# CHRON

## Cognitive Heterogeneous Reconfigurable Optical Network

Grant Agreement Nº 258644

# D6.4 Validation and performance evaluation of the CHRON concept through the physical testbed

## WP6 Integration and Experimental Validation

**Version:** 1.0
**Due Date:** 30/09/2013
**Delivery Date:** 07/11/2013
**Nature:** Report
**Dissemination Level:** Public

**Lead partner:** AIT
**Authors:** AIT, CREATE-NET, DTU, UVa.

**Internal reviewers:** UVa

**www.ict-chron.eu**

SEVENTH FRAMEWORK PROGRAMME

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement n° 258644

## The CHRON project consortium groups the following organizations:

| Short Name | Partner Name | Country |
|---|---|---|
| DTU | Danmarks Tekniske Universitet | Denmark |
| HWDU | Huawei Technologies Duesseldorf GmbH | Germany |
| TP SA | Telekomunikacja Polska S.A. | Poland |
| AIT | Research and Education Laboratory in Information Technologies | Greece |
| UVa | Universidad de Valladolid | Spain |
| CREATE-NET | CREATE-NET (Center for Research and Telecommunication Experimentation for Networked Communities) | Italy |

## CHRON Contacts:

| Partner name | Name | Email | Role |
|---|---|---|---|
| UVa | Rubén Lorenzo Toledo | ruben.lorenzo@tel.uva.es | Project Coordinator |
| DTU | Idelfonso Tafur Monroy | idtm@fotonik.dtu.dk | Technical Coordinator |

| | |
|---|---|
| **Deliverable Title** | Validation and performance evaluation of the CHRON concept through the physical testbed |
| **Deliverable Number** | D6.4 |
| **Keywords:** | Cognitive testbed, emulated testbed, cognitive decision. |

**Executive Summary:**

This deliverable describes the validation and the performance evaluation of the CHRON concept through the physical testbed. Specifically, it describes the testing scenarios that were performed in order to validate the CHRON using the physical testbed. It also presents the results of these scenarios and the advantages of the CHRON. Overall, it is shown how the cognitive system integrated into the optical network can be used to reduce significantly the time for the restoration path of the optical links. The performance evaluation shows the results for both simple cases such as OOK and high modulation formats such as QPSK and 16QAM.

# Table of Contents

# 1 Introduction

The CHRON project has proposed a novel architecture for cognitive heterogeneous reconfigurable optical networks. The aim of the CHRON project is to develop and showcase a network architecture and a control plane which efficiently use resources in a heterogeneous scenario while fulfilling QoS requirements of each type of services and applications supported by the network. This cognitive network observes acts, learns and optimizes its performance, taking into account its high degree of heterogeneity with respect to QoS, transmission and switching techniques. For that aim, CHRON relies on cognition, so that control decisions must be made with an appropriate knowledge of current status, and supported by a learning process to improve performance with acquired experience. The core element of the CHRON cognitive architecture is the cognitive decision system (CDS). The CDS receives traffic demands, and determines how to handle them by taking into account both the current status of the network and past history, and instructs the control plane to configure network elements accordingly. The advantages of the CDS have been shown in the past in several network topologies using both heterogeneous and homogeneous networks with many nodes. However, the CDS has been only validated in simulation and emulation modes.

In the report associated to deliverable D6.3, the design of the physical testbed that can be used for the validation and performance evaluation of the CHRON has been described extensively. In this deliverable (D6.4), we describe the validation and the performance evaluation of CHRON using the physical testbed that consists of real optical nodes. Specifically, **D6.4 describes the methodology and the scenarios that have been implemented in order to validate and demonstrate the advantages of a cognitive system in a reconfigurable heterogeneous optical network**. The main challenge in the demonstration of a cognitive system for the optical networks is that the scenarios must demonstrate how the cognitive system has learned and has populated its knowledge base in order to optimize its performance. Therefore, the scenarios that have been implemented are divided into two parts. In the first part we setup the network and we create artificially several alerts (e.g. high BER, fiber cut, etc.). During this phase the CDS explores several options and populates its knowledge base. In the second phase, we create again artificially an alarm and we validate that the CDS will use its knowledge base to optimize the operation of the network.

The main achievement of this deliverable is that we are able to demonstrate the operation of the CHRON in a real network and to demonstrate that CHRON can reduce the restoration time of a broken link by over 48%.

# 2 Validation and Performance evaluation of the CHRON concept through the physical testbed

To validate and evaluate the operation of the CHRON networks, we have developed a physical testbed integrating the control and management plane (CMP), the data plane and the Cognitive Decision System (CDS) as it is described in the report associated to D6.3. Specifically, the physical testbed is composed of the following systems:

- **Control plane:**

The control plane is implemented in 4 separate workstations emulating the GMPLS-based control functions and the cognitive decision system. The workstations are interconnected via an Ethernet switch and each one has a direct serial interface with a separate physical optical switch. Message exchange is realized as in the case of the emulated test-bed taking into consideration feedback that is received by the data plane (e.g. monitors).

- **Data plane:**

The data plane is implemented with physical interconnection of four switching nodes incorporating 8x8 MEMS switches. Two of the nodes accommodate input traffic from FPGA-controlled transmitters and the other two nodes are used to receive and measure the characteristics of the optical signals (BER, power, etc.). Depending on the switch configuration and the testing scenarios multiple paths can be realized with this set-up from source to the destination. Physical layer power monitors are also implemented on selected links in support of the testing scenarios.

- **Cognitive Decision System (CDS)**

The cognitive decision system is implemented in one of the four workstations.

Figure 1 depicts the overall architecture of the physical testbed, integrating the CDS, the control and management plane and the data plane. As it is shown in this Figure, four nodes are used for the validation and the performance evaluation. Nodes A and C are connected with the traffic generators from the FPGA, while nodes B and D are connected with the receivers of the FPGA and can be used to measure the quality of the channels (BER). The report associated to D6.3 describes in more detail the implementation of the physical testbed.

