Implementation of an OBS access node supporting multiple services

Víctor López¹,³, Georgios Zervas², Yixuan Qin², Sergio Lopez-Buedo¹, Dimitra Simeonidou², Javier Aracil¹ and Juan Fernandez-Palacios³

¹Universidad Autónoma de Madrid
²University of Essex
³Telefónica I+D

July, 2012
Index

- Motivation
- MAINS Reference Architecture
- Prototype Implementation Architecture
  - Use case for the OBS access node
  - Implementation
- Prototype Behavioural Validation
- Conclusions
Motivation

- Operators are interested on Network Centric Services (NCS)
- What is a NCS?
  - Combined user of both network and IT resources.
  - Examples:
    - PC virtualization, VoD, 3D Internet gaming, SaaS and SAN
Network Centric Service: Virtual PC
Metro Architecture

- **Centralized cloud approach:** Few servers located in core nodes
  - High bandwidth consumption
  - Low latency: minimum delay
  - Just required bandwidth
  - Low cost servers

- **Distributed cloud approach:** Multiple multipurpose servers located in metro and core nodes
  - High computation
  - Just required bandwidth
  - Low cost servers
Motivation

Why NCS?

- Operator’s perspective:
  - Network scalability
  - New business opportunity
  - CAPEX and OPEX optimization

- End user’s perspective:
  - Availability
  - Mobility
  - IT maintenance outsourcing
  - QoE: low latency and high bandwidth

Increment of network traffic will impact on metro network

- New metro architecture is required to support such services.
Motivation

Metro Architectures enabling Subwavelengths (MAINS) project proposes a new metro architecture based on two pillars:

- Subwavelength optical switching technologies in the Data Plane.
- Enhanced GMPLS architecture in the Control Plane.

The objectives of such architecture are:

- Reduce cost and energy consumption.
- Improve reliability and latency.

Index

- Motivation
- MAINS Reference Architecture
- Prototype Implementation Architecture
  - Use case for the OBS access node
  - Implementation
- Prototype Behavioural Validation
- Conclusions
MAINS Reference Architecture

- Application Client/Server
- Middleware
- MAINS Network Service Interface (MNSI)
  - Middleware
  - XML Interface
  - MNSI Gateway
- Data transport
- Access (LAN + last-mile)
- OPST Ring
- TSON Mesh
- OPST Ring
- CPE
- MNSI Gateway

Subwavelength-enabled GMPLS CP

Metro/regional segment
VM transference

1. Middleware information exchange

4. Control plane configuration

Transport Network (OPST/OBST)

User request a Virtual PC service
Index

- Motivation
- MAINS Reference Architecture
- Prototype Implementation Architecture
  - Use case for the OBS access node
  - Implementation
- Prototype Behavioural Validation
- Conclusions
Prototype Implementation Architecture

- Metro network supports multiple services at the same time.

- A prototype is developed with the following requirements:
  - Data plane burst transmission
  - VLAN support for multiple services
Use case for the OBS access node

1. **Service configuration** (VLAN, Burst_Size, Burst_Timer)

2. **Data transmission**

3. Any threshold is exceeded (Timer VLAN2)
Implementation: Board details

- Xilinx® Virtex™-5 PCI Express Development Kit (Avnet)
  - Xilinx Virtex-5 XC5VSX95T-FF1136 FPGA
Implementation: Modules
Implementation: TX modules

- **Eth2Scheduler:** get VLAN_id and packet size while storing the incoming packet
- **Scheduler:** burst generation and management
- **Scheduler2RocketIO:** burst adaptation and transmission
Implementation: SERDES interfaces

- SERDES interfaces are Serializer/Deserializer interfaces.
  - Input – output parallel interfaces has 16 bits.
  - Physical transmission is done with a differential signal.

Implementation: SERDES interfaces

- There is a 8B/10B module to provide redundancy.
- Channel management uses K-Characters
  - TXCHARISK signal is asserted if TXDATA is a K-Character.
How to create a burst?

- There are two solutions to create a burst:
  - Send burst structure information in the burst control packet.
  - Insert K-characters in the burst while transmitting.

- There are two special characters for our burst:
  - End of burst character
  - Start of packet character
Implementation: RX Modules

- Functions:
  - **Burst2Eth:**
    - Extract packets from the burst
    - Transmission over GbE interface
Index

- Motivation
- MAINS Reference Architecture
- Prototype Implementation Architecture
  - Use case for the OBS access node
  - Implementation
- Prototype Behavioural Validation
- Conclusions
Burst size threshold validation

- For this experiment, the application server 1 sends traffic to the application server 2 with a fixed packet size of 128 bytes.

- The burst size threshold is set to 6400 bytes (50 packets). Based on the PCAP file captured at Server 2, the burst is correctly transmitted.
Support to multiple services

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>15.537620</td>
<td>12.0.0.40</td>
<td>24.171.69.203</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>21</td>
<td>15.537706</td>
<td>12.0.0.40</td>
<td>125.188.97.203</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>22</td>
<td>15.537829</td>
<td>12.0.0.40</td>
<td>38.253.115.224</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>23</td>
<td>15.537842</td>
<td>12.0.0.40</td>
<td>83.50.205.9</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>24</td>
<td>15.537951</td>
<td>12.0.0.40</td>
<td>208.68.226.226</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>26</td>
<td>16.037834</td>
<td>12.0.0.40</td>
<td>24.171.69.203</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>27</td>
<td>16.037837</td>
<td>12.0.0.40</td>
<td>125.188.97.203</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>28</td>
<td>16.037839</td>
<td>12.0.0.40</td>
<td>38.253.115.224</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>29</td>
<td>16.037841</td>
<td>12.0.0.40</td>
<td>83.50.205.9</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>30</td>
<td>16.037842</td>
<td>12.0.0.40</td>
<td>208.68.226.226</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>41</td>
<td>30.378246</td>
<td>12.0.0.40</td>
<td>141.37.27.251</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>42</td>
<td>30.383435</td>
<td>12.0.0.40</td>
<td>4.226.99.80</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>43</td>
<td>30.628439</td>
<td>12.0.0.40</td>
<td>141.37.27.251</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>44</td>
<td>30.628461</td>
<td>12.0.0.40</td>
<td>4.226.99.80</td>
<td>IP</td>
<td>IP</td>
</tr>
</tbody>
</table>

- Frame 26 (60 bytes on wire, 60 bytes captured)
- Ethernet II, Src: 00:00:00_e0:01:04:03 (00:00:00:01:04:03), Dst: e4
- 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 10
- Internet Protocol, Src: 12.0.0.40 (12.0.0.40), Dst: 24.171.69.20
- Data (22 bytes)
Index

- Motivation
- MAINS Reference Architecture
- Prototype Implementation Architecture
  - Use case for the OBS access node
  - Implementation
- Prototype Behavioural Validation
- Conclusions
Conclusions

- MAINS project is exploring sub-wavelength technologies as an efficient solution for metro networks.
- The architecture for a multi-service OBS access node based on a Virtex 5 FPGA is detailed explained.
- The modules of the access node are described and their functionality to support multiple services on an access node.
- A behavioural validation of the prototype under traffic from two different services is presented.
Thank you!!
Questions?