Distributed Virtual Networking Laboratory and it's Exploitation at VSB-TUO

VSB-Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Computer Science
Regional Cisco Networking Academy

Petr Grygárek et al.

Agenda

- Goals of Virtual Networking Laboratory
- Basic Architecture
- Virtual Crossconnect
- Current Implementation Status and Exploitation
- ASSSK1 & ASSSK2 - Virtual Crossconnect Switching Elements for WAN ports (Petr Sedlář)
- Dynamic Search for Laboratory Devices Used for Particular Reservation (Jan Vavříček)
- Distributed Multiple-Site Architecture (Tomáš Hrabálek)

Virtual Laboratory Goals

- Give students more opportunity to exercise and make their own experiments with networking laboratory equipment using remote access via Internet
- Allow distant-learning students to do practice laboratory exercises
- Utilize costly laboratory devices more efficiently
- Share special and/or expensive devices between universities or possibly industrial companies
- Build a platform for cooperation with industry
  - common solution of real-world problems, staff training, …

Development Team

Virtlab system is developed and maintained by MSc. and Ph.D. students at Department of Computer Science under supervision of Petr Grygarek:

Pavel Nemec, Roman Kubin, David Seidl, Martin Milata,
Jan Vavricek, Tomas Hrabalek, Tomas Kucera,
Petr Sedlar

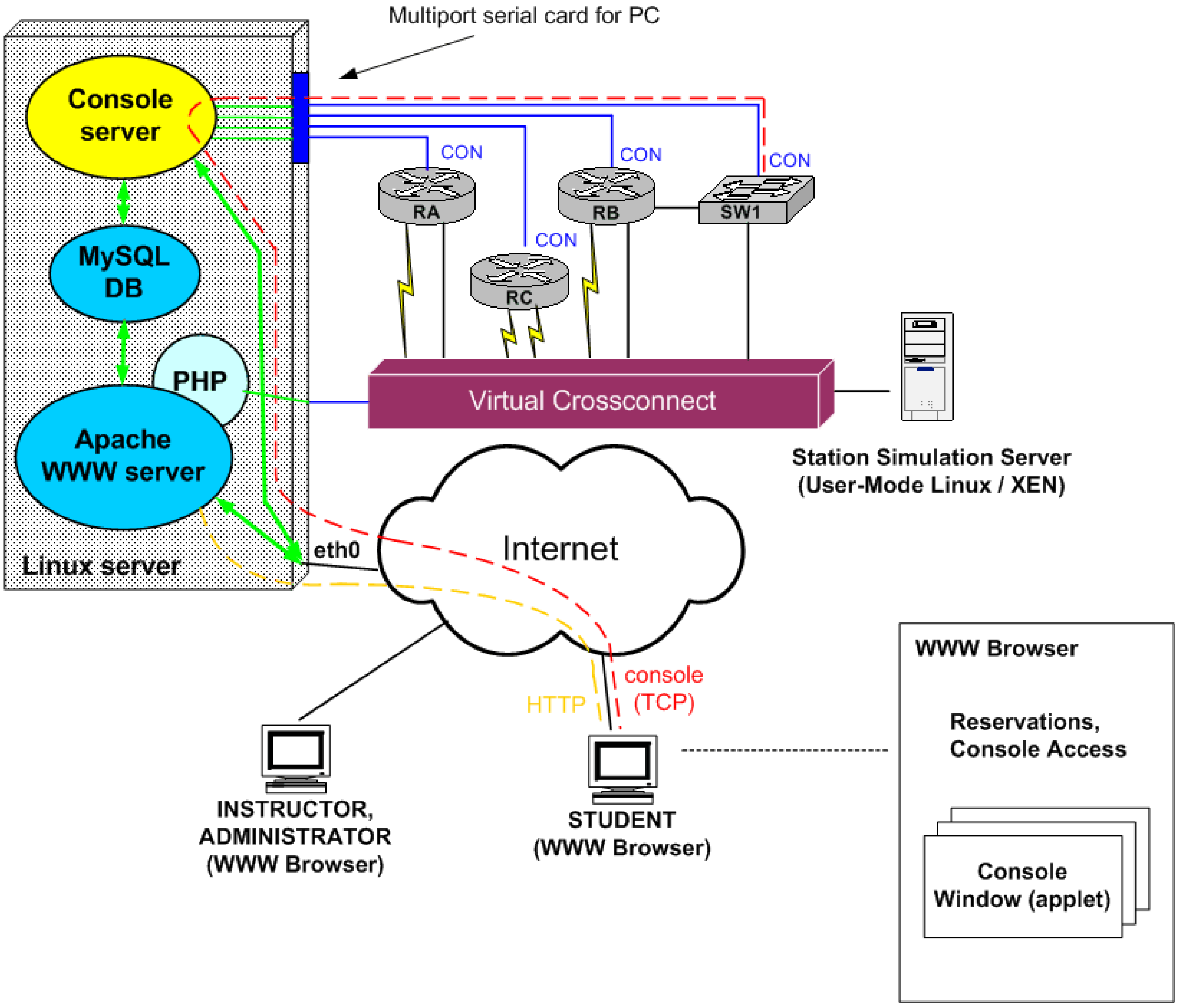
Team was established in 2004 and continually grows since then.

New interested members are always welcomed !

Basic Ideas

- Controlled remote access to consoles of lab devices
  - Includes access reservation system
- Extensible set of tutor-defined tasks
  - students commonly don’t know what exactly they could to experiment with without a such guidance
- Ability to handle various network topologies
  - fixed topology common for all tasks proved limiting
- Implementation using opensource platforms
  - On the client side, only standard Java-enabled WWW browser is required

Original Single-Site Architecture



Access Management System

Original Access Management System Philosophy

- Task-oriented device access
  - Task is defined by objectives/goal specification, devices the student may access and device interconnection topology
- Time is divided into timeslots with fixed duration. Various tasks are offered for reservation at individual timeslots
- Students may reserve timeslots to solve particular tasks

Tasks

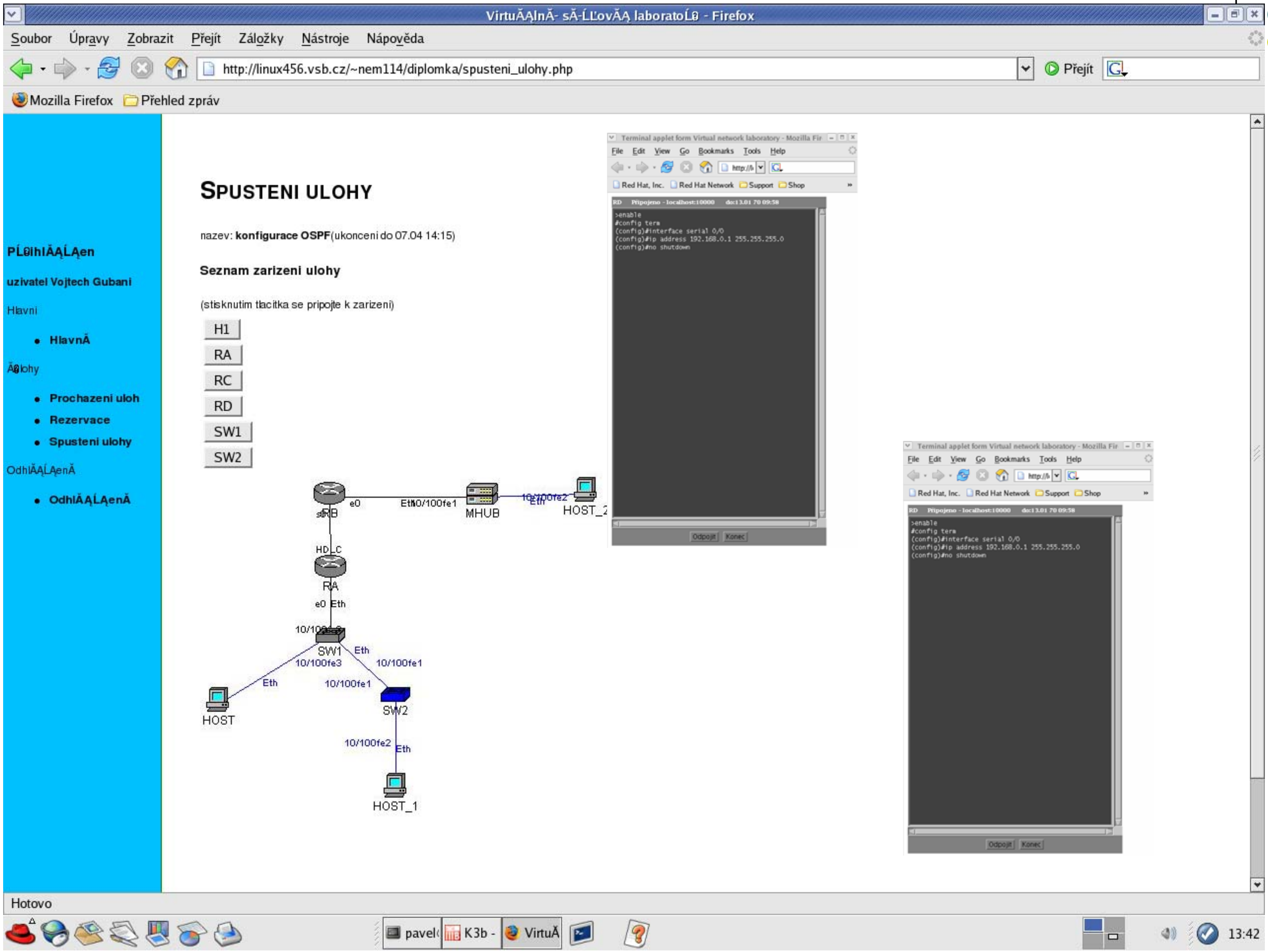
- Student may choose from extensible set of task created by tutors
  - Special case of task: student may define his/her own topology
- Task completely described using XML/HTML
  - objectives may include not only text and topology picture but also additional multimedia elements
  - tasks may be organized using user-definable category system
- Multiple students may cooperate on task solution
- Task solutions (correct configuration files) may be made available to students

