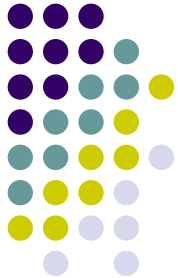
VIRTLAB Project

Virtual Networking Laboratory

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

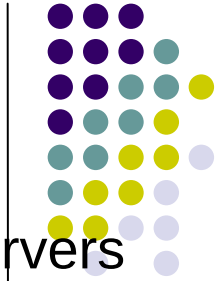
Project Leader: Petr Grygárek

Pedagogical Goals

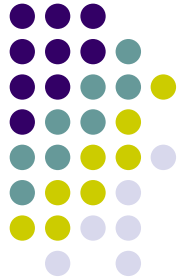


- Give students more opportunity to exercise and make their own experiments with networking laboratory equipment
  - laboratory is occupied in working hours
- Allow distant-learning students to do practice laboratory exercises
- Utilize costly laboratory devices more efficiently
  - allow remote access during non-working hours
- Share special and/or expensive devices between universities

Why Not to Use Commercial Solutions for Infrastructure Technology ?

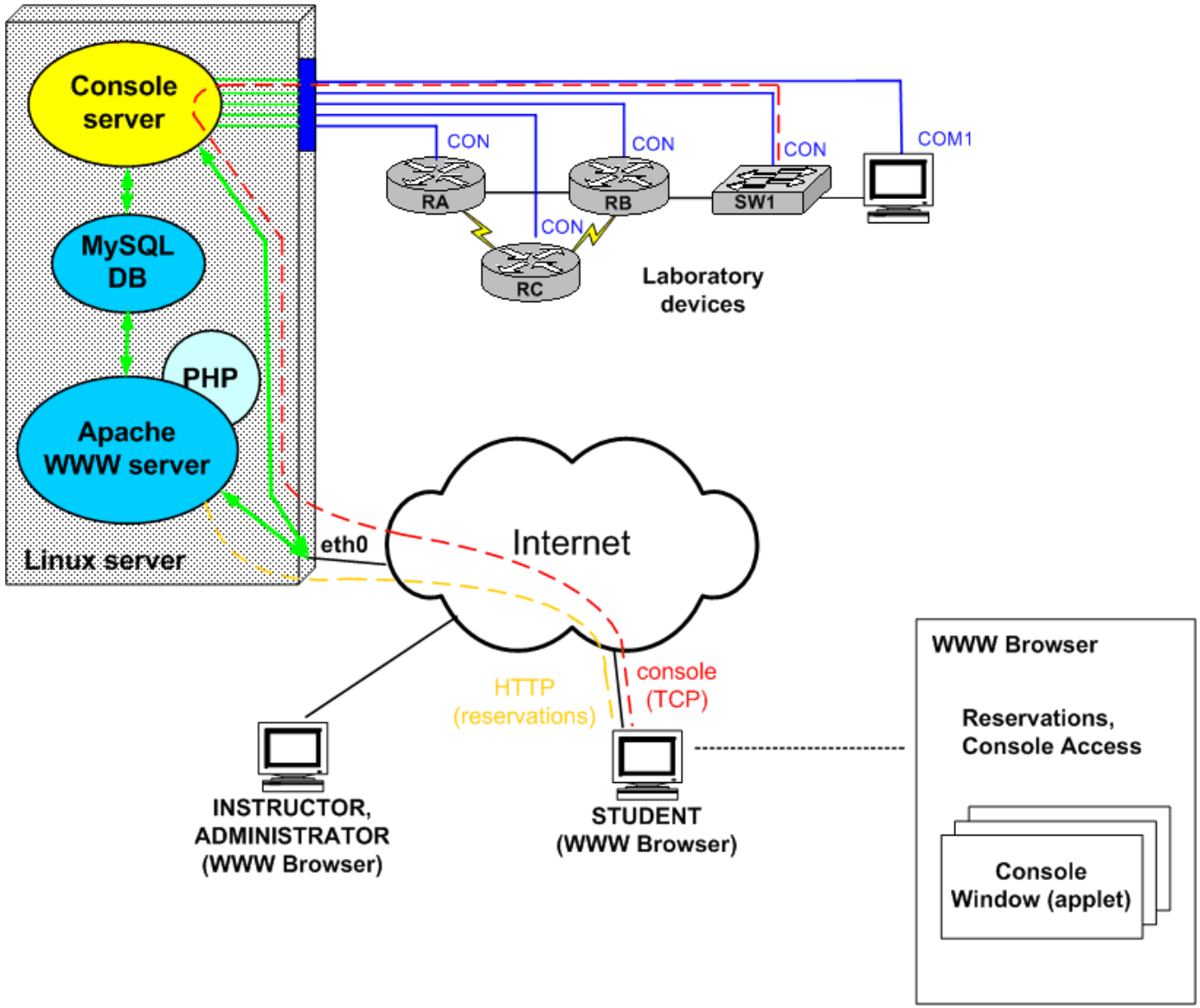


- Remote access solutions using hardware-based terminal servers were implemented by many universities
  - expensive
  - provides little flexibility - low potential for advanced feature implementation (e.g. tutor support)
  - often lacks access management system at all
- Software like Netlab requires costly periodic license renewal and can not by extended to implement features user-specific required
- We wanted to develop task-based, multiple-site variable-topology solution
- We aimed to avoid Adtran-like devices
  - expensive, non-extensible, closed architecture, sometimes behaves poorly
- We wish to keep academic nature of the project
  - gain knowledge from developing and researching our own architecture

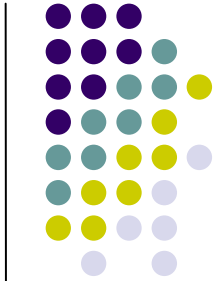


The Original Single-Site Architecture

The Original Idea and Architecture

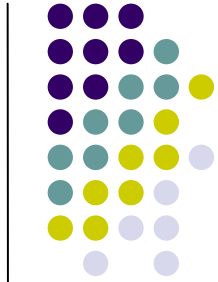


Basic Goals

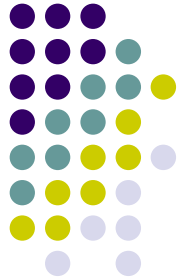


- Make consoles of lab devices accessible remotely
- Create access management system to securely share devices among students
- Provide a set of tutor-defined tasks
  - students commonly don’t know what exactly they could to experiment with
- Be able to handle various network topologies
  - fixed topology common for all tasks proved limiting

Implementation Platform

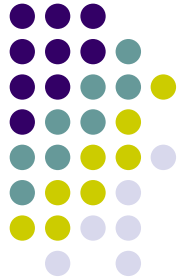


- Open-source technologies
  - Apache+PHP, ANSI C/C++, MySQL, XML, Java applets
- Server side runs on Linux
  - Transfer to other platforms possible, but not expected
- On the client side, only standard WWW browser is required
  - Java applet support is needed (Java 1.4)



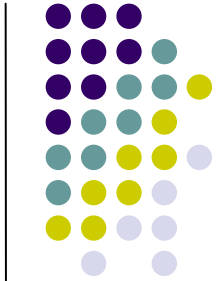
Access Management System

Original Access Management System Philosophy



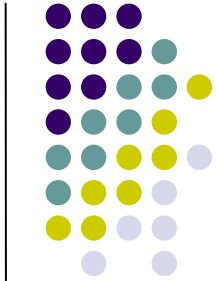
- Task-oriented device access
- Task defined by objectives, devices the student may access and network device interconnection topology
- Particular tasks are offered to students in timeslots chosen by administrator using electronic notice-board
  - Division of time into timeslots advantageous if task topologies are different and have to be connected manually
- Students may reserve timeslots to solve particular tasks
  - up to student’s weekly quota

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
- Multiple students may cooperate on task solution
- Task solutions (correct configuration files) may be made available to students

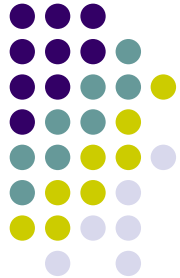
Roles of Access Management System Users



- Task Creator (commonly instructor)
- Task Scheduler
- Student
- System Administrator
  - administers user, devices, adjusts system parameters
- Tutor

