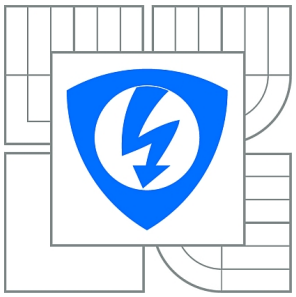
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

SYSTÉM SPRÁVY KONFIGURACÍ PÁTEŘNÍ SÍTĚ CISCO

CONFIGURATION MANAGEMENT FOR CISCO BACKBONE NETWORK

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

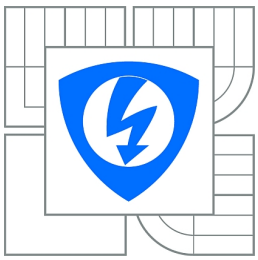
AUTOR PRÁCE — Bc. JAN BARTOŠ

AUTHOR

VEDOUCÍ PRÁCE — Ing. MICHAL POLÍVKA

SUPERVISOR

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Jan Bartoš	***ID:***	78504
Ročník:	2	***Akademický rok:***	2009/2010

NÁZEV TÉMATU:

Systém správy konfigurací páteřní sítě Cisco

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte základy problematiky IP sítí, konfigurace Cisco směrovačů, principy operačního systému IOS. Zaměřte se na možnosti připojení k síťovým zařízením, jejich výhody a nevýhody (protokoly, šifrování, autentizace). Navrhněte systém správy konfigurací laboratorní sítě složené ze sedmi směrovačů, jednoho přepínače a tří IP telefonů. V jazyce Java implementujte rozhraní pro síťovou komunikaci mezi externím zařízením (počítačem) a směrovačem potřebné pro přenos jeho konfigurace přes tftp protokol. Pro připojení ke směrovači využijte protokol telnet. Aplikace bude umožňovat ověření existence spojení mezi počítačem a směrovačem (využití protokolu ICMP). Zabývejte se možností několikanásobného připojení k síťovým zařízením pro potřebu hromadného stahování/nahrávání konfigurací vybraných síťových zařízení. Pro aplikaci navrhněte vhodné grafické rozhraní umožňující budoucí rozšíření programu, zejména práci s konfiguračními soubory směrovačů a přepínače.

DOPORUČENÁ LITERATURA:

[1] WENDELL, Odom, HEALY, Rus, MEHTA, Naren. Směrování a přepínání sítí : Autorizovaný výukový průvodce. David Krásenský. 1. vyd. Brno : Computer Press, a.s., 2009. 879 s. ISBN 978-80-251-2520-5.
[2] DOSTÁLEK, Libor, KABELOVÁ, Alena. Velký průvodce protokoly TCP/IP a systémem DNS. 3. vyd. Praha: Computer Press, 2002. 542 s. ISBN 80-7226-675-6.
[3] ZAKHOUR, Sharon, et al. Java 6 : Výukový kurz. Brno : Computer Press, 2009. 536 s. ISBN 978-80-251-1575-6.

Termín zadání:	29.1.2010	***Termín odevzdání:***	26.5.2010

Vedoucí práce: Ing. Michal Polívka

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

ABSTRAKT

Cílem diplomové práce bylo vytvořit aplikaci pro správu konfigurací páteřní sítě od firmy Cisco. Tato aplikace měla umožnit uživateli jednoduchou a přehlednou manipulaci s konfiguračními soubory. Jejím úkolem bylo komunikovat se zařízením pomocí vzdáleného připojení a stahovat podle potřeb uživatele konfigurační soubory. Program měl splňovat i funkci opačnou – nahrávat konfigurační soubory zpět na zařízení. Dále měla aplikace poskytovat možnost ověření dostupnosti zařízení pro vzdálené připojení. V realizaci byl pro ověření dostupnosti použit protokol ICMP, vzdálené připojení obstarává protokol telnet a pro přenos konfiguračních souborů se používá protokol TFTP. Program je realizován multivláknově v jazyce Java s podporou grafické knihovny Swing. Podporuje verzování konfiguračních souborů. Uživateli je umožněno ovládat program za chodu i pomocí editace textového souboru, do kterého program ukládá svou konfiguraci. Program a jeho vlastnosti jsou podrobně popsány v druhé části práce. První část je věnována teoretickým poznatkům z oblasti IP sítí a konfigurace zařízení Cisco v páteřní síti pomocí vzdáleného připojení.

KLÍČOVÁ SLOVA

Konfigurace páteřních sítí, IOS, TFTP, telnet, SSH.

ABSTRACT

The main aim of the master thesis was to create an application for backbone network configuration management. The priority of the application was to be effective and easy to take. Program has to be able to communicate with network devices by remote control and to download and upload configuration files. The program has enable also availability verification for remote control. For this purpose the ICMP protocol was used. The remote control is provided by telnet protocol and the protocol TFTP has been chosen for file tranfers providing. The application is a multithreading application based on the Java programming language. The Java graphic library Swing was used for GUI creation. The application supports file versioning, the configuration of the program can be saved as a text file, which can be easy changed by user. The program and its features are described in the second part of this thesis. The first part deals with theoretical facts about the IP networks and remote control of the Cisco backbone network devices.

KEYWORDS

Remote control, Backbone network, IOS, TFTP, telnet, SSH.

BARTOŠ, J. *Systém správy konfigurací páteřní sítě Cisco.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 59 s. Vedoucí diplomové práce Ing. Michal Polívka.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Systém správy konfigurací páteřní sítě Cisco“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojù, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným zpùsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., všetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne . . . .

(podpis autora)

PODĚKOVÁNÍ

Rád bych na tomto místě poděkoval vedoucímu práce panu Ing. Michalu Polívkovi za vedení a pomoc a za poskytování konzultací po celou dobu vypracovávání diplomové práce.

V Brně dne . . . .

(podpis autora)

Obsah

Úvod ... 10
1 Problematika IP sítí ... 11
1.1 Topologie sítí ... 11
1.2 Připojení uživatele k síti ... 13
1.3 Možnosti řízení ... 14
1.4 Bezpečnost ... 15
1.4.1 Mechanizmy zabezpečení páteřních sítí ... 15
1.5 Vzdálená správa zařízení ... 16
1.5.1 Telnet ... 17
1.5.2 Secure shell ... 18
1.6 Autentizace uživatele ... 18
1.6.1 Autentizace pomocí lokálních záznamů ... 19
1.6.2 Kerberos ... 19
1.6.3 RADIUS ... 22
1.6.4 TACACS+ ... 24
1.7 Cisco IOS ... 26
1.7.1 Módy Cisco IOS ... 27
1.7.2 Verzování Cisco IOS ... 30
1.7.3 Volba systému IOS ... 31
1.8 Konfigurace síťových zařízení ... 31
1.8.1 Práce s konfiguračními soubory ... 31
1.8.2 Podpora protokolů pro přenos souborů ... 33
2 Program pro správu konfigurací ... 34
2.1 Koncept ... 34
2.2 Požadavky programu ... 36
2.3 Funkce programu ... 37
2.3.1 Uživatelské funkce ... 37
2.3.2 Systémové funkce ... 44
2.4 Rozšíření programu ... 47
3 Závěr ... 50

Literatura 51
Seznam příloh 54
A Seznam balíčků a tříd programu 55
A.1 Balíček (default package) 55
A.2 Balíček database 55
A.3 Balíček exceptions 55
A.4 Balíček fileProcessing 56
A.5 Balíček gui 56
A.6 Balíček netCommunication 57
B Ukázky zdrojových kódů programu 58
B.1 Jádro ověřování dostupnosti zařízení 58
B.2 Definování barev pro sloupec tabulky připojení k zařízení 59
B.3 Funkce obstarávající verzování 59

Seznam obrázků

Obr. 1: Kerberos autentizace 21
Obr. 2: RADIUS autentizace 23
Obr. 3: Cisco Feature Navigator 27
Obr. 4: Hlavní části programu 35
Obr. 5: Okno po spuštění 37
Obr. 6: Zobrazení uživatelských skupin 38
Obr. 7: Menu programu 39
Obr. 8: Ověření připojení 40
Obr. 9: Nová skupina 41
Obr. 10: Editace skupiny 42
Obr. 11: Odebrání skupiny 42
Obr. 12: Nové zařízení 43
Obr. 13: Odebrání zařízení 44
Obr. 14: Vývojový diagram vlákna pro komunikaci se síťovým zařízením 45
Obr. 15: Příklad konfiguračního souboru 46
Obr. 16: Zdrojový kód - operace s vlákny při ověřování dostupnosti zařízení 58
Obr. 17: Dynamické barvení sloupce tabulky podle stavu připojení 59
Obr. 18: Funkce pro verzování souborů 59

Úvod

Konfigurace a správa konfigurací síťových zařízení je dnes rutinní praxí v případě malých i rozsáhlých sítí. Zařízení jsou na zásahy do konfigurace velmi citlivá, proto je u nich více než užitečné vytvářet zálohy konfigurací. Při poruše zařízení je nutné nejen zachovat dostupnost sítě, ale zároveň je důležitý mechanizmus, s jehož pomocí lze síť uvést opět do původního stavu, což závisí i na existenci záložních kopií konfiguračních souborů. Stejná situace nastává, pokud jsou v konfiguraci nebo přímo v systému prováděny nejrůznější změny.

Cílem diplomové práce je vytvořit program, který bude umožňovat jednoduchou práci s konfiguracemi síťových zařízení. Program bude využívat připojení k zařízením pomocí protokolu telnet a bude podporovat stažení konfigurace do programu a naopak protokolem pro přenos souborů TFTP. Součástí komunikace bude i možnost ověřit připojení k jednotlivým zařízením pomocí protokolu ICMP. Program bude podporovat několikanásobné připojení pro dosažení větší efektivity obsluhování. Jednoduchost ovládání bude docílena pomocí grafického rozhraní.

Diplomová práce je rozdělena na dvě základní části. Teoretická část se zabývá problematikou IP sítí. Obsahuje základní informace o síťové architektuře a bezpečnosti, podrobněji jsou zde rozebrány způsoby autentizace uživatele v síti, možnosti připojení uživatele k zařízení (fyzicky a virtuálně) a operační systém použitých síťových zařízení Cisco IOS.

Druhá část popisuje realizaci programu pro správu konfigurací páteřní sítě Cisco. Nejprve jsou představeny základní části programu, dále vlastní podoba programu – jeho vzhled a všechny možnosti, které program uživateli nabízí. Je popsán proces připojování a stahování konfigurace, aby byl čtenář dostatečně informován o tom, jak přesně program pracuje, které protokoly při práci používá, za jakým účelem a proč nejsou použity protokoly jiné. Pro grafické rozhraní jsou stanoveny ovládací prvky a jejich funkce. Praktická část je zakončena úvahou na téma rozšíření programu. Jsou popsány základní možnosti rozšíření, jejich potenciál a složitost, a také důvody, proč nebyly implementovány v této práci.

1 Problematika IP sítí

Počítačové sítě jsou v současné technické době brány jako samozřejmost. Historie propojování počítačů je dlouhá necelých 50 let a za tuto dobu došlo k obrovskému rozmachu počítačových sítí a jen velmi těžko si lze představit, co by způsobil například kolaps globální sítě Internet. IP síť je označení pro datovou (paketovou) síť, která využívá pro komunikaci na síťové vrstvě protokol IP (architektura TCP/IP). Síť, která využívá protokol IP, může být veřejná i privátní (také virtuální privátní tzv. VPN) a protokol IP zde zajišťuje přepojování paketů pomocí logických adres IP. Tyto sítě mohou být využívány pro mnohé aplikace, které poskytují například přenos hlasu či videa. Důležitými vlastnostmi takových sítí je jejich rychlost a spolehlivost, ale také míra zabezpečení. Následující kapitola se zabývá architekturou sítě, která hraje ve zmíněných oblastech významnou roli.

1.1 Topologie sítí

Uvažování nad tím, jak vlastně jednotlivé počítače propojit, vedlo k ustálení 3 základních topologií sítí – hierarchický, redundantní a bezpečný [29] .

Hierarchický model

Hierarchický model vychází z vrstvové architektury (podobně jako komunikační vrstvy v modelu OSI nebo TCP/IP). Výhodou těchto modelů je přehlednost sítě (zejména při zjišťování závad), snadná rozšiřitelnost, jasně definované rozdělení úloh a také finanční úspora. Hierarchický model dělí síť na vrstvu jádra, distribuční a přístupovou. Vzhledem k různě rozsáhlým sítím je možné vrstvy sjednotit – jedná se o velmi jednoduché sítě s jedním nebo dvěma prvky plnícími úlohy všech 3 vrstev.

Vrstva jádra sítě (core layer) tvoří základní prvek celé sítě (často také nazývána jako páteřní z angl. backbone). Charakteristickým rysem této vrstvy je vysoká rychlost a spolehlivost. Rychlost je docílena vysokorychlostními rozhraními a kvalitním hardwarem (jedná se často o nejdražší směrovací prvky v síti), který umožní vysokou rychlost zpracování směrovaných nebo přepínaných paketů. Spolehlivosti je docíleno pomocí redundance. Páteřní síť by měla být přizpůsobivá a neměla by provádět složité filtrační operace.

Distribuční vrstva (distribution layer) vytváří mezivrstvu mezi jádrem a přístupovou částí sítě. V této části sítě jsou implementována pravidla sítě a směrování mezi virtuálními lokálními sítěmi (VLAN). Na této úrovni se řeší sjednocení použitých technologií (jako Ethernet, token ring, ISDN), dále překlady

privátních a veřejných adres (NAT) a nejrůznější metody filtrování paketů v síti. Do této vrstvy je také zahrnováno zajištění kvality služeb (Quality of Services, QoS).

Třetí a nejnižší vrstvou v hierarchii je vrstva přístupová. Úkolem přístupové vrstvy je zajistit přístup uživatelům k síti. Do této oblasti se zahrnuje i připojení serverů nebo vzdálené připojení uživatele do podnikové sítě pomocí virtuálního připojení (VPN). Příkladem jsou domácí sítě SOHO (small office/home office).

Redundantní model

Redundantní model nachází uplatnění v kritických místech sítě. Podle umístění lze redundanci rozdělit podle [29] na redundantní komunikaci pracovních stanic se směrovači (routery), redundanci serverů, redundanci cest a redundanci médií.

Redundantní komunikace je založena na několika různých protokolech, kterými stanice se směrovačem komunikuje. Základní komunikace je pomocí protokolu ARP (Address resolution protocol) nebo protokolu RDP (Router discovery protocol) či OSPF (protokol podporuje mimo jiné provoz na pracovních stanicích). Pokud je směrovač nedostupný, umožňuje komunikaci stanic protokol HSRP (Hot Standby Router Protocol, protokol pro horkou zálohu směrovačů). Tento protokol vytváří fiktivní směrovač (fantom), který pak jednotlivé stanice vydávají za svou bránu.

Redundance serverů znamená znásobení datových úložišť se stejným obsahem přístupných v síti. Při vytváření násobných úložišť dat je důležitá jejich fyzická nezávislost (různé disky, různé počítače) při zachování dokonalé synchronizace.

Redundance cest zajišťuje rozložení zátěže a dostupnost síťových prvků při výpadku sítě. Cisco zařízení podporují rozložení zátěže až mezi 6 linek (výchozí hodnotou bývá 4). U výpadků prvku sítě rozhoduje mimo redundanci linek také rychlost konvergence sítě (zajišťují směrovací a linkové protokoly). Z hlediska redundance linek lze vytvořit topologii úplného grafu, označovanou v angličtině jako full mesh topologie.

K doplnění popisu typů redundance zbývá redundance médií. Příkladem takové redundance může být koexistence bezdrátové sítě s „klasickou“ technologií Ethernet v jedné budově.

Bezpečný model

Bezpečný model využívá při své činnosti prvky jako šifrování nebo přístupové seznamy. Do této sítě patří standardně firewall s implementovanými mechanizmy pro zabránění nepovolanému přístupu k síťovým zdrojům. U bezpečných modelů sítě se uplatňuje tzv. trojdílný firewallový systém [29] **,** který je tvořen z izolační sítě LAN, vnitřního paketového filtru a vnějšího paketového filtru.

Izolační síť (označována také jako demilitarizovaná zóna) obsahuje veškeré služby, které jsou určeny pro sdílení s vnějším světem. Často se jedná o HTTP, FTP nebo DNS servery. Je pro ni typická odlišná IP adresace a statické směrování v síti. Je jedinou částí vnitřní sítě, která je přístupná veřejnosti.

Vnitřní paketový filtr tvoří hranici mezi izolační sítí a vnitřní sítí. Směrovače (nebo firewally) tvořící tento hraniční bod často obsahují pravidla založená na blokování nové příchozí komunikace a povolování odchozí komunikace. Zároveň povolují tato zařízení příchozí komunikaci, která byla vytvořena z vnitřní sítě (tzv. založená či navázaná spojení) – často TCP pakety odpověď (ACK) nebo reset (RST).

