with. So in order for packets to be forwarded on a link, it must have a valid fcl. So let us represent this interaction in the following tuple format: **src, fcl, destination**.

We now show how data will be routed from Node X to Node Y, via the core endpoints Core A and Core B. A connection request is first sent by X to Core-Endpoint A; then the related tuples for the connection are created by the Attach-Link command:

Connection Request(Node X,Node Y)

TUPLE 1=Attach-Link(**Node X,fcl(XA),Core A)**
This allows Node X to send packets to the Core Endpoint A.

TUPLE 2=Attach-Link(**Core A,fcl(AB), Core B**)
This tuple allows Core endpoint A to send data to Core Endpoint B.

TUPLE 3=Attach-Link(**Core B,fcl(BY), Node Y**)
The final tuple allows Core Endpoint B to send packets to Node Y.

So we can now create a path between X and Y by joining the link tuples:

Path(XY)=Create-Path(**TUPLE 1 + TUPLE 2 + TUPLE 3**)

We can now create a data-flow using this path:
Data-flow(XY)=Create-Flow(**Path(XY),XA,**
Merge-Flow(XA,AB),BY = Demerge-Flow(AB,XA))

This means that the data-flow moves data along the physical path denoted by PATH(XY). However, the specification indicates that the initial flow XA can be merged into a compound flow that is going from Core Endpoint A to Core Endpoint B. The flow is then de-merged at Core Endpoint B and forwarded to Node Y.

Once the data-flow has been created; the connection is started by activating all the relevant links if they are not already activated.

Conn(XY)=Act-Link(**TUPLE 1 + TUPLE 2 + TUPLE 3**)

Finally, a message is sent to X to indicate that the Connection has been created.

fcl(X,A)= Connection_granted(**Node X,Node Y**)

So fcl(X,A) is now used by Node X to send data to Node Y using the path that has been set up for that connection.

A. Handling Handover

In this section, we show how handover is handled by NMCP. The setup is shown in Figure 4. Node X decides to handover from Core Endpoint A to Core Endpoint C. It sends a handover request to Core Endpoint A.

Handover_request(**X,Core A: X,Core C**)

This means that Node X wants to be handed over from Core Endpoint A to Core Endpoint C. Hence the Connection(XY) needs to be re-routed. Core Endpoint A sends a request to Core Endpoint C to create two new links:

NewTUPLE 1=Attach-Link(**Node X,fcl(X,C),Core C**)

NewTUPLE 2=Attach-Link(**Core C,fcl(C,B),Core B**)

We now need to modify the path from X to Y:

NewPath(XY)= Modify-Path(**XY,NewTUPLE 1**
+ NewTUPLE 2 + TUPLE 3)

NewData-flow(XY)= Modify-Flow(**NewPath(XY),XC,**
Merge-Flow(XC,CB),BY= Demerge-Flow(CB, XC))

We can now activate the new links:
NewConn(XY)=Act-Link(**NewTUPLE 1,NewTUPLE 2**)

In order to deallocate the old data-flow and the related links we ask that a notification be generated when a new packet is received on the new link:

Set-Note-Event(PACKET_ARRIVAL,NewTUPLE 1)

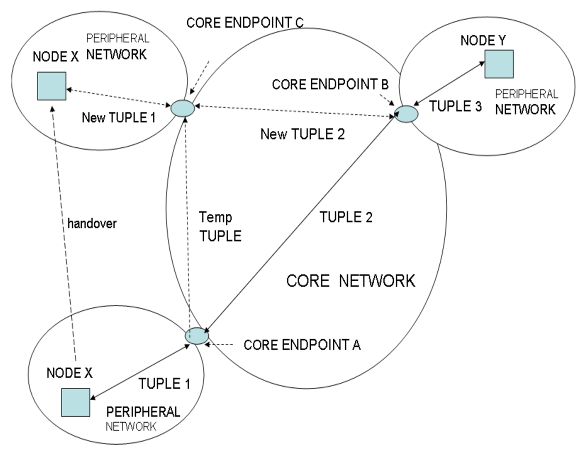

Fig. 4: Core Endpoint Handover Using NMCP

We then set a Conditional Execute instruction to handle this event:

Conditional Execute:
(Event == PACKET_ARRIVAL,NewTUPLE 1)
begin action 1
Delete-Path(XY)
Delete-Flow(XY))
Path(XY) = New Path(XY)
Data-flow(XY) = NewData-flow(XY)

Conn(XY) = NewConn(XY)
Deact-link(TUPLE 1)
Delete-Link(TUPLE 1)
end action

However, before handover starts we must specify that packets sent from Node Y to Node X during handover can no longer be routed to Node X using TUPLE 1 as the node is no longer in that network. So packets reaching Core Endpoint A must be routed to Core Endpoint C. So we must first create a temporary Link between Core Endpoint A and Core Endpoint C.