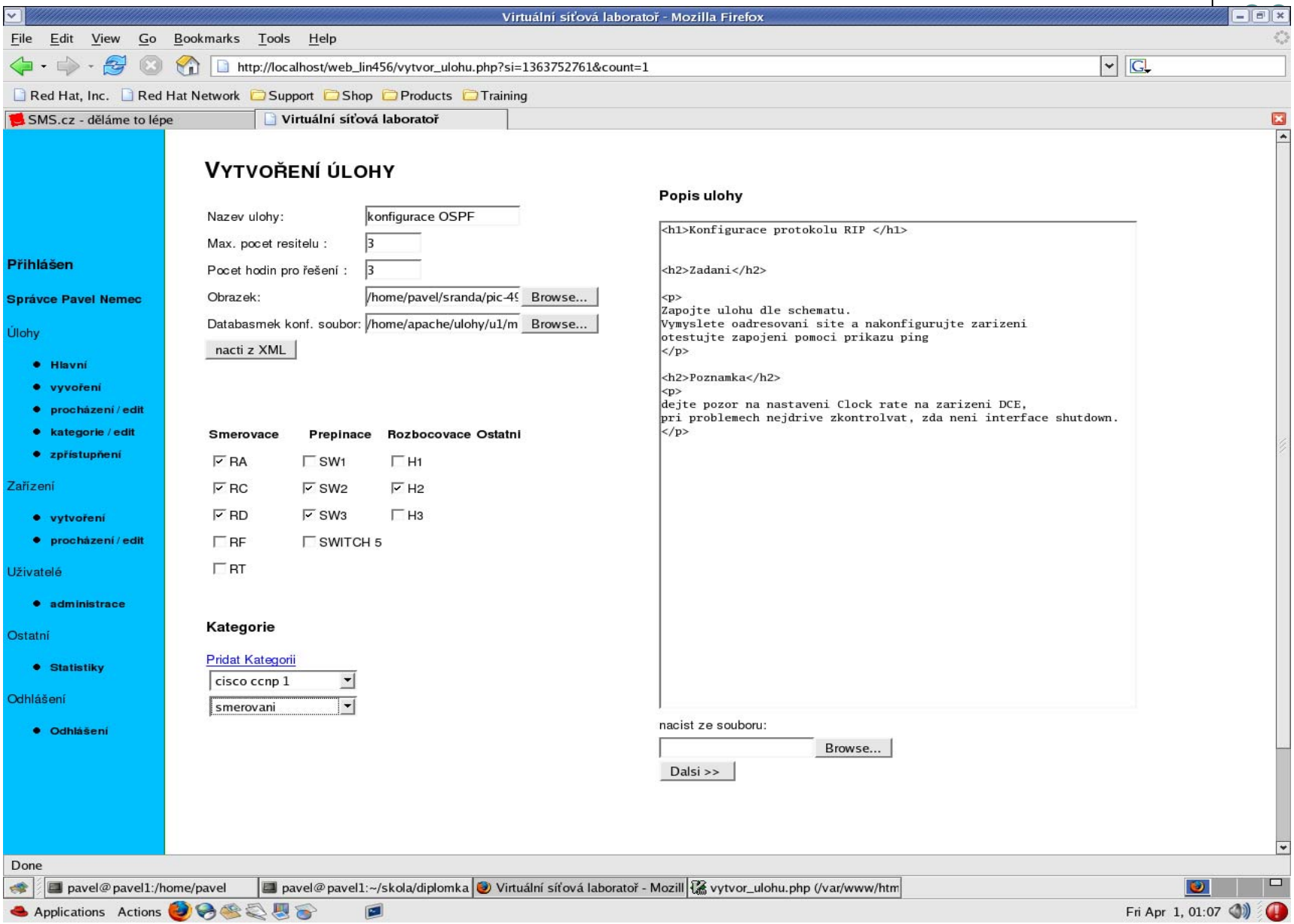
An user may have multiple roles simultaneously.

Task Creator Role

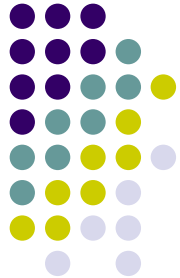


- Creates, edits and deletes tasks
  1. using Web form
  2. offline creation and insertion of archive with prescribed contents
     - HTML pages + XML-based description
  - tasks may be organized using user-definable category system

Task Creation - Screenshot

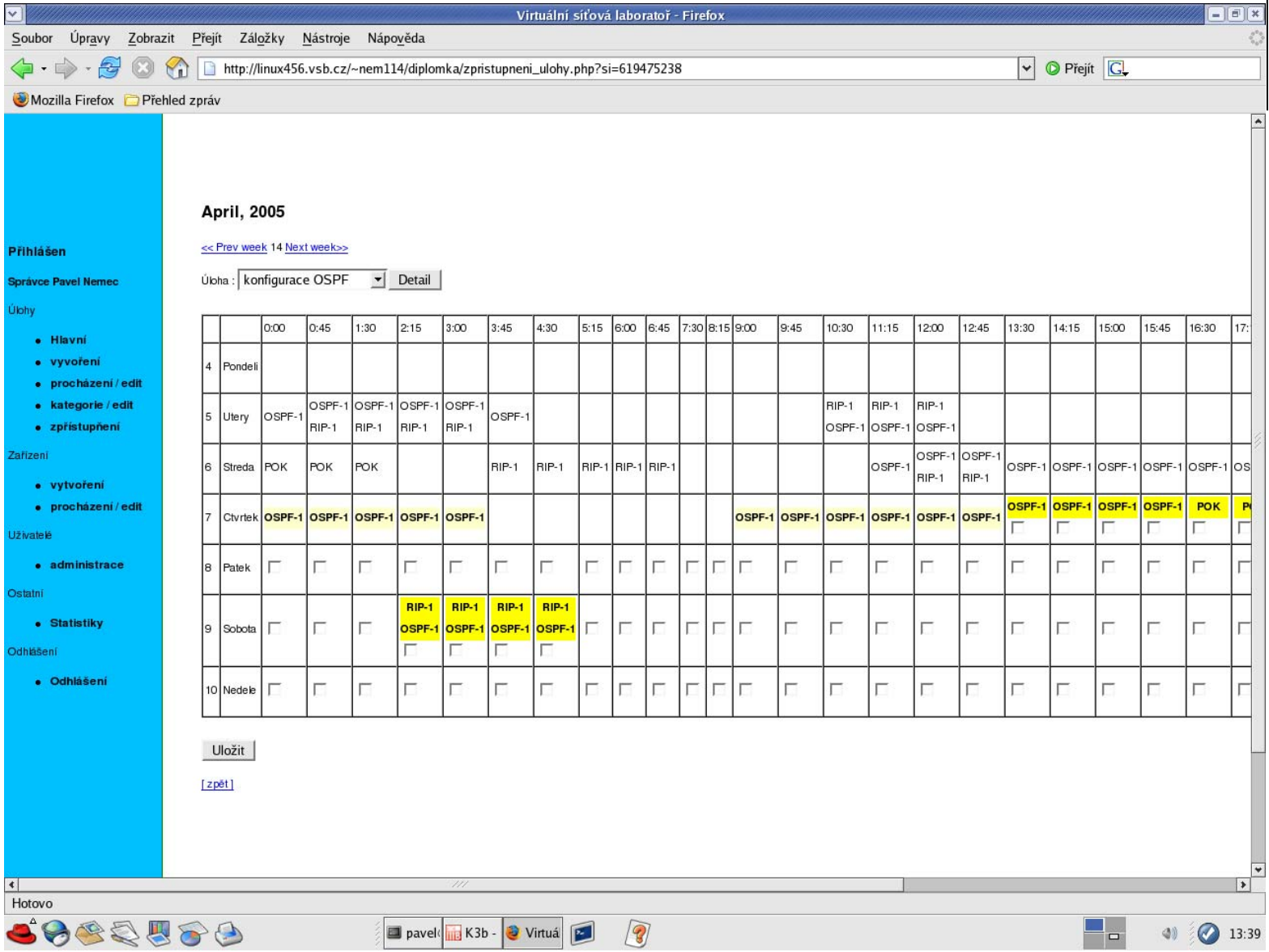


Task Scheduler Role

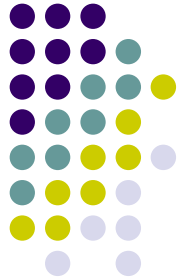


- Defines which task is accessible in particular timeslot
- Task schedule is available to students on electronic notice-board
  - Students may reserve particular timeslots
- Time divided to fixed-size timeslots (45 mins)