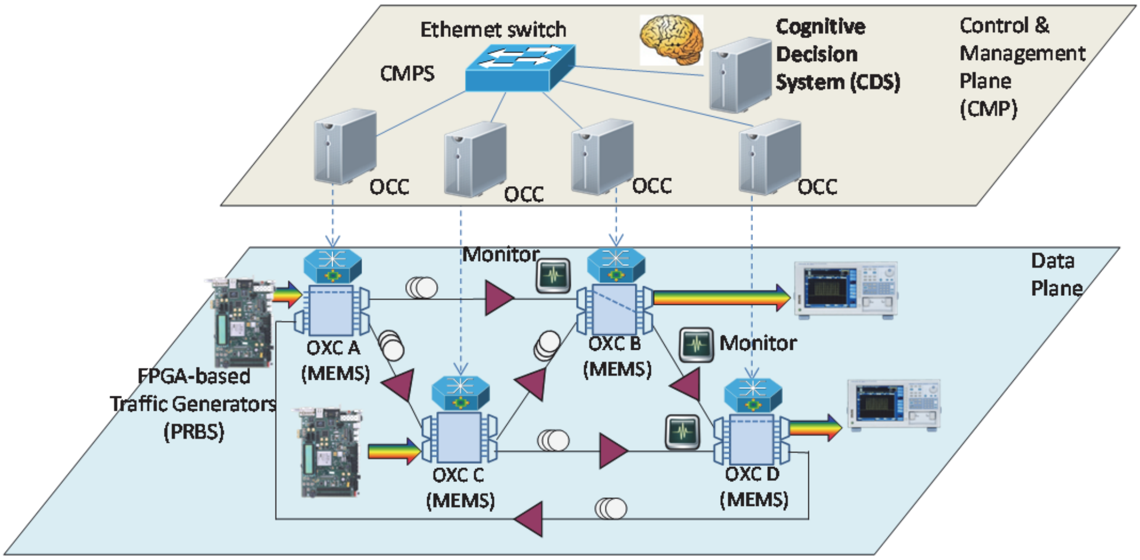


**Figure 1. The physical layer testbed used for the validation and performance evaluation**

## 2.1 Initial Set-up of a lightpath

The first measurement for the validation of the CHRON concepts was the initial setup of single channel between two nodes (Node A and Node B). The channel was based on a simple OOK lightpath. After the request of the establishment of a single lightpath between A and B, the cognitive engine (CDS) produces the *path_request* and forwards this request to the control plane (CMP). The control plane communicates with the nodes and the optical cross-connects and establishes the connection between the two nodes. Specifically, the CMS establishes a connection between the first port of node A and the first port of node B. The topology of this scenario is depicted in Figure 2. The total time to setup a lightpath between two nodes is 968 ms (from the time it receives the path request) as it is depicted in Figure 3. The detailed description of the establishment of this lightpath describing the CDS, the RSVPD and the Hardware interface configuration (HINT) is depicted in Figure 4.

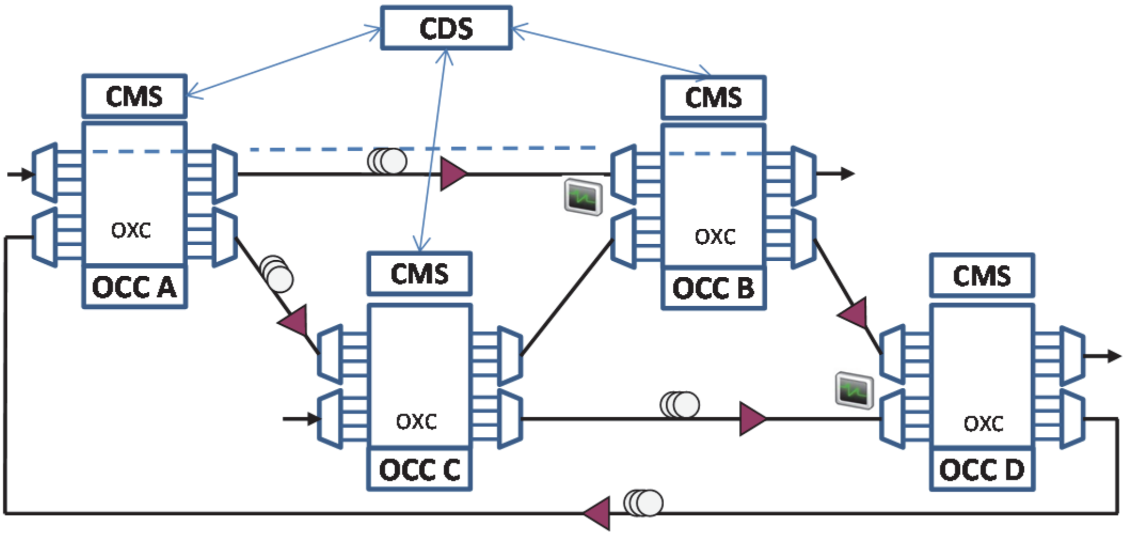


**Figure 2. Setup of a lightpath between nodes A and B**

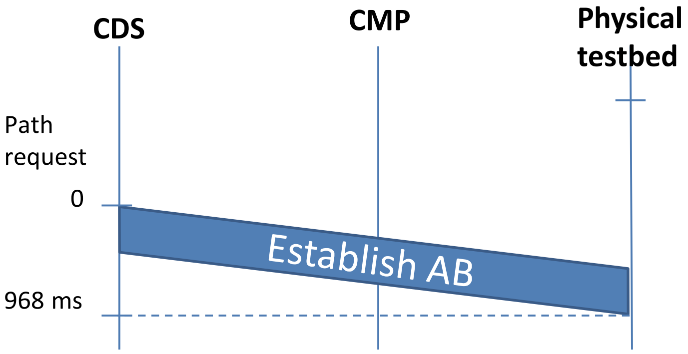


**Figure 3. Total time for setup of a single lightpath between two nodes**

| Node | Module | Time | Duration | Message | Ack | Description |
|---|---|---|---|---|---|---|
| Node_A | CDS | 0 | 0 | PATH_REQUEST | None | CDS produces the PATH_REQUEST |
| Node_A | CDS | 14 | 14 | PATH_REQUEST | None | CDS forwards the PATH_REQUEST to the database |
| node_A | DRAGON | 138 | 124 | Commit_PATH | None | DRAGON GMPLS interface receives the request and generates the PATH msg |
| node_A | DRAGON | 138 | 0 | PATH_msg_sent | None | DRAGON GMPLS interface sends the PATH msg to RSVP-TE module |
| node_A | RSVPD | 138 | 0 | PATH_msg_received | None | RSVP-TE receives the PATH msg |
| node_A | RSVPD | 139 | 1 | CACRM_ResCheckReq | None | RSVP-TE requests CACRM to check resource availability |
| node_A | RSVPD | 139 | 0 | CACRM_response | OK | CACRM sends a positive response |
| node_A | RSVPD | 139 | 0 | PATH_msg_sent | None | RSVP-TE forwards the PATH msg to the next hop |
| node_B | RSVPD | 140 | 1 | PATH_msg_received | None | RSVP-TE receives the PATH msg |
| node_B | RSVPD | 147 | 7 | CACRM_ResCheckReq | None | RSVP-TE requests CACRM to check resource availability |
| node_B | RSVPD | 147 | 0 | CACRM_response | OK | CACRM sends a positive response |
| node_B | RSVPD | 147 | 0 | PATH_msg_sent | None | RSVP-TE forwards the PATH msg to the next hop: being destination node, it forwards it to DRAGON GMPLS Interface |
| node_B | DRAGON | 147 | 0 | PATH_msg_received | None | DRAGON GMPLS Interface receives PATH msg and stores data |
| node_B | DRAGON | 147 | 0 | RESV_msg_sent | None | DRAGON GMPLS Interface generates RESV msg and forward it to RSVP-TE |
| node_B | RSVPD | 148 | 1 | RESV_msg_received | None | RSVP-TE receives the RESV msg |
| node_B | RSVPD | 156 | 8 | CACRM_ResResvReq | None | RSVP-TE requests CACRM to allocate resources |
| node_B | HINT | 159 | 3 | MEMS_CONFIG_REQ | C10000000 | CACRM requests FPGA to configure MEMS |
| node_B | HINT | 194 | 35 | None | A10000000 | FPGA sends a positive response |
| node_B | RSVPD | 204 | 10 | CACRM_response | OK | CACRM sends a positive response |
| node_B | RSVPD | 204 | 0 | RESV_msg_sent | None | RSVP-TE stores data locally and forwards the RESV msg to its previous hop in the path |
| node_A | RSVPD | 206 | 2 | RESV_msg_received | None | RSVP-TE receives the RESV msg |
| node_A | RSVPD | 220 | 14 | CACRM_ResResvReq | None | RSVP-TE requests CACRM to allocate resources: being the source node, MEMS configuration is stored locally, but not performed. It will be triggered by PATH_ACTIVATION request |
| node_A | RSVPD | 275 | 55 | CACRM_response | OK | CACRM sends a positive response |
| node_A | RSVPD | 275 | 0 | RESV_msg_sent | None | RSVP-TE stores data locally and forwards the RESV msg to its previous hop in the path: being source node, it forwards it to DRAGON GMPLS Interface |
| node_A | DRAGON | 275 | 0 | RESV_msg_received | None | DRAGON GMPLS Interface receives RESV and stores data, then sends response to CDS |
| Node_A | CDS | 751 | 476 | PATH_RESPONSE | 0 | CDS receives positive response (0) |
| Node_A | CDS | 772 | 21 | PATH_RESPONSE | 0 | CDS forwards the response to database |
| Node_A | CDS | 836 | 64 | PATH_ACTIVATION | None | CDS sends a PATH_ACTIVATION request message |
| Node_A | CDS | 854 | 18 | PATH_ACTIVATION | None | CDS forwards PATH_ACTIVATION request message to db |
| node_A | RSVPD | 869 | 15 | CACRM_ResResvReq | None | RSVP-TE requests CACRM to configure source node MEMS, activating the path |
| node_A | HINT | 880 | 11 | MEMS_CONFIG_REQ | C10000000 | CACRM requests FPGA to configure MEMS according to the stored configuration |
| node_A | HINT | 914 | 34 | None | A10000000 | FPGA sends a positive response |
| node_A | RSVPD | 925 | 11 | CACRM_response | OK | CACRM sends a positive response: RSVP-TE sends a positive PATH_ACTIVATION response to CDS |
| Node_A | CDS | 939 | 14 | PATH_RESPONSE | 30 | CDS receives positive PATH_ACTIVATION response (30) |
| Node_A | CDS | 968 | 29 | PATH_RESPONSE | 30 | CDS forwards the PATH_ACTIVATION response to database |