Podobnou politiku zachovává i vnější paketový filtr, který se nachází mezi izolační sítí a veřejnou sítí. Nastavení tohoto směrovače však musí povolit i novou příchozí komunikaci pro dané komunikační porty služeb, na kterých izolační síť zpřístupňuje veřejnosti své služby. Jelikož se hraniční prvky mohou stát prvním terčem útoků hackerů, doporučuje se blokovat u firewallů veškerý síťový provoz do vnitřní sítě. Na tomto místě se také zakazují všechny nepotřebné služby (TFTP server, proxy ARP, dynamické směrování – používá se statické [29] .

Realizace sítí

Současné sítě jsou vytvářeny jako hybridní tj. obsahují prvky všech základních typů. Podnikovou síť s 1 000 počítači nelze vytvořit a spravovat bez vytvoření síťové hierarchie; zároveň minimálně farma serverů firmy musí splňovat atributy bezpečné sítě. Redundance médií pomocí technologií Ethernet a Wifi a samozřejmě již zmíněná redundance cest u distribuční a páteřní sítě mohou být pro zachování správného chodu sítě v kritických chvílích nezbytné.

Realizace distribuční sítě

Mezisíťová komunikace může s sebou přinášet vysoké nároky na přenosovou kapacitu fyzické vrstvy sítě; zároveň musí být hardware dostatečně výkonný na to, aby co nejrychleji zpracoval hlavičky paketů a předal je dále (propustnost). Zařízení obsahují zejména vysokorychlostní porty a celková přenosová kapacita zařízení (tedy ze všech portů) se pohybuje nad 100 Gb/s. Příkladem může být např. řada přepínačů (switchů) Cisco Catalyst 4500 nebo Catalyst 6500, doporučovaná pro distribuční (řada 6500 páteřní) sítě malých a středních firem, s kapacitou 136 Gb/s pro řadu 4500 a 720Gb/s pro řadu 6500.

1.2 Připojení uživatele k síti

Připojení uživatele k síti je úkolem přístupové sítě. Každá síť je koncipována tak, aby umožnila co možná nejjednodušší možnost rozšíření; k tomu patří i dostatečný počet portů na přepínači nebo směrovači. Zařízení mívají proto vyšší počet portů

nižších rychlostí (FastEhternet) a celkově není vysoký ani jejich výkon, s čímž souvisí také nižší pořizovací náklady. Příkladem může být např. switch Cisco Catalyst z řady 2960 nebo 3560.

Vlastní připojení uživatele do sítě je z uživatelského hlediska jednoduché, z pohledu operačního systému a sítě se jedná o netriviální problém spojený s ověřováním uživatele, který požaduje dostupnost mnoha služeb (HTTP, FTP, DNS, multimediální přenosy). Při ověřování uživatele (autentizaci) se často používá uživatelské jméno a heslo, pro ověření počítače (nebo jiného síťového prvku) jednoznačná identifikace jeho síťového adaptéru – MAC adresa. Selhání uživatele (počítače) při autentizaci může vést k odmítnutí poskytnutí některých síťových služeb, nebo dokonce k úplné blokaci portu na zařízení, ke kterému je uživatel připojen (jedná se o systém port security). Cisco IOS umožňuje např. definovat konkrétní MAC adresy pro jednotlivé porty nebo maximální počet MAC adres, které je možné registrovat na daném portu).

Pokud je uzel ověřen, zajišťuje přístupová síť pomocí aplikačních protokolů další služby nutné pro správné směrování paketů k uzlu. Příkladem takových protokolů jsou DHCP (protokol pro přiřazovaní IP adres) nebo protokoly pro překlad adres (NAT protokoly). Přístupová vrstva musí zároveň umožnit přístup do virtuálních sítí (VLAN).

Při navyšování počtu uživatelů či počtu síťových aplikací je nutné zajistit dostatečné množství síťových zdrojů (přenosová kapacita, výkon zařízení). Nejdražší variantou je výměna starých zařízení za nové a výkonnější.. Pokud to umožňuje technologie zařízení, je možné doplnit zařízení o nové zásuvné moduly (modulární přepínače), nebo zařízení agregovat (tj. spojit starší zařízení s novým) a zvýšit tak počet spojů mezi jednotlivými uzly.

1.3 Možnosti řízení

Síťové zařízení lze řídit několika způsoby. Přímý přístup získává uživatel pomocí připojení ke konzoli a přes port AUX (přídavný port, z anglického auxiliary; slouží pro připojení uživatele k zařízení pomocí modemu). Podle [9] umožňují Cisco směrovače připojení pomocí konektorů RJ-45 a DB-25 (verze DCE a DTE). Pro přístup ke konzoli doporučuje Cisco pro svá zařízení základní konfiguraci terminálu, která se nijak neliší od nastavení jiných zařízení (rychlost 9600 baudů bez parity).

Velmi rozšířeným v praxi je vzdálené připojení pomocí aplikačních protokolů secure shell (SSH) a telnet (pojednáno v kapitole 1.5). Komunikace přes vzdálené připojeníje vedena na rozdíl od konzolového připojení souběžně s ostatním síťovým provozem a často vyžaduje vyšší míru zabezpečení (zajišťuje šifrování komunikace u SSH). Z tohoto důvodu je telnet často pro účely konfigurace síťových zařízení zakázán. O protokolech telnet a SSH bude pojednáno níže.

1.4 Bezpečnost

Otázka bezpečnosti je složitá a řeší se na různých úrovních sítě různě. U páteřní sítě je velmi důležitá rychlost, a velký význam má také objem přenesených dat, proto se klade velký důraz na jejich kvalitu (tj. zabezpečení přenosu dat a jejich a bezchybnost) a podstatné je také, od koho data směřují. Z tohoto pohledu se setkáme s největší benevolencí v síti přístupové. Zde se uplatňují např. zmíněné filtrace MAC adres. V distribuční síti je kladen důraz na filtrování paketů podle aplikačních portů nebo aplikace kvality služeb (QoS).

1.4.1 Mechanizmy zabezpečení páteřních sítí

Zabezpečení sítí hraje důležitou roli v konfiguraci sítě obecně. Protože se práce zabývá konfigurací páteřních sítí, je zaměřena tato kapitola na zabezpečení právě tohoto typu sítí.

SNMPv3 protokol

Protokol pro správu sítě (simple network management protokol), je určen pro monitoring a řízení síťových zařízení a pro řízení konfigurací, výkonu a zabezpečení. Přístup k zařízením zajišťuje kombinací autentizace a šifrování paketů, kontroluje se také integrita zpráv. Maximální možnou ochranu SNMP3 paketů je možné docílit hashováním s algoritmem MD5 s šifrovacím algoritmem DES (56b klíč).

Received ACL

Mechanizmus chrání procesor a sdílenou paměť od zpracovávání nevynuceného síťového provozu (např. pakety nezabezpečených nebo nepoužívaných protokolů). Využívá základních a rozšířených access listů pro filtraci paketů z nevyžádaných sítí (tj. sítí, se kterými nechceme komunikovat) či aplikačních a síťových protokolů. Zajímavou vlastností ACL je monitoring sítě a omezování příchozího provozu na základě zatížení odchozích portů nebo časové omezení přístupu.

Control plane policing

Technologie Control plane policing (CoPP) přináší zlepšení access listů. Zatímco ACL definuje filtraci paketů pouze na základě splnění požadavků pro propuštění či zahození, CoPP definuje jednotlivé třídy filtrování jako obdobu QoS, kdy uživatel (správce) definuje společně se třídou filtrování i maximální dostupnou kapacitu systémových prostředků, které směrovač poskytne pro síťový provoz dané služby (rate-limit). CoPP je ochranou proti útokům DoS (Denial of Services = odepření služby). Náročnost pro zpracování procesorem je srovnatelná s ACL a filtrace

paketů je citlivější. Cisco technologie CPP umožňuje rozšíření CoPP o možnost filtrace na základě informací z hlavičky jednotlivých síťových protokolů.

Další možnosti zabezpečení

Všechny moderní směrovací protokoly podporují zabezpečení pomocí algoritmu MD5 [1] mezi jinými protokol BGP. Identifikace a verifikace zdroje je v této oblasti velmi důležitá. Síťová zařízení algoritmus MD5 plně podporují, proto se používají v BGP pro autentizaci peerů.

Další je ochrana proti paketům z nealokovaných adres (bogon), které mohou obsahovat nebezpečná data. Problém této metody spočívá v ručním zadávání povolených adres a synchronizaci s aktuální databází alokovaného adresního prostoru. Možností je také zakázat určitý prefix adres, ovšem to může způsobit vážný problém s dostupností služeb pro všechny uživatele.

Zabezpečení páteřní sítě je realizováno též ověřováním administrančních systémů. V rámci doporučení Cisco Systems, Inc. je součástí zabezpečení sítí blokování paketů z neověřených administrančních systémů a blokování paketů s cestami (pakety AS_PATH), které pocházejí z externích BGP uzlů (eBGP, od verze 12.0.26S).

Mezi další konfigurace BGP patří Route Flap Damping (RDF). Při přeposílání nejrůznějších aktualizací databází protokolů dochází k přechodným špičkám a nerovnoměrnému zatěžování tras (protokoly neposílají update periodicky). Zmíněná technika umožňuje snižování těchto špiček tak, aby se zvýšila efektivita využití sítě. U páteřních sítí je předpokladem, že router musí být zabezpečen proti náhlým změnám a musí být schopen čelit velkému vytížení při směrování, přičemž se nelze spoléhat na ostatní routery a jejich zabezpečení. Alternativou k RFD je používání časovačů při přeposílání aktualizací u jednotlivých protokolů, ale to má za následek zhoršení konvergence celé sítě. V praxi se používá kombinace obou metod, časovače jsou nastaveny s malou periodou, pro představu v BGP-4 30 s.

Mezi další způsoby zabezpečení páteřní sítě patří např. omezení doby TTL, maximální zabránění přístupu do sítě (techniky pro „zneviditelnění“ uzlů) a pravidelné hlídání konfigurace uzlů sítě.

1.5 Vzdálená správa zařízení

Vzdálená správa zařízení poskytuje správci sítě mocný nástroj pro jejich konfiguraci. Z jednoho místa tak lze prakticky ovládat připojení uživatelů na celém světě. V komerčním sektoru umožňuje vzdálená správa ušetřit na lokálních podporách, neboť není nutné být fyzicky u zařízení. Nevýhoda tohoto spravování vyplývá z jeho podstaty. Pokud dojde k přerušení komunikace poškozeného zařízení s okolím (např. vinou špatné konfigurace), nelze aplikovat rychle a účinně

žádnou formu řešení. V takových případech se ukazuje, jak kvalitně je síť realizována (redundance, záložní systémy, nastavení klíčových uzlů sítě pro jednotlivé směrovací a aplikační protokoly apod.). Pro vzdálenou správu se nejčastěji využívají 2 aplikační protokoly – telnet a SSH.

1.5.1 Telnet

Telnet je protokol TCP/IP modelu, který se používá pro vzdálené připojení k entitám, které mají přiřazenu IP adresu a je na nich spuštěn telnet server. Telnetové připojení je z uživatelského pohledu stejné jako standardní konzolové připojení z terminálu, definuje tzv. síťový virtuální terminál (NVT). Uživatel se jako klient připojuje k serveru telnet, který je spuštěn na operačním systému na opačném konci komunikační linky.

Telnetové spojení využívá TCP transportní protokol (streamovaný protokol) a je založeno na 3 hlavních ideích [21] a [8] – koncept virtuálního terminálu, princip nastavení možností, shodný pohled z terminálu a procesu. Koncept pochází ze standardu RFC 854 z roku 1983.

Síťový virtuální terminál

NVT je komunikačním prostředkem telnetu. Klientská aplikace na terminálu přitom zodpovídá za správnou interpretaci příchozích NVT kódů na display. NVT využívá 7b kódů pro zasílání standardních znaků (základní tabulka ASCII). Poslední nejvýznamnější bit je nastaven na nulu. Konec řádku je přenášen jako znak CR (carriage return – vrací na začátek řádku) společně se znakem LF (line feed – posouvá kurzor na další tisknutelný řádek). Abecedu znaků, které používá NVT, využívají i protokoly SMTP nebo FTP.

Protokol používá sadu příkazů pro řízení spojení. Identifikace těchto příkazů je dána nejvýznamnějším bitem, který je nastaven na 1. Z příkazů uvedených ve zdroji [8] lze vybrat například příkaz Go ahead (kód 249), který výzývá druhou stranu k přenosu (pro poloviční duplex), určitě příkaz Do (kód 253), který indikuje žádost související s následujícím kódem. Jednotlivé příkazy jsou označeny v dokumentaci jako IAC znaky (interpret as command characters).

Možnosti nastavení

Příkazy pro nastavení telnetu udávají nastavení komunikace obecně a mohou být upravovány kdykoli během komunikace. Na rozdíl od specifikace RFC 854 má každé nastavení svůj standard. Nejdůležitějšími jsou asi „suppress go ahead“ a „echo“.

Suppress go ahead je příkaz, který povoluje duplexní komunikaci (full duplex). Telnet byl navržen v době, kdy byl pro komunikaci běžně využíván poloviční duplex (half duplex). Dnes je proto tato volba nastavena jako aktivní.

Příkaz echo znamená, že všechny příchozí znaky jsou posílány zpět k odesílateli a umožňuje tak uživateli vidět, jaká data vlastně přes terminál poslal. U poloviční duplexu se odesílají znaky až po dokončení celého řádku, plný duplex posílá znaky jednotlivě, zatímco znaky z echa přijímá.

Pokud je zapotřebí změnit konfiguraci komunikace, posílá jedna z komunikujících stran 3B paket, obsahující IAC, typ operace (zda se má něco udělat či zakázat) a konkrétní operaci. Paket může být v případě potřeby doplněn o další bajty souvisejícími s operací definovanou ve třetím bajtu paketu.

1.5.2 Secure shell

SSH je velmi jednoduše řečeno zabezpečená forma telnetu. Umožňuje přenášet data zabezpečeným kanálem mezi dvěma zařízeními. Jeho implementace se liší v jednotlivých systémech. Původně byl SSH protokol používán v unixovských systémech, kde sloužil a slouží k zabezpečení účtů. Měl nahradit telnet a další nezabezpečená připojení. Implementacemi v operačních systémech v dnešní době jsou např. PuTTy a OpenSSH podporované platformami Windows, Linux a Mac OS.

SSH se vyskytuje ve 2 verzích. První secure shell (SSH-1, SSHv1) vymyslel Tatu Ylönen (Finsko) v roce 1995 s úmyslem nahradit jím rlogin, telnet a rsh protokoly. Díky volnému šíření (SSH byl od začátku freeware) se SSH stalo brzy základním bezpečnostním prvkem mnoha aplikací. Druhou verzi již vyvíjela skupina z oganizace IETF. Nový standard vznikl v roce 1996 a není zpětně kompatibilní s SSHv1. Při revizi došlo k několika významným změnám v architektuře (SSHv2 odděluje přenos, autentizaci a spojování ve zvláštních protokolech), integrita dat se kontroluje pomocí kryptování, lze měnit heslo. Rozšířil se počet podporovaný autentizačních mechanizmů. SSH podporuje autentizaci s využitím asymetrických (RSA) a symetrických klíčů (PuTTy umožňuje použití klíčů AES, Blowfish, 3DES a staré šifrování DES a Arcfour), dále podporuje použití veřejných certifikátů. Klíč relace se periodicky obměňuje [1] . Internetovým standardem se stalo SSH až v roce 2006. V současné době zahrnuje SSH více než 10 RFC dokumentů.

Ačkoli se zdá varianta SSH ideální, nemá její použití vždy smysl. Pokud například probíhá komunikace po privátní síti, která je považována za dostatečně chráněnou proti útokům, není třeba vytvářet oproti telnetu složitější komunikaci, která zahrnuje vytvoření a výměnu klíčů a následné šifrování a dešifrování symetrickými klíči.

1.6 Autentizace uživatele

Autentizace znamená ověření, zda v komunikaci vystupuje ten uživatel, za kterého se komunikující osoba vydává. Patří společně s autorizací a účtování do služeb AAA

[3] Autentizační proces zahrnuje minimálně tři strany. Žadatele o autentizaci (klient), příjemce žádosti o autentizaci (server) a důvěryhodnou autoritu s databází oprávněných žadatelů.

Při nastavování autentizace se musí nejprve nastavit celý mechanizmus AAA. U mechanizmu AAA na směrovači je důležité pořadí metod, které má zařízení používat. Pokud dojde při užívání jedné metody k chybě, použije směrovač automaticky další metodu v pořadí. Totéž platí pro komunikaci s jednotlivými autentizačními (a jinými) servery. Protokoly RADIUS a TACACS+ umožňují servery řadit do skupin [20] .

Systémy IOS podporují většinou 4 typy autentizace – lokálně pomocí záznamů v zařízení a dynamicky pomocí protokolů Kerberos, Radius nebo Tacacs+ (v textu budou zmíněny i další metody, které většinou podporují prvky sítě se specifickou funkcí např. firewally). Metody autentizace se aktivují pomocí příkazu „aaa new-model“. Dále je třeba nastavit používaný autentizační protokol (viz níže). Následuje výčet, na jaké příkazy a přístupy se bude autentizace uplatňovat. Vytvoří se seznam metod pro typ autentizace a následně se tento seznam přiřadí lince či síťovému rozhraní zařízení. Detaily práce se seznamy lze nalézt ve zdroji [20] .