Task Scheduling - Screenshot

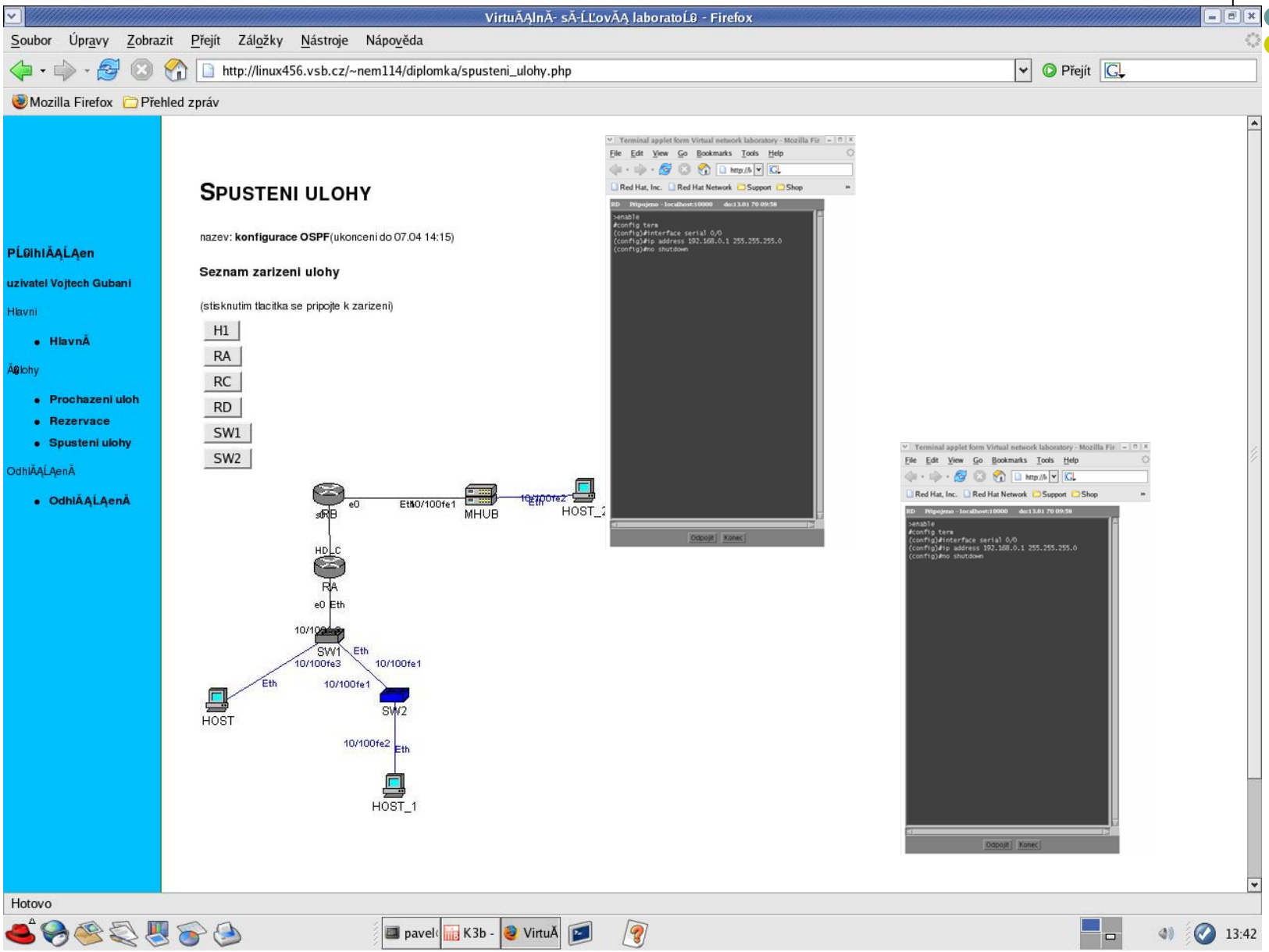


Student Role

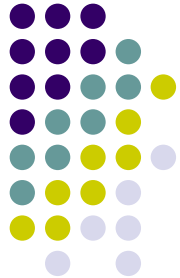


- Reserves timeslots offered using electronic notice-board for himself/herself
  - may also lists colleagues who he/she wants to solve the task with him/her
- Can access devices' consoles of reserved task in his/her timeslot using Java applet running in his/her WWW browser

Remote Device Access - Screenshot

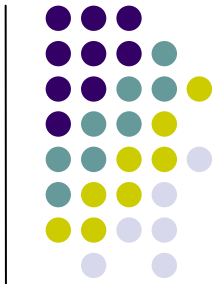


Java Applet Access Client

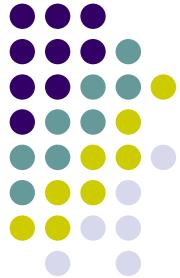


- Launched by access management system in student's browser for each remotely-accessed device in separate window
- Provides access to laboratory device's console and additional functions
  - Allow student alternately connect and disconnect to the console to let multiple students share single device
    - Single student holds the console at each instant
    - student is informed who holds the device at current instant
  - Disconnection warnings before timeslot ending
  - Protects from entering prohibited commands
    - (defined by system administrator)
  - Allows input/output capture to local file system
  - Allows cut&paste insertion of commands into console
  - Informs that tutor started/stopped to access the same device and optionally displays tutor's activity

Tutor Role

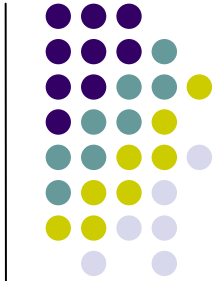


- Tutor can access console of any laboratory device anytime
  - Demonstration mode – student can see what tutor does
  - Hidden mode – student only knows that tutor took over his/her console
    - used for student examination purposes
- Future plan: tutor can passively watch student’s activities at any laboratory device



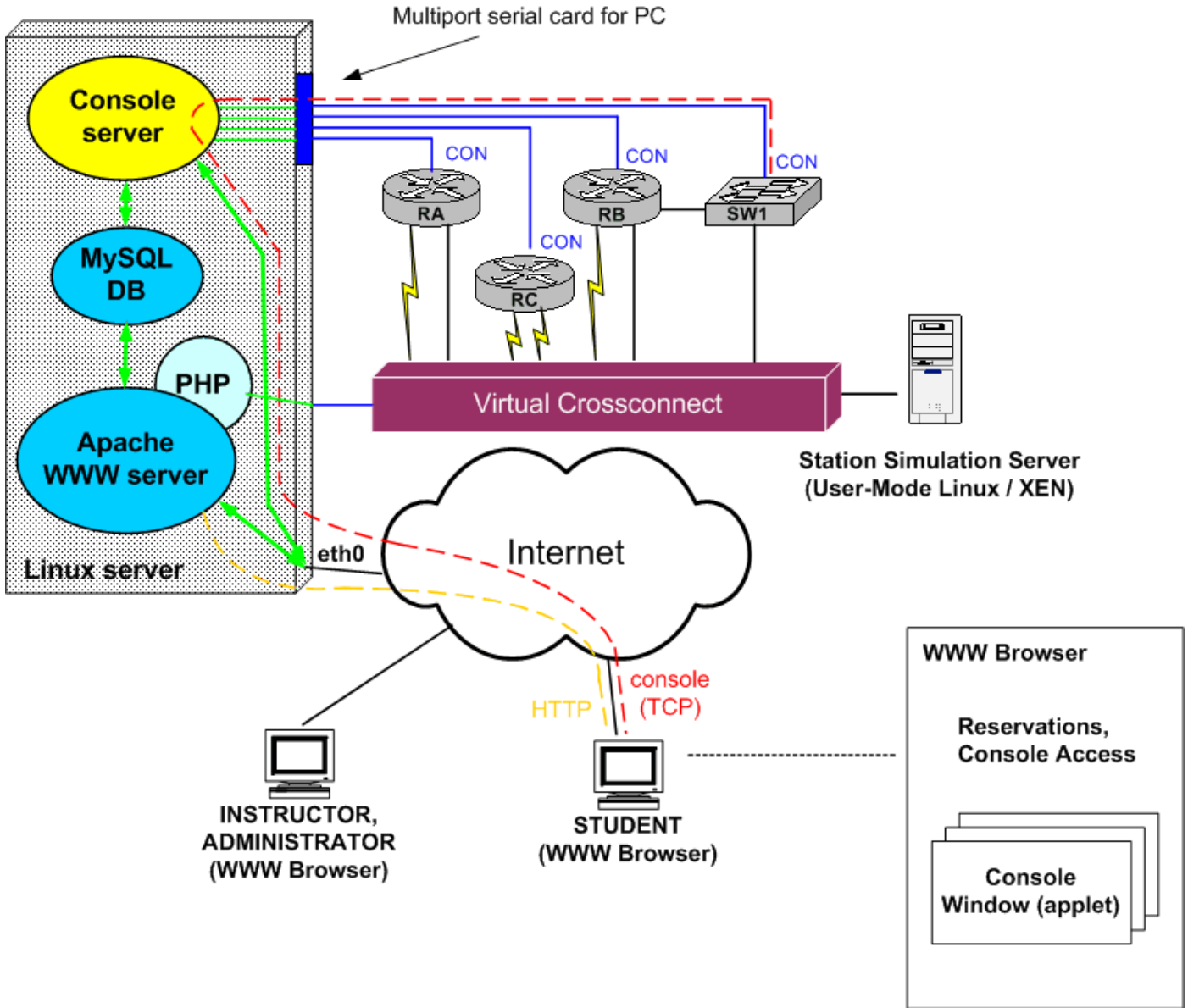
Automatic Topology Interconnection System

Device Topology Interconnection

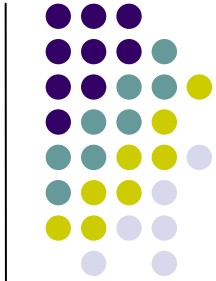


- Before task is made accessible to student(s), required topology has to be interconnected
- We use our own “Virtual Crossconnect”
  - integrates various switching elements
- Topology may be also physically connected by dedicated person informed to do so via email generated by access management system
  - but it is proved unacceptable for real operation and will be abandoned in the future

Position of Virtual Crossconnect in the System Architecture

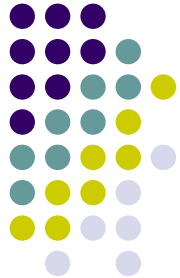


Why “Virtual” Crossconnect ?