Roles of Users in Access Management System

- Task Creator (teacher/instructor)
  - online (Web form) or offline (specification upload) task creation
- Task Scheduler
  - defines which task is accessible in particular timeslot (using electronic noticeboard)
- Student
  - reserves timeslots and accesses devices in previously reserved timeslot using Java applet running in his/her WWW browser
- System Administrator
  - administers user, defines devices, adjusts system parameters
- Tutor
  - can access consoles in parallel with students, multiple modes

An user may have multiple roles simultaneously.

Remote Device Access – Client Side



Virtual Crossconnect

The Automatic Topology Interconnection System

Device Topology Interconnection

- Before task is made accessible to student(s), required topology has to be interconnected
- We use our own “Virtual Crossconnect”
  - integrates various switching elements
    - comercially available or designed by us
- We can now connect connect any topology necessary for a task offered in particular timeslot automatically before timeslot start
  - interconnection of topologies by dedicated person proved to be unacceptable for real operation and we abandoned it
- Automation elimintates errors during topology interconnection made by human

Why do we Call the Crossconnect "Virtual" ?

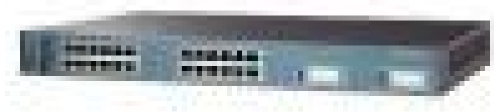
- Implemented using various and multiple physical switching elements
  - ASSSK, Cisco Catalyst 3500, Catalyst 1900, …
- Treated like single entity ("virtual crossconnect") by other parts of system
- All LAN/WAN ports of laboratory devices connected to ports of Virtual Crossconnect's switching elements
- Description of required topology is completely independent of types of actually used switching elements

Switching Technologies Currently Used in Virtual Crossconnect

- Serial Ports – our own microprocessor-controlled crossconnect implementation
  - ASSSK-1, ASSSK-2: will be discused later
- Ethernet Ports
  - VLAN-based interconnection between L3 devices
  - VLAN-tunneling (802.1 QinQ) between switches and other L2 protocols-sensitive devices
    - allows interconnection of trunk links
    - transparent to Spanning Tree and other L2 control protocols

Currently Used Virtual Crossconnect Switching Elements

ASSSK-1 & 2 (developed by us)

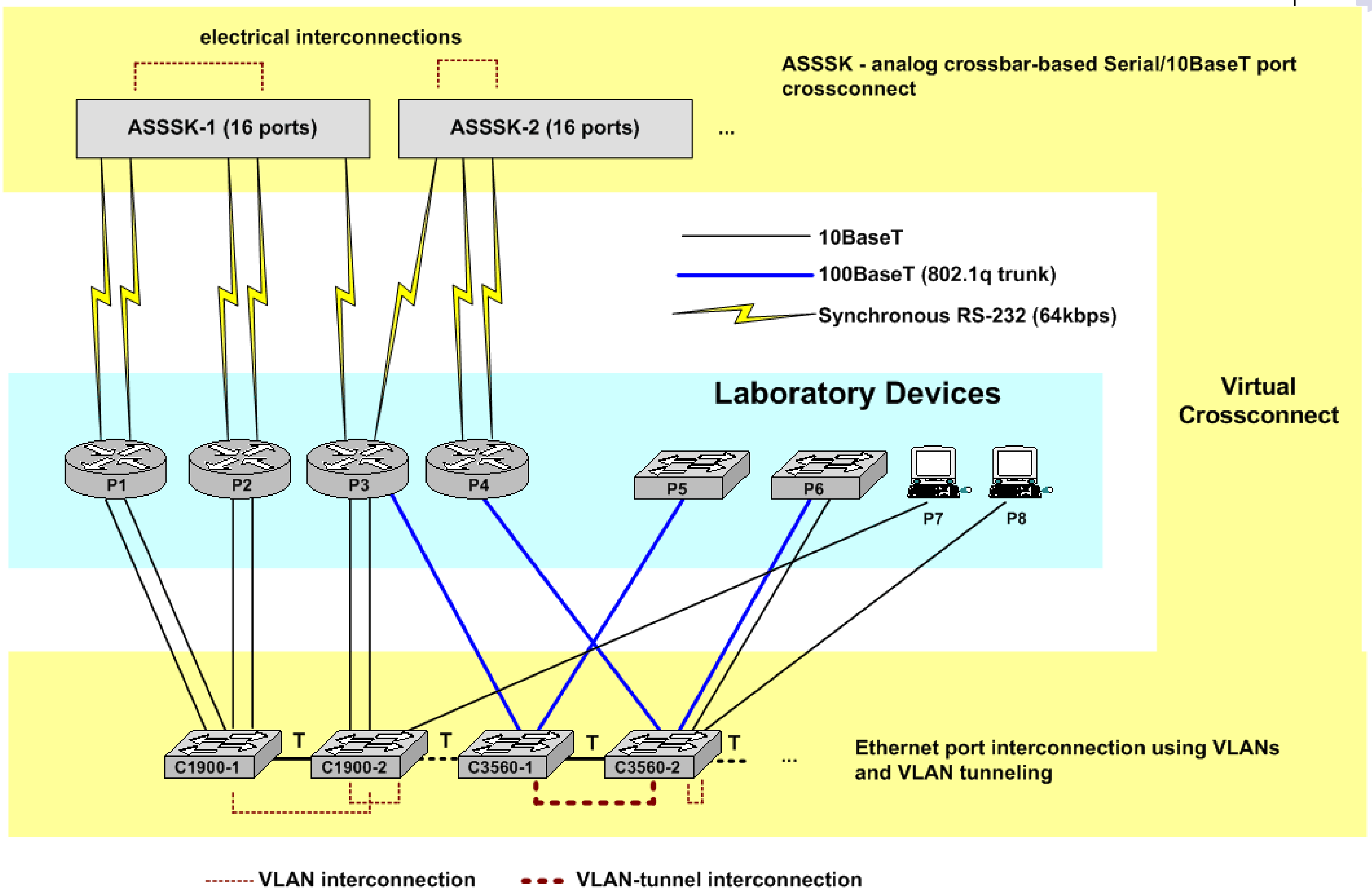


Cisco Catalyst 3550
(supports VLAN tunneling and L2 protocol transparency)