1.6.1 Autentizace pomocí lokálních záznamů

Jedná se o autentizaci pomocí databáze, která je uložená lokálně na zařízení. Výhodou je, že nevyžaduje žádný externí server, který musí být v činnosti. Zároveň je však velmi náročné udržovat tento způsob autentizace u více zařízení (u rozsáhlých globálních sítí je to absolutně nemyslitelné). Lokální autentizace by měla sloužit jako záložní, když všechny ostatní metody selžou – je tak zaručeno, že se správce zařízení k ovládání směrovače či přepínače vždy dostane.

Konfigurace lokální databáze

Pro přidání uživatele s určitými právy (0-15 od základních po práva administrátora) stačí napsat v globálním konfiguračním režimu zařízení příkaz „username *jméno uživatele* privilege *úroveň přidělených práv* password *heslo*“.

1.6.2 Kerberos

Systém Kerberos byl vytvořen v projektu Athena (USA, 1983), jeho tvůrci byly Jennifer G. Steiner, Clifford Neuman a Jeffrey I. Schiller. Byl zveřejněn v roce 1988 a jeho poslední verze pochází z roku 2005 (verze 5, RFC 4120 [22]). Jedná se o systém zajišťující autentizaci a autorizaci. Pro zabezpečení komunikace využívá symetrických klíčů (DES).

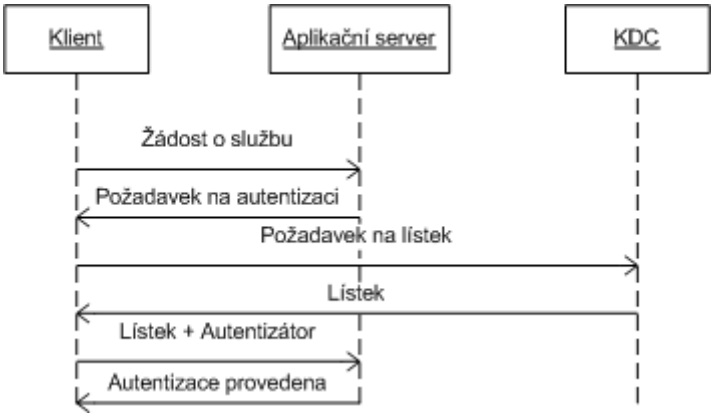
Pro pochopení funkce systému Kerberos jsou v následujícím seznamu vysvětleny pojmy, které se v Kerberosu vyskytují nejčastěji [24] :

- Centrum pro distribuci klíčů (KDC, Key Distribution Center, někdy označován jako Kerberos server) – jádro systému Kerberos. Jeho úkolem je přihlašování uživatelů. V síti se může vyskytovat více KDC serverů (jeden hlavní (master) a podřízené (slave) servery). Jeho součástí je databáze všech uživatelů, kteří mohou vstupovat do autentizačního procesu, a jejich klíčů.
- Aplikační server – server, který pro poskytnutí služby vyžaduje ověření uživatele.
- Klient – libovolný počítač, který žádá o lístky. Většinou se jedná o aplikaci spuštěnou na počítači v síti.
- Lístek (ticket) – zašifrovaný blok sloužící k autentizaci v systému Kerberos.
- TGT (Ticket Granting Ticket – "lístek na lístky") – speciální typ lístku umožňující získávání dalších lístků.
- TGS (Ticket Granting Service – služba pro vydávání lístků) – server pro vydávání TGT, součást KDC.
- Principál - řetězec, který jednoznačně identifikuje uživatele nebo službu.

Kerberos je založen na běžné formě komunikace klient-server se zprávami dotaz a odpověď. Dotaz se skládá z lístku a autentizátoru. Autentizátor obsahuje mj. aktuální čas, kontrolní součet a šifrovací klíč (volitelně), vše zašifrované klíčem relace. Během komunikace dokazuje klient serverům, že zná číslo relace obsažené v lístku. Při přijímání žádosti verifikátor (aplikační server) dešifruje lístek pomocí svého klíče a pomocí klíče relace dešifruje autentizátor. Pokud souhlasí kontrolní součet, považuje verifikátor autora autentizátoru za uživatele uvedeného v lístku, a tedy za vlastníka relačního klíče. K úspěšné autentizaci zkontroluje verifikátor navíc časovou značku, zda je ještě autentizátor platný. Pokud je časová značka přibližně shodná s aktuálním časem (rozdíl do 5 minut) a pokud tato značka dosud nebyla zjištěna u jiné žádosti, autentizuje verifikátor uživatele. Pokud je zapotřebí, aby si uživatel ověřil totožnost serveru, generuje server odpověď tím, že vyjme klientský čas z autentizátoru a vrátí autentizátor klientovi se všemi informacemi, vše zašifrované pomocí relačního klíče.

Uživatel potřebuje ke komunikaci s různými servery pokaždé jiný relační klíč. Pro získání lístku zasílá klient žádost o autentizaci (authentication request), ve které uvádí svou identifikaci, název verifikátoru, požadovanou dobu platnosti a libovolně zvolené číslo kvůli identifikaci odpovědi. V odpovědi posílá autentizační server relační klíč, přiřazenou dobu expirace, identifikátor odpovědi (číslo extrahované ze žádosti), jméno verifikátoru a další informace z lístku, vše zašifrované pomocí hesla uživatele registrovaného na autentizačním serveru. K tomu patří i lístek obsahující podobné informace, který je určen pro verifikátor jako součást žádosti o aplikaci. Zmíněné 4 zprávy tvoří základ protokolu Kerberos.

Z bezpečnostního hlediska není ideální, pokud uživatel používá k získání lístků neustále heslo. Z tohoto důvodu podporuje Kerberos systém SSO (single sign-on). Jedná se o způsob přihlášení, který umožňuje uživateli zadat své přihlašovací údaje pouze jednou, a poté již využívat jiné formy autentizace. U Kerberosu se toto zajišťuje pomocí tzv. credentials (dokladů, osvědčení). Když se uživatel poprvé přihlašuje, místo standardní autentizační odpovědi mu zašle autentizační server lístek a relační klíč pro službu vydávání lístků. Doba platnosti takového lístku (lístek pro vydávání lístků) se pohybuje v hodinách (výchozí hodnota 8 h). Pokud potřebuje uživatel získat nový relační klíč, použije se pro šifrování odpovědi z autentizačního serveru relační klíč pro TGT.



Obr. 1: Kerberos autentizace

Důležitou součástí autentizace je autentizace uživatele ke klientovi (často hraniční směrovač). Uživatel otevře PPP připojení ke směrovači a posílá mu uživatelské jméno a heslo. Směrovač následně ověří pomocí dotazu do KDC (dotaz o TGT). Odpověď od KDC je zašifrovaná pomocí klíče vytvořeného z uživatelského hesla a obsahuje identitu uživatele. Směrovač se pokusí rozbalit odpověď pomocí hesla zaslaného uživatelem a při úspěšném dekódování uživatele autentizuje.

Konfigurace systému Kerberos v systémech IOS

Zdroj [24] popisuje výše uvedený proces. Pro konfiguraci je podle [14] nejprve nutné vytvořit záznamy o uživatelích v KDC. Na směrovači je pak nutné definovat sféru serveru Kerberos (příkaz „kerberos local-realm *sféra*“). Dále se definuje server pro tuto sféru pomocí IP adresy nebo názvu serveru (příkaz „kerberos server *sféra doménový název/IP adresa port*“). Volitelně lze ještě namapovat DNS doménu na danou sféru (příkaz „kerberos realm *doména/název sféra*“).

Konfigurace pokračuje importováním souboru se SRVTAB záznamy (tento záznam musí mít každé zařízení používající protokol Kerberos), které obsahují mimo jiné hesla či vygenerované klíče pro všechny uživatele vstupující do KDC procesu. Tyto klíče musí být sdílené, aby bylo možné autentizaci provozovat.

Soubor obsahují všechna zařízení v dané sféře. Soubor se přenáší pomocí příkazu „kerberos srvtab remote *IP/název serveru název souboru*“.

1.6.3 RADIUS

Radius (Remote authentication dial in user service) je v současné době asi nejpoužívanějším systémem pro autentizaci z vnější sítě (např. virtuální privátní sítě nebo bezdrátové sítě). Uživatel, který žádá o danou službu, musí v tomto případě komunikovat s hraničním prvkem vždy (jedná se o agenta, také NAS – network access server), který vystupuje v protokolu jako jeho zástupce (klient v komunikaci se serverem RADIUS). Stejně jako Kerberos používá RADIUS transportní protokol UDP (pro autentizaci standardně port 1812). Pokud první spojení s primárním RADIUS serverem selže, připojuje se klient ihned k sekundárnímu serveru).

Použití RADIUSu lze uplatnit podle [15] v sítích, které obsahují prvky různých technologií (RADIUS je velmi rozšířen). RADIUS lze provázat např. se zabezpečením Kerberos u uživatelů s vytáčeným připojením. RADIUS podporuje stejně jako Kerberos použití smart karet. Narozdíl od Kerberosu podporuje účtování (doba připojení, množství stahovaných dat). RADIUS server může sloužit jako proxy server. Podporuje autentizaci uživatele pomocí PPP protokolů (PAP, CHAP, EAP), SLIP, ARAP (AppleTalk remote access protocol) a dalších [23] .

Uživatel v systému RADIUS je požádán o vložení uživatelského jména a hesla, která jsou poslána do RADIUS serveru, který odpoví jednou z následujících zpráv:

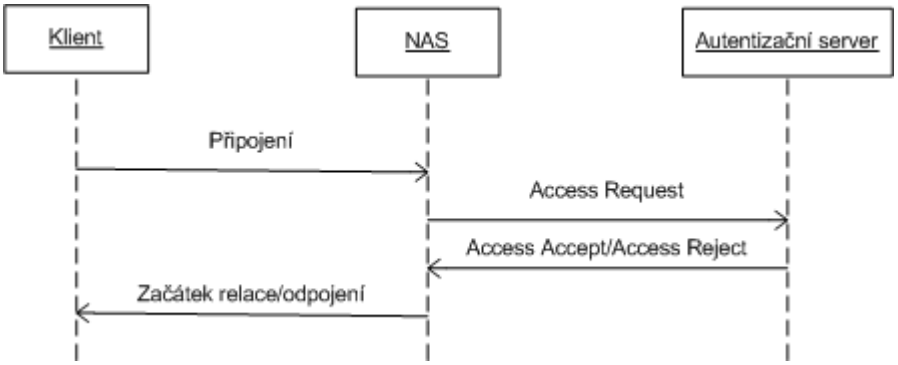
- Accept – uživatel je autentizován.
- Reject – uživatel není autentizován a je vyzván k zopakování jména a hesla, nebo je odmítnut.
- Challenge – RADIUS server požaduje další informace od uživatele.
- Change-password – uživatel je žádán o změnu hesla.

RADIUS paket se skládá z kódu zprávy (1 B), identifikátoru zprávy (1 B), délky (2 B), pole autentizátoru (16 B) a atributů. Klient vytvoří zprávu Access-Request, která obsahuje uživatelské jméno a heslo. Protokol nespecifikuje proces generování zpráv; standardně se do pole identifikace zprávy vkládá inkrementované číslo poslední zprávy. V poli autentizátoru je 16B řetězec jako jednoznačný identifikátor. Paket Access-Request je nezabezpečený až na pole s uživatelským heslem. To je chráněno pomocí sdíleného tajemství mezi klientem a serverem. Tento klíč společně s autentizátorem je pomocí hashe (algoritmus MD5) zašifrován a projde společně s heslem funkcí XOR. Pokud je heslo delší než 16 B, pak je operace XOR opakována, ale MD5 hash je tentokrát vytvořen z tajemství a

šifrovaného textu z předchozí operace). Všechny zašifrované bloky jsou poté přeneseny v atributu zprávy jako uživatelské heslo.

Server přijme paket a ověří, zda server vlastní sdílené tajemství s klientem. Pokud ne, je paket zahozen. V opačném případě server ověří, zda uživatelské jméno a heslo souhlasí. Pokud ano, posílá zprávu Access-Accept zpět, v opačném případě posílá Access-Reject. Obě zprávy musí mít v poli identifikátoru stejnou hodnotu. Do pole autentizátoru vkládá server MD5 hash zřetězené předchozí zprávy a sdíleného tajemství.

Jakmile klient obdrží odpověď, porovná identifikátor s poslední odeslanou zprávou. Pokud se ID liší, pak zprávu zahazuje, v opačném případě vypočítá hash stejným způsobem jako server a porovná pole autentizátoru. Pokud je vše v pořádku, uživatel je považován za autentizovaného.



Obr. 2: RADIUS autentizace

RADIUS PROXY

RADIUS umožňuje komunikaci pomocí proxy serverů. Funkce spočívá v přeposílání dotazů a odpovědí mezi klientem a serverem. Základním použitím je roaming. Pomocí proxy je umožněno autentizovat uživatele i mimo „domovskou" administrativní oblast.

Klient posílá žádost na forwarding (přeposílací) server, který přepošle žádost na vzdálený server. Vzdálený server odešle odpověď na forwarding server, který ji přepošle k NAS. Atribut s uživatelským jménem může obsahovat identifikátor přístup do sítě (Network Access Identifier) pro operace RADIUS proxy. Rozhodnutí, který server obdrží přeposlaný dotaz, by mělo být realizováno na základě sféry autentizace, která by měla být součástí identifikátoru. Alternativně lze definovat jiné kritérium (například číslo sféry).

RADIUS server může fungovat jako forwarding server i jako vzdálený server pro různé sféry. Počet serverů, na které lze odesílat, je neomezen, servery lze libovolně řetězit. Jediným problém je riziko přeposílací smyčky, která může vzniknout.

Forwarding server pracuje s atributy proxy od jiných serverů jako s transparentními daty. Jeho činnost jimi nesmí být ovlivněna. Pokud byly

v klientské žádosti atributy proxy, musí forwarding servery přidat tyto atributy k odpovědi klientovi. Forwarding server může vložit atributy proxy do žádosti, pokud ji přeposílá, popřípadě je může vynechat. Pokud atributy vynechá při přeposílání dotazu, musí je vložit při přeposílání odpovědi zpět.

Komunikace RADIUS s využitím forwarding serverů (podle [23])

NAS pošle dotaz k forwarding serveru. Server dešifruje heslo (nemusí být přítomno) pomocí sdíleného klíče. Pokud zpráva obsahuje atribut CHAP-Password (při komunikaci přes protokol CHAP – atribut obsahuje uživatelské jméno CHAP, identifikaci CHAP a heslo pro CHAP) a zároveň neobsahuje atribut CHAP-Challenge (s identifikací výzvy CHAP), nesmí změnit obsah pole autentizátoru. Zároveň může být zpráva doplněna o atribut proxy-state (maximálně 1 atribut), kterými se ostatní forwarding servery nemusí řídit, ovšem nesmí jej mazat ani změnit. Server zašifruje heslo pomocí klíče, které sdílí se vzdáleným serverem, nastaví identifikátor a pošle zprávu ke vzdálenému serveru.

Pokud je vzdálený server konečnou destinací pro zprávu, ověří uživatele pomocí uživatelského hesla (nebo hesla CHAP či podle použité metody) a vrací zpět zprávu o přijetí, odmítnutí či žádost o další data. Posílá zpět odpověď společně se všemi proxy atributy zkopírovanými ze žádosti.

Forwarding server stejným způsobem ověří identifikaci vzdáleného serveru a odstraní poslední atribut proxy-state (pokud jej v dotazu do zprávy vložil). Dále zašifruje autentizátor pomocí klíče sdíleného s klientem a zprávu mu pošle.

Konfigurace systému RADIUS v systémech IOS

Cisco IOS podporuje globální nastavení časovače, opětovného posílání a sdíleného klíče pro všechny RADIUS servery. K tomu slouží příkazy „radius-server timeout“, „radius-server retransmit“ a „radius‑server key“. Pro konfiguraci RADIUS serveru se používá příkaz „radius-server host *název/IP adresa*“ s volitelnými parametry pro autentizační port (auth-port), účtovací port (acc-port), alias a sdílený klíč (key). RADIUS poskytuje mnoho možností, které ale nejsou vyžadovány (pomocí mnoha typů atributů). Lze tak například definovat nové atributy pro danou technologii. V systému IOS lze například nastavit míru „skrytí“ sdíleného klíče (0 pro volný text v konfiguraci, 7 pro skrytý).