**Figure 4.Detailed description of the setup of a new lightpath between two nodes. Times are given in ms**

## 2.2 Tear-down of a lightpath

Similarly, one of the first scenarios for the validation and the correct integration of the CHRON testbed was the measurement for the tear-down of a lightpath. When the CDS receives from the user the command to tear-down a lightpath, it forwards the *tear_down_request* to the control plane (CMS). The control plane communicate with the optical cross connect modules and configures the MEMS using the hardware interface (HINT). Finally, the CDS receives the positive response from the RSVP-TE that the channel was been torn down. The total time for the tear-down of the lightpath between two nodes is 281 ms. The detailed description of the all the steps are shown below.

| Node | Module | Time | Duration | Message | Ack | Description |
|---|---|---|---|---|---|---|
| Node_A | CDS | 0 | 0 | PTEAR_REQUEST | None | CDS produces the PTEAR_REQUEST |
| Node_A | CDS | 32 | 32 | PTEAR_REQUEST | None | CDS forwards the PTEAR_REQUEST to the database |
| node_A | DRAGON | 75 | 43 | TearDown_PATH | None | DRAGON GMPLS interface receives the request and generates the PTEAR msg. |
| node_B | RSVPD | 75 | 0 | PTEAR_msg_received | None | RSVP-TE receives the PTEAR msg. |
| node_A | RSVPD | 75 | 0 | CACRM_ResReleaseReq | None | RSVP-TE requests CACRM to free the associated resources |
| node_A | HINT | 99 | 24 | MEMS_CONFIG_REQ | C00000000 | CACRM request FPGA to configure the MEMS |
| node_A | HINT | 135 | 36 | None | A00000000 | FPGA sends a positive response |
| node_A | RSVPD | 156 | 21 | CACRM_response | OK | CACRM sends a positive response |
| node_A | RSVPD | 156 | 0 | PTEAR_msg_sent | None | RSVP-TE updates its internal data and forwards the PTEAR msg to next hop |
| node_B | RSVPD | 156 | 0 | PTEAR_msg_received | None | RSVP-TE receives the PTEAR msg. |
| node_B | RSVPD | 156 | 0 | CACRM_ResReleaseReq | None | RSVP-TE requests CACRM to free the associated resources |
| node_B | HINT | 162 | 6 | MEMS_CONFIG_REQ | C00000000 | CACRM request FPGA to configure the MEMS |
| node_B | HINT | 202 | 40 | None | A00000000 | FPGA sends a positive response |
| node_B | RSVPD | 238 | 36 | CACRM_response | OK | CACRM sends a positive response |
| node_B | RSVPD | 244 | 6 | PTEAR_msg_sent | None | RSVP-TE updates its internal data and forwards the PTEAR msg to next hop: being the destination node, it forwards the request to the local DRAGON GMPLS Interface |
| node_B | DRAGON | 254 | 10 | PTEAR_msg_received | None | DRAGON GMPLS Interface receives the PTEAR msg, update its local configuration and sends a response to CDS |
| Node_A | CDS | 269 | 15 | PTEAR_RESPONSE | 0 | CDS receives positive response from RSVP-TE |
| Node_A | CDS | 281 | 12 | PTEAR_RESPONSE | 0 | CDS forwards positive response from RSVP-TE |

**Figure 5. Detailed description of the tear-down of a lightpath. Times are given in ms**

## 2.3 Scenario 1: Setup of 4 channels

After the validation of the CHRON testbed using the setup and the tear-down of the lightpaths, we implemented a more sophisticate scenario in which 4 channels are established and based on the user requirements, some of the channels are torn-down and later re-established, possibly using different resources. Specifically, in this scenario the following steps are performed :

1. LSP 1 is established from node A to node D.
   CDS choose a route A -> B -> D, using wavelength 2
2. LSP 2 is established from node C to node D.
   CDS choose a route C -> D, using wavelength 4
3. LSP 3 is established from node C to node B.
   CDS choose a route C -> B, using wavelength 1
4. LSP 4 is established from node A to node B.
   CDS choose a route A -> B, using wavelength 3
5. Two path teardown requests are sent by the USER, requesting both paths 2 and 3 (i.e., the two paths starting from node C). 5 seconds passed from the first PTEAR_REQUEST to the second one.
6. After 15 seconds the system requests the establishment of path 5, from node A to node B. The new path is created on along path A -> B, using wavelength 1.
7. After other 10 seconds, the system requests another paths, from A to D. LSP 6 is established along route A -> B -> D, using wavelength 4.
8. Finally after 10 seconds, 4 requests for the teardown of the active paths are sent, with an interval of 10 seconds between two consecutive deletions.
9. After that, the system has explored all the feasible paths, given the testbed specific configuration, and has returned to the initial situation, with no paths established.