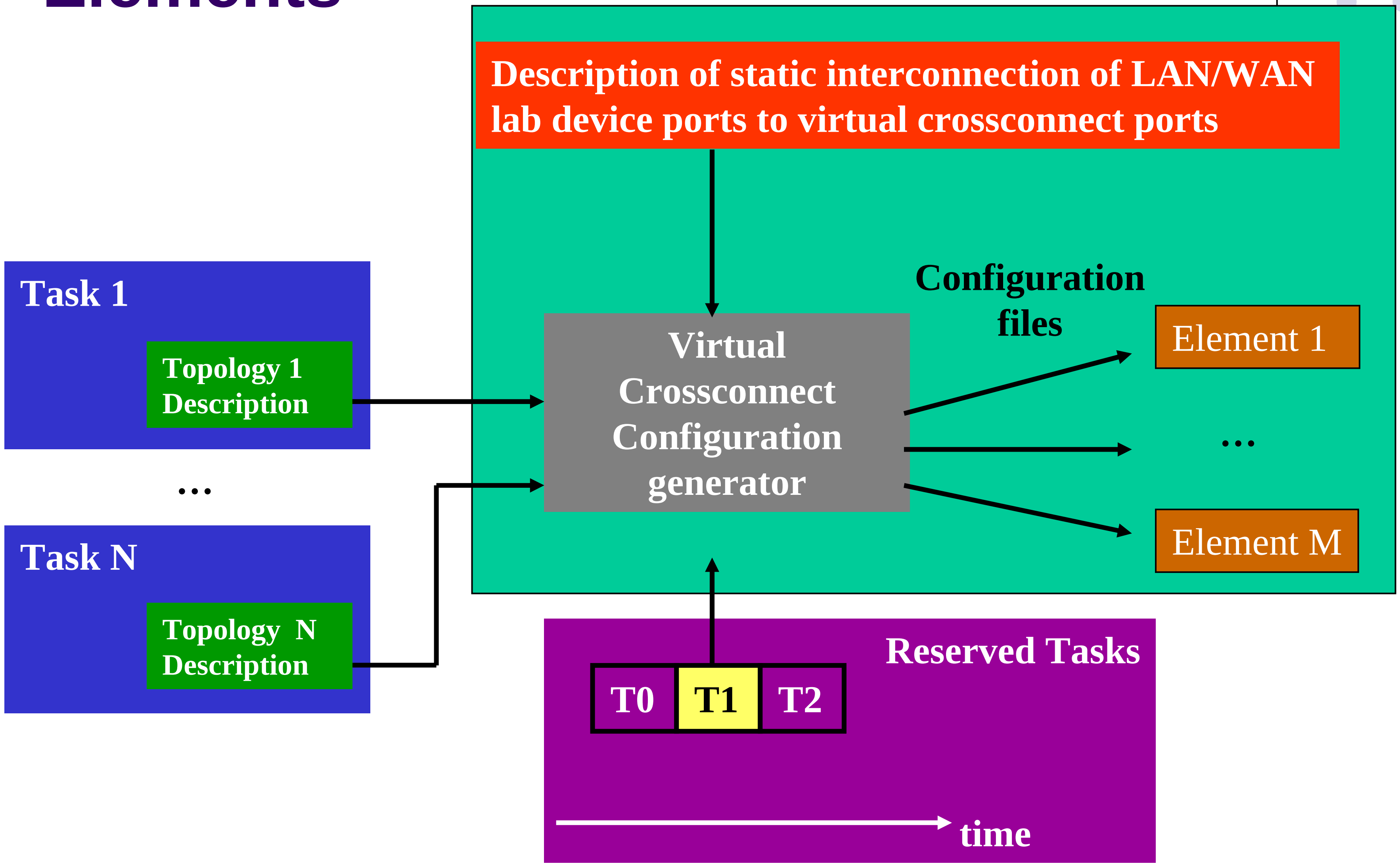
… possibly other switching elements in the future

Cisco Catalyst 1900
(cheap solution)

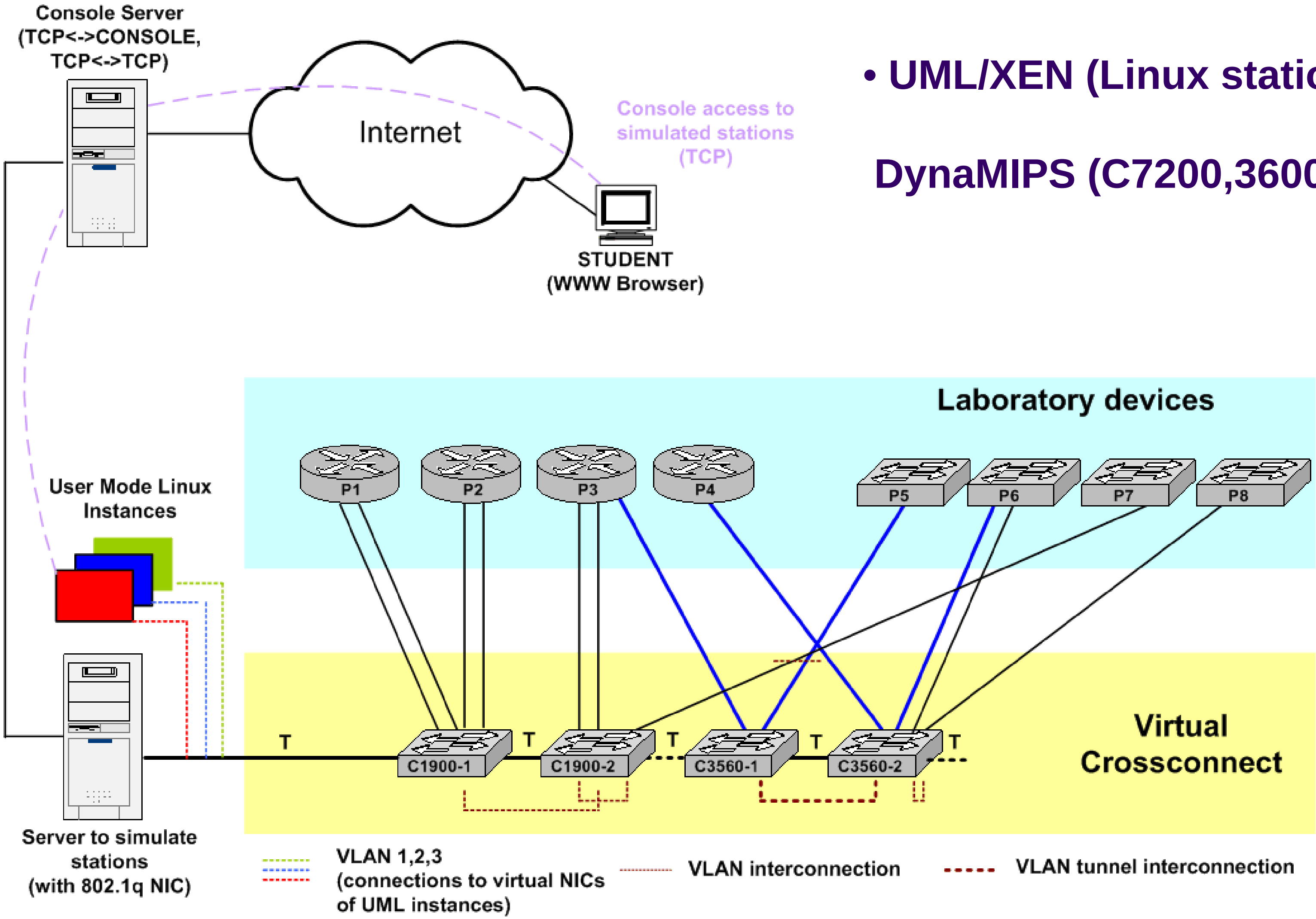
Interconnection of Laboratory Devices with Virtual Crossconnect Switching Elements



