5th Int. Conference on

Emerging e-learning Technologies and Applications

The High Tatras, Slovakia
September 6-8, 2007

AUTOMATIC WAN TOPOLOGY INTERCONNECTION AND IT'S USAGE IN CNAP NETWORKING LABORATORIES

Petr Grygárek, David Seidl
Regional Cisco Networking Academy
Faculty of Electrical Engineering and Computer Science
Technical University of Ostrava, Tř. 17. listopadu, 708 33 Ostrava, Czech Republic
Tel.: (+420) 597 323 263, Fax: (+420) 597 323 099
E-mail: petr.grygarek@vsb.cz, http://www.cs.vsb.cz/grygarek

Abstract.
In practical education in networking laboratories it is useful to be able to interconnect various WAN topologies quickly and efficiently. In the article we present our hardware devices and architectures we developed for such automatic interconnection of WAN topologies - either in the scope of a single laboratory or fully distributed. Basic experiences gained during practical usage of developed prototypes in CNAP teaching and planned improvements are also described.

Keywords: Communication Technologies, Virtual Laboratory, CNAP

1. INTRODUCTION

During practical education in networking laboratory, it is necessary that students work on various topologies of networking devices. Unfortunately, the process of physical connecting of network topology is both time-consuming and error-prone and commonly prevents students to concentrate on the configuration of particular protocol or technology which is the main objective of the respective lesson. Our experience revealed that although it is inevitable to let students connect network topologies manually at the beginning of their study to give them concrete idea about WAN/LAN interfaces usage, it is much more effective to concentrate on upper-layer protocols and don't waste time with repeated physical layer troubleshooting later.

The another issue with frequent topology changes reported by many LCNAs is that students often do not manipulate network interface connectors with enough care. This often results to mechanical damage of rather expensive WAN interface cables or router's connectors and a need of their replacement. This problem is most serious in cases of non-modular routers often present in earlier CNAP bundles, where damaged interface cannot be easily replaced. The frangibility of new Cisco Smart Serial connector types is also an issue.

To avoid the above mentioned problems, we searched for methods how to interconnect network topologies automatically without physical human interaction. Although the original motivation of our research was the need to automatically change topologies in our Distributed Virtual Networking Laboratory [7] which resulted to the Virtual Crossconnect architecture [2], we soon realized the potential of developed prototypes to solve problems described above.

For Ethernet ports interconnection, we use standard LAN switches and VLAN-based interconnection. Using VLAN tunnelling (also called dot1QinQ sometimes), we are even able to interconnect trunk links of various devices and have the switching element be completely invisible for the laboratory devices [8]. This way we can transparently pass CDP, STP, VTP, PAgP/LACP and other L2 protocols discussed in CNAP curriculum between connected devices. Unfortunately, there exists no similar commercially-available and cheap solution for WAN links. This is why we decided to develop a series of our own devices for WAN port automatic interconnection. The basic ideas, architectures and experiences with these devices will be discussed in the following article.

2. THE FIRST GENERATION CROSSCONNECTS

All WAN port crossconnect prototypes we finished and use up to now have the similar philosophy (fig. 1). All network devices' WAN ports are connected to interfaces of a single crossconnect which can be configured to interconnect arbitrary pairs of connected ports. The configuration is accomplished via RS-232 console port using a simple "IOS-style" command line interface (CLI) available to instructor or lab administrator or possibly directly to students. As in Cisco IOS, command completion, command abbreviations and context-based help system were implemented. The configuration is maintained in RAM but may be also stored into internal flash memory so that it can be loaded automatically when the crossconnect device is powered on. The device is controlled by Atmel 8051ED2 microprocessor which acts as a CLI command interpreter and configures switching array according to user's requirements. The CLI allows instructors or other authorized person just to Cut&Paste one of the previously prepared topology configurations into the crossconnect using HyperTerminal, Minicom or similar terminal emulator program.