1.6.4 TACACS+

TACACS+ (Terminal access controller access control system) je v současné době poslední verze protokolu TACACS a proprietární protokol firmy Cisco, který nikdy nebyl standardizován (zůstal pouze jako RFC předloha v ukončenou platností v roce 1998 [1]). Jedná se o protokol pracující stejně jako RADIUS nebo Kerberos

na modelu klient-server, ale narozdíl od předešlých využívá na transportní vrstvě protokolu TCP. Princip, na kterém funguje, je podobný jako u protokolu RADIUS. Opět zde funguje komunikace NAS a autentizačního serveru. Narozdíl od dřívějších verzí (TACACS a XTACACS) a protokolu RADIUS odděluje server TACACS+ striktně všechny tři funkce AAA (u protokolu RADIUS jsou pevně spojeny autentizace s autorizací). To umožňuje velmi dobře kombinovat systém s Kerberosem (Kerberos pro autentizaci a TACACS+ pro autorizaci a účtování). NAS se autentizuje u Kerberosu a může si pak vyžádat autorizaci u TACACS+ serveru bez nutnosti nové autentizace. NAS informuje server, že byl úspěšně autentizován od Kerberosu a server poskytne NAS informace týkající se autorizace. TACACS+ používá stejně jako RADIUS šifrování pomocí MD5 hashů, ale zatímco RADIUS šifruje pouze heslo a ostatní části zprávy zůstávají otevřené, protokol TACACS+ šifruje zprávu celou (v hlavičce paketu je indikuje šifrování vyhrazené pole). Výhodou TACACS+ systému je podpora více platforem oproti systému RADIUS podle [22] se jedná o protokoly ARA (AppleTalk Remote Access), NetBios Frame Protocol Control, NASI (Novell Asynchronous Services Interface) a X.25 PAD connection.

Jelikož se RADIUS a TACACS+ svým chováním liší, jsou odlišné i komunikace mezi klientem a serverem. Komunikace protokolu probíhá na portu 49 (rezervován pro TCP i UDP komunikaci). Hlavičku protokolu tvoří 12 B - jedná se o pole hlavní a vedlejší verze protokolu (4 bity + 4 bity), typu (1 B), sekvenčního čísla (1 B), značky (flag, 1 B), identifikace relace (4 B) a délky (4 B). Hlavní verze má číslo 12, vedlejší číslo je vyhrazeno pro nové revize protokolu. Standardně má hodnotu 0, ostatní čísla mohou být použita pouze, pokud to daná služba vyžaduje. Typ označuje službu (1 pro autentizaci, 2 pro autorizaci a 3 pro účtování). Sekvenční číslo je číslo paketu pro danou relaci, vždy musí číslování začít od 1, další pakety mají vždy hodnotu inkrementovanou o 1. Klient tak posílá pouze „liché" pakety a server „sudé". Pokud komunikace dosáhne bodu s posledním dostupným číslem paketu, relaci je nutno restartovat a vést od čísla 1. Značky obsahují mj. informaci o šifrování obsahu zprávy, pokud je značka nastavena na 1, pak k šifrování nedochází. Nešifrovaná komunikace slouží zejména pro testování a není doporučená pro běžné použití. identifikace relace přiřazuje komunikaci jednoznačné číslo. Volba tohoto čísla by měla být náhodná. Zde je potenciální problem protokolu, jelikož míra náhodnosti generování ovlivňuje zabezpečení [22] . Pole délky obsahuje délku celého pole s uživatelskými daty (tj. bez záhlaví paketu).

K šifrování datové části, jak již bylo zmíněno, slouží šifrovací algoritmus MD5. Konečný kód je dán zřetězením 16b hashů MD5. První 16b hash je vytvořen z polí číslo relace, klíč, verze a číslo sekvence. Další, obdobně jako u protokolu RADIUS, vznikne spojením již zmíněných polí a předchozího hashe.

Standard TACACS+ definuje 3 typy paketů pro autentizaci. Klient odesílá pakety „start" a "continue" a třetím paketem je „reply" jako odpověď od serveru. Autentizace začíná tím, že klient odešle zprávu „start". Tato zpráva popisuje typ autentizace a může obsahovat uživatelské jméno a další autentizační data. Smí být odeslána pouze jako první zpráva v relaci popř. po restartu relace (tj. vždy má identifikační číslo 1). V odpovědi posílá server paket „reply" jako indikaci, zda byla autentizace dokončena, či zda pokračuje. V druhém případě reaguje klient odesláním zprávy „continue". Server musí na zprávy od klienta vždy odpovědět, jedinou výjimkou je, pokud klient ve zprávě „continue" zruší spojení. Tehdy se komunikace přeruší.

Konfigurace serveru TACACS+ na Cisco zařízení ([1])

Nejprve se nastavuje na směrovači globální sdílený klíč (nepovinné) pomocí příkazu „tacacs-server key *klíč*". Klíč se zadává v nešifrované podobě a může obsahovat mezery. Dále se definují autentizační TACACS servery pomocí příkazu „tacacs-server host *název*" s nepovinými parametry pro port, časovač a sdílený klíč. Výchozí hodnota portu je, jak již bylo zmíněno, 49 (TCP). Časovač slouží k nastavení době čekání na odpověď od serveru. Pomocí parametru „key" lze definovat specifický sdílený klíč.

1.7 Cisco IOS

Cisco IOS (Internetwork Operating Systém) je operační systém běžící na téměř všech směrovacích zařízeních (míněny směrovače, přepínače a firewally) firmy Cisco. Operační systém je na první pohled obdobou operačních systémů u počítačů, tentokrát však s tím rozdílem, že IOS poskytuje funkce pro účely síťového uzlu. Jedná se o směrování, přepínání, koordinaci v rámci sítě a telekomunikační funkce.

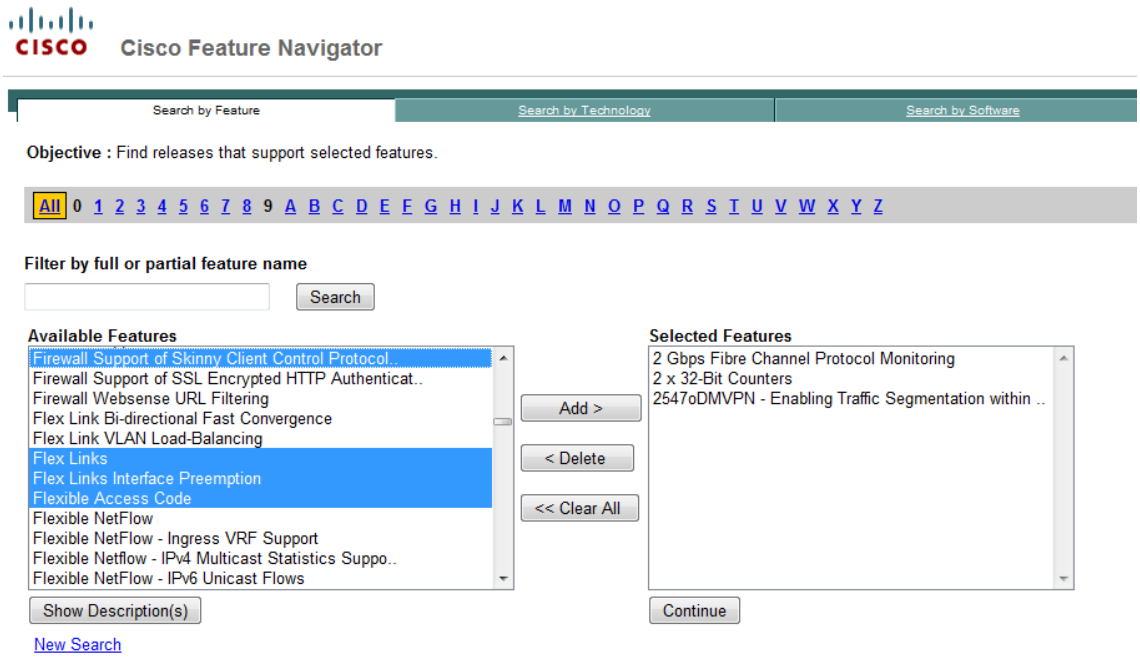
Komunikovatelnost se systémem je zařízena zejména prostřednictvím příkazového řádku CLI (command line interface). CLI podporuje stejně jako příkazové řádky u PC značné množství příkazů, s jednotlivými verzemi se liší i podpora.

Cisco nabízí několik grafických nástaveb, pomocí nichž lze snadněji ovládat některá nastavení. Příkladem může být například nástroj Cisco SDM (Security Device Manager), který je podporován od verze IOS 12.2(11), případně 12.3; závisí na typu zařízení. SDM slouží ke konfiguraci (W)LAN a WAN sítí na směrovači a konfiguraci zabezpečení [6] .

V případě konfigurace systémů IOS je užitečné použít nástroj IOS Feature Navigator. Cisco IOS Feature Navigator je aplikace, která je určena pro vyhledávání a zobrazení vlastností, které jsou k dispozici, a to pro jednotlivé verze systému IOS nebo hardwarové platformy. Navigator naviguje uživatele pomocí jednoduchých

obrazovek, aby uživatel získal informace o systému IOS, IOS XE nebo CatOS. Všechny tyto grafické nástroje využívají HTTP server, který je běžnou součástí IOSu na zařízeních.

Cisco IOS se vyskytuje v mnoha verzích, které se od sebe liší podporou protokolů a funkcí. Vzhledem k šíři výčtu všech vlastností nelze verze v této práci popsat; Cisco na svých stránkách umožňuje vyhledávat verzi IOSu s potřebnými vlastnosti pomocí nástroje Cisco Feature Navigator [12] , který je volně dostupný na internetu. Pomocí tohoto nástroje lze vyhledávat vhodný software pro všechna zařízení firmy Cisco podle různých kritérií (funkce, technologie) viz Obr. 3.



Obr. 3: Cisco Feature Navigator

1.7.1 Módy Cisco IOS

Se systémem IOS lze pracovat v několika základních módech. Základní přístupový mód se nazývá User exec mode. Je výchozím bodem pro vstup do privilegovaného režimu (Privileged exec mode), ze kterého se lze již bez jakéhokoli hesla přesunout do ostatních režimů (globální konfigurační režim – Global configuration mode, další konfigurační režimy pro jednotlivé oblasti nastavení). Všechny základní režimy jsou popsány níže.

User exec mode

User exec mode je výchozím módem pro práci se zařízením. V tomto režimu je možné provádět základní monitoring sítě ze zařízení (např. příkazy traceroute a ping), a také lze zde vysledovat základní vlastnosti nastavení zařízení (pomocí příkazů show – detaily o rozhraní (show interfaces), směrovací tabulka (show ip

route) a další). Samozřejmostí je nápověda. Množství získaných informací o nastavení zařízení je omezeno pouze na informace, které jsou považovány jako méně důvěrné (jejich šíření není považováno za bezpečnostní problém). Tento režim je běžně chráněn heslem a funguje jako první bezpečnostní bariéra pro případné útočníky. Zároveň je neoprávněný vstup do tohoto režimu hodnocen podobně jako neoprávněné vniknutí uživatele do počítače a porušení soukromí.

Privileged exec mode

Privileged exec mode je režim pro základní konfiguraci zařízení. Uživatel se do tohoto módu přepne příkazem enable, z uživatelského módu. Privilegovaný mód je druhou bezpečností zónou zařízení a jeho vstup je standardně stejně jako základního uživatelského módu taktéž zaheslován. Z tohoto módu se již lze pohybovat v nastavení zařízení bez použití hesel. IOS podporuje vytváření databáze uživatelů a hesel, kdy práva pro jednotlivé uživatele jsou oddělena podle privilegií. Zároveň lze vysledovat, kdy který uživatel na zařízení pracoval. Uživatel s přiřazenou úrovní privilegií (pro orientaci user exec mode je úroveň 1, privileged exec mode na úrovni administrátora má úroveň 15 = nejvyšší) může ovlivňovat pouze vymezenou oblast nastavení; vznikají tak skupiny uživatelů pro specifické oblasti konfigurace (např. pouze konfigurace určitých rozhraní). Celkem existuje 16 úrovní (0 nejnižší, 15 nejvyšší) [19] , [7]).

Možnosti práce v privilegovaném režimu jsou mnohem větší než u uživatelského módu. Všechny příkazy z User exec mode zde lze plně uplatnit. Navíc přibývají příkazy spojené se zobrazením detailů o hardware zařízení (show inventory), příkazy pro detailní monitoring sítě (příkazy debug, show logging) a příkazy spojené s prací s pamětí (copy, replace, format, erase) a mnoho dalších příkazů, spojených se základní funkcí směrovače (reload, reset). Obecně lze shrnout, že narozdíl od režimu user exec mode, který je určen zejména pro sledování, umožňuje privilegovaný mód změnu základních nastavení zařízení.

Global configuration mode

Tento režim je navazuje na předchozí, liší se možnostmi konfigurací, které je zde možné provádět. Na této úrovni se přistupuje ke konkrétním oblastem konfigurace, a tím je také možno tento režim dále větvit. Přes globální konfiguraci se přechází ke konfiguraci ACL, rozhraní, směrovacích protokolů, různých serverů typu NAT nebo DHCP atd. Přístup do konfiguračního režimu je možný přes terminál (standardní použití), paměť nebo síť.

Interface configuration mode

Interface configuration mode (režim konfigurace rozhraní) je jedním z mnoha, do kterých je možné se přepnout z předchozího režimu. Slouží, jak již z názvu vyplývá,

pro konfiguraci rozhraní zařízení tj. zařazení do LAN (IP adresa), VLAN (podporované sítě VLAN, mód přepínání) a samozřejmě základní ovládání jako aktivace a deaktivace portu. IOS podporuje i ovládání několika portů najednou (tzv. „interface range“) s omezenými funkcemi.

Setup mode

Tento režim slouží pro nastavení konfiguraci při spuštění zařízení. Vstoupit do něho lze pomocí příkazu setup z privilegovaného režimu, nebo při prvním spuštění (spuštění bez nastavení) zařízení. Umožňuje uživateli základní nastavení nutné pro ovládání (název routeru, přístupová hesla, nastavení protokolu SNMP, aktivaci rozhraní). Nevýhoda módu spočívá v tom, že neumožňuje našeptávání (pomocí „?“ a tabulátoru).

Tab. 1 uvádí vybrané příkazy, které lze uplatnit v jednotlivých módech. Dostupnost některých příkazů může záviset na verzi IOS (např. u příkazů „show logging“ nebo „show mac-address-table“) a na nastavení práv uživatelů v módech.

Rommon mode a boot mode

Režimy ROMmon (z angl. ROM monitoring – monitoring paměti ROM) a boot (spouštěcí) jsou nouzové režimy, které se používají pro obnovu IOSu nebo pro diagnostické účely. Jedná se o režimy s velmi omezenou a specifickou možností konfigurace. V režimu ROMmon lze konfigurovat pouze 1 port (tzv. management port), pomocí kterého lze zařízení připojit k TFTP serveru, ze kterého se stahuje IOS. Zařízení se dále restartuje a v konfiguračním režimu se dále konfiguruje podle potřeb uživatele.

Mód	Příkaz	Popis
User exec mode	Enable	Přepnutí do privilegovaného režimu.
	Login	Přepnutí uživatele.
	ssh/telnet	Vytvoření spojení pomocí ssh/telnet.
	traceroute/ping	Základní příkazy pro monitoring sítě.
	show inventory	Zobrazení fyzického inventáře zařízení.
	show version	Zobrazení detailu o HW a SW.
	show arp	Zobrazení ARP tabulky.
Privileged exec mode	show mac-address-table	Zobrazení tabulky fyzických adres připojených zařízení.
	show logging	Zobrazuje logovací informace.
	debug	Aktivuje funkce debuggeru (mnoho voleb).
	configure	Přepnutí do globálního konfiguračního režimu.
	reload	Restartuje zařízení.
	more	Zobrazí obsah souboru.
	mkdir/rmdir/dir	Vytvoření/odstranění/zobrazení souborů.
	format/erase	Formátování/vymazání systému souborů.
Global configuration mode	router	Aktivace směrovacího procesu.
	vtp	Nastavení protokolu vtp.
	spanning-tree	Nastavení protokolu STP.
	login	Nastavení zabezpečení při logování.
	errdisable	Nastavení sledování chybových vypnutí rozhraní.
Interface configuration mode	IP address	Nastavení síťové adresy rozhraní.
	channel-group	Přidání zařízení do rozhroní Ether-channel.
	standby	HSRP konfigurace.
	snmp	Změna SNMP parametrů.

Tab. 1: Dostupnost vybraných příkazů z jednotlivých režimů Cisco IOS

1.7.2 Verzování Cisco IOS

Tvar verze IOS se dá symbolicky napsat jako a.b(c)d [12] . Hlavní verzi označuje první číslo „a“, následuje vedlejší číslo „b“ a v závorce aktualizace verze. „D“ je vždy písmeno, je nepovinné a značí rozšíření hlavní (stabilní) verze – např. podporu nového hardware a nové možnosti systému. Stávající nejnovější hlavní verze Cisco IOS pro směrovače a přepínače je verze 15.1 (k dubnu 2010).