Generation of Configurations for Virtual Crossconnect Switching Elements



Integration of Simulated Devices with Virtual Crossconnect

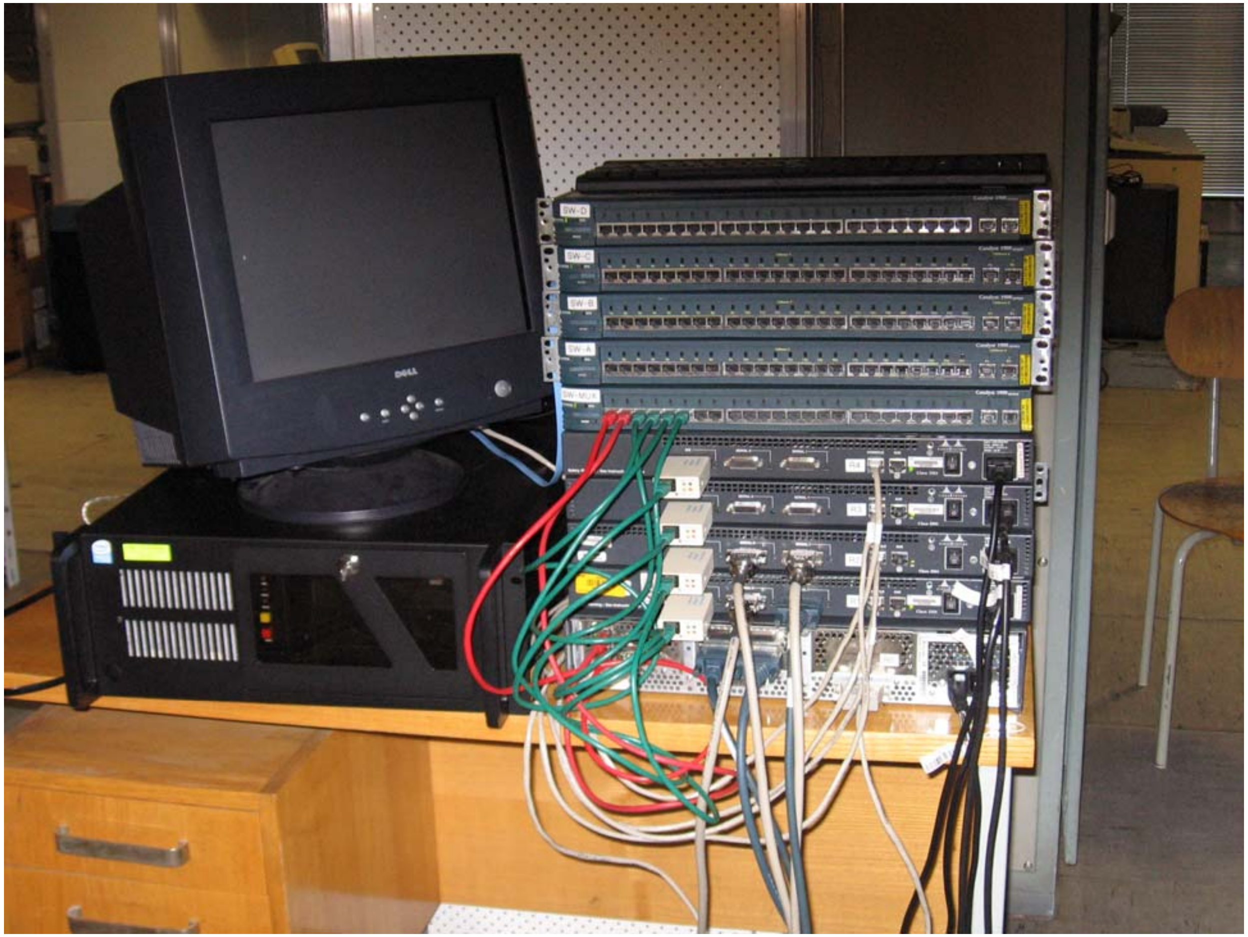


- UML/XEN (Linux stations)

  DynaMIPS (C7200,3600,2600)

Today’s Functional Single-Site Implementation

How We Started:
The Very First Virtlab Implementation

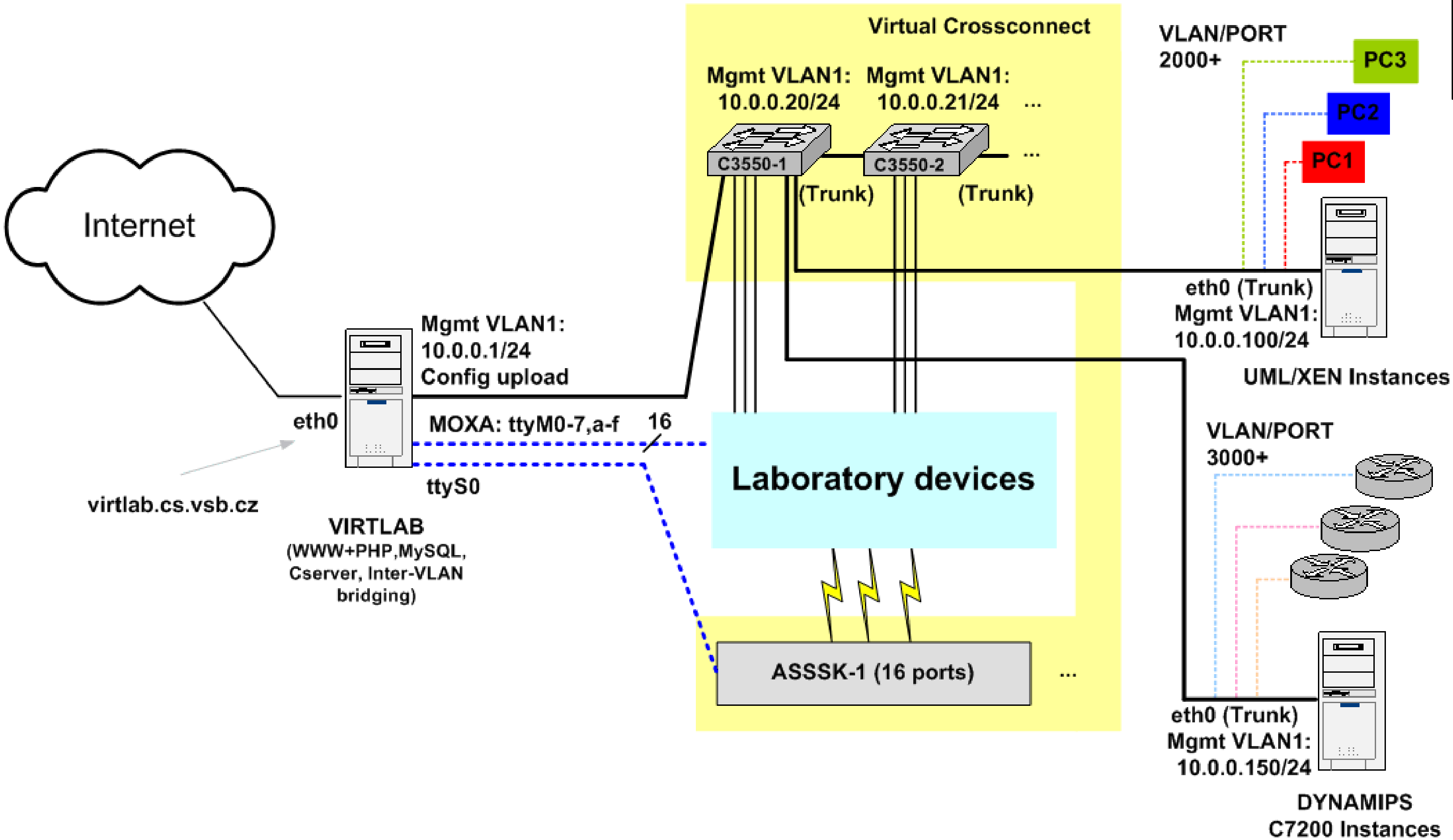


Today’s Virtlab

Two 45U racks incorporating

- Access Management System Server
- Virtual Crossconnect
  - (1x C3550, ASSSK1)
- UML server
- DynaMIPS server
- Cisco 2500/2600/4000 routers
- Cisco Catalyst 1900 switches
- Cisco Catalyst switches
- Cisco 5500 switch with RSM