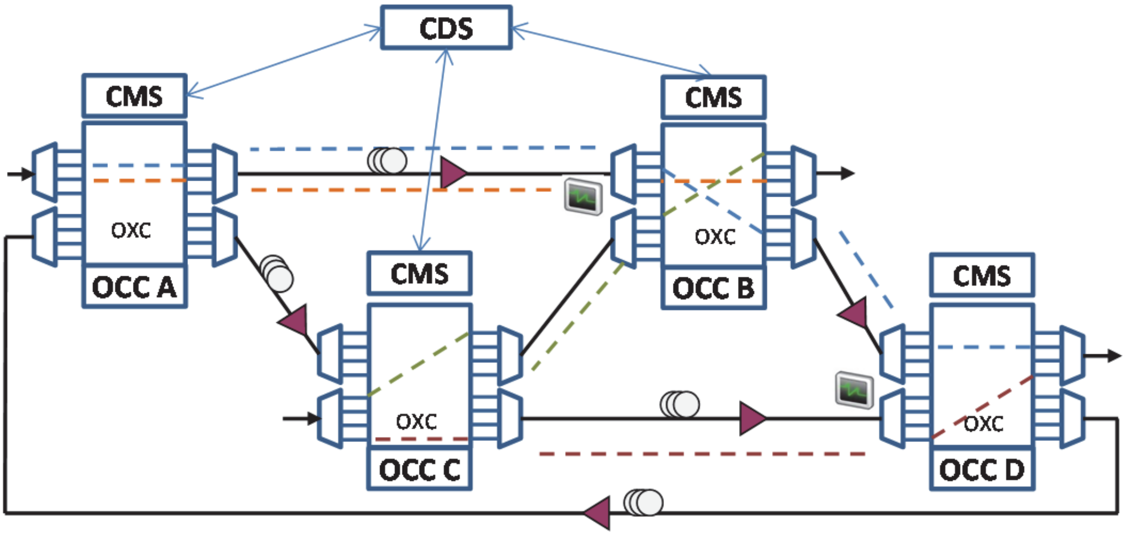


**Figure 6. Scenario 1: Setup and teardown of 4 channels**

Figure 6 shows the initial setup of the four channels using three different wavelengths.

The following measurements have been taken for this scenario:

- **Average setup time: 2001 ms** (from USER_PATH_REQUEST arrival at CDS to MONITOR_SETUP messages reception at NMSA agent side)
    - Average CDS PATH evaluation time: 495 ms (from USER_PATH_REQUEST arrival at CDS to PATH_REQUEST sending to RSVP moduler)
    - Average PATH SETUP time: 735 (from PATH_REQUEST sending to RSVP to PATH_RESPONSE arrival to CDS)
- **Average CDS PATH ACTIVATION evaluation time: 305 ms** (from PATH_RESPONSE arrival at CDS to PATH_ACTIVATION request sending to RSVP moduler)
    - Average PATH ACTIVATION time: 102 (from PATH_ACTIVATION request sending to RSVP to PATH_RESPONSE for activation arrival to CDS)
    - Average PATH ACTIVATION time: 90 (from PATH_ACTIVATION positive response to last MONITOR_SETUP message arriving at destination NMSA agent module)
- **Average USER PATH RESPONSE time: 274** (from from last MONITOR_SETUP sent to USER_PATH_RESPONSE message asending to USER module)
- **Average teardown time: 498 ms** from USER_PTEAR_REQUEST arrival at CDS to USER_PTEAR_RESPONSE messages sending to USER module)

## 2.4 Scenario 2: Dynamic reconfiguration due to failure and high BER

To validate the correct operation of the CHRON using the cognitive engine under the event of a failure and a high BER, the following simple scenario has been setup which consists of simple OOK channels and shown in Figures 2-4:

1. The network needs an OOK connection between node A and node D.
2. The CDS decides that the best path for this link is the path ABD.
3. The CDS set-ups this link by using the CMP control packets.
4. At some point, the fiber link between A and B is cut (failure).
5. The power monitors in links AB and BD, notify the Cognitive Decision System (CDS) through the Control and Management Plane (CMP) about the problem in the links.
6. The CDS processes the data from the monitors and decides that the best alternative path is route ACBD.
7. The CDS sends the new path to the CMP and the CMP is used to setup the new path by reconfiguring the MEMS at nodes A, B, C, and D.
8. The new connection (ACBD) however has very low quality (high BER). The BER monitors in the receiver of node D, sends an alarm signal to the CMP about the problem with the high BER.
9. The CMP notifies the Cognitive Decision System (CDS) for the high BER of the specific link (ACBD).
10. The CDS decides that the alternative path is ACD. The CMP is used to control the reconfiguration of the MEMS switches in order to establish the new path.
11. The quality of this new link is very high (low BER) and the CDS writes to the database the results of the new path.
12. When the problem in fiber link AB is fixed, then the CDS establish the original connection using the original link ABD.
13. When the same problem appears in the connection AB at some point, the CDS checks the database to look if the same problem has appeared in the past. Using the database info, the CDS knows that the best alternative path is ACD and not ACBD.

14. The CDS establish directly the new connection ACD since it is already known that the connection ACBD will result to high BER.

The operation of this scenario is depicted in Figures 2-4 and the timing diagram is shown in Figures 5-7. In the first appearance of the problem in link ABD, the restoration took 7542 ms. The CDS received the alarm for the fiber problem by the power monitor after 89 ms, and it received the decision for the new connection after 2272 ms. The reconfiguration of the optical MEMS switches took 35 ms in order to rear down the connection ABD (this time includes the time for all of the optical MEMS switches). At time 2700, the CDS initiates the new connection and at time 3797, the new connection has been established. At time 3832, the BER monitors send an alarm message to the CDS due to the high BER of this channel. The CDS initiates the tear down of this connection at time 6096 ms. After 106 ms, the ACBD channel has been torn down and at time 6503 ms the new path is initiated. After 845 ms, the new path ACD is established and the BER monitors send the BER of the new established path.

On the other hand, using the CDS, when the same problem appear in the network, the CDS has “learnt” what is the best alternative path and it directly setups the best alternative path. The time for the restoration in the second instance of the same problem is as low as 3872 ms. As it is shown in this Figure the use of the CDS reduces the time of the link restoration by over 48%.

The specific scenario shows a very simple case with just 4 nodes for the reconfiguration of the optical network based on the cognitive engine. However, the main target of the CHRON testbed is to demonstrate that the developed cognitive engine (CDS) and the enhance control plane that has been developed can be realized in a physical testbed and to demonstrate the main advantages of the cognitive engines.