- Implemented using various and multiple physical switching elements
  - ASSSK1, 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 Virtual Crossconnect ports
- Required topology description is completely independent of types of actually used switching elements

Virtual Crossconnect Switching technologies



- Serial Ports – our own microprocessor-controlled analog crossbar implementation
  - Current model called ASSSK-1
  - New FPGA-based model under development
- Ethernet Ports
  - VLAN-based interconnection between L3 devices
  - VLAN-tunneling (802.1 QinQ) between switches
    - allows interconnection of trunk links
    - transparent to Spanning Tree and other L2 control protocols

Currently Used Virtual Crossconnect Switching Elements

ASSSK-1 (developed by us)

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

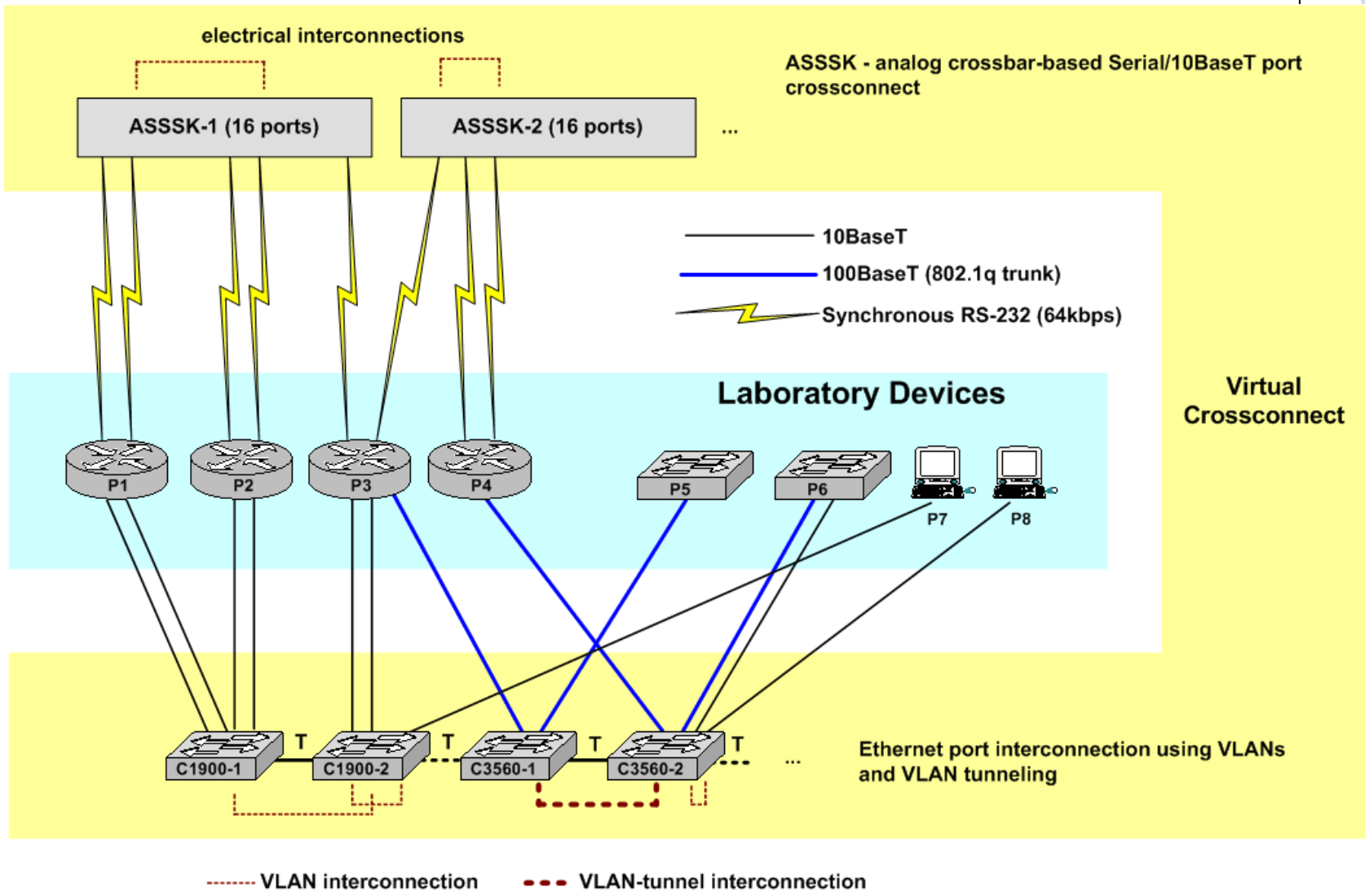
… possibly other switching elements in the future

Cisco Catalyst 1900
(cheap solution)

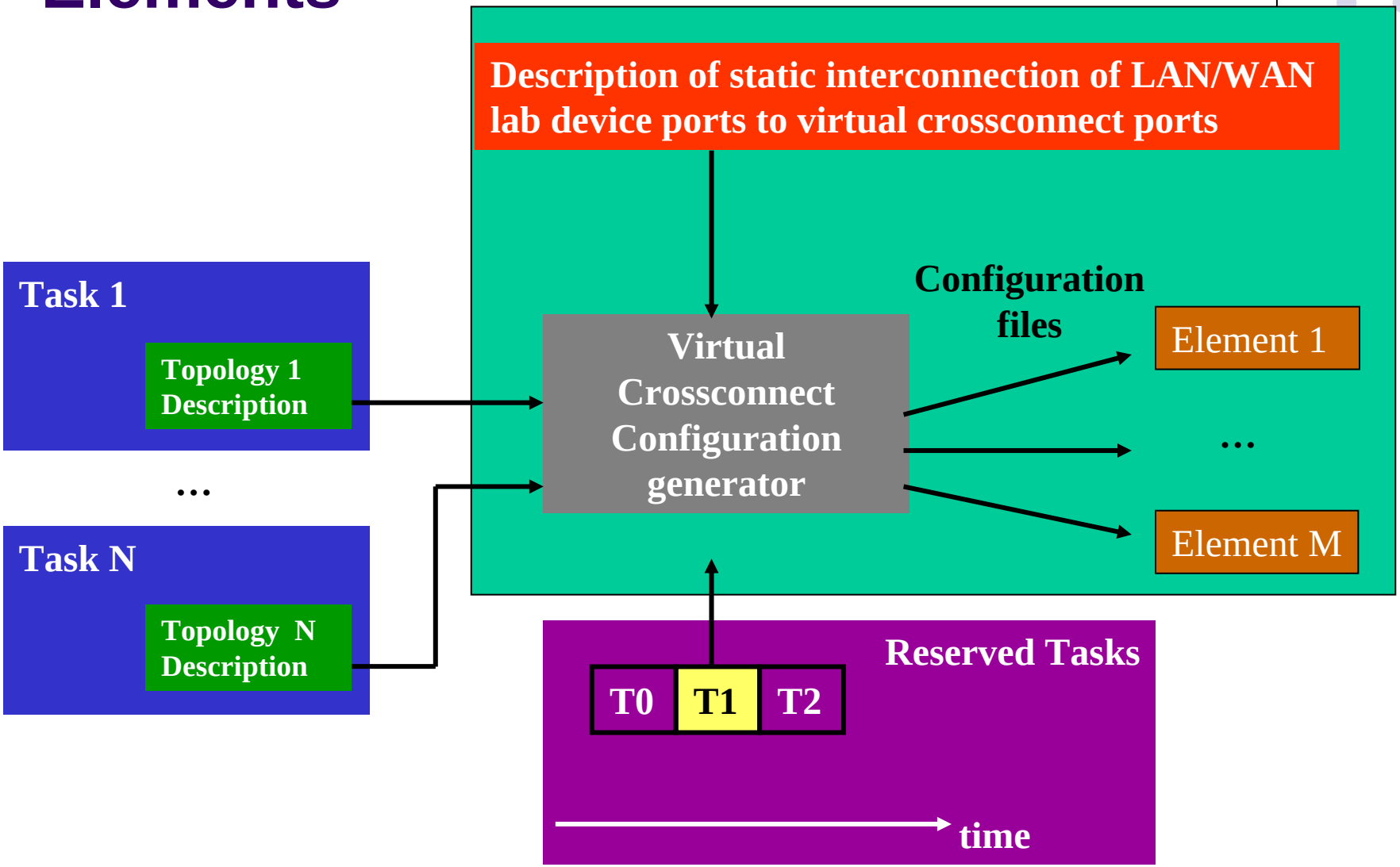
ASSK-1 Switching Element

- Modular design
- 16 Serial/Ethernet ports
- Serial ports behave like DCE
- Controlled using IOS-style CLI

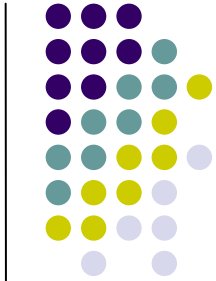
Virtual Crossconnect Components in Action



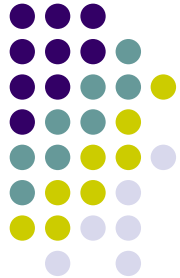
Generation of Configurations for Virtual Crossconnect Switching Elements



Virtual Crossconnect - Future plans

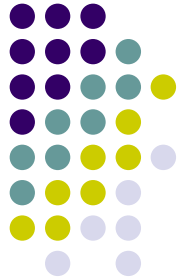


- ASSSK-1
  - Various clockrates on serial lines
  - WAN links flapping simulation to let students get experience with real-word WAN troubleshooting
- ASSSK-2 (under construction now)
  - Redesign of ASSSK1 using FPGA
    - smaller, cheaper
  - Only for serial ports inteconnection
    - for Ethernet ports, VLAN-tunneling using standard switches proved more efficient
- ASSSK-3 (under investigation)
  - serial-port crossconnect implementation based on multiport serial card for PC, Linux HDLC/PPP drivers and bridging software
- Enhancement of element configuration generator scripts
  - implement semantic checks of topology definition provided by student who requests his/her own topology



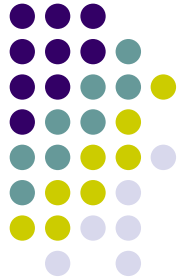
Incorporation of Simulated Network Devices

What Simulated Devices do we Use ?