Today’s Single-Site Virtlab Internal Architecture



Accessible at *http://virtlab.cs.vsb.cz*
for the second year

ASSSK1 & ASSSK2

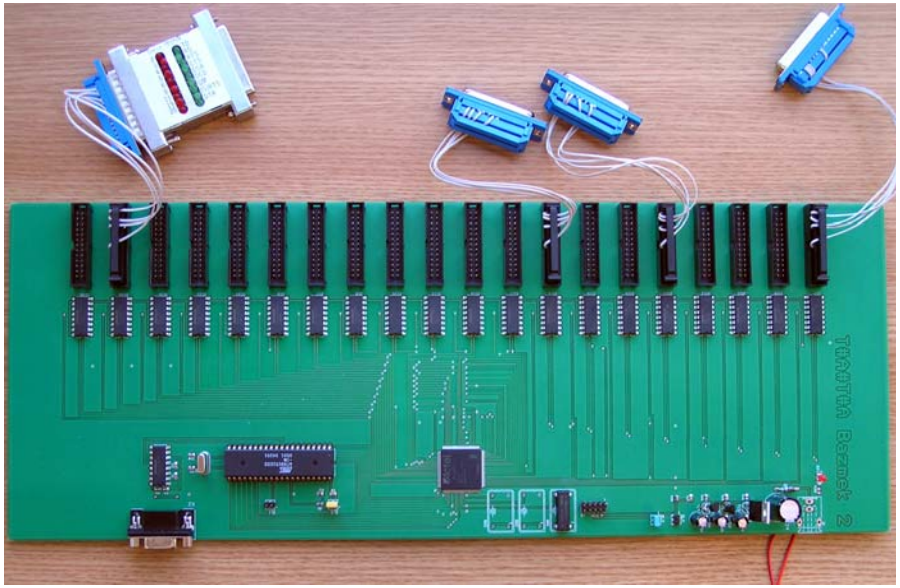
Our Own Virtual Crossconnect Switching Elements for WAN Ports

Petr Sedlář (MSc. Thesis)

ASSSK-1 Switching Element

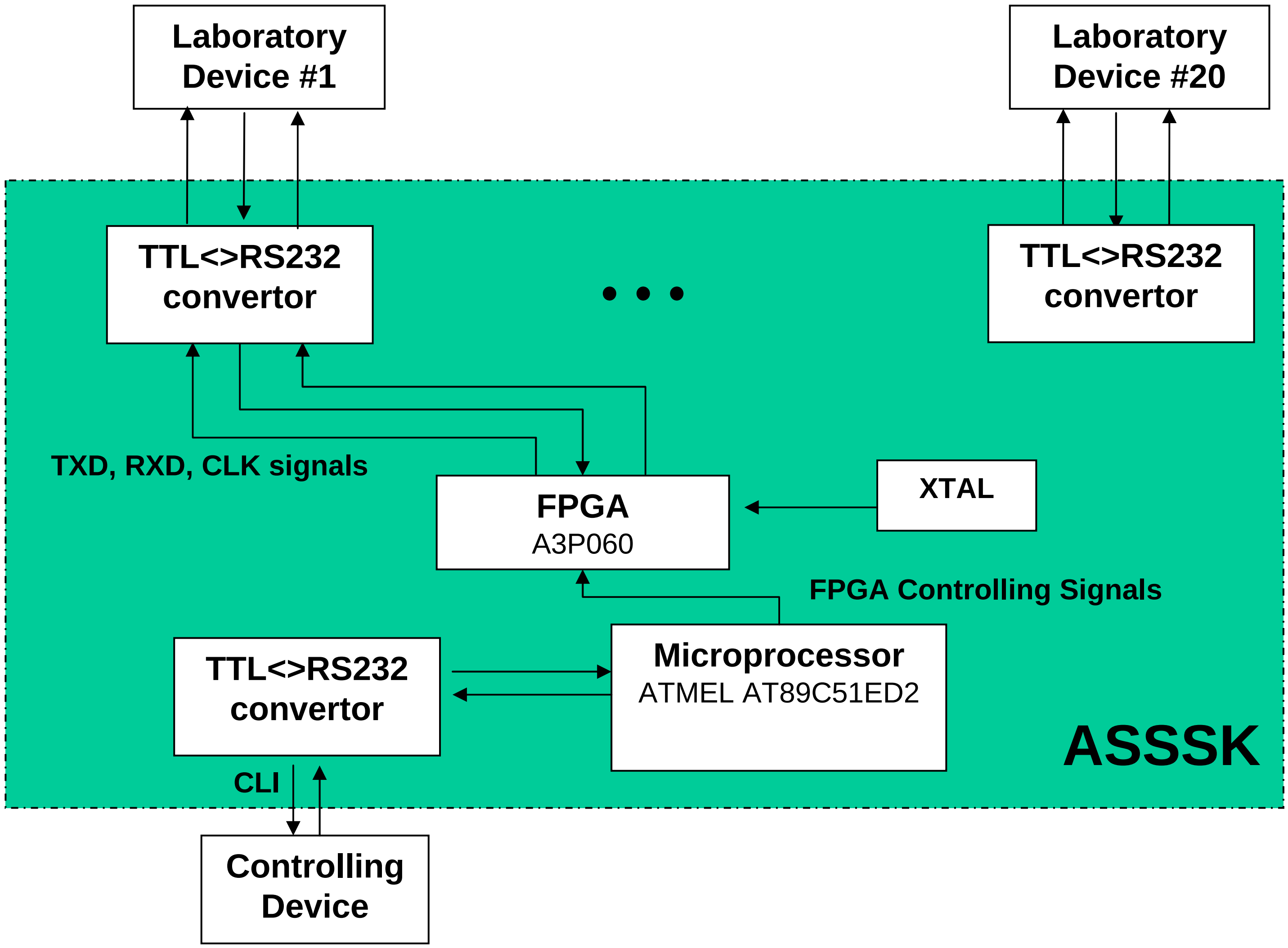
- Developed by David Seidl (MSc. thesis)
- Uses analog switch array
- Modular design
  - 16 Serial/Ethernet ports
  - Serial ports behave like DCE
- Controlled using IOS-style CLI via RS-232 port

ASSSK-2 Switching Element



- Redesign of ASSSK1 using FPGA
  - smaller, cheaper, higher switchable bitrates
  - VHDL-based design allows easy changes
- Only for serial ports inteconnection
  - for Ethernet ports, VLAN-tunneling using standard switches proved more efficient
  - Various clockrates can be set on individual serial lines

ASSSK2 Block Diagram



Future plans with Virtual Crossconnect

- ASSSK-3 (under investigation)
  - serial-port crossconnect implementation based on multiport serial card for PC, Linux HDLC/PPP drivers and bridging software
  - will allow WAN traffic tunelling across Internet
    - (HDLC/PPP frames encapsulated in UDP datagrams)
- Enhancement of element configuration generator scripts
  - implement semantic checks of topology definition provided by student who requests his/her own topology
- WAN links flapping simulation to let students get experience with real-word WAN troubleshooting

