# Demonstration of IP Based Control and Management for a Reconfigurable Photonic Access Network

**Ronald Teune, Rajeev Roy, *Student Member, IEEE*, Wim van Etten, *Senior Member, IEEE***
*Telecommunication Engineering, University of Twente, PO Box 217, 7500 AE Enschede, The Netherlands*
*Tel: (0031) 53 489 2819, Fax: (0031) 53 489 5640, e-mail:r.roy@ewi.utwente.nl*

**ABSTRACT**
An implementation of control and management for a reconfigurable photonic access network is presented. An out of band control channel is used on which an IP communication is established to communicate with remote elements. A Headend based master controller communicates with a far end embedded processor, this in turn is used to control and manage remote devices in the network.

**Keywords**: Control and Management , FPGA, embedded processor , EPON, DWDM PON.

## 1. INTRODUCTION

The Broadband Photonics project under the Freeband consortium of projects looks into design of agile reconfigurable access networks. The proposed network is viewed as a stack of logical PONs with each PON operating on a unique wavelength pair. This stacking of PONs increases the aggregate bandwidth offered and adds another dimension to increase the flexibility in the network as such, but this also increases the parameters which need to be monitored and controlled for optimal network performance. A practical implementation for the control and management of such a network has been proposed [1]. This paper presents a concept demonstrator in which remote elements are controlled with IP based communication from the Headend (HE).

Figure 1 illustrates the schematic of the network. The Optical Line Termination Units (OLTs) are placed at the HE. Each OLT operates on a unique wavelength pair. The number of OLTs that can be used depends on the number of wavelength pairs that can be supported in the network. The network uses optical micro-ring resonator based [2] reconfigurable optical add-drop multiplexers (ROADMs) to add/drop wavelength pairs towards Optical Network Units (ONUs). The Remote Nodes (RNs) which house the ROADMs act as the split points as in legacy PONs. The use of ROADMs allows for add/drop of any wavelength pair towards any ONU. The ONUs associate with the OLT which operate on that particular wavelength add/drop pair. Every OLT and the associated ONUs form a logical PON. The selective add/drop of wavelengths towards the ONUs can be used to dynamically change the number of ONUs in every such logical PON and this can be used for dynamic inter-PON bandwidth allocation for optimal bandwidth availability to the end user [3].

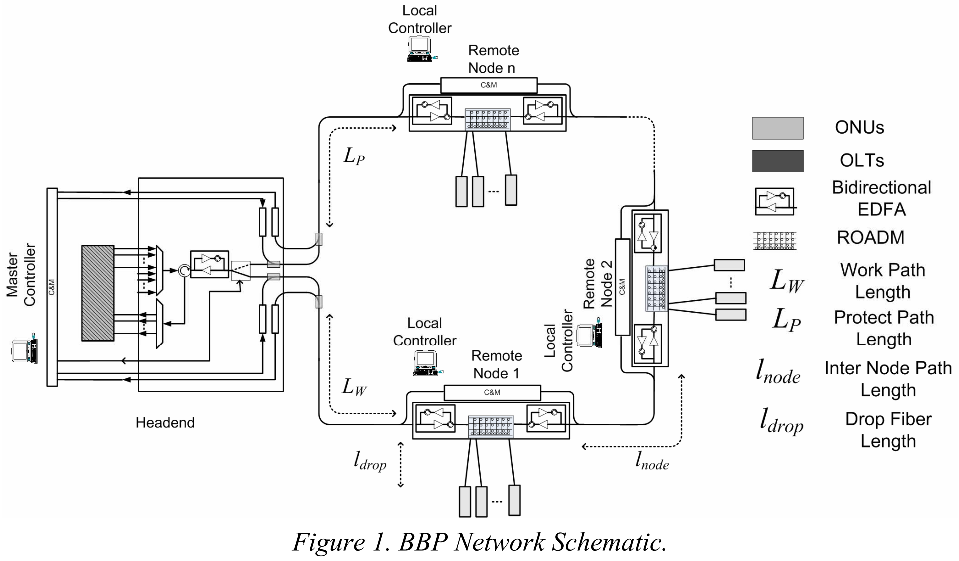


*Figure 1. BBP Network Schematic.*

## 2. CONTROL AND MANAGEMENT

An IP (Internet Protocol) based control and management channel gives the network operator flexibility to use commercially available equipment and a standardized protocol stack to do remote control and management. A 100BASE-X MAC communication is used between the Master Controller (at the HE) and the Local Controllers (at the RNs) . The physical channel for this communication will be on an out of band 1310/1490 nm

The work is funded by the Dutch Ministry of Economic affairs through the BSIK Freeband Project.

optical channel. Figure 2. illustrates the protocol stack running in the Master controller (at the HE) and in the Local Controllers (at the RN).