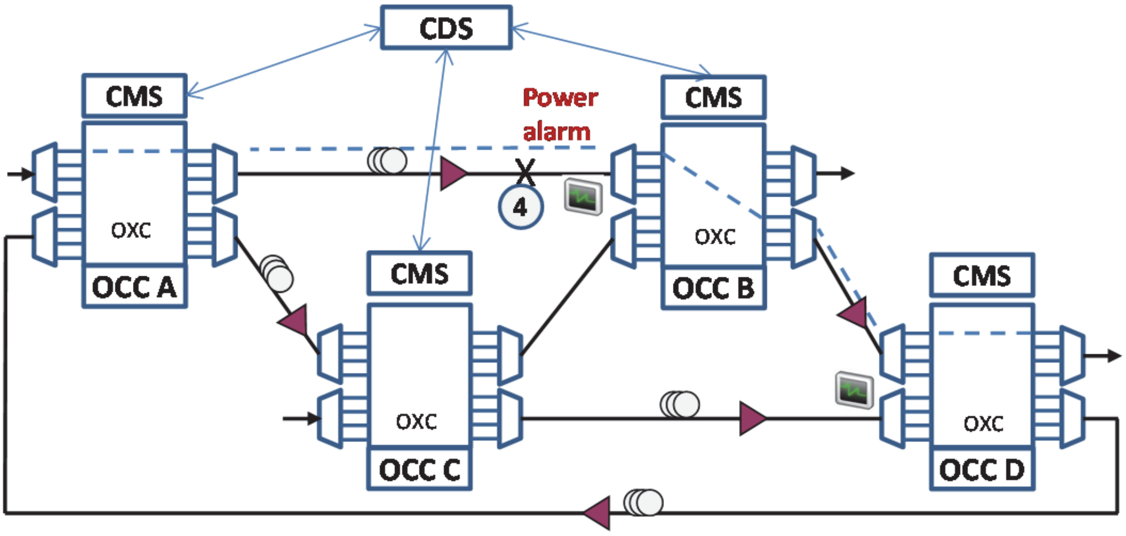


Figure 7. Initial setup and fiber cut

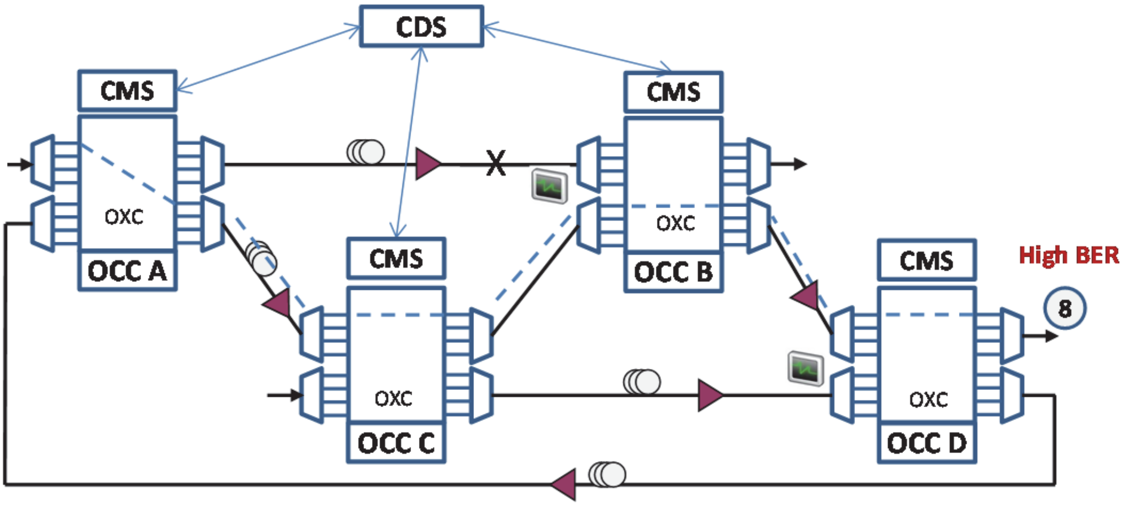


Figure 8. First alternative path: ACBD, High BER alarm

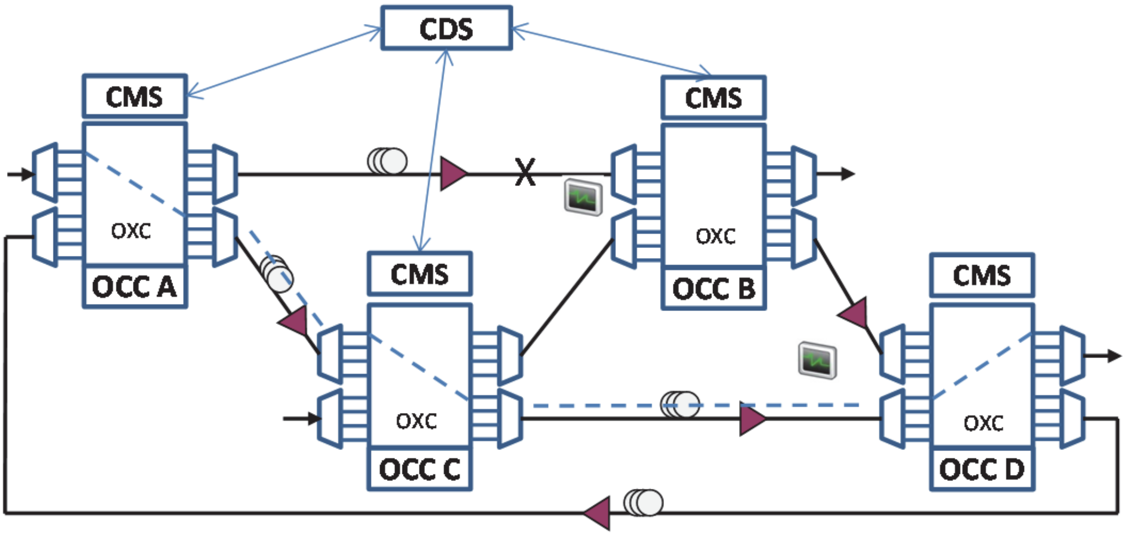


Figure 9. Final path establishment: ACB. High quality of the channel (low BER)