Systém IOS se vyskytuje v několika typech. Typ ED (z angl. Early Deployment), jindy označován jako „T“ verze přináší nové vlastnosti a podporu nových platforem a rozhraní pro stávající hlavní verzi. Tímto typem je označena většina vedlejších verzí. Typ GD (z angl. General Deployment) je používán pro označení konečné fáze hlavních verzí operačního systému a je určený pro zákazníka. Tyto verze používají v běžném provozu a jsou výsledkem spolupráce

zákazníků a firmy Cisco. Typ LD (Limited Deployment) označuje mezistupeň hlavní verze mezi počáteční komeční podobu (typ FCS – First Commercial Shipment) a finální podobou (GD). Posledním typem DF (deferred – odložený). V těchto verzích byly objeveny vážné nedostatky a není je po označení DF stáhnout z internetu, použití takového software se nedoporučuje.

1.7.3 Volba systému IOS

Systémy IOS se vyskytují vždy jako jeden soubor (většinou typu bin). Při výběru je nutné dbát na platformu, pro kterou je systém určen, a samozřejmě na nároky operačního systému.

Obrazy systému IOS se svými požadavky na hardware velmi liší. Je zapotřebí, aby pro systém IOS bylo dostatek flash paměti i dynamické paměti. Poslední verze (15) určená pro směrovače c2951 vyžaduje Flash o velikosti 256 MB a DRAM 512 MB, samotný obraz je velký cca 54 MB. Verze 12.4 v poslední podobě má velikost 10 MB a vyžaduje flash 12 MB a DRAM 64 MB.

K posuzování vhodnosti systému se také doporučuje posoudit rizika vybrané verze viz [1] . Na internetu lze pracovat se seznamem chyb nalezených u jednotlivých verzí IOS pomocí nástroje „bug tool“ (vstup pouze pro registrované uživatele). Stejným způsobem lze vyhledávat i nejrůznější poznámky k verzi.

1.8 Konfigurace síťových zařízení

1.8.1 Práce s konfiguračními soubory

Konfigurace zařízení se vyskytuje standardně minimálně ve dvou podobách. První je startup-config, který obsahuje konfiguraci, jež se načítá během startu z paměti NVRAM do systémové paměti RAM v podobě označované často jako running-config, se kterým se dále pracuje. Soubory jsou textové a tedy snadno čitelné. Z bezpečnostního hlediska je proto nutné šifrovat použitá hesla, která jsou nedílnou součástí souborů. Navzdory předpokladu se do konfiguračních souborů nezapisuje kompletní konfigurace zařízení. Síťové prvky mají svá výchozí (defaultní) nastavení, v konfiguracích se pak nacházejí pouze odlišné konfigurace od tohoto nastavení. Existují také příkazy, které se do konfiguračního souboru nezapisují nikdy (např. příkaz pro aktivaci rozhraní „no shutdown“).

Samotná práce s konfiguračními soubory je stejná jako práce s jakýmkoli souborem v počítači. Soubory je možné libovolně kopírovat, mazat a přesunovat; problémem může být dopad neopatrného zacházení, kdy se tyto soubory odstraní a směrovač přejde po restartu do výchozího „továrního“ nastavení. Postup při aplikování konfigurace je velmi jednoduchý. Soubor je vlastně skript, takže se do systému vkládají příkazy ve stejné podobě, jako kdyby je psal uživatel (funguje

například kopírování částí konfiguračních souborů a jejich vkládání do CLI). Při vkládání konfigurace může být problémem výchozí režim pro vkládání příkazů (často globální konfigurační režim), uživatel proto při vkládání příkazů musí dbát odezvy ze zařízení, zda byla konfigurace aplikována bez hlášení chyb. Zároveň je nutné sledovat odezvu systému v případě, kdy byla změněna verze operačního systému IOS (příkaz nebo jeho atribut nemusí již být podporován v podobě, která fungovala v předchozích verzích).

Nahrazení a obnovení konfigurace

Popis nahrazení a obnovení konfigurace je podrobně popsán ve zdroji [8] . Pro nahrazení a obnovení platí následující omezení:

- Pokud směrovač nemá dostatek volné paměti (tzn. jeho paměť je nižší než součet velikostí obou konfiguračních souborů), nelze uplatnit proces nahrazování.
- Některé příkazy konfiguračního souboru vztahující se na fyzická rozhraní (např. „interface ethernet 0“) nemohou být vykonány, pokud se na příslušném zařízení takové rozhraní nenachází.
- Ve specifických případech nelze uplatnit okamžité změny nastavení bez restartování směrovače (změna konfiguračního registru).

Konfigurační soubor lze jednoduchým způsobem archivovat. Samotné archivaci by mělo předcházet zálohování na lokální či vzdálené zařízení pomocí standardního vytvoření kopie. Configuracion Replace a Rollback umožňují automaticky ukládat konfigurace do archivu. Archivační soubory slouží jako kontrolní body pro konfigurace a mohou se uplatnit při obnově konfigurace. Archivace konfigurací poskytuje základní správu konfigurací – lze omezit počet archivačních souborů, k názvu konfiguračního souboru se připojuje číslo a často časová značka pro lepší orientaci správce zařízení. Cesta a název pro archivační soubory jsou dané předem (nebo je lze nastavit). Archiv lze ukládat buď do flash, další vnitřní a externí paměti či na servery jako (t)ftp nebo http.

Pro nahrazení aktuální konfigurace jinou slouží příkaz „configuration replace“. Lze jím vráti zpět konfiguraci do pevného archivovaného bodu, pokud je archivace aktivní. Příkaz „configure replace“ vyvolá porovnání obou konfigurací s tím, že rozdíly jsou pak volány jako nové konfigurační příkazy. Aplikováním změn a ne shodných konfiguračních prvků se směrovač vyhýbá potenciálním přerušením poskytování služeb. Algoritmus zajišťuje, že budou všechny příkazy správně aplikovány také zaručením správného pořadí (pomocí několikanásobného průchodu konfigurací). Počet opakování je ohraničen 5 opakováními z důvodu nebezpečí vzniku smyčky.

Alternativou k příkazu „configure replace“ je příkaz „copy“. Kopírování je však slučovací operace – zdrojová a soudobá konfigurace vytvoří směs nastavení. Každý řádek zdrojové konfigurace se aplikuje do současné bez ohledu na to, zda se již v současné konfiguraci vyskytuje adekvátní nastavení či ne. Neefektivita algoritmu může vést i k výpadku služby. Při aplikaci příkazu „configuration replace“ naopak dojde pouze k vyřazení starého nastavení, které již v novém konfiguračním souboru není obsaženo; aplikují se vždy pouze změny. Příkaz „copy *zdrojová konfigurace* running config“ lze účinně použít pouze v případě potřeby doplnění současné konfigurace o částečné nastatvení.

Mechanizmus je velmi citlivý na manipulaci v konfigurací během změny konfigurace. V současné době systémy IOS obsahují bezpečnostní zámek, který blokuje změny v konfiguraci během nahrazovacího procesu. Zámek je standardně nastaven, lze jej však odstranit. Automaticky se zámek odemkne po skončení nahrazování konfigurace.

Možnosti nahrazovat konfiguraci a její archivaci se vyskytují od verzí 12.2 a 12.3 (podle typu). Současné verze IOS 12.4 funkci nahrazování plně podporují.

Pokud nahrazujeme soubory pomocí příkazu „configuration replace“, je třeba si uvědomit, že tento příkaz mění pouze aktuální nastavení v paměti RAM, pro trvalé uložení změn je třeba uložit nahrazenou konfiguraci do souboru startup- config (pomocí příkazu copy).

1.8.2 Podpora protokolů pro přenos souborů

Konfigurační soubory je společností Cisco doporučeno přenášet pomocí jednoduchého protokolu TFTP. Takový přenos je bezproblémový v případě, že není třeba zabezpečení. Navíc jsou konfigurační soubory relativně malé (stovky kB), proto je využití takového protokolu ideální. Problém nastává, pokud chceme takovýto přenos zabezpečit, a to třeba i v případě, že je vytvořen virtuální tunel. Protokol TFTP v sobě neobsahuje žádnou formu autentizace, navíc je nespolehlivý a nevhodný pro přenos objemově větších souborů (např. obrazy IOS). Proto zařízení nabízejí přenos pomocí jiných protokolů. Podle [13] lze přenášet pomocí protokolů TFTP, FTP, HTTP, HTTPS, SCP a RCP. Všechny tyto protokoly fungují na zařízeních jako server i klient podle potřeby (příklad využití HTTP(S) serveru je konfigurace pomocí grafického rozhraní). Problémem jsou nouzové stavy. Například v režimu ROMmode lze operační systém nahrát (nebo načíst) pouze pomocí protokolu TFTP. Mimo vzdálenou správu lze využít pro přenos souborů externí zařízení – v dnešní době jsou na platformách přítomny sloty USB, u starších zařízení lze nahrávat z pamětí Compact Flash. Použití paměťových karet může mít své úskalí ve formátování na správný systém souborů. Ideální je proto formátovat kartu přímo v zařízení, které bude z karty číst.

2 Program pro správu konfigurací

Tato kapitola popisuje praktickou část diplomové práce. Budou zde popsány vlastnosti aplikace, koncept, na kterém je postavena a všechny funkce, které tento program vykonává a nabízí uživateli. Dále bude rozvedeno, jak je možné program rozšířit – od jednoduchých funkcí ke složitějším. Program Správa konfigurací Cisco páteřní sítě bude dále označován jako Správa konfigurací.

2.1 Koncept

Hlavním úkolem programu bylo umožnit uživateli snadno a přehledně spravovat konfigurační soubory na síťových zařízeních. Při analýze problému došlo k vytvoření konceptu 5 pilířů programu viz Obr. 4.

Jádro programu

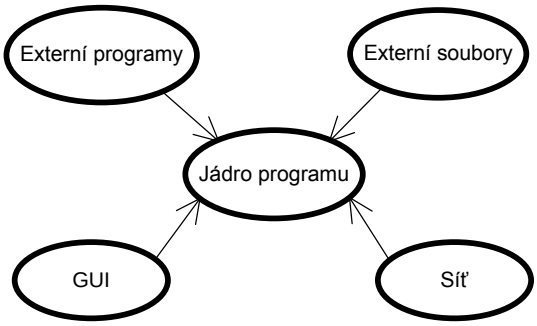
Jádro programu tvoří základ celé aplikace. Umožňuje spolupráci instancí všech objektů a jejich členských funkcí.

Síť

Správnou funkci z hlediska síťové komunikace zajišťuje pilíř Síť. Tvoří jej všechny objekty vázané na komunikaci pomocí protokolů ICMP, telnet a TFTP. Protokol ICMP slouží pro ověření spoje mezi síťovými entitami a je jedinou variantou pro splnění tohoto úkolu. Pro připojení k jednotlivým zařízením bylo voleno mezi protokoly telnet a SSH. Vybrán byl protokol telnet. Skutečnost, že je tento protokol nezabezpečený, není v případě vytvářené aplikace důležitá, neboť se všechna zařízení nacházejí v privátní síti, a dokonce v jedné místnosti. Nebezpečí odposlechu dat a jejich zneužití je tímto limitováno pouze na oblast zabezpečenou proti vnějšímu vniknutí (laboratoř). Důležitým rozhodovacím faktorem je jednoduchost tohoto protokolu (bez vytváření klíčů a zajištění jejich výměny), která byla při implementaci jeho hlavní výhodou. Pro přenos souborů používají Cisco zařízení několik přenosových protokolů. Nejjednošším z nich je protokol TFTP, který používá UDP transportní protokol, je nezabezpečený, ale stále představuje velmi důležitý a používaný protokol. Server je v současné době realizován pomocí externího freewarového programu, který podporuje vícevláknovou komunikaci.

Externí programy

Externí programy zahrnují aplikační server telnet, který je nutný pro samotný transfer konfiguračních souborů, a operační systém, který je nutný pro běh hlavní aplikace.



Obr. 4: Hlavní části programu

Grafické rozhraní

Uživatelské grafické rozhraní (GUI) slouží pro splnění požadavku přehlednosti programu. Příkazový řádek svými možnostmi nemůže převýšit způsob zobrazení pomocí grafických objektů. Pro grafickou nástavbu byla použita knihovna Java swing, která obsahuje všechny nezbytné komponenty pro vytvoření kvalitního grafického uživatelského rozhraní.

Práce s externími soubory

Externí soubory jsou stejně jako předchozí pilíře velmi důležité pro správný chod programu. Zahrnují práci s konfiguracemi síťových zařízení a s konfigurací samotného programu. Aplikace pracuje výhradně s textovými soubory, které umožňují uživateli jednoduše zasahovat do konfigurace zařízení a částečně i celé aplikace. Soubory představují vždy velmi efektivní řešení pro případ vypínání programů. Celou konfiguraci pak není nutné znovu vytvářet.

Programovací jazyk

Program byl vytvořen v jazyce Java. K rozhodnutí vedly jeho následující vlastnosti:

- Jazyk je objektivní – v jazyce lze programovat objekty téměř nezávisle na sobě vznikají tak celky, jenž lze velmi snadno měnit, aniž by docházelo k vlivu na ostatní části programu. Zároveň je zdrojový kód přehlednější.
- Multiplatformní – i když byla aplikace vytvářena pro systém Windows, lze kód bez zásahu do zdrojového kódu zkompilovat v jiném operačním systému.
- Podpora více vláken a grafických objektů – tento jazyk plně podporuje vytváření vícevláknových (multithreading) aplikací, což jej předurčuje pro

tvorbu síťových programů, zejména serverů, u kterých bývá multithreading často nezbytnou podmínkou pro správnou funkci. Tato podpora zahrnuje i velmi jednoduchou synchronizaci sdílených dat mezi objekty, což může v jiných jazycích (např. v jazyce C++) vést k vážným problémům.

Jako programovací prostředí byl vybrán program Eclipse 3.4.1. Jeho výraznou výhodou je volná dostupnost programu (freeware) včetně různých rozšíření (tvorba grafického rozhraní, vykreslování různých typů diagramů), i když některé jsou dostupné pouze pro krátkobé vyzkoušení. Výhodou Eclipse je též jeho jednoduchost a „přívětivosti“ (user friendly).

Objekty programu

Z pohledu zdrojového kódu se aplikace skládá ze 6 balíčků (package) – default, gui, fileProcessing, exceptions, netCommunication a database. Jednotlivé balíčky jsou popsány v příloze A. Jsou zde uvedeny třídy, které balíčky obsahují, a jejich stručný popis.

2.2 Požadavky programu

Program potřebuje přibližně 10 MB fyzické paměti pro chod hlavní aplikace, TFTP server 7,7 MB fyzické paměti. Nároky na hardware jsou tedy minimální, aplikace bude funkční i na méně výkonných počítačích (jeden procesor, fyzická paměť < 1 GB). Podmínkou je podpora vícevláknových aplikací od operačního systému, což je v dnešních OS naprostým základem.

Aplikace nebude funkční bez TFTP serveru, proto musí být takový program bezpodmínečně nainstalován v počítači, na kterém je spuštěna Správa konfigurací. Aplikace TFTP nejsou komplikované, a proto by jimi počítač neměl být výrazně zatěžován. Důležité je, aby byl server spuštěn na stejném počítači jako Správa konfigurací a byl provozován na primární IP adrese počítače (standardní nastavení serveru pro případ, že se na PC nachází více síťových karet fyzických či virtuálních), kterou Správa konfigurací používá při komunikaci se síťovým zařízením pro lokalizaci TFTP serveru.

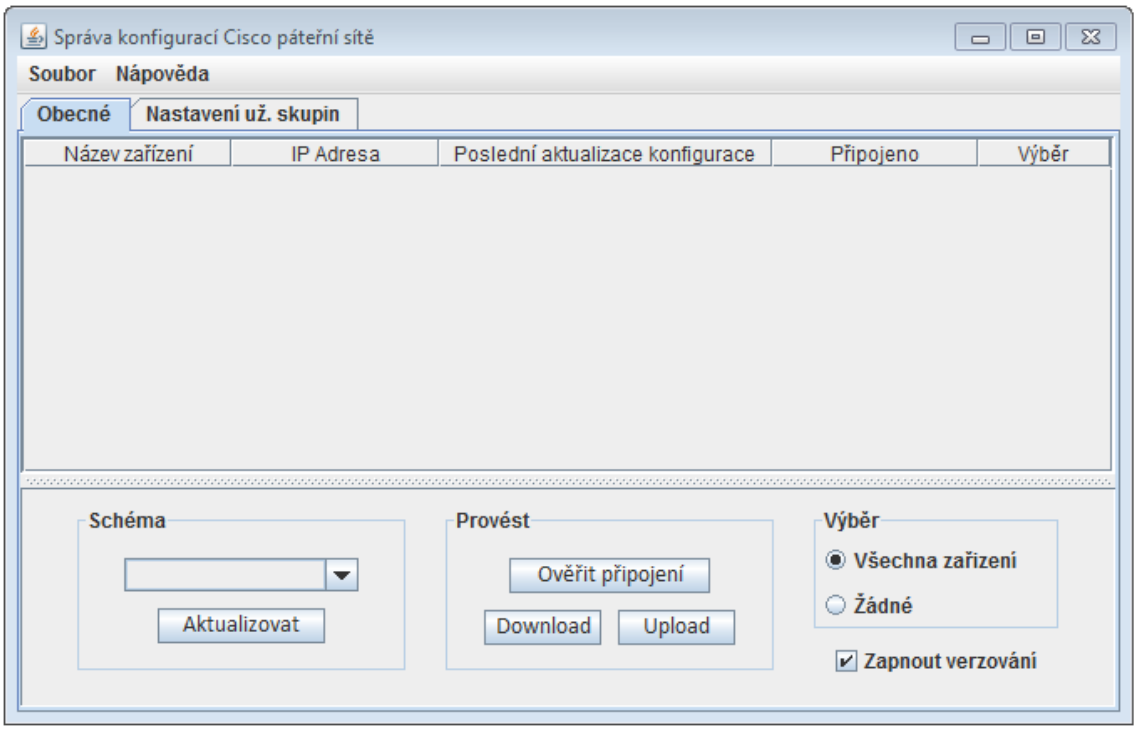
2.3 Funkce programu

2.3.1 Uživatelské funkce

Zobrazení aplikace po spuštění

Po spuštění programu se uživateli zobrazí základní okno aplikace (viz Obr. 5). Uživateli se zobrazí prázdná šablona záložky Obecné. Záložka se skládá z tabulky zobrazující název zařízení, IP adresu, času poslední aktualizace konfiguračního souboru na serveru, stav připojení k zařízení a výběr zařízení pro operace. Další část obrazovky tvoří objekty pro volbu schématu (skupiny zařízení), rychlý výběr všech nebo žádného zařízení, volba zapnutí verzování a tlačítka pro aktivací operací (aktualizace tabulky, ověření připojení, stažení souboru či naopak nahrávání souboru ze serveru na zařízení).