In some cases, it is useful to be able to control the crossconnect not only by directly connected RS-232 terminal, but also remotely via intranet or Internet. It allows us to share access to the crossconnect's control port by multiple users. In this way students may modify crossconnect configuration by themselves using connection over laboratory LAN, which is useful especially during unattended practicing of groups of CNAP students. Our solution for RS-232 to TCP conversion was to use relatively cheap commercially-available modules, in this case Charon II [5] for mutual RS232 to Ethernet conversion and Sollae EZL80c [4] for RS-232 to WiFi (802.11b) conversion, as can be seen at fig. 1. Our current aim is to extend function of Charon II module by a simple proxy authentication capability, which will authenticate the user before giving him/her an access to the crossconnect's control port. The Web-based GUI to isolate user from crossconnect's text-based configuration language might be also implemented into Charon II module quite easily.

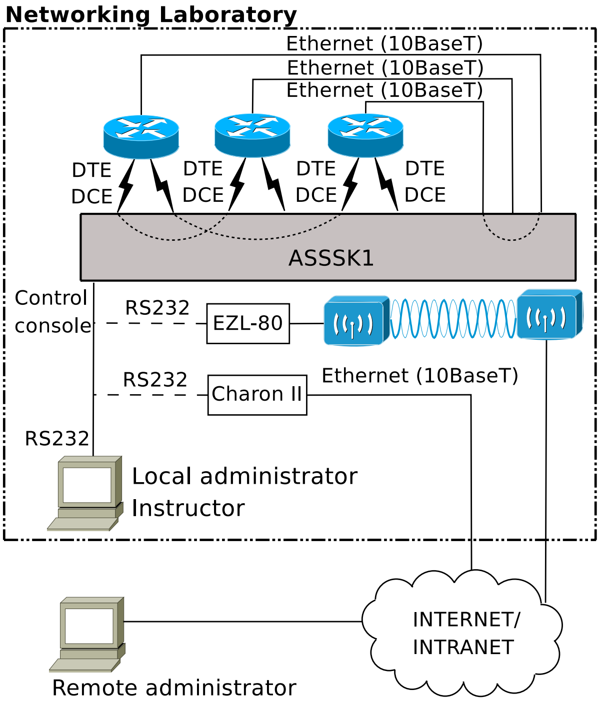

Fig. 1. Basic Crossconnect Philosophy

As will be described in detail later, we subsequently tested and implemented various approaches for actual signal switching, but the substantial part of controlling software has always been reused in the individual prototype devices.

2.1 WAN INTERFACE TYPE SELECTION

One of the main issues of crossconnect design was a selection of physical WAN interface type our device should support. Since we use our crossconect primarily to interconnect WAN ports of Cisco routers which can provide multiple physical interfaces and let the user to choose the required one only by usage of an appropriate WAN cable, we could choose from ITU-T X.21, ITU-T V.35, EIA/TIA RS-232 and EIA/TIA RS-499 (all synchronous). Since we wanted to limit a number of signals to switch and also take the interface's connector availability into account, we chose to use RS-232 because it doesn't use symmetrical signals neither for data nor control and it's CANNON DB-25 connector is cheap and easily available. The Cisco DB60/SmartSerial to RS232 DTE cables are also typically much more cheaper than other WAN cable types. Since we use "null modem" interconnection we need only to switch RxD and TxD signals and provide clock signals, as will be discussed later.

Another important issue we had to solve was the problem of clocking. In reality, routers at both sides of the leased line provided by telco act as DTEs and are connected with synchronous modems using „DTE cable". Clocking signal for both routers is provided by respective modem. Since it is inefficient to have so many modems in the laboratory, most people commonly connect WAN interfaces of Cisco routers directly, using a pair of two different cables for DCE and DTE side. The router connected with a "DCE" cable provides clocking for both directions of the communication if instructed to do so by an IOS command. It means that for that type of direct interconnection we are only able to connect a pair of interfaces if one interface is connected with a DCE cable and another one with a DTE cable. So we have a problem with cable type to use between router's WAN port and crossconnect port if we want to be able to connect arbitrary pairs of connected WAN ports by the crossconnect.

To allow our crossconnect to connect arbitrary pair of WAN ports, we decided to take completely different approach, which much more resembles the real usage of leased WAN links. All WAN ports are connected to the crossconnect using DTE cables and the crossconnect itself provides clocking for all devices, exactly as would a modem at the end of leased line do. In the newer prototype, we are even able to set various clock rates for individual pairs of connected ports. From the student's perspective, the crossconnect may be viewed as a telco cloud which provides leased line services including clocking and he/she does not have to take care about clocking at all a the router side.