TempTUPLE= Attach-Link(Core A,fcl(A,C),Core C)

TempPath(YX)= Modify-Path((YX),TUPLE 3 + TUPLE 2 + TempTUPLE + NewTUPLE 1)

TempData-flow(YX)= Modify-Flow(TempPath(YX),YB, Merge-Flow(YB,BA),AC= Demerge-Flow(BA,YB),CX)

So we now activate the new reverse path during handover:

Set-Note-Event (Handover_granted(X,Core A: X,Core C))

Conditional Execute:
(Event == Handover_granted(X,Core A: X,Core C)
begin action
TempConn(YX) = Act-Link(TempTUPLE)
Delete-Path(YX)
Delete-Flow(YX)
Path(YX) = TempPath(YX)
Data-flow(YX)= TempData-flow(YX)
Conn(YX) = TempConn(YX)
end action

We also need to delete this set up when the handover has been completed:

Conditional Execute:
(Event == PACKET_ARRIVAL,NewTUPLE 1)
begin action 2
Delete-Path(YX)
Delete-Flow(YX)
Path(YX) = New Path(XY)
Data-flow(YX) = NewData-flow(XY)
Conn(YX) = NewConn(XY)
Deact-link(TempTUPLE, TUPLE 2)
Delete-Link(TempTUPLE, TUPLE 2)
end action

Finally we send a handover granted reply giving Node X the new fcl to talk to Core Endpoint C:

fcl(X,C)=Handover_granted(**X,Core A: X,Core C**)

Node X is now able to do the handover. After the handover, it uses fcl(X,C) to send packets to Core Endpoint C and this triggers the conditional execute actions which result in the deallocation of resources.

VII. Conclusions

This paper has presented an Implementation Framework for the Future Internet using SDN, NFV and the Y-Comm Framework. The approach explored will allow the evolution of both underlying network infrastructure in terms of the development of H-CRAN and CRRM as well as the development of new network services by third parties.

References

[1] G. Mapp, F. Shaikh, J. Crowcroft, D. Cottingham, and J. Baliosian, "Y-Comm: A Global Architecture for Heterogeneous Networking (Invited Paper)," in *3rd Annual International Wireless Internet Conference (WICON)*, October 2007.

[2] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," April 2012.

[3] HoKey Working Group, " Hokey: Security Support for IEEE 802.21: Media Independent Handover Services," 2010.

[4] N. Niebert, A. Schieder, A. Abramowicz, H. Malmgen, G. Sachs, J. Horn, C. Prehofer, and H. Karl, "Ambient Networks: An Architecture for Communication Networks Beyond 3G," in *IEEE Wireless Communications*, vol. 11, 2004.

[5] K. Masahiro, Y. Mariko, O. Ryoji, K. Shinsaku, and T. Tanaka, "Secure service and network framework for mobile ethernet," *Wireless Personal Communications*, vol. 29, 2004.

[6] M. Aiash, G. Mapp, A. Lasebae, J. Loo, F. Sardis, R. Phan, M. Augusto, R. Vanni, and E. Moreira, "A Survey of Potential Architectures for Communication in Heterogeneous Networks," in *Proceedings of Wireless Telecommunications Symposium (WTS 2012)*, April 2012.

[7] M. Aiash, G. Mapp, A. Lasebae, J. Loo, and R. Phan, "A Formally Verified AKA Protocol For Vertical Handover in Heterogeneous Environments using Casper/FDR," *EURASIP Journal on Wireless Comminications and Networking (Open Springer)*, April 2012.

[8] M. Aiash, G. Mapp, A. Lasebae, R. Phan, and J. Loo, "Integrating Mobility, Quality-of-Service and Security in Future Mobile Networks," *Electrical Engineering and Intelligent Systems: Lecture Notes in Electical Engineering*, vol. 130, pp. 195–206, 2013.

[9] G. Mapp, F. Shaikh, M. Aiash, R. Vanni, M. Augusto, and E. Moreira, "Exploring Efficient Imperative Handover Mechanisms for Heterogeneous Networks," in *Proceedings of the International Symposium of Emerging Ubiquitous and Persuasive Systems, Indianapolis, USA*, August 2009.

[10] F. S. Shaikh, *Intelligent Proactive Handover and QoS Management using TBVH in Heterogeneous Networks*, School of Engineering and Information Sciences, Middlesex University, January 2010, PhD Thesis.

[11] M. Aiash, G. Mapp, A. Lasebae, and A. Al-Nemrat, "Supporting LTE Networks in Heterogeneous Environment using the Y-Comm Framework," in *Proceedings of the 4th International Conference on Networks and Communications*, December 2012.

[12] Q. Duan, "Network-as-a-Service in Sodtware Defined Networks for end-to-end QoS provisioning," in *Proceeding of the 23rd Conference on Wireless and Optical Communication (WOCC)*. Newark, NJ, USA: IEEE, 2014, pp. 1–5.

[13] W. Chen, J. Liu, and H. Huang, "Implementation and performance study of IEEE 802.21 in Integrated IEEE 802.11/802.16 e newtorks," in *Proceedings of the 10th International Conference on Parallel and Distributed Systems*, 2004.

[14] Riley, L and Mapp, G, "yRFC3: The Specification of SP-Lite." [Online]. Available: http://www.mdx.ac.uk/research/science_technology/informatics/projects/ycomm.aspx(Lastaccessed:30/09/2014)

[15] N. Doraswamy and D. Harkins, "IPSec: The New Security Standard for the Internet, Intranets and Virtual Private Networks," in *Prentice Hall Security Series*. Prentice Hall, 2003.

[16] Wan, Tao and Ashwood, Peter, "A performance study of CPRI over Ethernet," *IEEE1904*, vol. 3, 2015.