Uživateli se nabízí dvě možnosti, jak nahrát konfiguraci do programu. Buď ji načte ze souboru, nebo ji vytvoří manuálně ve vedlejší záložce Nastavení už. skupin.

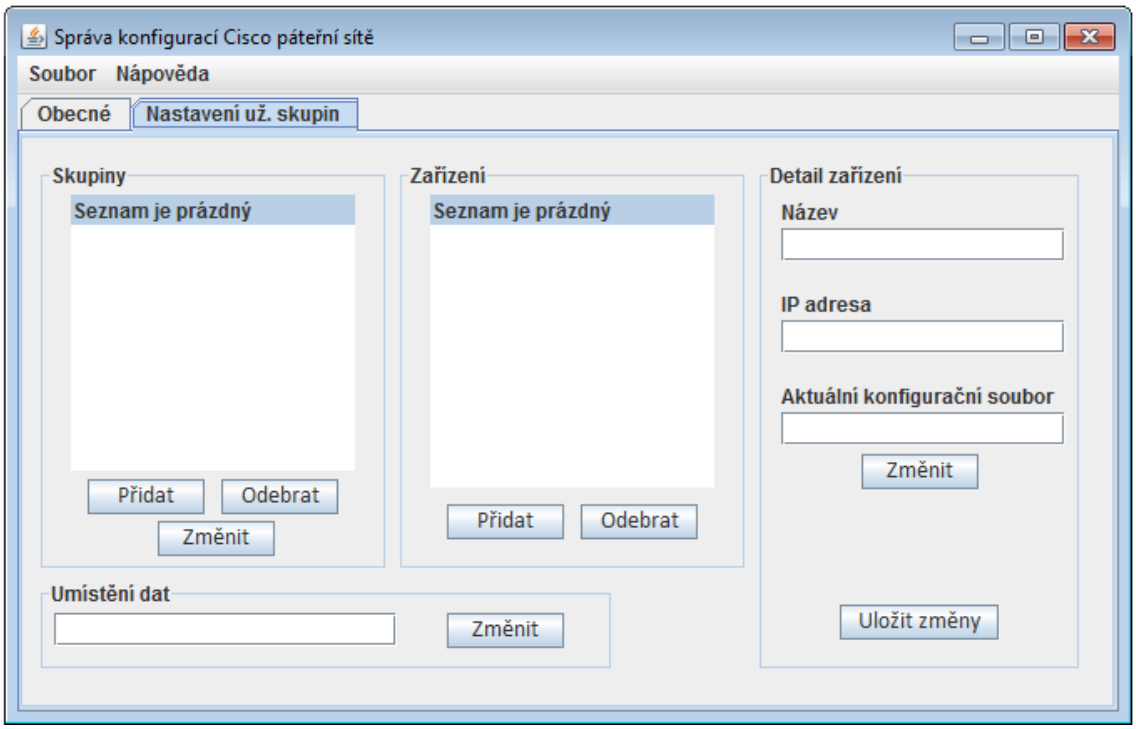


Obr. 5: Okno po spuštění

Zobrazení záložky Nastavení uživatelských skupin

Záložka je zobrazena viz Obr. 6. Obrázek zobrazuje stav po spuštění programu. Není definována žádná skupina ani zařízení. Pomocí tlačítka pro přidání skupiny lze přidat první skupinu, pro kterou lze dále definovat zařízení. Formuláře pro

zobrazení se naplní v případě nahrání konfigurace ze souboru (platí pro zobrazení skupin). Zařízení se zobrazují po kliknutí myši na položku v seznamu skupin. Detaily zařízení (název, IP adresa, soubor) se zobrazují po kliknutí na položku v zařízeních. Záložka Nastavení uživatelských skupin zobrazuje dále cestu ke konfiguračním souborům (místo, kam server ukládá na počítači soubory). Tato cesta je důležitá pro výběr konfiguračních souborů pro zvolené zařízení.



Obr. 6: Zobrazení uživatelských skupin

Načtení konfigurace

Konfiguraci lze načíst z libovolného souboru, který uživatel vybírá pomocí dialogu. Uživatel klikne na položku v menu Soubor> Načíst databázi ze souboru. Následně se zobrazí uživateli dialogové okno, kde může uživatel libovolně listovat v systému souborů. Pro výběr souborů není nastaven žádný filtr pro typ souboru, jedinou podmínkou kladenou na souboru je správná syntaxe zápisu jednotlivých položek. Pokud je soubor v pořádku, načte se do programu databáze všech skupin a jejich zařízení a cesta ke konfiguračním souborům. Zároveň se v záložce Obecné načte do tabulky první databáze ze souboru a uživateli je umožněno se schématem ihned pracovat.

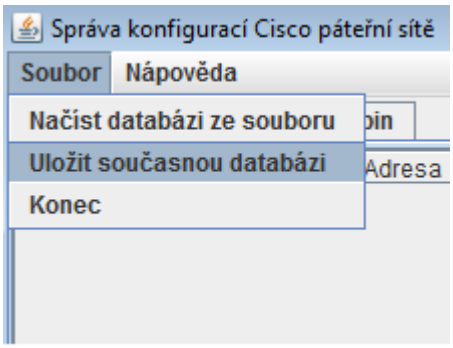
Uložení konfigurace

Aktuální konfiguraci skupin a zařízení lze uložit do souboru. Způsob je odobný jako při načtení konfigurace. Uživatel vybere z menu položku Soubor>Uložit současnou databázi a zobrazí se dialogové okno, kde uživatel zvolí cestu a pojmenuje

konfigurační soubor. Aplikace ukládá soubor bez ohledu na to, zda již existuje soubor se stejným názvem.

Ukončení aplikace

Aplikace se ukončuje kliknutím na křížek v horním pravém rohu aplikace, nebo kliknutím na položku Konec v menu Soubor.



Obr. 7: Menu programu

Nápověda

Nápověda obsahuje informace o aplikace v položce O programu. Součástí zobrazeného okna je logo Ústavu telekomunikací, kterému se v případě, že je nalezeno ve stejném adresáři s aplikací, přizpůsobí formátování.

Výběr schématu

Uživatel si vybírá pomocí seznamu schéma (skupinu), kterou si přeje zobrazit v tabulce. Pro vlastní zobrazení je nutné kliknout na tlačítko Aktualizovat.

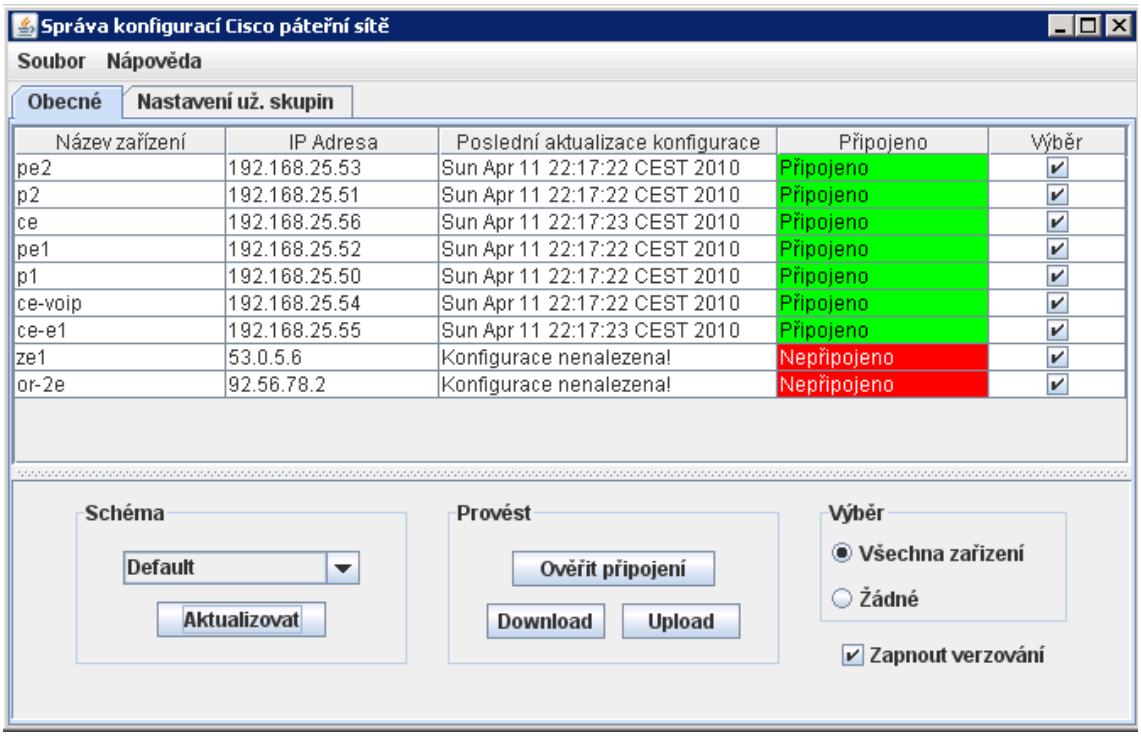
Aktualizace tabulky

Aktualizace tabulka spočívá v načtení vybraného schématu ze seznamu. Do tabulky se zobrazí detaily podle popisků jednotlivých sloupců. Datum poslední aktualizace konfiguračního souborů se zobrazí pouze v případě, že konfigurační soubor pro dané zařízení a danou skupinu (tj. soubor přidělený v detailu zařízení) je nalezen v místě, kam odkazuje nastavená cesta ke konfiguračním souborům. Zároveň s načtením záznamů do tabulky se všechny zařízení označí jako vybrané (poslední sloupec).

Ověření připojení

Jedna ze základních funkcí programu. Pomocí protokolu ICMP jsou ověřena spojení s vybranými zařízeními uvedenými v tabulce. Po stisku tlačítka Ověřit připojení dojde k zablokování okna tak, aby operace proběhla bez zásahu uživatele do vnitřního chodu programu. Po ověření připojení je přístup opět umožněn. Celá

operace trvá podle toho, jestli jsou jednotlivá zařízení připojena (při nedostupnosti zařízení trvá vyhodnocení připojení déle než 2 s – doba čekání na odpověď ICMP). Pro větší přehlednost je stav připojení zobrazován barevně (zelená – připojeno, červená – nepřipojeno, žlutá – ověření zatím nebylo provedeno) viz Obr. 8: Ověření připojeníObr. 8.



Obr. 8: Ověření připojení

Stahování konfiguračních souborů

Stahování konfiguračních souborů probíhá po kliknutí na tlačítko Download u vybraných zařízení v tabulce. Součástí stahování konfigurace i ověření připojení. Soubor(y) se stahují do adresáře, který je momentálně nastaven pro umístění dat. Na stahování souborů se uplatňuje verzování, pokud je zaškrtnuto v dolní pravé části záložky Obecné. Při stahování souborů (running-config) se uplatňuje příkaz „copy running-config tftp“ pro přenos souborů.

Nahrávání konfiguračních souborů

Nahrávání konfiguračních souborů je obdobné jako stahování. Pro oba typy práce se soubory platí, že souborem s konfigurací je soubor „running-config“ tedy aktuální konfigurace na zařízení (může se lišit od verze nahrané při startu systému na zařízení). Jak je uvedeno výše, pokud má být uplatněno nastavení uložené na serveru, nelze uplatnit příkaz copy, nýbrž příkaz replace, který zaručí přemazání stávajících nastavení. Pokud není stanoveno jinak, nahrává program na zařízení poslední verzi konfiguračního souboru.

Výběr zařízení

Uživateli je umožněno vybírat zařízení pomocí rychlých voleb; takto lze vybrat všechna zařízení, nebo žádné. Volba se ukládá po další načtení konfigurace nebo aktualizaci tabulky. Pokud uživatel vyžaduje práci s konfiguracemi u specifické množiny zařízení, je mu umožněno zaškrtávat v tabulce v sloupci Výběr u zařízení, se kterými se bude dále pracovat (sloupeček Výběr je jediný, který je uživateli přístupný k editaci).

Aktivace verzování

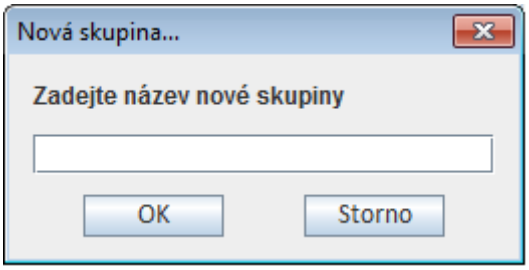
Program nabízí uživateli zapnout/vypnout verzování konfiguračních souborů. O tom, jak verzování funguje, je popsáno v kapitole o verzování viz níže. Pokud uživatel zruší volbu verzování, budou všechny konfigurační soubory při novém stahování přemazány.

Zobrazení skupin zařízení

Skupiny zařízení se zobrazují dynamicky podle změn, které jsou uživatelem či systémem nad němi prováděny (přidávání/editace/odebrání skupiny uživatelem, načtení konfigurace ze souboru systémem).

Přidání skupiny zařízení

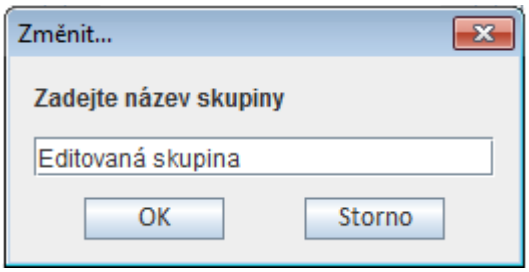
Uživatel přidává skupinu pomocí tlačítka Přidat v panelu Skupiny. Systém zobrazí následně dialogové okno viz Obr. 9, ve kterém uživatel vyplní název skupiny a potvrdí volbu pomocí tlačítka OK. Proces vytváření lze stornovat kliknutím tlačítka Storno. Nová skupina se následně objeví v seznamu skupin.



Obr. 9: Nová skupina

Editace skupiny zařízení

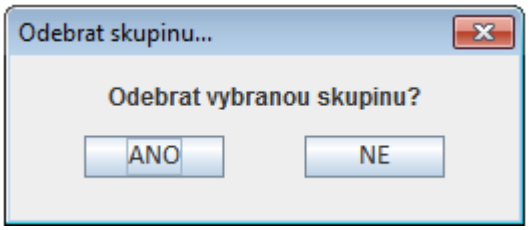
Uživatel edituje skupinu obdobně jako při vytváření skupin. Po označení skupiny klikne na tlačítko Změnit a je mu zobrazeno dialogové okno pro zadání nového názvu skupiny. Uživatel vyplní název a potvrdí volbu. Proces lze zrušit pomocí tlačítka Storno.



Obr. 10: Editace skupiny

Odebrání skupiny zařízení

Skupinu lze odebrat stejným způsobem jako ji lze skupinu editovat. Po označení skupiny se klikne na tlačítko Odebrat a uživatel je vyzván k potvrzení odebrání skupiny.



Obr. 11: Odebrání skupiny

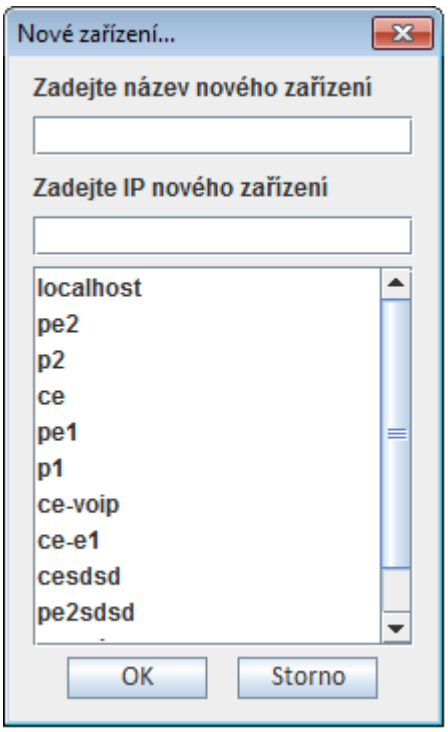
Zobrazení zařízení

Kliknutím na skupinu zařízení se zobrazí uživateli všechna zařízení ve skupině (v opačném případě se zobrazí, že seznam je prázdný). Detail zařízení se zobrazí kliknutím na položku v seznamu zařízení.