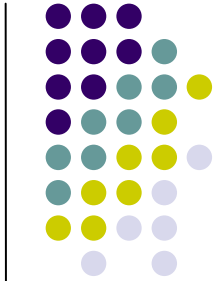
- Stations running Linux
  - full user control using text console as with console-controlled network devices
  - fast, efficient, deterministic and reliable networking configuration
- Simulated Cisco 7200-series routers
  - Implemented using open-source DynaMIPS/DynaGEN project
  - Flexible platform to test advanced routing features

Reasons to Incorporate User Stations into Lab Topologies



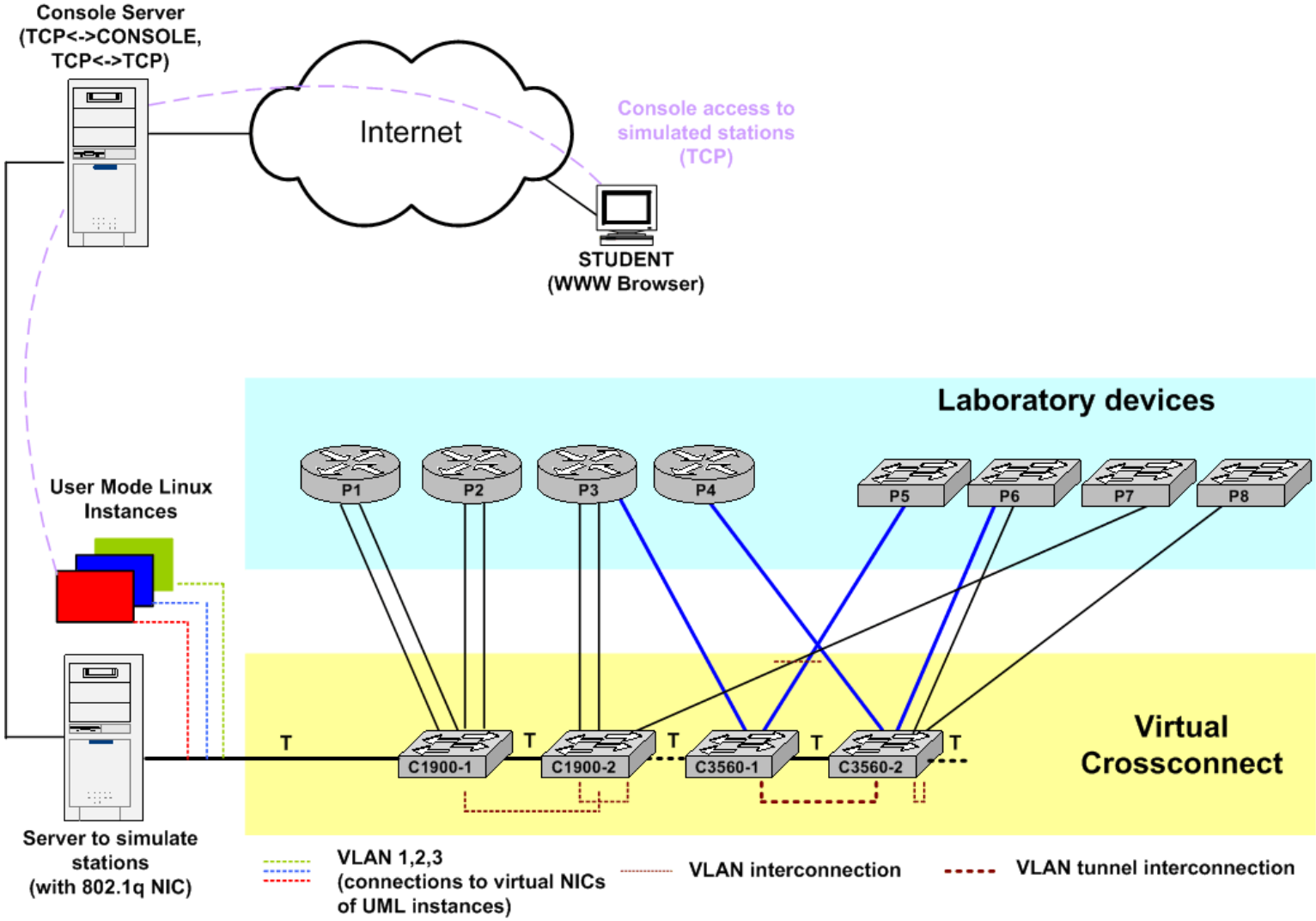
- User stations often necessary to test functionality of task solution
- Stations may serve to run various network services and/or practice configuration of these services
  - DHCP, DNS, Syslog, RADIUS, TACACS, …
- Servers needed to run applications to tests access lists configuration
  - WWW, FTP, Telnet, SSH, …

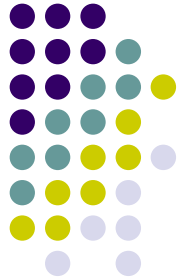
Advantages of Simulated Devices



- It is ineffective to use multiple physical PCs
  - Physical space requirements, energy consumption, complicated installation/administration
  - No processor or memory intensive application are expected to be ran
- Some virtualization mechanism proved to be useful
  - User-Mode Linux (UML) used currently, usage of XEN considered
- Simulated Ethernet interfaces connected to virtual crossconnect using VLANs
  - Virtual crossconnect architecture was generalized to be able to interconnect devices residing on fixed VLANs with real or other simulated devices

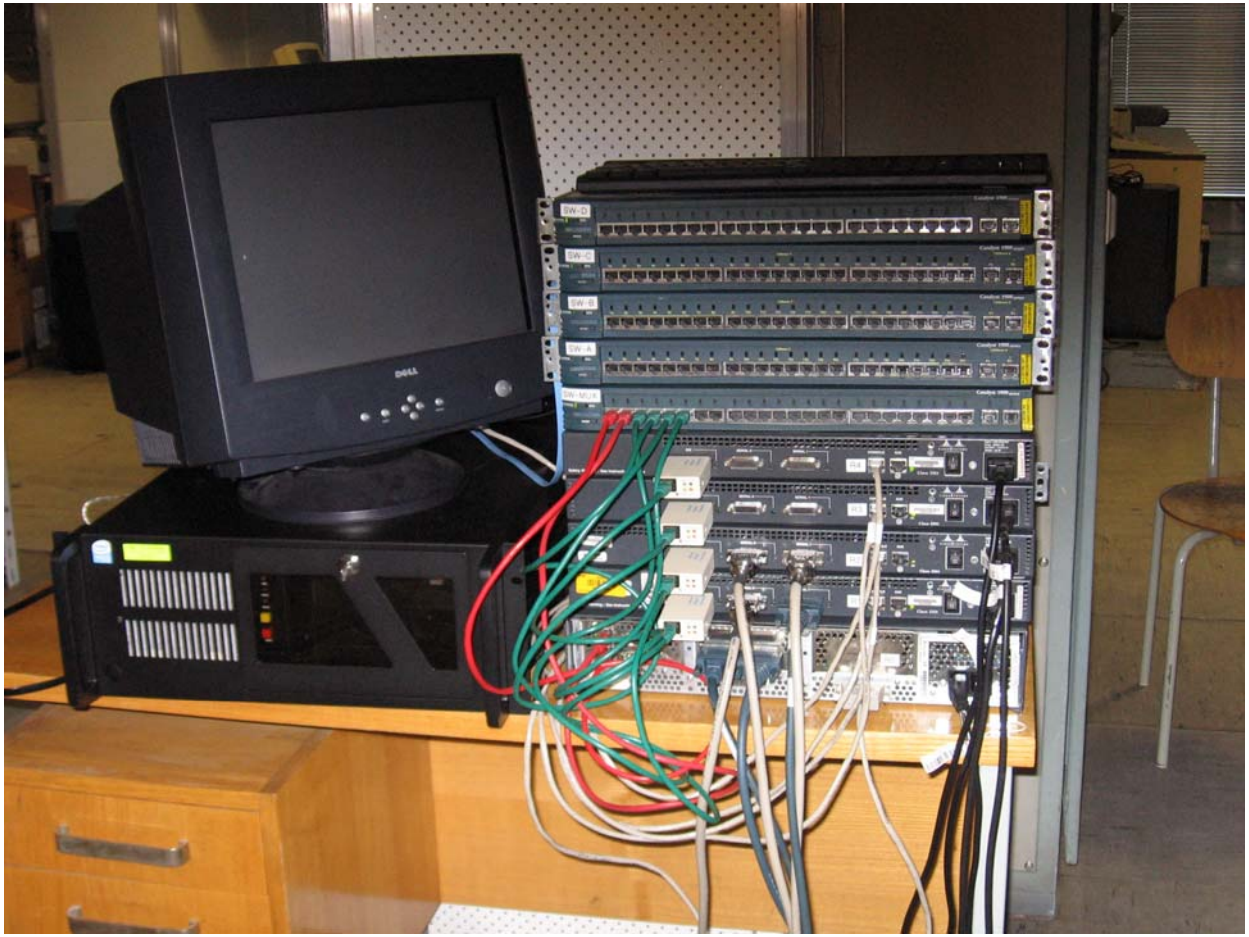
Integration of Simulated Devices with Virtual Crossconnect