2.2 THE ANALOG CROSSCONNECT

The very first version of our crossconnect (called ASSSK-1) was developed by David Seidl in his MSc. thesis [1]. The general aim of the thesis assignment was to develop a crossconnect based on analog switch array core suitable for connecting of signals of various networking technologies. Individual WAN ports are attached to the switching core via interface modules, which adapts various electrical interfaces' signals to the voltage range suitable for the switching core. The device (fig. 2) may accomodate up to 16 modules.

The switching core is composed of two Zarlink MT8816 analog switch array integrated circuits [6], which together from a 16 x 16 matrix. The matrix allows to connect each of 16 TxD signals to any of 16 RxD signals, so that arbitrary 8 pairs of modules may be connected together. It is even possible to loopback any interface for testing purposes.

Various interface modules may be developed to switch signals of individual interface types. The only limitation is the frequency range of the used switching array (30MHz).

Fig. 2. ASSSK-1: The Analog Crossconnect

That range proved sufficient for 10BaseT Ethernet and synchronous RS-232 up to 2 Mbps during our extensive testing. We developed a double-interface modules, which allows to connect either RS-232 WAN port or 10BaseT Ethernet. Because of different electrical characteristics and processing of RS-232 and Ethernet signals, it proved most effective to use a small mechanical relay to choose which of the two interfaces available at the module will be really connected to the switching core. The reachable bitrate of switched serial WAN interfaces is somewhat limited by RS-232 to TTL convertors (the capacitor-based charge pump used to create ±12V for RS-232 interface), but we currently plan to provide external 12V DC supply to avoid this problem.

The block diagram of the ASSSK-1 device is depicted at fig. 3. The Control Processor interacts with user using CLI and configures analog switching array accordingly. We decided to use separate microprocessor (the Clock Processor) to provide clocking for individual WAN lines.

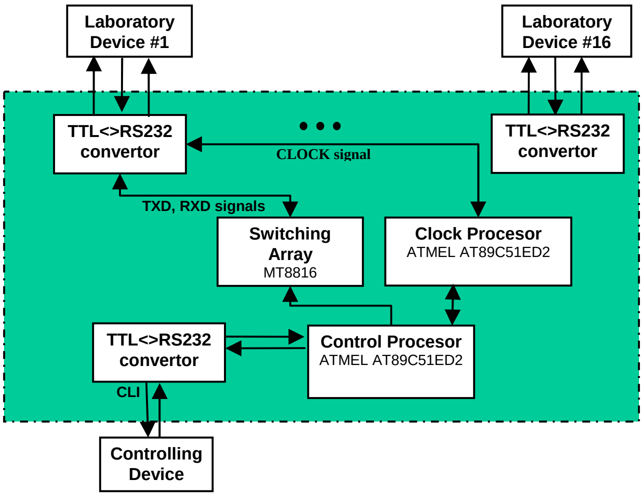

Fig. 3. ASSSK-1 Block Diagram

To support potential future extensions, we decided to provide I2C bus implemented by ATMEL microprocessor for the usage at the interface modules. It means that various I2C-enabled devices may be attached the similar way as interface modules and accessed from the control microprocessor. Currently we assess the attachment of external Flash memory to let instructors to store multiple pre-defined interconnection configurations and easily choose one of them when a need to change topology arises during laboratory work. By implementation of a simple user interface consisting from a numeric keyboard and LCD display, we also plan to provide a mechanism to let the instructor change configurations very quickly and efficiently without an external control terminal (most probably PC).

The cost of electrical components and other material to produce the ASSSK-1 crossconnect was about 12 thousands CZK.

2.3 FPGA-BASED CROSSCONNECT

After a period of usage of ASSSK-1, we decided to redesign an architecture based on the experience with the original prototype. Switching of Ethernet ports proved inefficient, because we need to switch faster interfaces than 10BaseT today, which is not possible because of frequency limitation of the used analog switch array. This is why we decided to concentrate on WAN interface switching in the future hardware development and interconnect Ethernet ports using standard VLAN-aware 10/100/1000 Ethernet switches and VLAN/802.1q tunnelling approach, as mentioned above.