Přidání zařízení

Uživateli je umožněno přidávat nová zařízení. Přidat zařízení je však možné pouze tehdy, pokud existuje alespoň jedna skupina zařízení. Přidání zařízení odpovídá přidávání skupiny. Uživatel je po kliknutí na tlačítko Přidat v panelu Zařízení vyzván k zadání údajů o zařízení – názvu zařízení a IP adresy zařízení. Uživatel může pole vyplnit běžným způsobem, nebo může využít možnosti si vybrat mezi zařízeními definovanými v systému. Tato možnost usnadňuje a zrychluje proces vytváření zařízení). Údaje uživatel potvrdí klikem na OK. Uživatel musí vyplnit oba údaje o zařízení; kromě toho kontroluje program, zda uživatel nevytváří ve skupině jiné zařízení se shodným názvem. V případě, že nové zařízení neprojde verifikací, není uloženo a uživatel je vrácen zpět do dialogového okna viz Obr. 12. Pokud verifikace proběhne bez problémů, je zařízení uloženo do databáze zařízení a zároveň zařazeno mezi ostatní zařízení zvolené skupiny. Při vytváření zařízení

systém automaticky doplní podle předdefinované šablony název konfiguračního souboru pro zařízení.



Obr. 12: Nové zařízení

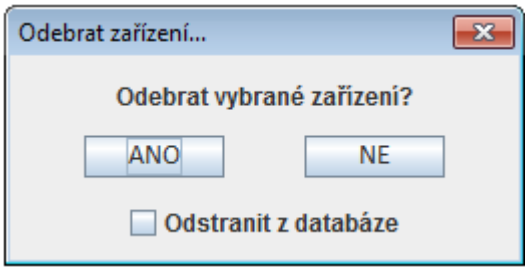
Editace zařízení

Editace zařízení probíhá jiným způsobem než editace skupiny. Jelikož je možné zařadit jedno zařízení do několika skupin, není dovoleno editovat název ani IP adresu zařízení v rámci skupiny. Pokud chce uživatel přidělit zařízení jiné jméno či jinou adresu, musí vytvořit nové zařízení pomocí způsobu popsaném výše. Stejná zařízení v jednotlivých skupinách se tak odlišují pouze konfiguračním souborem (resp. jeho názvem). Pro editaci zařízení je nutné zařízení vybrat ze seznamu zařízení skupiny (musí být tedy definováno minimálně jedno zařízení ve skupině). Název konfiguračního souboru bývá předdefinován, ale je možné jej libovolně měnit v panelu „Detail zařízení“. Zároveň lze zařízení přiřadit konfigurační soubor jiného zařízení nebo přidělit jiná než nejnovější verze souboru. Při výběru souboru pomocí dialogového okna (po kliknutí na tlačítko Změnit je uživatel omezen na výběr v adresáři daného cestou ke konfiguračním souborům, uvedenou v panelu „Umístění dat“ v editačním okně. Změny detailu zařízení se uloží po kliknutí na tlačítko „Uložit změny“.

Odebrání zařízení

Zařízení lze odebrat ze skupiny nebo jej odstranit úplně z databáze všech zařízení. Pro odebrání je potřeba kliknout na tlačítko Odebrat v panelu Zařízení. Uživateli se

následně zobrazí dialogové okno s volbou, zda chce uživatel odstranit zařízení také z databáze (výchozí hodnota nezaškrtnuto). Uživatel potvrdí volbu kliknutím na ANO a systém odstraní zařízení podle uživatelovy volby.



Obr. 13: Odebrání zařízení

Změna uložení konfiguračních souborů

Umístění konfiguračních souborů lze zvolit pomocí tlačítka Změnit v panelu „Umístění dat“. Uživateli je zobrazen dialog pro výběr adresáře, kde uživatel svou volbu potvrdí a systém umístění uloží.

Změna velikosti okna

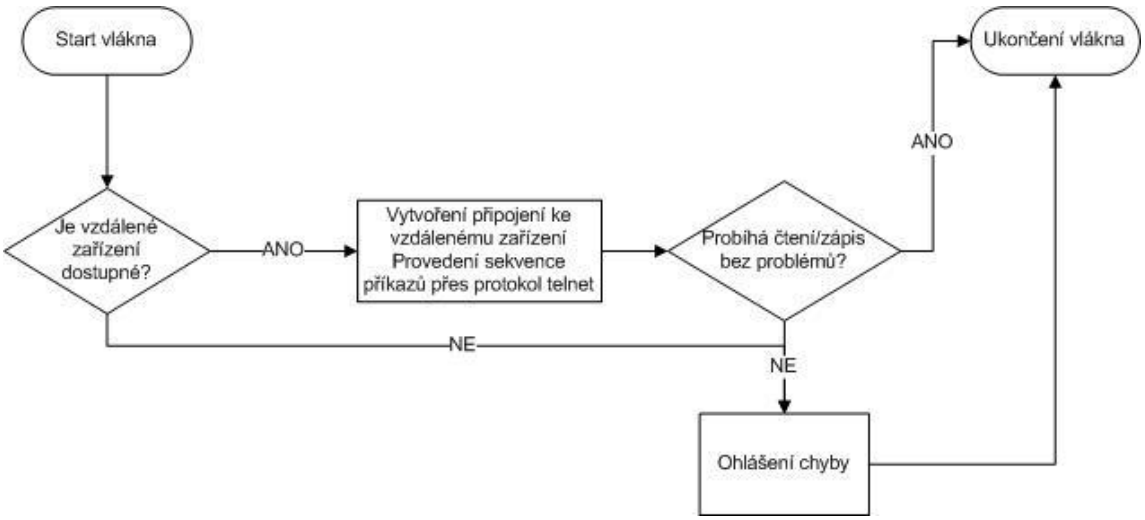
Grafické rozhraní bylo řešeno tak, aby umožnilo uživateli měnit velikost okna (tj. okno není zamčené). Uživatel by však měl používat aplikaci ve výchozím zobrazení, které je nejpřehlednější. Výchozí rozměry hlavního okna aplikace (675x430 px) zdaleka nepřesahují rozlišení současných monitorů. V opačném případě (rozlišení obrazovky je příliš veliké a položky jsou špatně viditelné) je doporučeno použít lupu.

2.3.2 Systémové funkce

Připojování k zařízení

Pokud systém rozpozná kliknutí na tlačítko, načte si databázi všech zařízení, které byly uživatelem vybrány pro stažení konfigurace. Po vytvoření vektoru s těmito zařízeními vytvoří nová vlákna programu s nastaveními, jenž jsou nezbytná pro správné vykonání operace. Průběh vlákna lze sledovat viz Obr. 14. V rámci vlákna dojde nejprve k ověření dostupnosti zařízení. Pokud je dostupnost potvrzena, připojí se program k zařízení pomocí protokolu telnet. Pomocí předem definovaného skriptu pro účely nahrávání (nebo stahování) konfiguračního souboru se na zařízení vykonají příkazy, které vyvolají komunikaci zařízení rozhraní a TFTP serveru. Pokud dojde při navazování komunikace k chybě, pak je tato chyba vypsána, spojení se přerušuje a zařízení je v tabulce označeno jako nedostupné (nepřipojené). Pokud dojde k chybě při komunikaci přes protokol

TFTP, zobrazí se tato chyba na serverové aplikaci. V programu lze úspěšnost stažení souborů vysledovat podle data poslední aktualizace souborů (zobrazí se aktuální datum). Každé vlákno vždy skončí, proto nikdy nedojde k zablokování programu a uživatel bude moci učinit nový pokus o požadovanou operaci.



Obr. 14: Vývojový diagram vlákna pro komunikaci se síťovým zařízením

Vytváření textových souborů

Podoba textových souborů byla inspirována konfiguračními soubory, které vytvářejí aplikace v operačních systémech linux. Tyto soubory poskytují výborné možnosti konfigurace chodu programu bez toho, aniž by uživatel měnil konfiguraci přímo v aplikaci. Editace konfiguračního souboru Správy konfigurací tak umožňuje plnohodnotnou alternativu k záložce Nastavení uživatelských skupin v grafickém rozhraní.

Textový soubor se skládá s tagů, parametrů, a dokonce i komentářů. Příklad takového souboru ukazuje Obr. 15. Pro úspěšné čtení konfiguračního souboru je nutné dodržet několik pravidel:

- Tagy (značky) jsou uzavřeny do standardních závorek <, > a jsou vždy nepárové. Platnost tagu ruší následující tag.
- Parametry jsou vždy umístěny po značce, ke které náleží. Jejich hodnota se nastavuje pomocí operátoru '='.
- Pokud nabývá parametr několika hodnot, jsou tyto hodnoty odděleny čárkou.
- Kombinace zadávaných parametrů je oddělena pomocí operátoru ':'.
- Zařízení přidělená do skupin musí být definována dříve než skupiny.
- Tagy a parametry jsou case-sensitive (záleží na velikosti písma), musí začínat malým písmenem.
- Komentář začíná vždy znakem '#' a platí pro celý řádek.

Program podporuje 4 různé tagy – <general> pro obecná nastavení. Obsahuje parametr path, který určuje cestu ke konfiguračním souborům. Tag <device> slouží pro označení zařízení. Obsahuje parametr name pro název zařízení a parametr ip pro IP adresu zařízení. Tag <group> je určen pro označení skupiny. Obsahuje parametr name a kombinaci parametrů device:filename, které slouží pro přiřazení definovaného zařízení a názvu konfiguračního souboru.

```
#this is an automatic configuration generated by system
<general>
path=C:\TFTP-Root
<device>
name=localhost
ip=127.0.0.1
<device>
name=pe2
ip=192.168.25.53
<device>
name=p2
ip=192.168.25.51
<device>
name=ce
ip=192.168.25.56
<device>
name=pe1
ip=192.168.25.52
<device>
name=p1
ip=192.168.25.50
<device>
name=ce-voip
ip=192.168.25.54
<device>
name=ce-e1
ip=192.168.25.55
<group>
name=Default
device:filename=localhost:localhost.conf,pe2:Defalutni-pe2.confg,p2:Defaultni-
p2.confg,ce:Defaultni-ce.confg,pe1:Defaultni-pe1.confg,p1:Defaultni-p1.confg,ce-
voip:Defaultni-ce-voip.confg,ce-e1:Defaultni-ce-e1.confg
```

Obr. 15: Příklad konfiguračního souboru

Verzování

Verzování souborů zamezuje neustálému přemazávání konfigurací při jejich stahování. Pokud se například stažený soubor jmenuje soubor-conf, následující soubor se bude jmenovat soubor-conf_v1. Před stažením souboru program hledá podle tohoto schématu poslední verzi souboru. Pokud se nalezena verze, která se již fyzicky nevyskytuje na disku, je její označení přiděleno stahovanému souboru. Pokud je verzování vypnuto, je poslední verze souboru přemazána nově staženým souborem. Do souborů na disku není vhodné zasahovat nevybíravým způsobem tj. pokud např. uživatel odstraní z disku z pěti verzí tu třetí (tj. soubor-conf_v2), tak příští stažený soubor ponese právě toto označení bez ohledu na to, zda existují také verze s vyšším pořadovým číslem.

Doplňování názvu konfigurací

Systém usnadňuje uživateli označení konfiguračních souborů pro zařízení skupiny. Jakmile uživatel přidá zařízení do skupiny, použije se název skupiny a název souboru pro vytvoření názvu konfigurace. Např. u skupiny Group bude pro zařízení Gevice název konfiguračního souboru group-device.conf. Vzhledem k tomu, že se ve skupině nesmí vyskytovat dvě zařízení stejného označení a duplicitní skupiny jsou zakázány, zajišťuje tento způsob jezdnoznačnost v názvosloví souborů.

Načítání schémat při změně tabu

Pokud uživatel změní skupiny v záložce Nastavení uživatelských skupin a překlikne na záložku Obecné, načte se do seznamu schémat aktuální stav skupin. Pokud došlo ke změnám v právě zobrazovaném schématu, musí uživatel pro zobrazení těchto změn použít tlačíko Aktualizovat a změny u zařízení jsou do tabulky promítnuty.

Blokování uživatele

V rámci ochrany chodu programu jsou uživatelské zásahy při síťové komunikace progamu blokovány po stisknutí tlačítka. Uživatel tak ztrácí na několik vteřin kontrolu nad programem. Aplikace je však navržena tak, aby v případě jakékoli chyby došlo k odblokování programu.

2.4 Rozšíření programu

Tato kapitola pojednává o možnostech rozšíření programu v budoucnosti. Tyto funkcionality nebyly zatím v různých důvodů implementovány, v budoucnu by však mohly nalézt v programu větší či menší uplatnění.

Podpora více typů transferů souborů

TFTP není zdaleka jediný aplikační protokol, který je možné využít pro přenos konfiguračních souborů. Podpora volby protokolu v aplikaci představuje velmi jednoduchou implementaci. Realizována však nebyla z následujících důvodů:

- Laboratorní podmínky a fyzické umístění zařízení nevyžadují použití zabezpečeného protokolu (např. SFTP).
- Konfigurační soubory jsou velmi malé (v řádu kB). U takto malých souborů je velmi malá pravděpodobnost chyby při přenosu, proto není třeba používat protokoly doporučované pro přenos objemných souborů.

- Vzdálenost mezi zařízeními je natolik malá, že pravděpodobnost vzniku chyby vlivem vzdálenosti je opět zanedbatelná. Tento a výše uvedený důvod vede k rozhodnutí nepoužívat protokol FTP, který zabezpečuje přenos použitím transportního protokolu TCP.

Podpora vkládání skriptů protokolu telnet

Vkládání skriptů protokolu telnet představuje možnost ovládat komunikaci programu se síťovým zařízením. Uživatel by si mohl vybrat, jaký skript použít pro nejrůznější konfiguraci zařízení (kam patří i přenos souborů). Otázkou zůstává, zda není lépe složitější operace řešit individuálně pomocí „klasického“ připojení pomocí klienta telnet. Účelem tohoto programu bylo zajistit komunikaci se zařízením, která vede ke stažení či nahrání konfiguračního souboru. Tento účel program plní. Implementace takové funkcionality není obtížná (v současné době již program umí číst textové konfigurační soubory se skupinami a zařízeními), důležité je však pro ni najít využití.

Podpora pravidelného sledování dostupnosti zařízení

Tabulka zobrazující stav připojení zařízení v současné době není pravidelně aktualizována. Program byl napsán tak, aby bylo možné vytvářet plánované úlohy jakou je i aktualizace. Plánované úlohy však nebyly jednou z priorit programu.

Přenosy jiných než konfiguračních souborů

Pokud by bylo nutné přenášet jiné než konfiguračních soubory, velmi by záleželo na konkrétním požadavku uživatele. Pokud je požadavkem přenos jednoho konkrétního souboru (jako je tomu právě v případě konfiguračních souborů), pak je implementace jednoduchá. V případě, že uživatel předem nezná soubor, který by mohl v budoucnu nahrávat na zařízení, pak lze implementovat podporu výběru libovolného souboru z adresářové struktury. Složitější problém však představuje situace, kdy uživatel pořebuje stáhnout soubor ze síťového zařízení soubor, jeho název předem nezná (např. jednu z verzí operačního systému IOS na zařízení). Situaci může zkomplikovat i konkrétní úložiště, ze kterého chce uživatel data čerpat (u konfiguračního souboru je situace jednoduchá – soubor se nachází v paměti NVRAM) – zařízení obsahují několik typů pamětí, ke kterým patří i externí paměťové karty. Pokud by uživatel potřeboval zobrazovat vždy obsah všech pamětí, vyžaduje implementace poměrně sofistikovaný přístup. Pro úspěšnou extrakci názvu souborů a jejich úložiště je nutné implementovat širokou podporu protokolu telnet. Po úspěšném vyjmutí názvů souborů je nutné uchovat je v databázi a zobrazit vhodným způsobem uživateli, aby si mohl vybrat, který soubor chce vlastně stáhnout. Zde je nutno podotknou, že protokol telnet definuje časový interval, po jehož uplynutí dojde k přerušení spojení. Je proto nutné zajistit,

aby se uživatel včas rozhodl, nebo spojení přerušit a po uživatelově volbě vytvořit novou instanci telnetu, ve které k samotnému přenosu dojde. Každá funkcionalita navíc vyžaduje dostatečná ošetření proti chybám.

Implementace protokolu TFTP do programu

Vytvořit samostatně v programu server TFTP bylo původně součástí návrhu aplikace. V současné době se však vyskytují volně dostupné programy, které problematiku TFTP zvládají na více než dostatečné úrovni, a není proto zapotřebí se tímto problémem zabývat přímo v programu. TFTP server patří navíc k základnímu vybavení správce síťových zařízení, proto nebyla implementace TFTP serveru do programu stanovena jako prioritní.