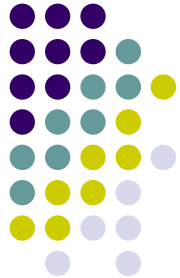


Today's Functional Single-Site Implementation

How We Started:
The Very First Virtlab Implementation

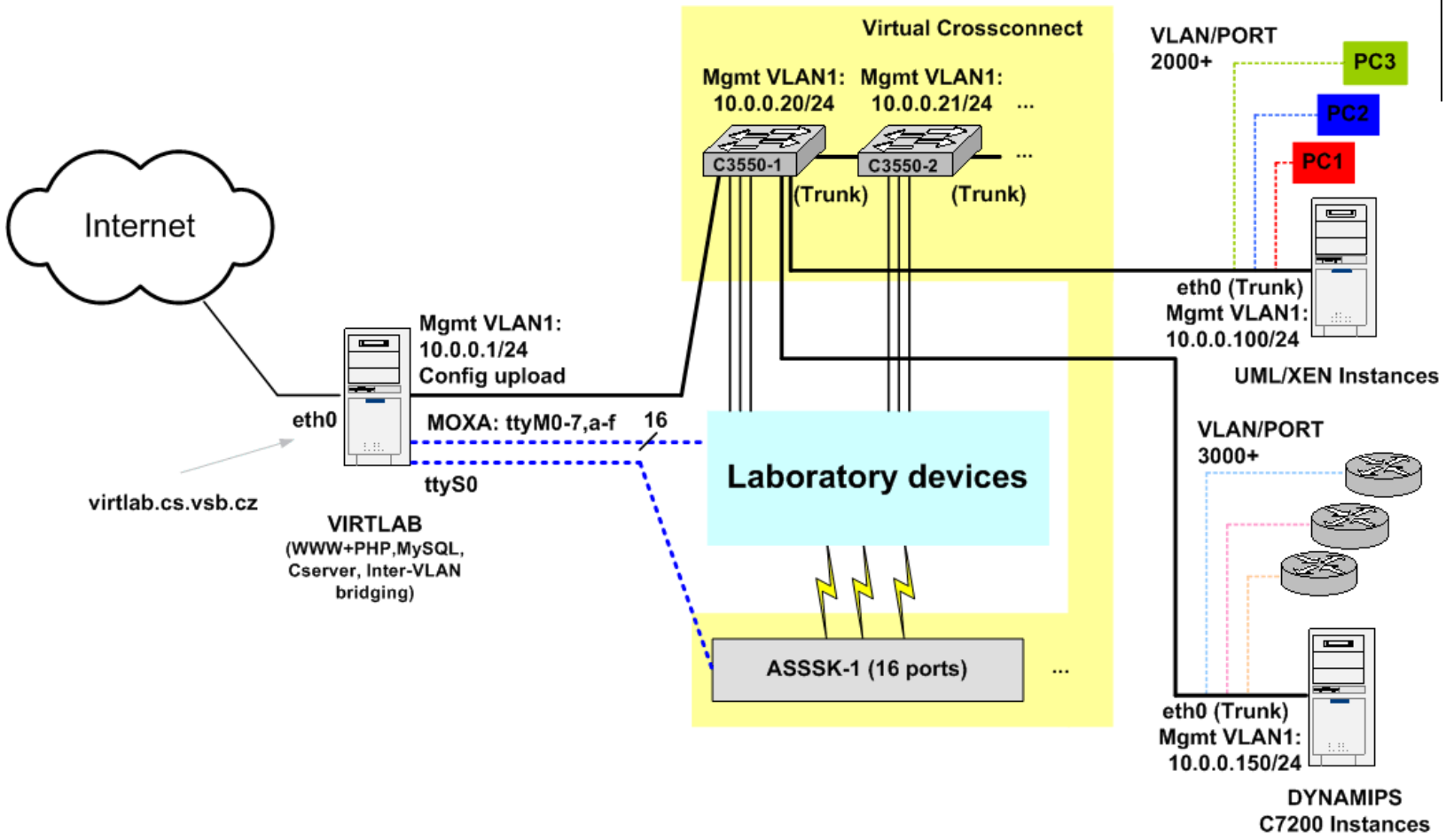


Today's Virtlab

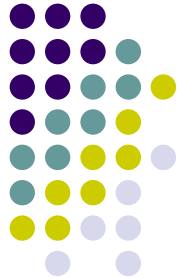


- Two 45U racks incorporating
  - Access Management System Server
    - (Virtlab Server)
  - Virtual Crossconnect
  - UML server
  - DynaMIPS server
  - C2500/2600/4000 routers
  - C1900 switches
  - C2500 switches
  - C5500 switch with RSM

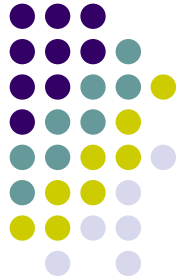
Today's Virtlab Internal Architecture



System Security



- Web interface – uses HTTPS
  - users authenticated using passwords stored in database or through LDAP
- Console access
  - One-time password authentication
  - Digitally-signed applet
- Access via firewalls taken into account
  - single fixed TCP used for console access
- Extensive logging of user's activity
  - Access reservation system logins
  - Console access

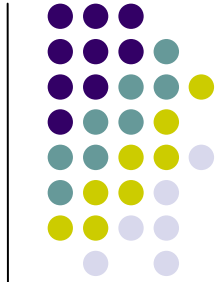


Architecture Extension

Dynamic Mapping of Network Devices used for Particular Reserved Tasks

(Current Work)

The General Idea (1)

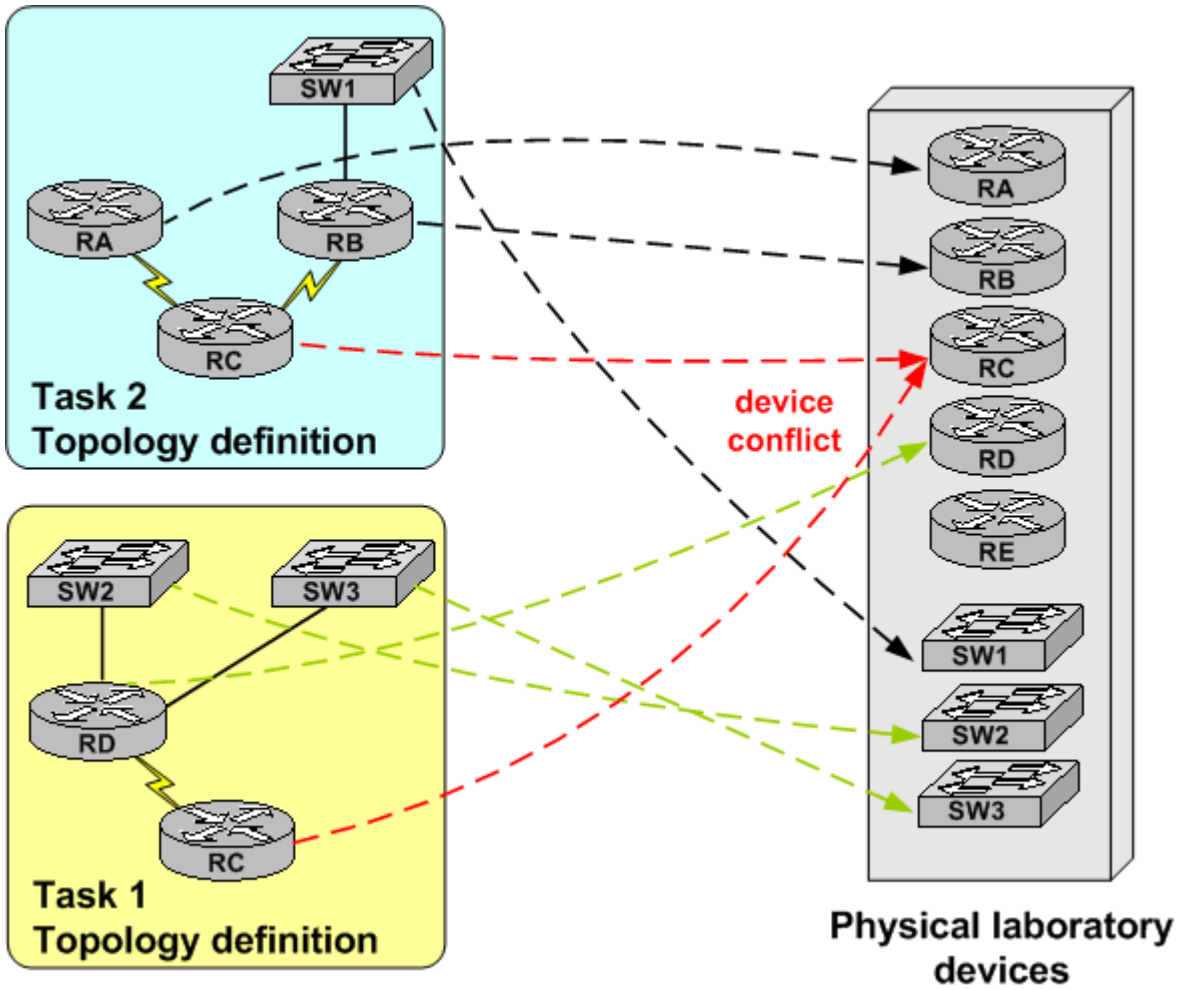


- Fixed-length and fixed-beginnings timeslots proved inefficient
  - Students are not interested in some scheduled tasks but there is a contention for timeslots with other more interesting tasks
- Since we are able to interconnect topologies automatically, it is better to let students choose ANY task at ANY timeslot
  - fixed timeslots abandoned, student may reserve any time interval up to his/her weekly quota
- Task scheduler role no longer needed
  - notice-board used only to remember what task was reserved for what time interval