The main aims of the new crossconnect version was to make the device more replicable and compact, decrease it's production cost and increase flexibility. It also turned out unnecessary to be able to switch various WAN physical interface types, since RS-232 proved most efficient during usage period of the ASSSK-1. This is why we abandoned the modular architecture and decided to implement interface circuitry on the baseboard instead of on modules. We also used FPGA technology and VHDL to implement the device much more efficiently. The most important change is that the FPGA-based switching core is now fully digital. Except the switching function, the FPGA circuit also provides clocking for individual ports, based on the frequencies pre-set to it's configuration registers by control microprocessor. This way we can easily simulate WAN links of various speeds which is necessary in many labs and case studies of CNAP courses, especially CCNP1 and CCNP2.

The prototype (fig. 4) was developed by Petr Sedlar in his master thesis [3], produced and successfully tested. Much higher bitrates are reachable with digital switching matrix than with the previous analog one. Up to twenty interfaces may be made available at the chassis. This allows us to interconnect up to 10 2-WAN-port routers, which is more than enough for a single CCNP bundle. In real teaching, the experience shows that it is often better to use the crossconnect to switch ports of multiple smaller and independent lab pods. The FPGA-based crossconnect is very easy to modify because of it's capability of in-system reprogramming of both Atmel control microprocessor and FPGA core. The block diagram of the FPGA-based device called ASSSK2 is depicted at fig. 5.

Because of implementation of interface circuitry on the baseboard instead on interface modules in ASSSK-2, the number of mechanical contacts was reduced considerably

Fig. 4. ASSSK-2

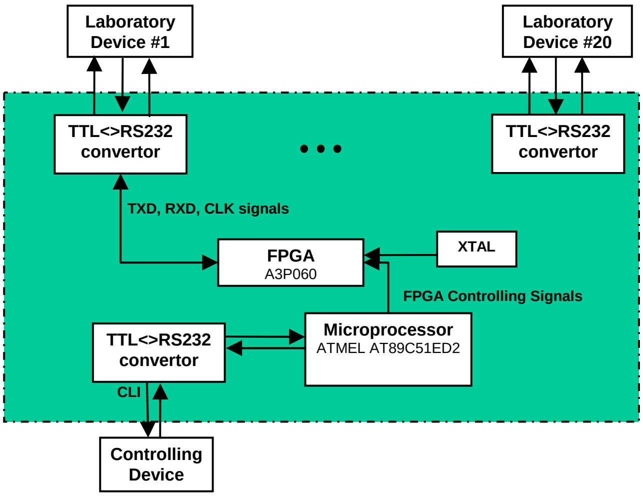

Fig. 5. ASSSK-2 Block Diagram

so the new crossconnect is not only cheaper and more compact, but also more reliable. It is important in particular in case of lab pods accessible for CNAP students remotely all the day without any instructor attendance. The material expenses to construct ASSSK-2 are about 8 thousands CZK.

In the future, we intent to integrate the whole logic (i.e. the switching array and control processor) into the more advanced type of FPGA integrated circuit. We expect that it will further decrease the cost, improve reliability and extend the flexibility of the crossconnect.

3. PASSING TRAFFIC BETWEEN MULTIPLE CROSSCONNECTS

The limitation of first generation of our crossconnect solutions was that there was no possibility to switch traffic across multiple crossconnect instances. Although there is 16 or 20 ports at the available models, the scalability is somewhat limited because even if we produce multiple crossconnect instances, we are only able to connect WAN ports of network devices connected to the same crossconnect. This is why we searched for solutions how to let traffic pass across multiple crossconects connected together.

The first idea was just to reserve some number of standard crossconnect ports for interconnetions between crossconnects. Unfortunately, this solution would be rather inefficient because 4 ports in total would have to be consumed for a single interconnection of two WAN ports connected to a pair of different crossconnects. It would be even more in case of a longer chain of crossconnects linked together. A non-trivial issue of clock synchronization between multiple crossconnect devices would have to be solved also. Although it is of course possible to connect crossconnects into more efficient hierarchical structures than a simple chain, we decided to take completely different approach.