Dynamic Search for Laboratory Devices used for Particular Reservation

Jan Vavříček, MSc. diploma thesis

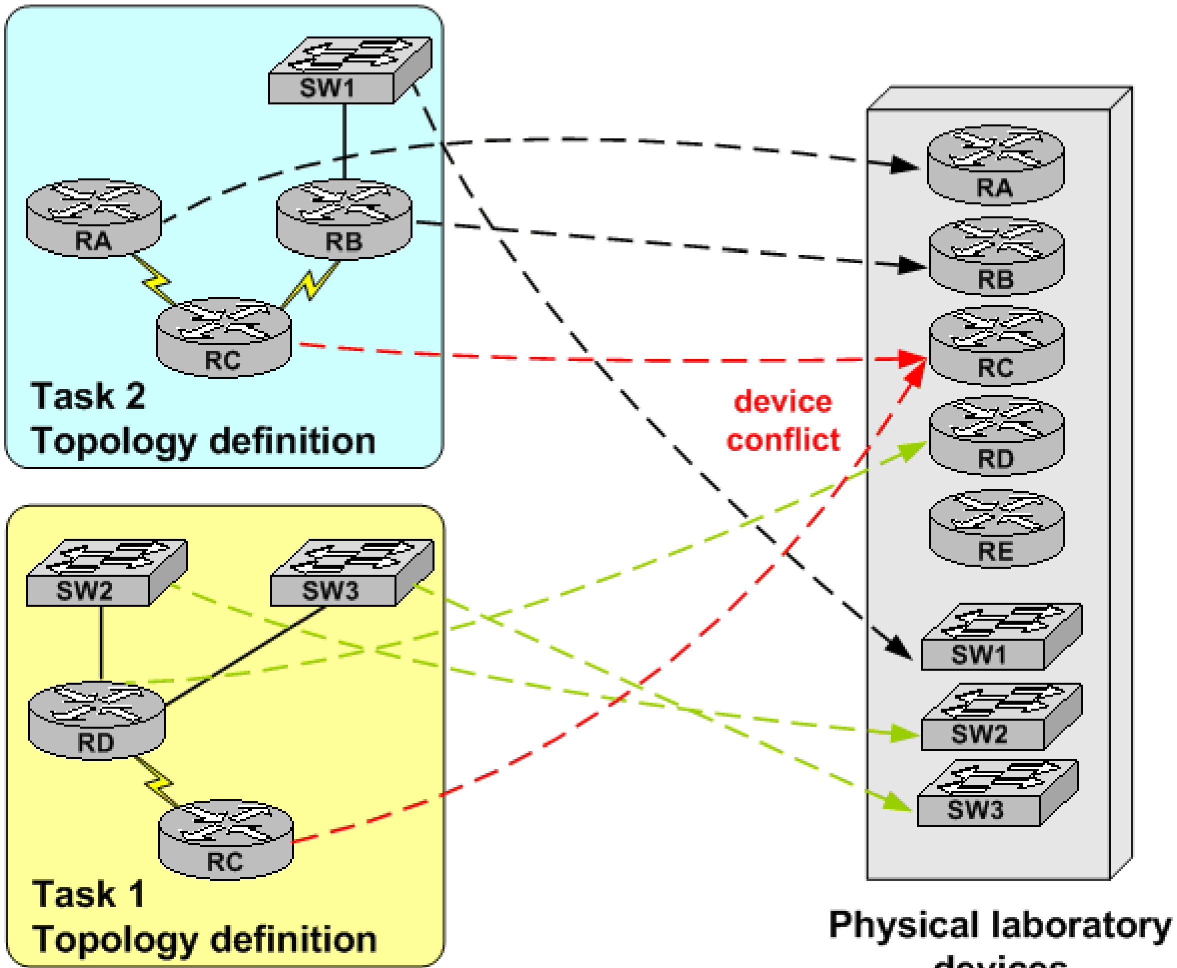
The General Idea (1)

- Fixed-length and fixed-beginnings timeslots proved inefficient
  - Students are not interested in some scheduled tasks but there is a contention for timeslots with other more interesting tasks
- Since we realized the full power of automatic topology interconnection, we decided to let students to choose ANY task at ANY timeslot (with arbitrary duration)
  - fixed timeslots abandoned, student may reserve any time interval up to his/her quota
- Task scheduler role no longer needed
  - notice-board used only to remember what task was reserved for what time interval

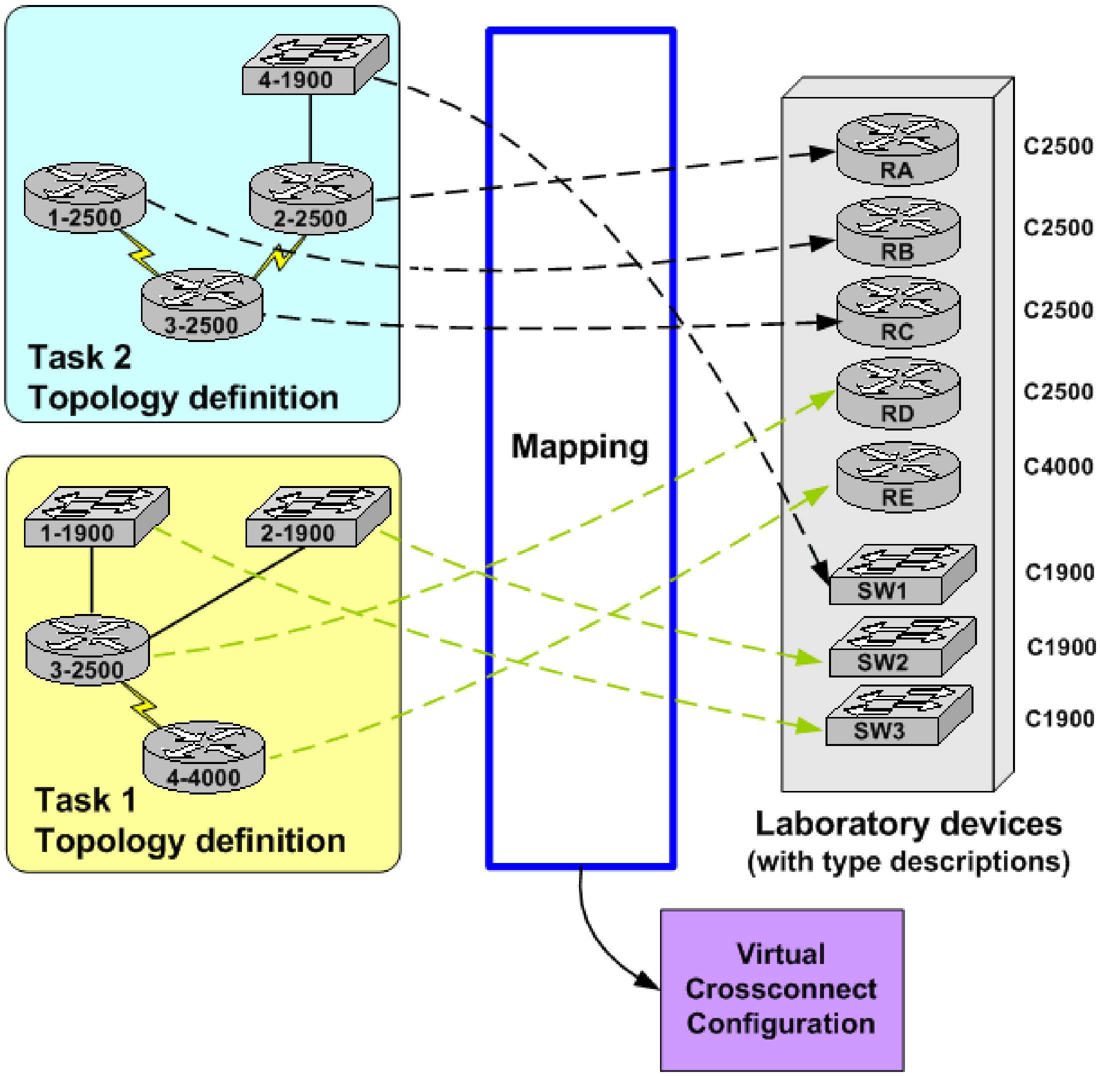
The General Idea (2)

- To utilize virtual laboratory equipment efficiently, it is useful to let multiple tasks be reserved in parallel if there is enough network devices
- To let multiple tasks be reserved in parallel without unnecessary limitations and device conflicts, it is needed to decouple task definitions from physical device identities
- Physical devices used for task reservation are searched dynamically with respect to usage of other devices at the same time

Problem of Task Definition Coupled with Physical Device Identities (Original Architecture)



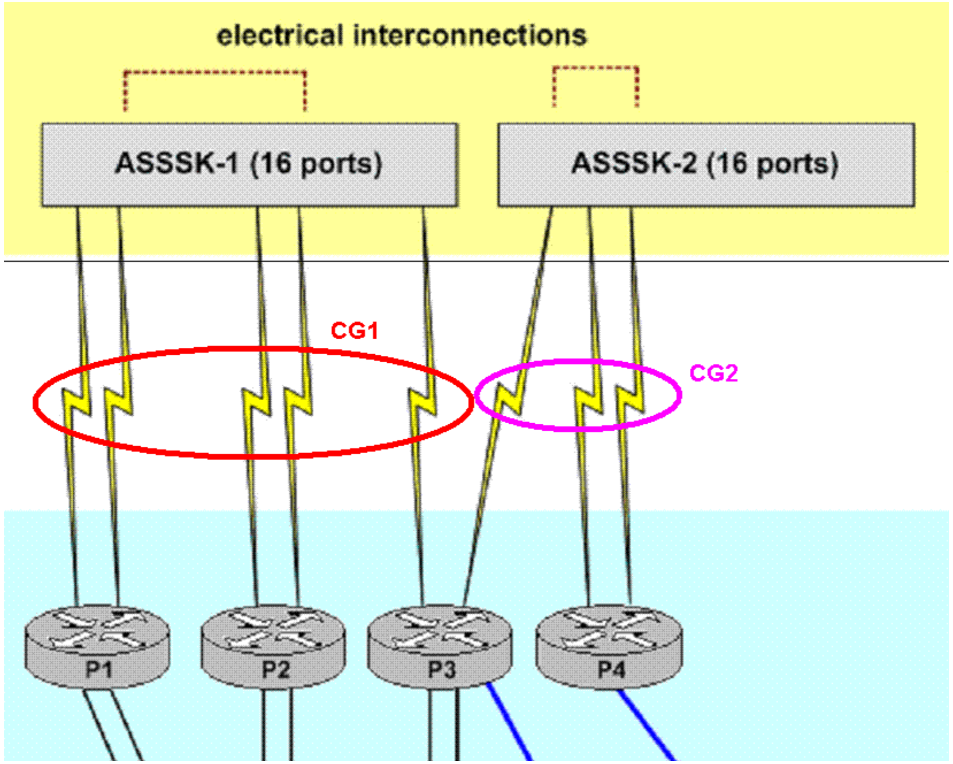
Dynamic Mapping of Task to Physical Devices During Task Reservation Procedure