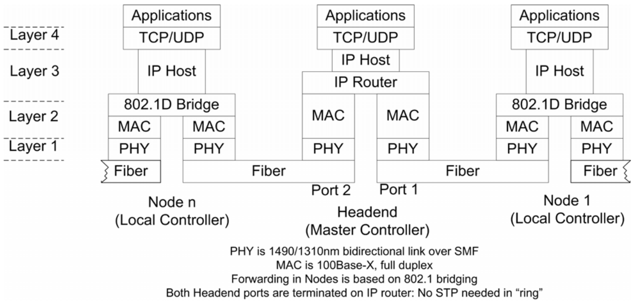

Layer 4
Layer 3
Layer 2
Layer 1
Applications
TCP/UDP
IP Host
802.1D Bridge
MAC
PHY
Fiber
IP Router
Port 2
Port 1
Node n
(Local Controller)
Headend
(Master Controller)
Node 1
(Local Controller)
PHY is 1490/1310nm bidirectional link over SMF
MAC is 100Base-X, full duplex
Forwarding in Nodes is based on 802.1 bridging
Both Headend ports are terminated on IP router: No STP needed in "ring"

*Figure 2. Protocol Stack at Headend and at Remote Nodes.*

## 3. DEMONSTRATOR

Figure 3. illustrates the schematic set up for the demonstrator. The Master Controller is implemented in a Pentium 3 desktop. The desktop communicates with an embedded processor implemented in an Field Programmable Gate Array (FPGA). The embedded processor or "soft" processor acts as the Local Controller. The Local Controller in turn is used to control a board having an array of Digital to Analog Convertors (DACs). The DACs in turn would be used to control the thermal tuning of the micro-ring resonator based ROADM. The selective tuning allows for add/drop of designated wavelengths at different ports of the ROADM. The communication channel between the Master and Local Controller is based on an arbitrary IP based connection. The Master Controller is used to issue HTTP based commands to the Local Controller which in turn translates these commands to suitable signals for the DAC interface. Figure 4. is a snapshot of the demonstrator set up.

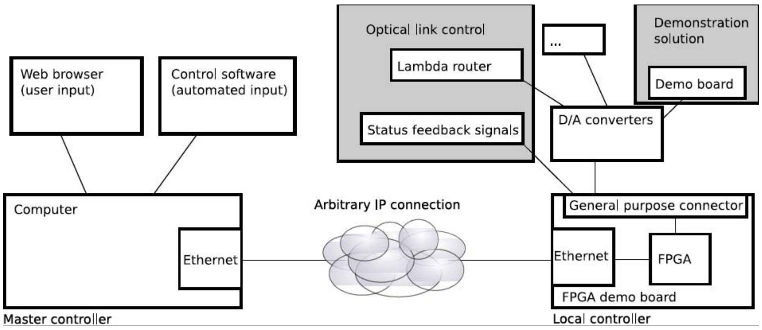
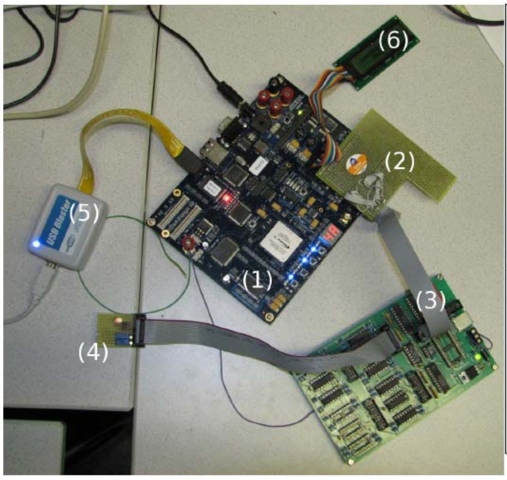


*Figure 3. Schematic of demonstrator set up.*



1) Altera NIOS II demonstration board

2) Daughter card to interface GPIO pins to (3)

3)DACDRIVE32 D/A convertor board

4) RGB LED indicators

5) USB JTAG

6) LCD board (not used)