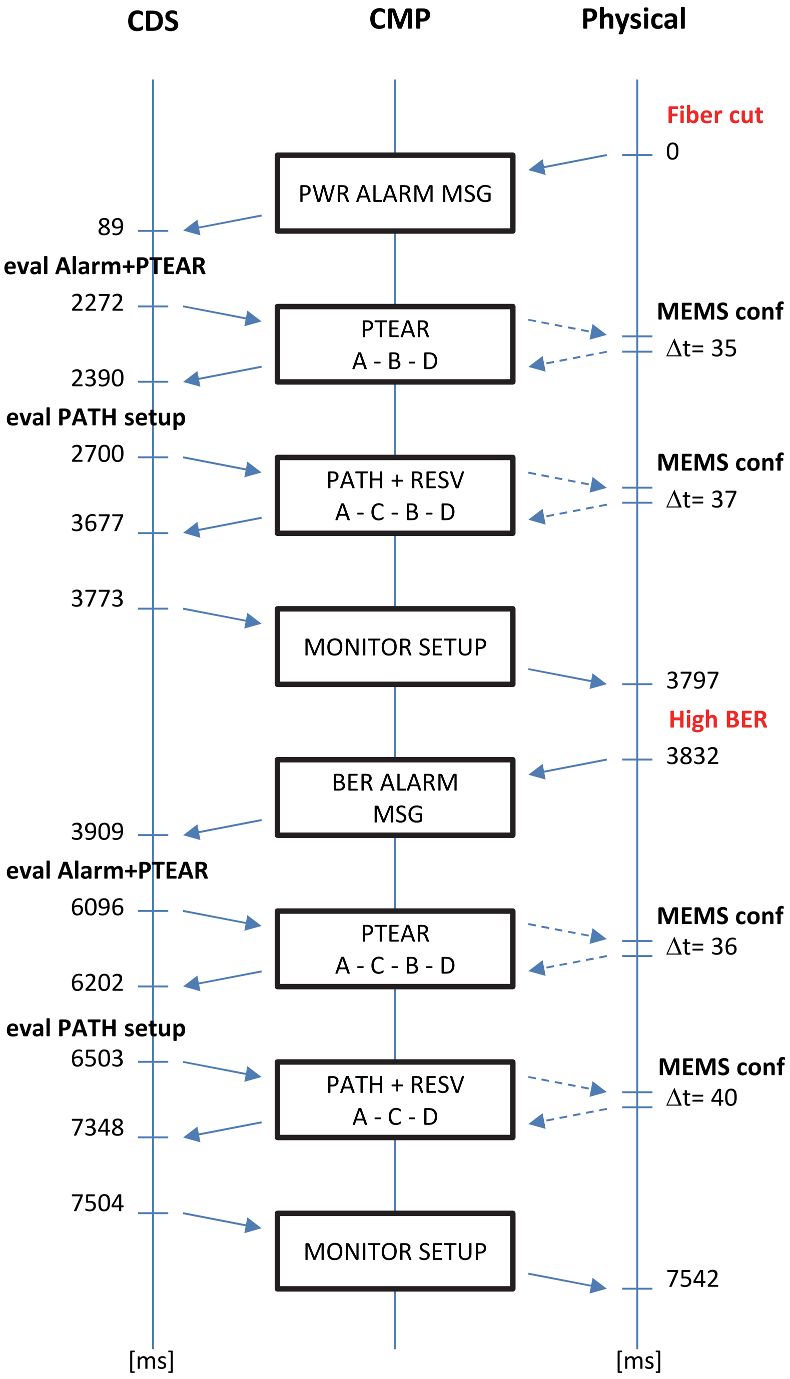


**Figure 10. Restoration of the lightpath AD (first attempt: learning)**

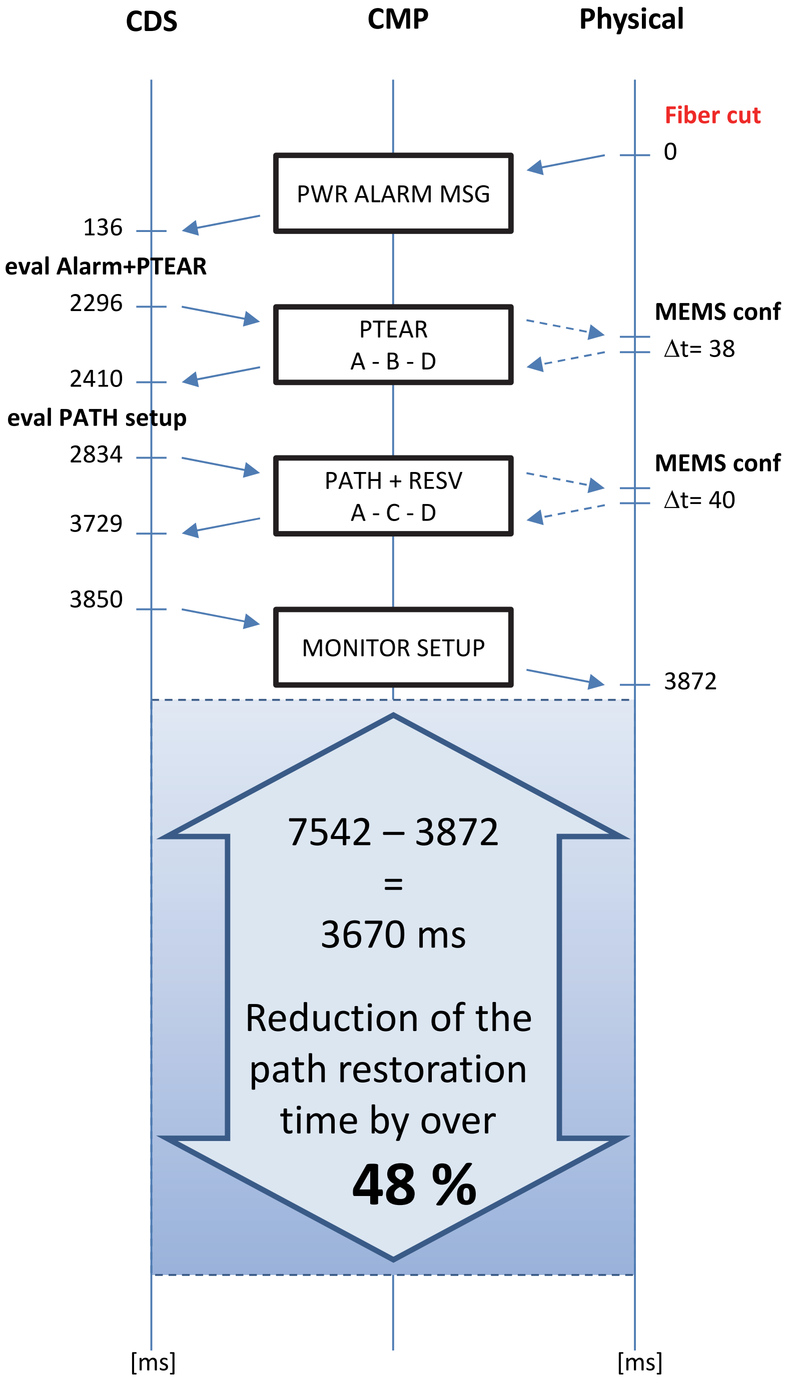


**Figure 11. Restoration of lightpath AD (second time)**

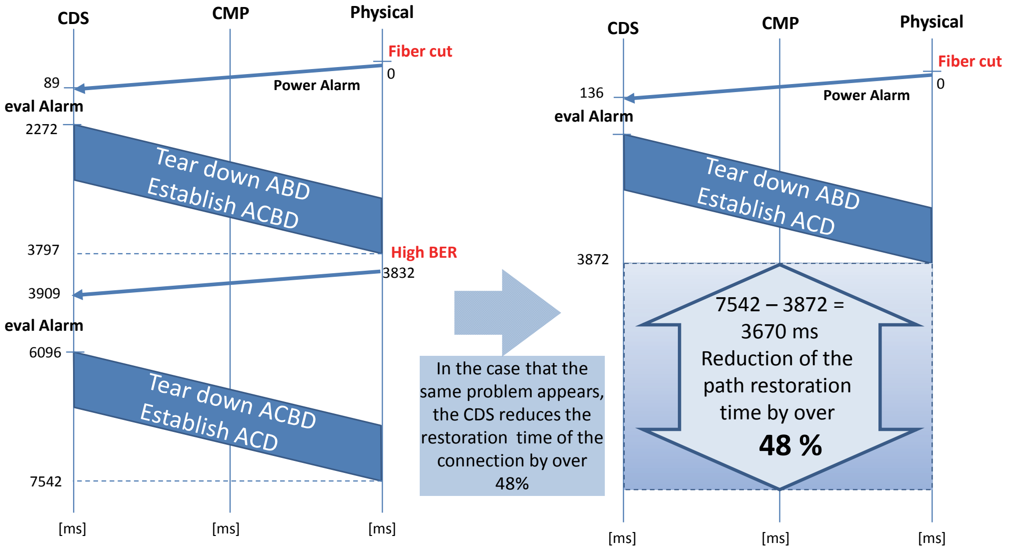


**Figure 12. Lightpath restoration using the CDS engine (showing both the first and the second attempt)**

## 2.5 Scenario 3: Virtual Topology Design

The last scenario that was used to validate the CHRON concepts was aiming to prove the capabilities of the Virtual Topology Design (VTD) module of the CDS. It shows the capabilities of the CDS to adapt to changing traffic matrixes during time. Actually, since the reconfigurations that can be done in the testbed are limited due to the low number of transmitters and receivers and the small size of the network, in the following test we want to demonstrate that the CDS is capable of doing reconfigurations. However, the complete design of virtual topologies with forecasting is done in the emulator (Deliverable 6.2).