The solution for passing traffic between multiple crossconnects we are working on now was influenced by a need to pass traffic between crossconnects located at various, physically distant sites. This requirement first appeared during our Distributed Virtual Laboratory development ([7]), but we also found useful to be able to connect together laboratory equipment located in multiple networking laboratories for some laboratory tasks, like CNAP final exams. The general idea is not just to interconnect physical signals, but read the content of passed PPP/HDLC frames, encapsulate them and tunnel over intranet or even Internet. Frames are decapsulated at the receiving side and sent out of the particular serial port to the WAN interface of the network device they are destined to. This approach scales well because the interconnections between individual crossconnects forms a logical full mesh and in case of sufficient capacity of the underlying LAN/WAN the number of interconnections of network devices connected to different crossconnects is potentially unlimited. We denote solutions adhering the approach described above as a second-generation crossconnects.

3.1. LINUX-BASED CROSSCONNECT

The simplest way to implement handling of HDLC/PPP frames and passing them between multiple switching devices is to use software-based approach. We decided to utilize PC with Linux for that purpose. The architecture we are now experimenting with is described at fig. 6. The PC is equipped with multiport synchronous serial card, which allows to connect WAN interfaces of network devices to be interconnected. We investigate now how standard Linux PPP/HDLC drivers could be used to allow a switching software developed for that purpose to switch PPP/HDLC frames between logical ppp/hdlc interfaces. The switching software will also be able to tunnel frames between multiple crossconnect PCs in UDP datagrams, so that we will be able to create virtual WAN links over campus LAN or the Internet. The remote configuration using Telnet is planned in the first prototype. We denote the above mentioned Linux-based crossconnect architecture as ASSSK-3.

The only serious limitation of this approach we encountered up to now is that commercially available synchronous serial cards are both costly and typically do not provide more than two ports. For this reason, we started to work on our own

FPGA-based card with a higher port density (8 ports as a minimum).

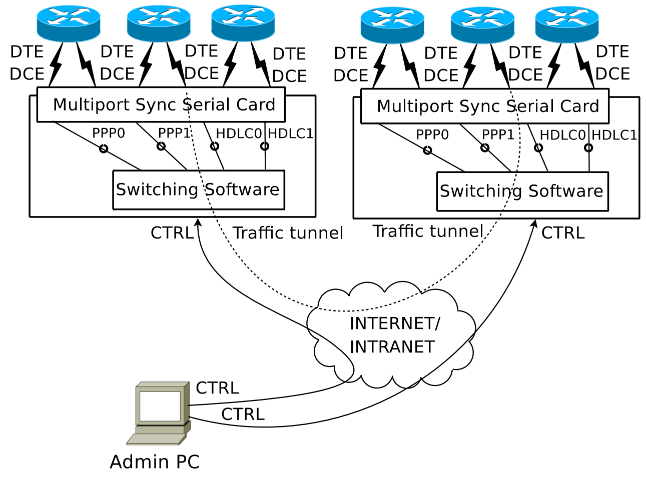

Fig. 6 –ASSSK3: The Linux-based Crossconnect

3.2. EXTENSION OF FPGA-BASED CROSSCONNECT

Another possible approach we are assessing now for processing of PPP/HDLC frames and tunnelling them across the Ethernet is to integrate frame processing intelligence into the current FPGA-based crossconnect (ASSSK-2). The frame separation logic capable of recognizing frame flags and handling bit stuffing seems to be relatively simple to integrate into FPGA. We plan to use some commercially-available embedded module for synchronous serial to Ethernet conversion in the prototype, such as above-mentioned Charon II module. The frame separation logic could be also implemented at this module if the processing power will suffice. The general idea is to reserve a couple of ports of FPGA-based switch array (denoted as internal ports) and connect them internally to the serial-port side of the serial-to-Ethernet conversion modules. The way how the modules will handle frames incoming from switch array and Ethernet interface will be programmed to the modules by crossconnect's control processor. In fact, only destination/source MAC (or IP) address to internal switching array port mapping will have to be configured. The FPGA switch array will then switch frames coming from another crossconnects through Ethernet LAN and Ethernet-to-serial module to it's internal port the same way as if it came from the regular port. Multiple serial-to-Ethernet convertors can be integrated together to limit the size and cost of the construction or a more-powerful convertor capable of handling multiple serial ports could be utilized. The architecture proposal is depicted at fig. 7. Although the number of ports that can be tunnelled via Ethernet is limited by a number of implemented internal ports in this approach, it requires relatively minor changes in the current ASSSK-2 crossconnect design. Except of the integration of convertor modules, only some changes in the control software will be required. We also expect that the controlling RS-232 console will be converted to Ethernet so that we will be able to control multiple crossconnects from a single control entity, which is useful for centralized creation of distributed virtual WAN topologies for education purposes.