*Figure 4.Demonstrator Setup.*

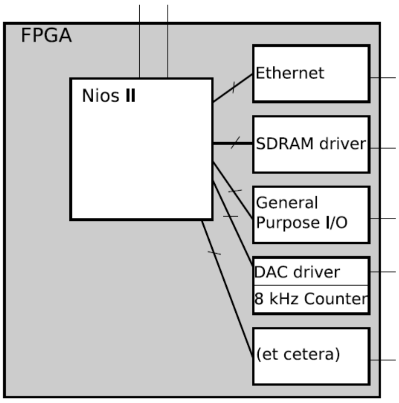
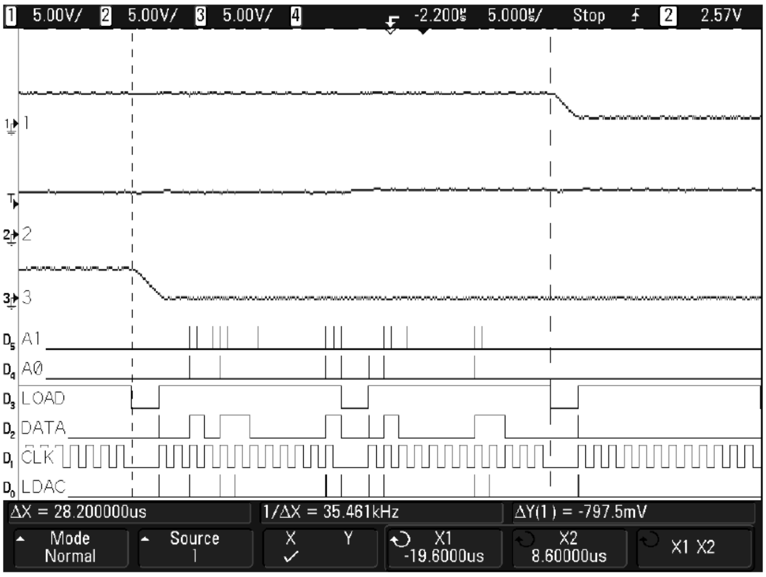


*Figure 5. FPGA Internals.*

### 3.1 FPGA and driver for DAC board

An FPGA based design has been chosen to implement the Local Controller. An Altera Nios II board with a Stratix II (EP2S60F672C5) FPGA is used. The device has much more logic elements (LEs) than what the design required, however since this is a prototype design to showcase concept implementation the available device has been used. The System on a Programmable Chip (SOPC) builder which is an integrated part of the Quartus II design environment has been used to program the FPGA. The FPGA design is built around an embedded Nios II processor. The processor is interfaced through an Altera Avalon bus to connect to an external SDRAM, external Ethernet MAC/PHY (SMSC LAN91C111) and a General Purpose Input/Output (GPIO) bus. Figure 5. illustrates a simplified schematic of the FPGA internals.

The Nios II is internally connected to a custom-made DAC driver block, combined with an 8 kHz counter that can be used to synchronously output analog signals. This block has a parallel interface to the Nios II CPU and converts this parallel input to a serial output signal for the TLC5628 D/A converters. The output of this driver block for 8 kHz, four channel quasi-simultaneous output, as measured with an Agilent MSO6014A mixed-signal oscilloscope is shown in Figure 6. The figure illustrates both the digital input signals and the analog output signals. After a new output value is received from the CPU by the TLC5628 driver, LOAD goes high and the clock signal starts to toggle. The TLC5628 is negative edge triggered and the data is toggled on the positive edge of the clock to meet setup and hold requirements. The first three bits being clocked out (that is, the three leftmost on the picture) on the DATA signal line are the address bits, to select one of the eight available D/A converters. The next bit is a RNG (range) bit; 0 selects the 0 to 5 V output range and 1 selects the 0 to 10 V output range. The next eight bits represent the output voltage in a linear way; 0x00 means 0 V and 0xFF means full scale (either 5 V or 10 V, depending on the value of the RNG bit). After these bits are clocked into the D/A converter, the LOAD signal is being pulled low to indicate completion of data transmission.



1 5.00V/ 2 5.00V/ 3 5.00V/ 4 -2.200µs 5.000µs/ Stop 2 2.57V

ΔX = 28.200000us 1/ΔX = 35.461kHz ΔY(1) = -797.5mV

Mode Normal

Source 1

X1 -19.6000us

X2 8.60000us

*Figure 6. Screenshot of MSO showing digital inputs and the corresponding analog outputs.*