Function of Dynamic Device Mapping

- Mapping takes place when student reserves task for particular time interval for himself/herself
- Network device types, numbers of interfaces and other features have to be taken into account
- Multiple tasks may be mapped in parallel
  - feasibility determined during mapping process performed at reservation time

Connectivity Groups of WAN Ports



- Currently, only WAN ports connected to the same ASSSK device can be interconnected
  - notion of Connectivity group was established to define what interconnections are possible
- In the future, we plan to tunnel HDLC/PPP frames between ASSSKs using Ethernet
  - Linux-based ASSSK-3
- There is no similar problem with Ethernet ports, since distributed virtual crossconnect allows to interconnect whichever two Ethernet ports

Mapping Algorithm: The Basic Steps

- Candidate selection
  - decides which physical device can potentially act as which logical device (based on interface numbers/types and other features)
- Calculation of “cost’ of each candidate
  - based on features and number/type of interfaces of the physical device
  - physical devices with lower cost are preferred
- Matrix mapping
  - recursive process with backtracking capability which searches the solution space for the ‘cheapest” one
  - connectivity groups have to be taken into account

New Implementation of Access Management System

- Complete redesign
  - implementation of general page generation framework
  - better extensibility, easier access rights enforcement, multi-language support
- Support for users/sites in different timezones
- Decoupling of logical and physical laboratory devices' identity
- Redesign of reservation system internal logic and GUI with regard to abolishment of fixed-timeslots philosophy
- Distributed nature of the system taken into account
  - will be explained next

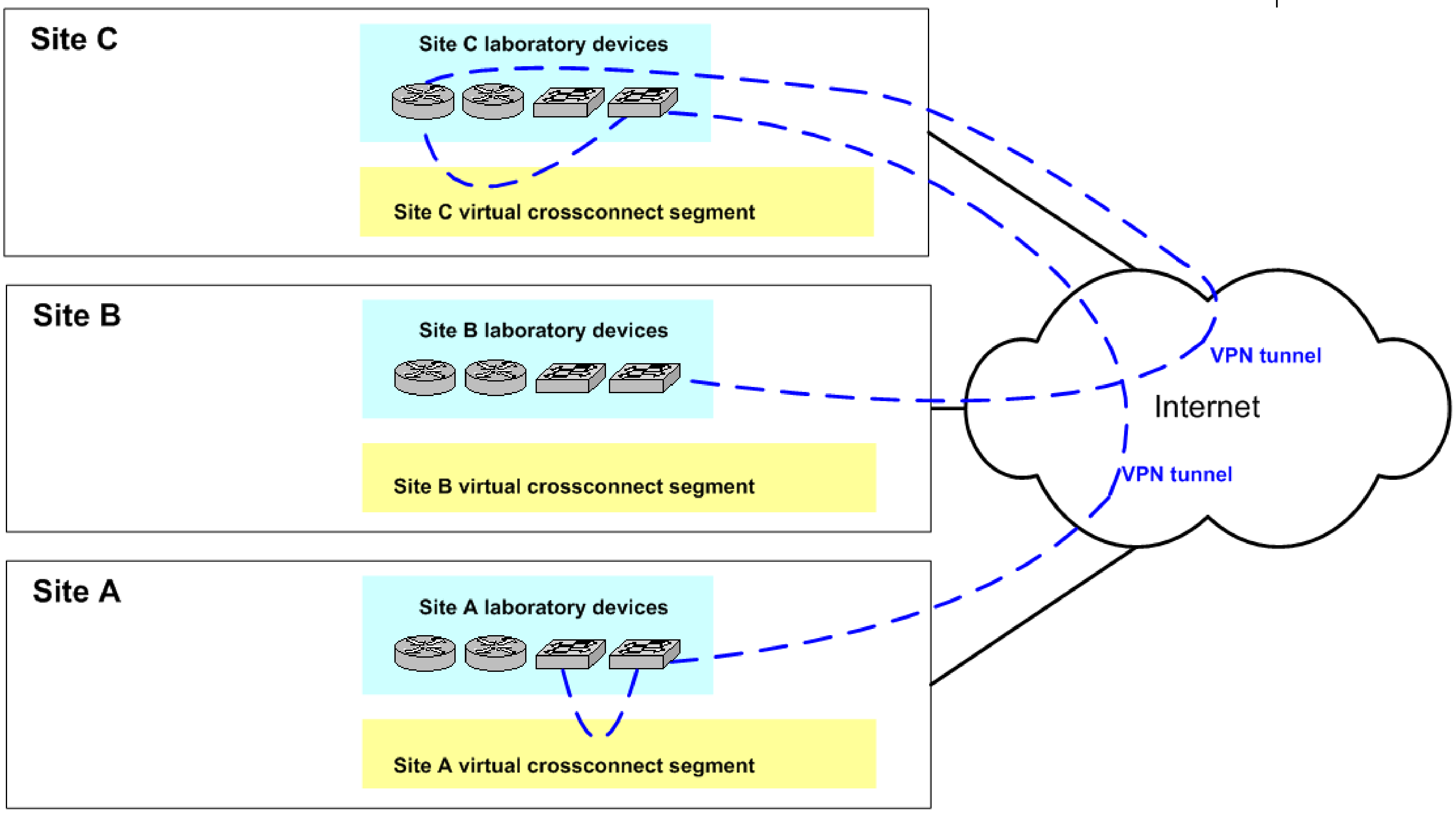
Distributed Multiple-Site Architecture

Tomáš Hrabálek, MSc. diploma thesis

Multiple-Site Architecture Basic Paradigm

- Multiple cooperating sites may share laboratory devices transparently
  - Constraints to offer site’s lab devices for other sites may be specified
  - Local mappings are preferred to limit inter-site traffic
- Creation of virtual topologies using tunnels over Internet
- Fully-decentralized architecture allows independent operation of individual sites if other sites become unavailable
- Distributed nature is hidden to student
  - he/she accesses device consoles the same way regardless of physical target device placement
  - virtual topology between devices of multiple sites behaves the same way as single-site topology