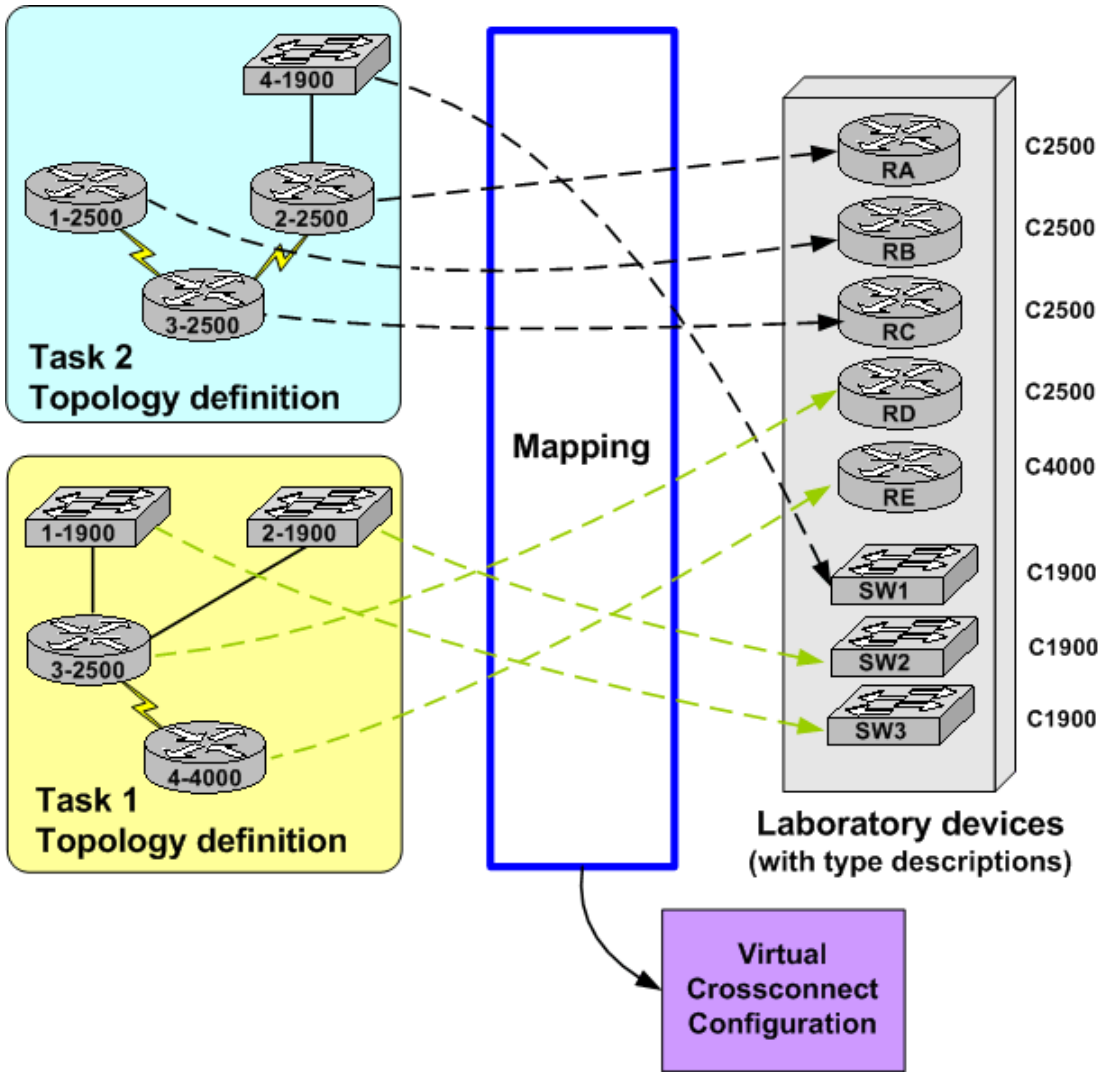
The General Idea (2)

- 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 are chosen dynamically with respect to other devices used at the same time

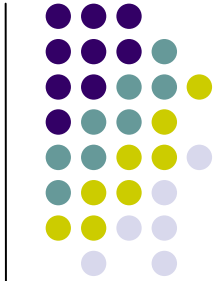
Problem of Task Definition Coupled with Physical Device Identities



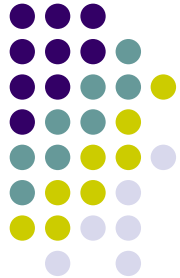
Dynamic Mapping of Task to Physical Devices During Task Reservation Procedure



Function of Dynamic Device Mapping



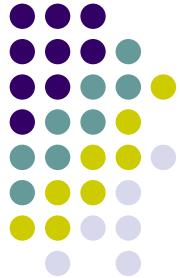
- Mapping takes place when student reserves task for particular time interval for himself/herself
- Network device types, numbers of interfaces and other features have to be taken into account
- Multiple tasks may be mapped in parallel
  - feasibility determined during mapping process at reservation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



Distributed Multiple-Site Architecture

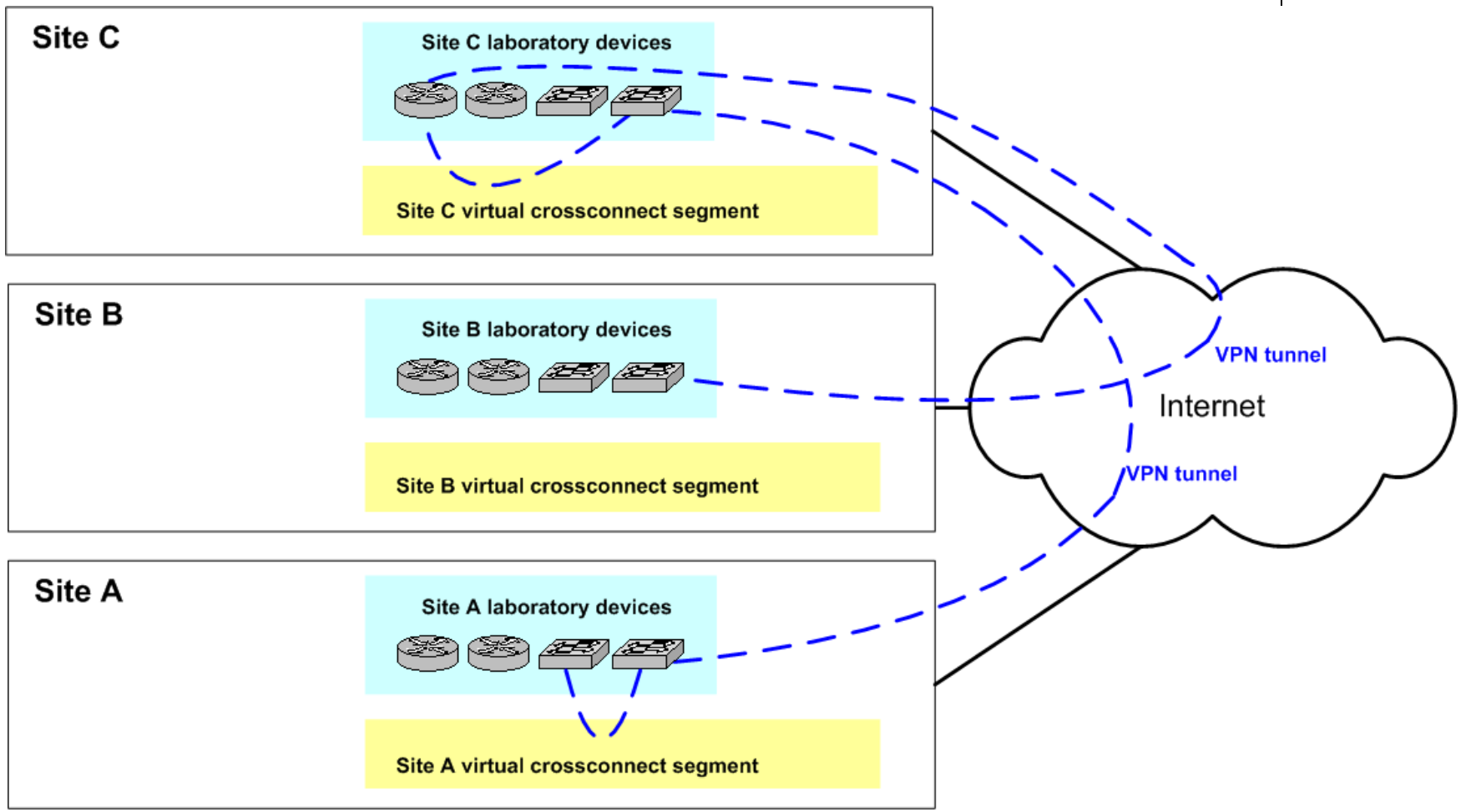
(Current Work)

Multiple-Site Architecture Basic Paradigm

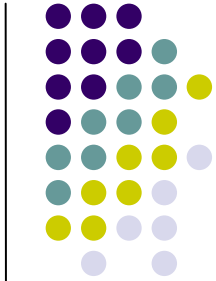


- Multiple cooperating sites may share laboratory devices transparently
  - Constraints to offer site's lab devices for other sites may be specified
  - Local mappings are preferred to limit inter-site traffic
- Fully-decentralized architecture allows independent operation of individual sites if other 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