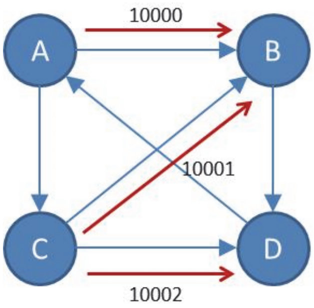
The algorithm that performs the Virtual Topology Design is a cognitive multiobjective algorithm that optimizes congestion, operational expenditure (OPEX) and the number of lightpaths to be changed between the current VTD and the new one.

To demonstrate the proper behavior of CHRON testbed in this scenario, the system is injected every 30 seconds with a traffic matrix. This process is repeated 5 times; therefore, 5 traffic matrixes are tested. After receiving each matrix, the CDS performs an evaluation with the Virtual Topology Design module which is going to design candidate virtual topologies for the incoming traffic matrix and it decides if it is worth to have a change of the network configuration and, if so, triggering the appropriate operations.

The foreseen reaction of the system to each of the matrix traffic is the following:

1. **Traffic Matrix 1 (**Figure 13**)**: The system is starting from scratch, i.e., no lightpath is established in the network. Taking into account the traffic matrix, the virtual topology designed and established consists of three lightpaths (10000 from A to B, 10001 from C to B and 10002 from C to D ). The resultant virtual topology shows a congestion (i.e. the traffic carried by the most loaded lightpath) equal to 1. The VTD evaluation time

was 1119 ms. The reconfiguration time (the time required to set up the lightpaths since the arrival of the traffic matrix) was 3134 ms.

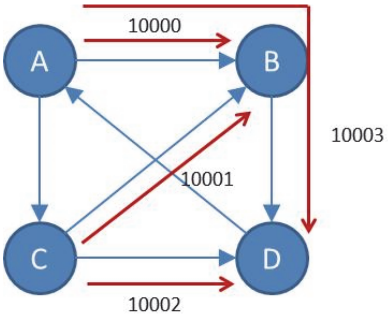


| s/d | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 1 | 0 | 0 |
| B | 0 | 0 | 0 | 0 |
| C | 0 | 0.5 | 0 | 0.5 |
| D | 0 | 0 | 0 | 0 |

Congestion = 1

**Figure 13. VT established for the Traffic Matrix 1. Blue lines are physical link while red lines are lightpaths**

2. **Traffic Matrix 2**: To satisfy this traffic matrix, the CDS decides to establish a virtual topology based on the established one but adding a new lightpath, 10003, from A to D (via B). In this case the congestion is 0.5 (the occupancy of all the lightpaths is 50%). The VTD evaluation time was 1430 ms. The reconfiguration time (setting up the new lightpath path) was 1196 ms.

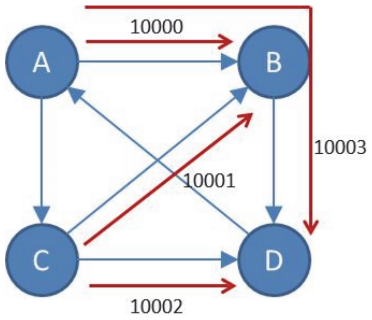


| s/d | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 0.5 | 0 | 0.5 |
| B | 0 | 0 | 0 | 0 |
| C | 0 | 0.5 | 0 | 0.5 |
| D | 0 | 0 | 0 | 0 |

Congestion = 0.5

**Figure 14. VT established for the Traffic Matrix 2. Blue lines are physical link while red lines are lightpaths**

3. **Traffic Matrix 3**: To satisfy this traffic matrix, the CDS decides not to reconfigure the virtual topology. The VTD evaluation time was 1241 ms. The reconfiguration time (doing nothing) was 0 ms.

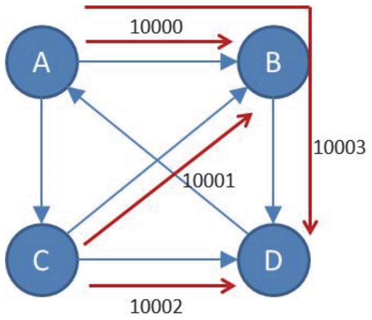


| s/d | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 0.5 | 0 | 0 |
| B | 0 | 0 | 0 | 0 |
| C | 0 | 0.5 | 0 | 0.5 |
| D | 0 | 0 | 0 | 0 |

Congestion = 0.5

**Figure 15. VT established for the Traffic Matrix 3. Blue lines are physical link while red lines are lightpaths**

4. **Traffic Matrix 4**: Similarly to Traffic Matrix 3, no reconfiguration was required. The VTD evaluation time was 1810 ms. The reconfiguration time (doing nothing) was 0 ms.

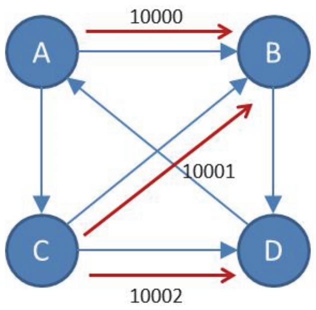


| s/d | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 0.5 | 0 | 0 |
| B | 0 | 0 | 0 | 0 |
| C | 0 | 0 | 0 | 1 |
| D | 0 | 0 | 0 | 0 |

Congestion = 1

**Figure 16. VT established for the Traffic Matrix 4. Blue lines are physical link while red lines are lightpaths**

5. **Traffic Matrix 5**: In this situation, the CDS decides to release one of the lightpaths (10003) of the current virtual topology, as it is not needed to carry the traffic. Hence, the reconfiguration consists of the tear down of one lightpath. The VTD evaluation time was 1681 ms. The reconfiguration time (deleting the path) was 570 ms.



| s/d | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 0.5 | 0 | 0 |
| B | 0 | 0 | 0 | 0 |
| C | 0 | 0 | 0 | 1 |
| D | 0 | 0 | 0 | 0 |

Congestion = 1

**Figure 17. VT established for the Traffic Matrix 5. Blue lines are physical link while red lines are lightpaths**

Therefore, with this test it can be concluded that the system is working as expected, establishing and tearing down lightpaths in order to reconfigure the network depending on the traffic demands and network resources.

## 2.1 Performance evaluation of QPSK and 16QAM channels

The scenarios using the OOK were mainly used to validate the correct operation of the cognitive system using homogeneous networks. To validate the operation of the cognitive system using more advanced scenarios we tested the CHRON testbed using the QPSK and the 16QAM channels. To use these channels, we first had to measure the quality of these channels.

The performance of QPSK and 16-QAM when transmitted in the testbed is shown in Table 1. The performance in terms of BER was estimated from error vector magnitude of the received constellation [3]. This allowed us to estimate very low BER for which the conventional BER measurement would be impossible.