Rozšíření grafického rozhraní

U grafického rozhraní lze vymýšlet nejrůznější změny a aktualizace stále. Program by mohl v budoucnu například stále zobrazovat veškeré konfigurační soubory, které se vyskytují v daném umístění (v současné době je toto zobrazováno uživateli pouze při změně konfiguračního souboru v detailu zařízení). Taktéž by bylo možné implementovat podporu pro zobrazení konfiguračního souboru nebo jeho významných částí. Složitější problém představuje odblokování uživatele během komunikace se síťovým zařízením. V průběhu vytváření se však nepodařilo úplného ochránění programu proti desynchronizaci a destabilizaci grafického rozhraní (zejména při vytváření několika – i když synchronizovaných – vláken programu).

3 Závěr

Cílem diplomové práce bylo vytvořit aplikaci pro správu konfigurací páteřní sítě Cisco. Program měl umožnit jednoduchou a efektivní práci s konfiguračními soubory, jejich zálohování a nahrávání. Pro vzdálenou správu měl být využit protokol telnet a pro přenos souborů protokol TFTP. Dále měl program umožnit ověření spojení mezi síťovými entitami pomocí protokolu ICMP.

Výsledkem práce je aplikace, která výše splněné atributy splňuje. Komunikace je řešena vícevláknově, což výrazně zkracuje dobu, po kterou program komunikuje. Při práci s konfiguračními soubory se stahuje či nahrává konfigurace aktuálně běžící na zařízení (tj. running-config). Při jejím stahování se používá příkaz copy, v opačném případě nelze konfiguraci jednoduše kopírovat, nýbrž dochází k jejímu nahrazení pomocí příkazu replace. Samotné přenosy souborů probíhají pomocí externího TFTP serveru, který je běžně dostupný v síti internet od několika výrobců včetně firmy Cisco. Pro snadné ovládání bylo vytvořeno uživatelské grafické rozhraní pomocí knihovny Swing. Grafická podoba programu poskytuje uživateli přehlednou manipulaci s konfiguračními soubory a jejich vazbami na síťová zařízení. Vytvořený systém podporuje vytváření skupin zařízení podle potřeb uživatele. Program navíc podporuje archivaci s využitím verzování konfiguračních souborů. Nastavení programu lze uložit do souboru, který lze jednoduše měnit v textovém editoru. Struktura souboru je podobná konfiguračním souborům linuxovských aplikací, podporuje i vkládání komentářů uživatele.

Struktura zdrojového kódu programu (objektivně orientovaný jazyk) umožňuje jeho další rozšíření bez znalosti kódu celého programu. Bez výrazných zásahů do ostatních oblastí lze tak odděleně vylepšovat grafické či síťové rozhraní, práci se soubory či práci s externími programy.

Program byl testován v laboratoři PA-429 na Ústavu telekomunikací FEKT VUT v Brně, kde prokázal plnou funkčnost dle popsaných parametrů v kapitole 2.

Na přiloženém CD se nachází mimo elektronickou verzi tohoto textu kompletní zdrojový kód programu, jeho zkompilovaná verze pro systém Windows (XP a novější verze) a instalační balíček pro TFTP server běžící v systému Windows.

Literatura

[1] BARRETT, Daniel J., SILVERMAN, Richard E., BIRNES, Robert G. *SSH Frequently Asked Questions* [online]. 2000, 2009 [cit. 2009-12-12]. Dostupný z WWW: <http://www.snailbook.com/faq/ssh-1-vs-2.auto.html>.

[2] BEHRINGER, Michael. *Securing a Core Network* [online]. 2009 [cit. 12. 12. 2009]. Dostupný z WWW: <http://www.ripe.net/ripe/meetings/ripe-49/presentations/ripe49-eof-security-tutorial.pdf>.

[3] BURDA, K. *AAA systémy a protokoly.* In *Elektrorevue* [online]. 29. 9. 2009 [cit. 5. 5. 2010]. Dostupné na WWW: <http://www.elektrorevue.cz/cz/clanky/informacni-techologie/0/aaa-systemy-a-protokoly/>.

[4] Cisco Systems, Inc. *SNMPv3* [online]. 2009 [cit. 12. 12. 2009]. Dostupný z WWW: <http://www.cisco.com/en/US/docs/ios/12_0t/12_0t3/feature/guide/Snmp3.html#wp4363>.

[5] Cisco Systems, Inc. *Authentication, Authorization, and Accounting Overview* [online]. 2009 [cit. 12. 12. 2009]. Dostupný z WWW: <http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/ft_ipacl.pdf>.

[6] Cisco Systems, Inc. *Downloading and Installing Cisco Router and Security Device Manager* [online]. 2009 [cit. 12. 12. 2009]. Dostupný z WWW: <http://www.cisco.com/en/US/products/sw/secursw/ps5318/prod_installation_guide09186a00803e4727.html>.

[7] Cisco Systems, Inc. Privilege Command Enhancement [online]. 2009 [cit. 12. 12. 2009]. Dostupný z WWW: <http://www.cisco.com/en/US/docs/ios/12_2t/12_2t13/feature/guide/ftprienh.html>.

[8] Cisco Systems, Inc. *Configuration replace and rollback* [online]. 2007 [cit. 4. 5. 2010]. Dostupné z WWW: <http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gtrollbk.html>.

[9] Cisco Systems, Inc. *BGP Enforce the First Autonomous System Path* [online]. 2009 [cit. 12. 12. 2009]. Dostupný z WWW: <http://www.cisco.com/en/US/docs/ios/12_2t/12_2t13/feature/guide/ftprienh.html>.

[10] Cisco Systems, Inc. *Cabling Guide for Console and AUX Ports* [online]. 2009 [cit. 12. 12. 2009]. Dostupný z WWW: <http://www.cisco.com/en/US/products/hw/routers/ps332/products_tech_note09186a0080094ce6.shtml >.
[11] Cisco Systems, Inc. *Using the setup configuration tool* [online]. 2009 [cit. 10. 5. 2010]. Dostupný z WWW: <http://www.cisco-secure.com/en/US/docs/ios/12_1/configfun/command/reference/frd1002.html>.
[12] Cisco Systems, Inc. *How to choose a Cisco IOS software release* [online]. 2010 [cit. 10. 5. 2010]. Dostupný z WWW: <http://www.cisco.com/en/US/products/sw/iosswrel/ps1834/products_tech_note09186a00800fb9d9.shtml>.
[13] Cisco Systems, Inc. *Software upgrade procedure* [online]. 2007 [cit. 10. 5. 2010]. Dostupný z WWW: <http://www.cisco.com/en/US/products/hw/routers/ps259/products_tech_note09186a00801fc986.shtml>.
[14] Cisco Systems, Inc. *Configuring Kerberos* [online]. 2009 [cit. 5. 5. 2010]. Dostupný z WWW: <http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfkerb.html>.
[15] Cisco Systems, Inc. *Configuring RADIUS* [online]. 2009 [cit. 5. 5. 2010]. Dostupný z WWW: <http://www.cisco.ws/en/US/docs/ios/12_2/security/configuration/guide/scfrad.html>.
[16] Cisco Systems, Inc. *TACACS+ and RADIUS comparison* [online]. 2001 [cit. 5. 5. 2010]. Dostupný z WWW: <http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a0080094e99.shtml>.
[17] ERDMAN, Joshua. *Cisco IOS* [online]. 2005 [cit. 12. 12. 2009]. Dostupný z WWW: <http://www.networkclue.com/routing/Cisco/IOS/index.aspx>.
[18] GOORDEN, M. *Internet accounting – TACACS+* [online]. 1999 [cit. 10. 5. 2010]. Dostupné z WWW: <http://ing.ctit.utwente.nl/WU5/D5.1/Technology/tacacs+/>.
[19] HAJNÝ, J.; PELKA, T. *Univerzální autentizační rámec.* In *Elektrorevue* [online]. 30. 3. 2010 [cit. 5. 5. 2010]. Dostupné na WWW: <http://www.elektrorevue.cz/cz/clanky/informacni-techologie/0/univerzalni-autentizacni-ramec/>.
[20] HUCABY, S.; McQUERY, D. *Konfigurace směrovačů Cisco*. 1. vyd. Brno: Computer Press, 2004.
[21] Microsoft Corp. *The Telnet Protocol* [online]. 2009 [cit. 12. 12. 2009]. Dostupný z WWW: <http://support.microsoft.com/kb/231866>.

[22] NEUMAN, C.; HARTMAN, S.; Raeburn, K. *The Kerberos network authentication service (V5)* [online]. 2005 [cit. 2. 5. 2010]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc4120.txt>.
[23] POSTEL, J., REYNOLDS, J.. *RFC 854 : TELNET PROTOCOL SPECIFICATION* [online]. 1983, 2009 [cit. 12. 12. 2009]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc854.txt>.
[24] RICCIARDI, F. Kerberos authentication protocol [online]. 2010 [cit. 10. 5. 2010]. Dostupné z WWW: <http://www.zeroshell.net/eng/kerberos/Kerberos-operation/>.
[25] RIGNEY, C. a kol. *Remote authentication dial in user service (RADIUS)* [online]. 2000 [cit. 5. 5. 2010]. Dostupný z WWW: <http://tools.ietf.org/html/rfc2865>.
[26] SEMPERBONI, Fabio. *CiscoZine: CoPP?! What is that?* [online]. 2009, 22. 7. 2009 [cit. 12. 12. 2009]. Dostupný z WWW: <http://www.ciscozine.com/2009/07/22/copp-what-is-that/>.
[27] SMITH, Roderick W. *Linux ve světě windows*. 2. Vyd. Praha: Grada Publishing, 2006.
[28] Sun Microsystems, Inc. *The Java Tutorials: How to Use Tables* [online]. 1995, 23. 9. 2009 [cit. 12. 12. 2009]. Dostupný z WWW: <http://java.sun.com/docs/books/tutorial/uiswing/components/table.html>.
[29] TEARE, D. *Návrh a realizace sítí Cisco*. 1. Vyd. Brno: Computer Press, 2003.

Seznam příloh

A Seznam balíčků a tříd programu 55
A.1 Balíček (default package) 55
A.2 Balíček database 55
A.3 Balíček exceptions 55
A.4 Balíček fileProcessing 56
A.5 Balíček gui 56
A.6 Balíček netCommunication 57
B Ukázky zdrojových kódů programu 58
B.1 Jádro ověřování dostupnosti zařízení 58
B.2 Definování barev pro sloupec tabulky připojení k zařízení 59
B.3 Funkce obstarávající verzování 59

A Seznam balíčků a tříd programu

A.1 Balíček (default package)

Mainclass

Tato třída slouží k volání funkce main, tedy ke spuštění celého programu – instance třídy MyWindow.

A.2 Balíček database

Device

Třída slouží pro definici zařízení – jeho názvu, síťové adresy a souboru, ve kterém je uložena konfigurace. Zároveň obsahuje všechny nezbytné členské funkce potřebné pro operaci s ním.

Group

Třída má stejnou funkci jako předchozí, ovšem pro skupiny zařízení.

MainDatabase

Třída slouží pro definování databází aplikace a obsahuje členské funkce potřebné pro její ovládání. Mimo definici všech zařízení a skupin je její součástí cesta ke konfiguračním souborům.

A.3 Balíček exceptions

FileIsEmpty

První z definovaných vyjímek. Volá se, pokud je konfigurační soubor prázdný.

NICNotFound

Systém zavolá tuto vyjímku v případě, že je zařízení s IP adresou vzdáleného zařízení nedostupné (tzn. selhal příkaz ping). Projevuje se barevnou změnou v tabulce a příslušným textem Nepřipojeno.

ReadingError

Vyjímka v případě, že nelze číst ze síťového soketu.

SocketNotBound

Výjimka se volá v případě, že nelze přiřadit soket. Tato chyba vzniká např. pokud je soket již obsazen jiným programem.

UndefinedCommand

Rezervovaná výjimka, která se volá v místech, kde dojde k výskytu nedefinovaných stavů.

WritingError

Vyjímka v případě, že nelze zapisovat na síťový soket.

A.4 Balíček fileProcessing

FileReader

Třída zajišťuje čtení konfigurace ze souboru. Mezi její funkce patří ale také například hledání posledních verzí souboru pro zobrazení posledního data aktualizace do tabulky.

Processsing

Třída slouží pro hledání konfiguračních souborů v daném úmístění. Zároveň slouží pro vlastní zpracování souborů při verzování.

A.5 Balíček gui

Balíček s grafickými komponentami zde nebude popisován. Grafické objekty vznikly vždy poděděním z objektů knihovny Swing. Balíček obsahuje následující seznam tříd:

- ColorColumnRenderer,
- DialogAbout,
- DialogError,
- DialogNew,
- DialogNewDevice,
- DialogRemove,
- GuiConstants,
- MainWindow,
- MyButton,
- MyFileFilter,
- MyFileSystem,

- MyTableModel,
- MyWindow,
- PanelConfig,
- PanelDeviceDetail,
- PanelItem,
- Task.

A.6 Balíček netCommunication

CommunicationManager

Třída slouží k řízení komunikace se zařízeními. Spouští se v ní jednotlivá řídící vlákna, která slouží pro stahování konfigurací na server či nahrávání na konfigurací na zařízení.

NetConstans

Třída obsahuje pouze statické proměnné. Tyto proměnné jsou například porty, velikost bufferu, označení práce s bufferem. Tato třída neobsahuje žádné členské funkce.

ReadWriteManager

Hlavním úkolem této třídy je zajistit síťovou komunikaci s konkrétním síťovým zařízením na protokolu telnet (TCP). Tato třída dědí vlastnosti z třídy Thread, běží vždy v novém vlákně.

Třída obsahuje zděděnou metodu run, která spouští nové vlákno programu a umožňuje tak provést připojení k více síťovým zařízením najednou. Nezbytnou součástí této třídy je metoda verifyIP, která kontroluje dostupnost vzdáleného zařízení.

SocketRead

Třída slouží pro čtení streamu z přicházejícího na soket serveru od vzdáleného zařízení. Klíčovou roli hraje členská funkce read.

SocketWrite

Funkce této třídy je opačná k třídě SocketRead. Úkolem členské funkce write je zapsat data ze serveru na soket a odeslat jako stream přes TCP protokol do vzdáleného zařízení.

B Ukázky zdrojových kódů programu

B.1 Jádro ověřování dostupnosti zařízení

```
    Vector<ReadWriteManager> rwmanager = new Vector<ReadWriteManager>();

    for (Device d : database) {
      rwmanager.add(new ReadWriteManager(NetConstants.READ, d.getIpAddress(),
true, d.getFilename()));
      rwmanager.lastElement().start();
    }

    while(rwmanager.size()!=0) {
      try {
        Thread.sleep(500);
        table.repaint();
      } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
      }

      for(int index=0; index<rwmanager.size(); index++) {
        if (rwmanager.get(index).getState()==State.TERMINATED) {
          for(int i=0; i<database.size(); i++) {

            if(database.get(i).getIpAddress()
                .equals(rwmanager.get(index).getIp())) {

              if (rwmanager.get(index).isReachable()==true)
                table.setValueAt("Připojeno", group.getDevices().indexOf(
database.get(i)), 3);
              else
                table.setValueAt("Nepřipojeno", group.getDevices().indexOf(
database.get(i)), 3);

              rwmanager.remove(index);
              break;
            }
          }
        }
      }
    }
```

Obr. 16: Zdrojový kód - operace s vlákny při ověřování dostupnosti zařízení

B.2 Definování barev pro sloupec tabulky připojení k zařízení

```
public Component getTableCellRendererComponent (JTable table, Object value,
boolean isSelected,
       boolean hasFocus, int row, int column) {
      Component cell = super.getTableCellRendererComponent
         (table, value, isSelected, hasFocus, row, column);


      if (x.getValueAt(row, column) == "Připojeno") {
      cell.setBackground( bkgndColor );
      cell.setForeground( fgndColor );
      }
      else if (x.getValueAt(row, column) == "Nepřipojeno" || x.getValueAt(row,
column).equals(false)){
        cell.setBackground( Color.red);
          cell.setForeground( Color.white);
      }
      else {
        cell.setBackground( Color.yellow);
          cell.setForeground( Color.yellow);
      }


      return cell;
   }
```

Obr. 17: Dynamické barvení sloupce tabulky podle stavu připojení

B.3 Funkce obstarávající verzování

```
public static String getNewVersion(String name, String path, boolean newVer) {

    String s = new String(name);
    s = s.split("_v")[0];


    File file = new File(path+"\\"+s);

    Integer v=0;
    while(file.exists()) {
      v++;
      if (v!=1)
        s = s.split("_v")[0];
      s += ("_v"+v.toString());
      file = new File(path+"\\"+s);
    }
    if (newVer)
      return s;
    else {
      v--;
      s = s.split("_v")[0];
      s += ("_v"+v.toString());
      return s;
    }
}
```

Obr. 18: Funkce pro verzování souborů