### 3.2 Web Based Control

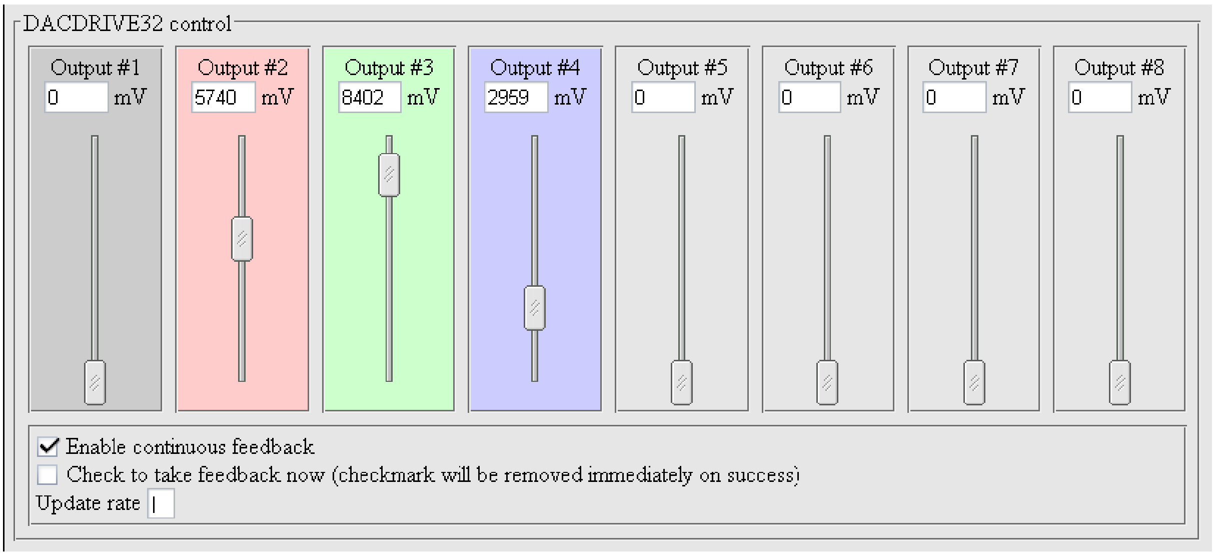
A web server is hosted at the Local Controller location in the FPGA. The Nios II processor has a real time MicroC/OS-II operating system with an InterNiche IP stack used for IP connectivity through the Ethernet port on the demo board.

#### 3.2.1 Server

The HTTP GET handling routines is given the web URL by a function call to an external parsing routine. If the URL is in the format http://ip-of-server/DAC?address=A&millivoltage=B (with 0 to7 as useful values for A, and 0 to 10000 as useful values for B), the millivoltage is converted into a RNG/data output pair and sent to the DACDRIVE32, together with the address. A feedback interface is also provided; at every request starting with "/DAC", the current memorized output values are returned in the headers of the returned web page. These headers have names "X-DAC-0" to "X-DAC-7", followed by a colon and the stored value in millivolts of the corresponding output. These header values can easily be parsed using the AJAX (Assymetric Javascript And XML) interface built into modern browsers.

### 3.2.2 Client

A graphical user interface to control the web server functionality has been created in the host PC (Master Controller) with the help of the WebFX Slider API5. A screenshot of this interface is shown in Figure. 7. The checkmark before "Enable continuous feedback" enables or disables the use of the returned X-DAC headers to set the values of the sliders. The "Check to take feedback now" checkmark is used to take feedback just once. The current slider value is not sent to the demonstrator board instantaneously, because its IP stack choked on the many, many updates that are sent when sliding the slider. The text box next to "Update rate" shows a rotating dash, which is used to show when the updates are sent (currently 2 times per second).



DACDRIVE32 control

Output #1 0 mV

Output #2 5740 mV

Output #3 8402 mV

Output #4 2959 mV

Output #5 0 mV

Output #6 0 mV

Output #7 0 mV

Output #8 0 mV

☑ Enable continuous feedback

□ Check to take feedback now (checkmark will be removed immediately on success)

Update rate |

*Figure 7. Screenshot website seen on host machine.*

## 4. CONCLUSIONS

A demonstrator concept for remote control of network elements in a reconfigurable access network has been demonstrated. The platform is developed on commercially available and standardised interfaces. A web based interface is provided to the network operator at the HE through which commands are issued to a remote Local Controller which in turn monitors and controls devices at the remote location. In the demonstrator a DAC array is controlled remotely by just means of an arbitrary IP communication from the Master Controller through an embedded processor at the remote location. The concept can be used to control multiple devices with different interfaces; standard and proprietary. This demonstrator shows flexibility in control and management of devices with differing interfaces (eg, RS232, USB etc).

## ACKNOWLEDGEMENTS

The authors acknowledge Dr Edwin Klein for designing and providing the DACDRIVE32 D/A board.
Thanks are due to Mr Eduard Bos for operational support with the demonstrator.

## REFERENCES

[1] R. Roy, *et al.*: Control and Management Scheme in a DWDM EPON, *ICTON 2006, Nottingham, UK, 2006,* pp. 151-154.

[2] D.H. Guezebroek, *et al*: 40 Gb/s Reconfigurable Optical Add-Drop Multiplexer based on Microring Resonators, in *Proc ECOC 2005, Glasgow, UK, 2005,* pp. 983-986.

[3] R. Roy, *et al.*, “Bandwidth re-distribution techniques for extended EPON based multi-wavelength Networks”, *ICTON 2007*, *Rome, Italy, 2007*, vol. 4, pp. 80-83.

[4] Introduction to Quartus II, Version 4.2, *2004.*

[5] Altera Nios development board Stratix II Edition Reference Manual, www.altera.com/literature/manual/mnl_nios2_board_stratixII_2s60_rohs.pdf .