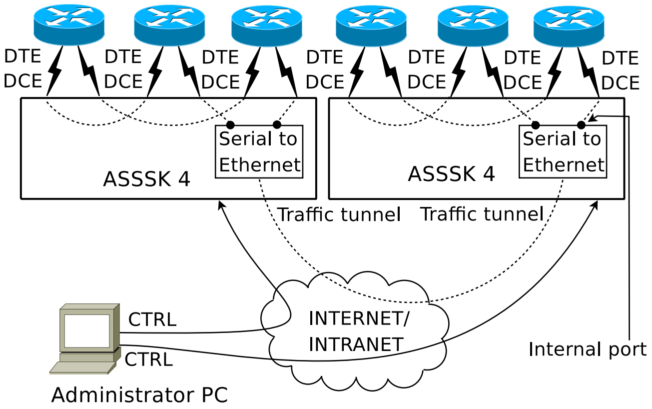

Fig. 7 – Basic Idea of Extension of ASSSK-2 for Frame Tunelling

4. FULLY DISTRIBUTED CROSSCONNECT ARCHITECTURE

As a natural extension of the above mentioned architectures, we proposed a fully-distributed crossconnect architecture, which we believe to be more flexible and cheaper in the practise, particulary for creation of virtual topologies built from equipment scattered in multiple rooms of the campus of even multiple LCNAs.

The architecture is based on a big number of small remotely configurable bidirectional synchronous serial to Ethernet convertors (fig. 8). These convertors are connected to individual WAN ports of network devices (one convertor may potentially handle more than one WAN port). Every convertor may be programmed remotely to which address it has to tunnel PPP/HDLC frames coming from the serial port. Ethernet ports from the convertors connected to WAN ports of network devices at the lab site will be connected together via standard Ethernet switch. Multiple independent lab sites may be connected together via Internet. The architecture requires a controller entity which will create and upload configurations into individual convertors based on the required virtual topology. The convertor modules may be built to provide clocking (i.e. behave as DCE) or to accept clock from the router, so the direct serial interconnection can be completely simulated. We can also pass additional information between modules, such as physical layer up/down state of the respective serial interface.

It is expected that there will be some troubles with passing traffic through firewalls of individual lab sites, so we plan to include proxy capability into convertor modules so that only a limited number of conduits will have to be configured at firewalls.

As depicted on fig. 8, our effort is to build a serial-to-Ethernet modules in such a way so that they can also just pass the serial interface signals through. It will allow to have these convertors connected to WAN ports of network devices permanently, so that students will be able to connect network devices physically during laboratory work. At the same time, we will be also able to connect topology automatically if necessary. The positive side effect of that solution is that serial ports on the routers will be protected from mechanical damage by relatively cheap convertor modules.

The described approach will also allow to tunnel traffic between Ethernet ports of network devices the similar way as for serial lines, so that we will be able to automatically connect the complete topology with the single technology. The only difference will be the interface type at the router's side of the convertor modules.

The experimental prototype of serial-to-Ethernet modules are just being constructed using commercially-available Charon module.