Distributed virtual topologies



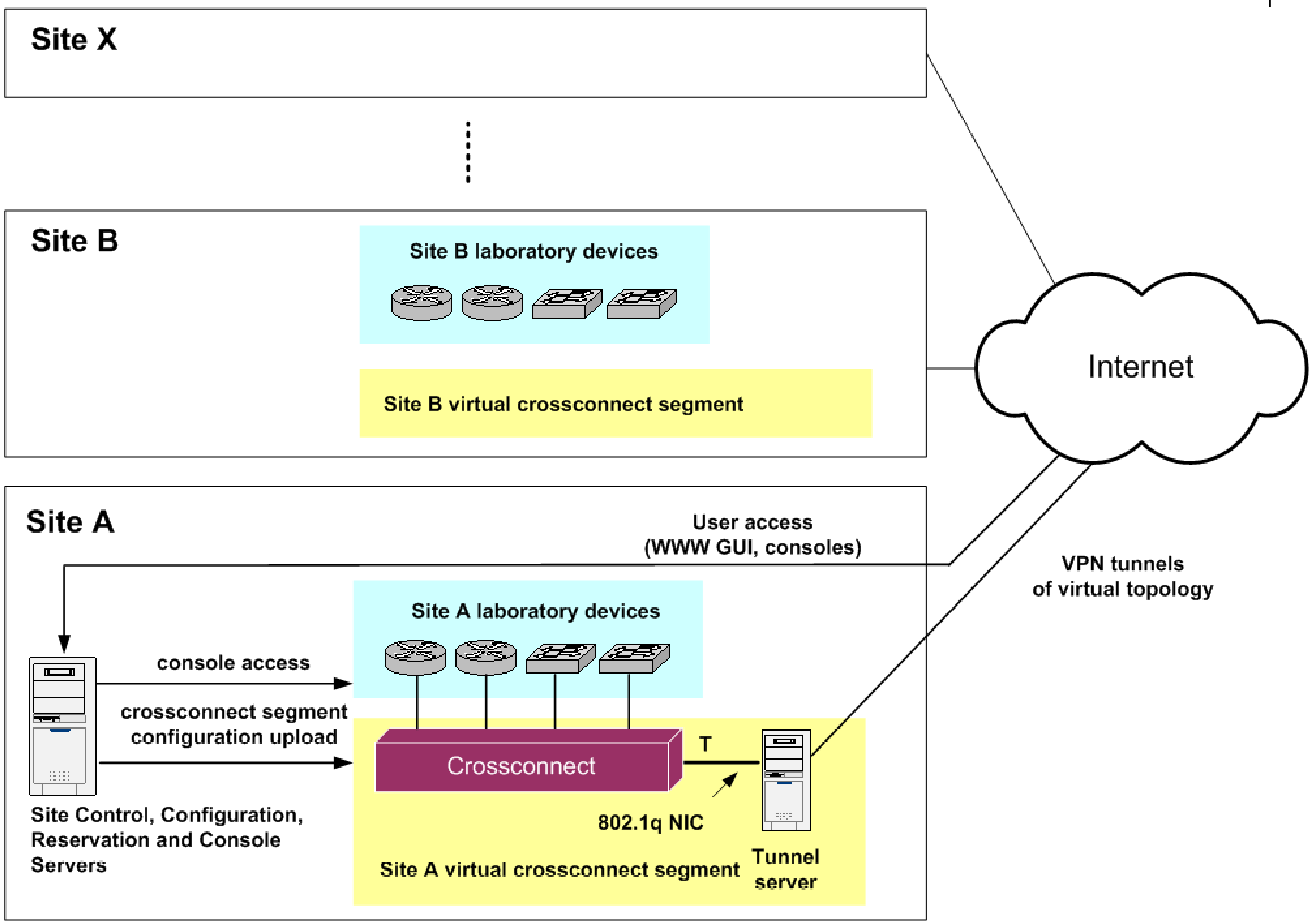
Technologies Used to Create Distributed Topologies

- Distributed virtual topologies may be constructed using Internet tunnels
  - Tunneling of Ethernet links traffic is implemented now
  - We are also working on HW/SW solution to tunnel traffic of serial (WAN) links
- Layer 2 frames tunneling has been chosen since it allows transparent operation of whatever layer 3 protocol

Architecture Extensions for Implementation of Distributed Topologies

- Virtual Crossconnect concept was generalized to support multiple-site topologies
- Our implementation of Tunnel Server allows to extend VLAN-based crossconnections over Internet tunnels between sites (802.1q in UDP)
- Virtual Crossconnect is now treated as set of Virtual Crossconnect Segments at individual sites which form a single Distributed Virtual Crossconnect together

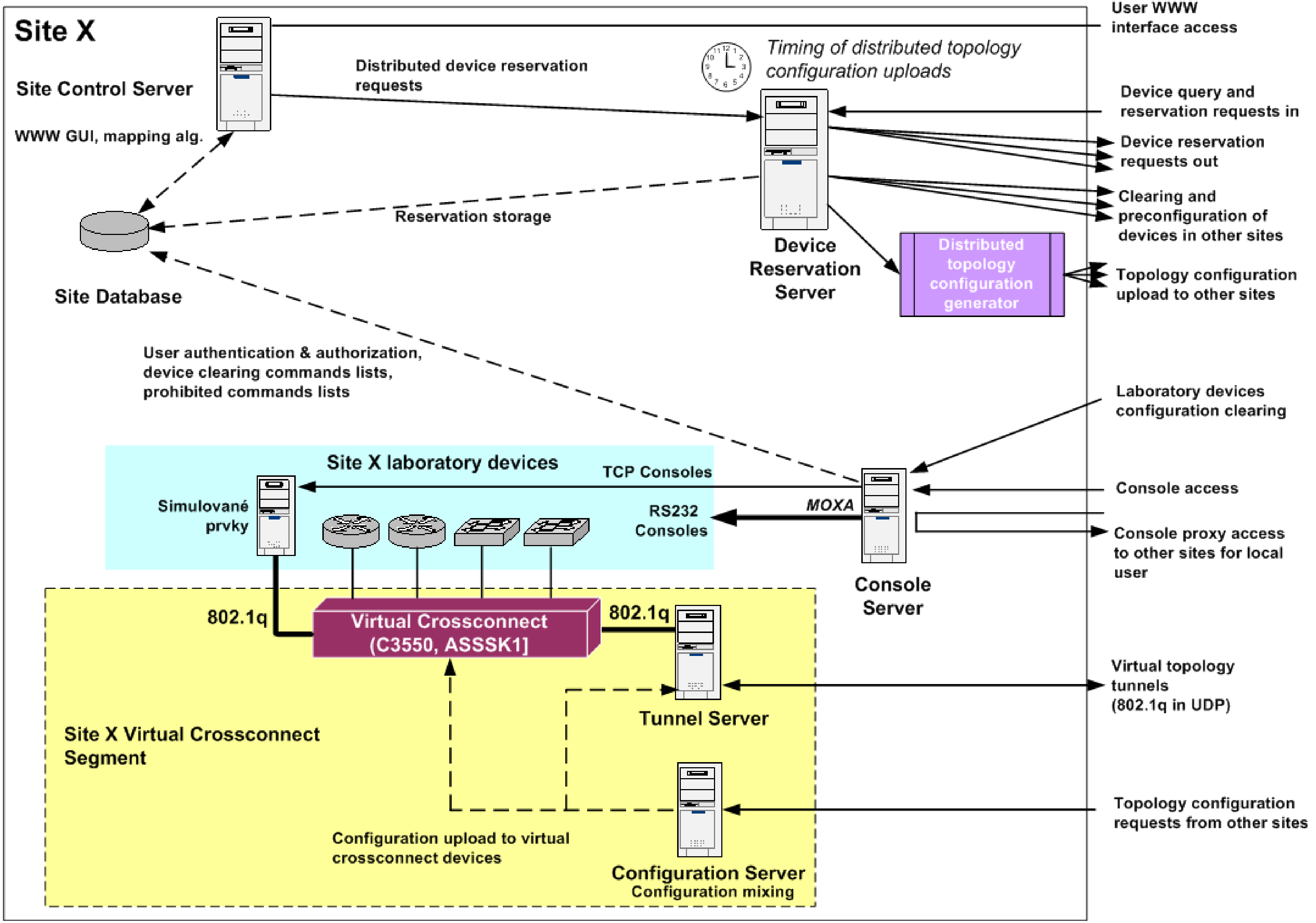
Multiple-Site Architecture Basic Components



Function of Distributed Architecture Components

- Virtlab Server
  - Provides access management system GUI, console access GUI and device mapping algorithm
- Reservation Server
  - Keeps track of site devices' reservations
- Configuration Server
  - accepts configuration requests for site's distributed virtual crossconnect segment
- Console Server
  - allows access to consoles of sites' devices
- Tunnel Server
  - Tunnels traffic between laboratory devices at different sites

Interaction of System Components



Advantages of Multiple-Site Architecture

Multiple cooperating institutions are able to

- implement large WAN topologies using devices available at all participant sites
  - behavior of real-word topologies may be studied
- specialize to buy expensive special devices and share them with others

Current Status and Planned Development Information

- See http://www.cs.vsb.cz/vl-wiki
- Any improvement and/or cooperation suggestions are welcomed at

  virtlab@cs.vsb.cz

Thank you for attention

Questions ?