**Table 1. Performance of 16QAM and QPSK in the CHRON testbed. Measured channel marked with dark gray.**

| Path | C31 193,1 GHz | C30 193 GHZ | C29 192,9 GHz | C28 192,8 GHz | BER | -lg(BER) |
|---|---|---|---|---|---|---|
| B2B 16QAM | — | — | 16-QAM | — | 3,2E-04 | 3,5 |
| AB | OOK | OOK | 16-QAM | OOK | 6,3E-04 | 3,2 |
| ACB | OOK | OOK | 16-QAM | OOK | 1,0E-03 | 3,0 |
| ACD | OOK | OOK | 16-QAM | OOK | 1,0E-03 | 3,0 |
| ACBD | OOK | OOK | 16-QAM | OOK | 2,0E-03 | 2,7 |
| ACBDAB | OOK | OOK | 16-QAM | OOK | 5,0E-03 | 2,3 |
| B2B QPSK | — | — | — | QPSK | 1,0E-16 | 16,0 |
| ACD | QPSK | QPSK | QPSK | QPSK | 3,2E-12 | 11,5 |
| ACBD | QPSK | QPSK | QPSK | QPSK | 2,5E-11 | 10,6 |
| ABD | QPSK | QPSK | QPSK | QPSK | 1,3E-14 | 13,9 |
| ABD | OOK | OOK | QPSK | OOK | 4,0E-15 | 14,4 |

Figure 18 shows the performance of 16QAM represented graphically, as a function of total distance travelled between source and destination nodes in the testbed. The performance of all, except for the longest ACBDAB lightpaths, is above the assumed FEC threshold of $3.8\times10^{-3}$.

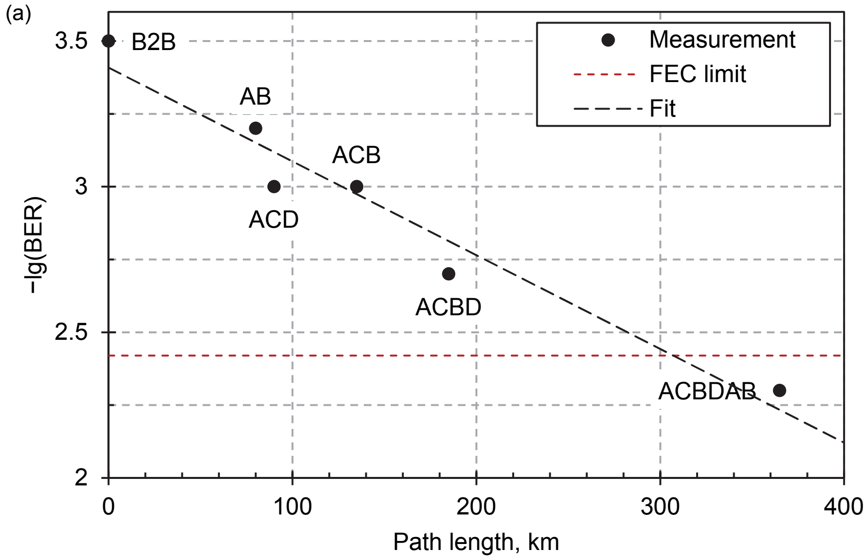


**Figure 18. Graphical representation of 16QAM performance in the testbed.**

Figure 19 shows the received constellation diagram for the lightpaths with worst BER.

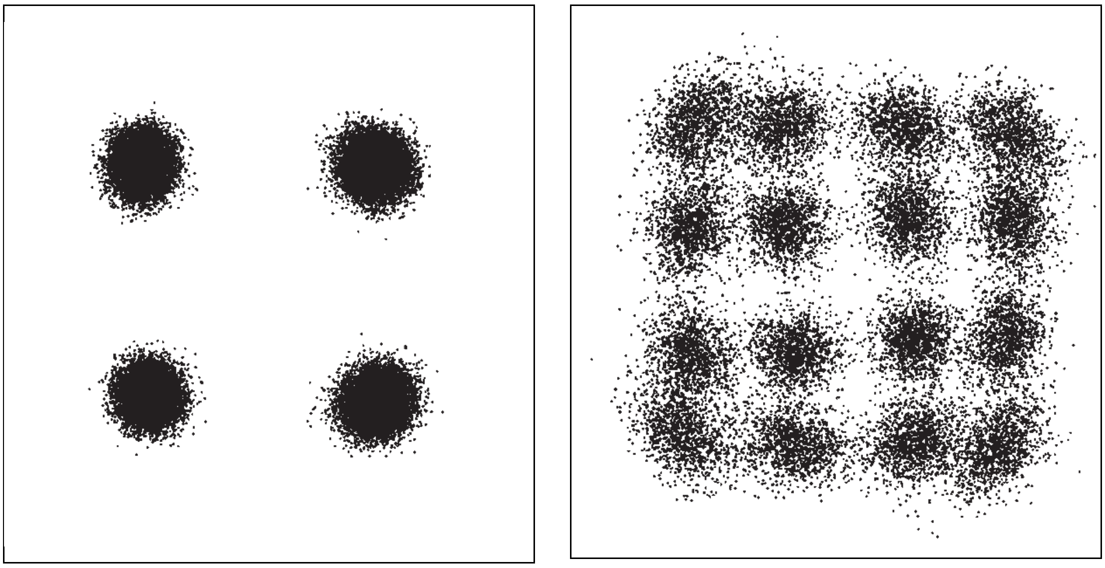


**Figure 19. Received 16QAM and QPSK constellations.**

The main aim of the integration of the physical testbed of the QPSK and 16-QAM channels was to demonstrate the correct operation of the CHRON with advanced modulation formats. However, it is noted that the setup of a path, the tear-down of a path or the reconfiguration of a path is modulation- and rate-independent. The data-rate and the modulation of the channels only affect the time for the cognitive engine (CDS) and not the time to setup the lightpath or to tear it down. Therefore, the same results that have been presented in the previous scenarios (average time for lightpath setup, etc.) can be also applied in the more advanced modulation formats.

# 3 Conclusions

The performance evaluation of the CHRON concepts using the physical testbed has demonstrated the realization of the CHRON in a real network. All of the components that have been developed in the CHRON (cognitive engine, novel control plane, physical electro-optical modules, etc.) have been implemented and integrated successfully into one unified physical testbed. Furthermore, the performance evaluation has shown that the proposed CHRON concepts can achieve to reduce significantly the restoration time of typical failures in the optical networks through the efficient use a cognitive engine.

# 4 References

[1] UVa, CEDETEL, CREATE-NET, TP SA, HWDU, "Deliverable 4.1: Feedback system between the monitoring and the cognitive systems", CHRON project, December 2010.

[2] Report associated to deliverable 6.3 “The integrated physical test-bed for the validation of the CHRON concept”, CHRON project, September 2013

[3] R. Schmogrow et al., “Error Vector Magnitude as a Performance Measure for Advanced Modulation Formats,” in IEEE Phot. Technol. Lett., 24(1), pp. 61-63, (2012).

# 5 List of Acronyms and Abbreviations

| | |
|---|---|
| BER | Bit error rate |
| CACRM | Channel Access Control/Resource Manager |
| CDS | Cognitive Decision Systems |
| CMP | Control and Management Plane |
| FPGA | Field Programmable Gate Array |
| GMPLS | General Multiprotocol Label Switching |
| HINT | Hardware interface |
| LSP | Label Switched Path |
| MEMS | Micro-Electro Mechanical Switch |
| NMSA | Network Management System Agent |
| OCC | Optical Cross-connect Controller |
| OOK | On-Off Keying |
| OXC | Optical Cross Connects |
| QAM | Quadrature Amplitude Modulation |
| QoS | Quality of Service |
| QPSK | Quadrature phase-shift keying |
| RSVP | Resource Reservation Protocol |
| VTD | Virtual Topology Design |