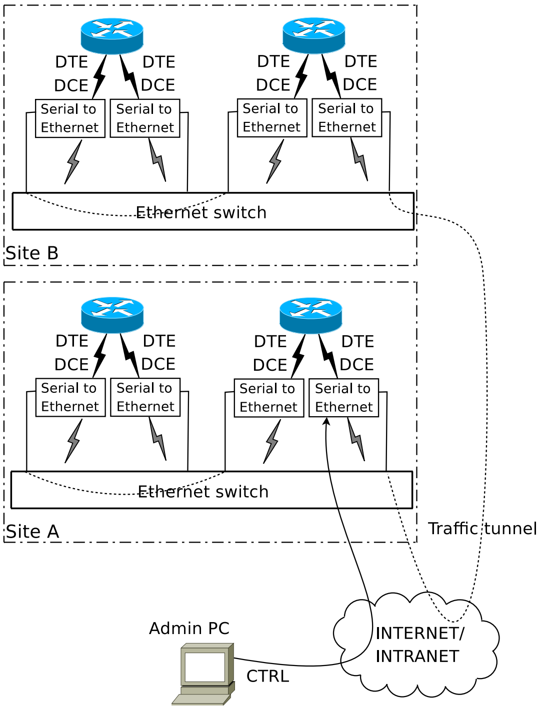

Fig. 8 – The Fully Distributed Crossconnect

4. CONCLUSION

In the article, we presented a couple of architectures and hardware device prototypes for automation of WAN topology interconnection. Lot of approaches presented here proved useful to make practical education in CNAP laboratories more efficient, to support integration of equipment of multiple laboratories and allow remote laboratory access for the purpose of distant learning. Some of the technologies presented here also form a technological basis of the Distributed Virtual Laboratory [7] developed at our university and piloted in cooperation with Silesian University of Opava with support of Czech Educational Scientific Network (CESNET), who provided funds to let LCNAs specialize on particular advanced technologies. This way we can share equipment between LCNAs and create high-quality CNAP virtual lab pod, particularly for CNAP security courses.

We also believe that the proposed technologies which promise to connect laboratory topologies both directly and remotely at the same time will make the lab device usage more efficient, since it will be no longer needed to maintain separate lab pods for direct and remote access.

8. REFERENCES

[1] Seidl, D.: System for Automatic Network Configuration Management. Master's Thesis, Faculty of Metallurgy and Materials Engineering, VŠB-TU Ostrava, 2005. [In Czech]

[2] Grygárek, P., Practical Experience with Implementiation of Virtual Computer Network Laboratory and Proposed Ways of its Further Development. Proceedings of Technologies for E-Learning conference, FEL ČVUT Praha, 2006, ISBN 80-01-03512-3, pp.58-68. [In Czech]

[3] Sedlář, P.: FPGA-Based Crossconnect for Serial Lines. Master's Thesis, Faculty of Electrical Engineering and Computer Science, VŠB-TU Ostrava, 2007. [In Czech]

[4] EZL-80 and EZL-80c: converter modules description http://www.hw-group.com/products/sollae/ezl80_en.html [online, April 2007]

[5] Charon 2 Ethernut embedded Ethernet module. http://www.hw-group.com/products/charon2/index_en.html [online, April 2007]

[6] Datasheet of analog switch array http://www.ortodoxism.ro/datasheets/zarlinksemiconductor/zarlink_MT8816_MAR_97.pdf' [online, April 2007]

[7] Grygárek, P., Milata M., Vavříček J., The Fully Distributed Architecture of Virtual Network Laboratory. 5th Int. Prepared for publishing at proceedings of Conference on Emerging e-learning Technologies and Applications, The High Tatras, Slovakia, September 6-8, 2007

[8] Gaura, J., Vavříček, J.: Tunnelling of VLANs and Auxliliary Layer 2 Protocols in Provider Network. Semestral Project of Routed and Switched Network subject,http://www.cs.vsb.cz/grygarek/SPS/projekty0506/QinQ.pdf [In Czech, online, May 2007]

THE AUTHORS

Petr Grygárek (Ph.D, MSc., CCNP) is Main Contact and CCNP/CCNA/NS instructor at RCNA VŠB-TU Ostrava. He also coordinates development group of Distributed Virtual Networking Laboratory used both in CNAP and for standard university networking courses.

David Seidl (MSc.) is a professor-assistant and Ph.D student at VŠB-TU Ostrava. He helps our RCNA with special hardware development.