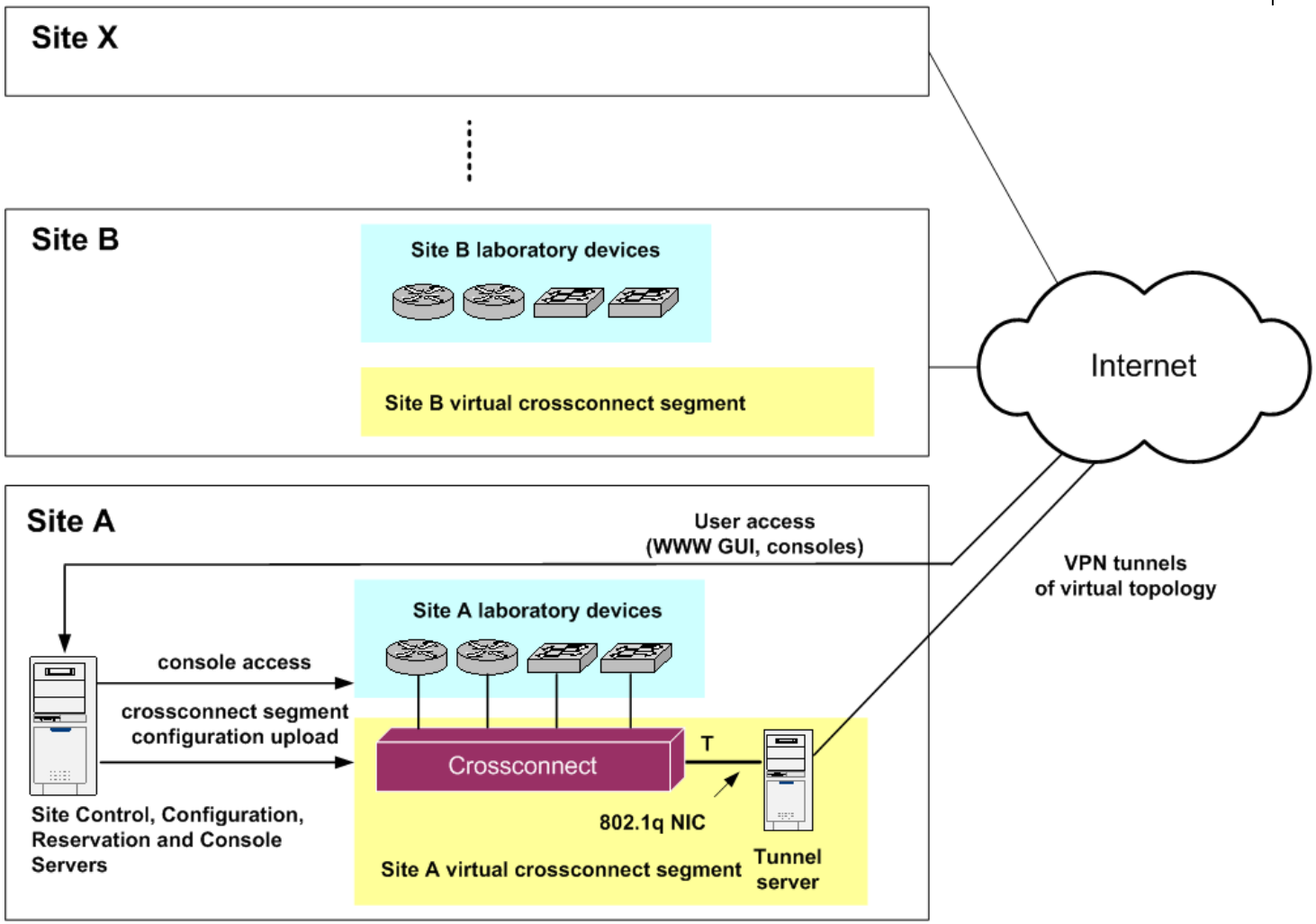


- Distributed virtual topologies may be constructed using Internet tunnels
  - Tunneling of Ethernet links is implemented now
  - We are also working on HW/SW solution to tunnel serial (WAN) links
- Layer 2 frames tunneling allows transparent operation of multiple layer 3 protocols

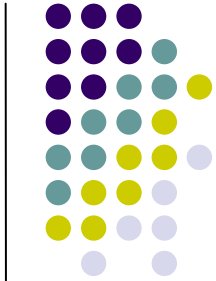
Extensions Proposed to Implement Distributed Topologies

- Virtual Crossconnect concept was generalized to support multiple-site topologies
- Tunnel Servers allow 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 in individual sites which form single Distributed Virtual Crossconnect

Multiple-Site Architecture Basic Components

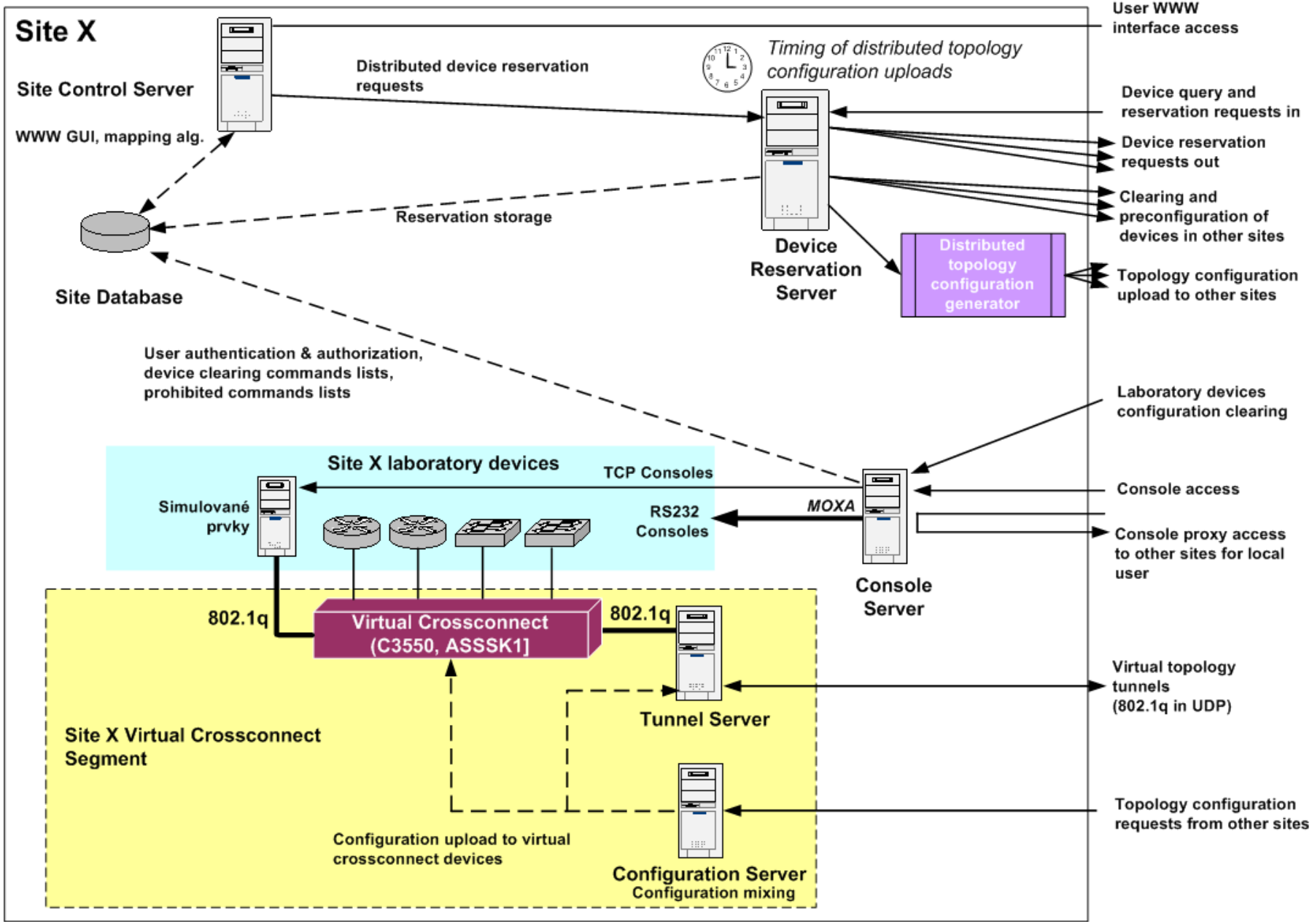


Function of Control Components

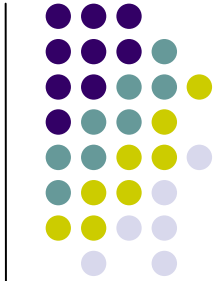


- Virtlab Server
  - Provides access management system GUI, console access GUI and device mapping algorithm
- Reservation Server
  - Keeps track of site devices' reservations
- Configuration Server
  - accepts distributed virtual crossconnect configuration requests
- Console Server
  - allows access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

Interaction of System Components



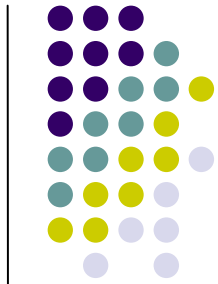
Advantages of Multiple-Site Architecture



Multiple cooperating teaching/research institutions are able to

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

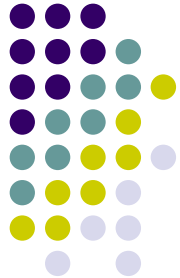
Development Team



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

- Pavel Nemec
- Roman Kubin
- David Seidl
- Martin Milata
- Jan Vavricek
- Tomas Hrabalek
- Tomas Kucera
- Petr Sedlar
- … and others

Current Status and Planned Development



- See http://www.cs.vsb.cz/vl-wiki