ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra telekomunikační techniky

Implementace dohledové nadstavby stávajícího zálohovacího systému

Implementation of supervisory add-on of an existing backup system

Diplomová práce

Studijní program:	KME – Komunikace, multimédia a elektronika
Studijní obor:	Sítě elektronických komunikací
Vedoucí práce:	Ing. Pavel Troller, CSc.
Konzultant specialista:	Ing. Jan Carva

Tomáš Pilík

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra telekomunikační techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Pilík Tomáš**

Studijní program: Komunikace, multimédia a elektronika
Obor: Sítě elektronických komunikací

Název tématu: **Implementace dohledové nadstavby stávajícího zálohovacího systému**

Pokyny pro vypracování:

Zpracujte ve své diplomové práci problematiku operátorsky spravovaných zákaznických služeb (Managed Services). Uveďte historická i současná dostupná řešení této problematiky. Navrhněte a implementujte dohledovou nadstavbu pro tento segment služeb s využitím dostupných nástrojů s otevřeným kódem.

Seznam odborné literatury:

[1] ČESKÝ TELEKOMUNIKAČNÍ ÚŘAD 2008. Obecná pravidla a doporučení pro využívání řízení datového provozu při poskytování služby přístupu k síti internet [online]. [cit. 2014-03-22]. Dostupné z: http://www.ctu.cz/cs/download/
[2] CARVA, Jan, 2011. Outsourcing v ICT. Praha. Bakalářská práce (Bc.). České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra ekonomiky, manažerství a humanitních věd. Vedoucí práce Pavel Náplava.

Vedoucí: Ing. Pavel Troller, CSc.

Platnost zadání: do konce letního semestru 2015/2016

prof. Ing. Boris Šimák, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 13. 11. 2014

Poděkování

Děkuji vedoucímu práce panu Ing. Pavlu Trollerovi, Ph.D. za konstruktivní kritiku, cenné rady a hlavně za to, že mi pomohl realizovat tuto diplomovou práci, která je z praxe. Dále děkuji externímu vedoucímu práce panu Ing. Janu Carvovi za návrh, podporu a vedení. Díky patří v neposlední řadě mým nejbližším, kteří mi po celou dobu byli podporou.

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a v souladu s Metodickým pokynem o dodržování etických principů pro vypracování závěrečných prací, a že jsem uvedl všechny použité informační zdroje.

Nemám námitky proti použití tohoto školního díla ve smyslu § 60 zákona č. 121/2000 Sb., o autorských právech a právech souvisejících, ve smyslu pozdějších znění tohoto zákona.

V Praze dne 11. 1. 2016 ……

Abstrakt

Práce se zaměřuje na problematiku řízených služeb (managed services) a výběr vhodného dohledového systému pro rozvoj řízených služeb. V této práci jsou uvedeny vlastnosti a funkce nejznámějších dohledových systémů, kterými jsou Cacti, Dude, Nagios, rConfig, Systém Center Operations Manager 2012 a Zabbix. Podle požadavků řízených služeb byl vybrán vhodný dohledový systém pro rozšíření současné služby. K dispozici byl reálný server s předinstalovaným operačním systémem Linux CentOS Red Hat. Na tomto serveru proběhla reálná instalace dohledového systému rConfig. Nástroj rConfig je primárně určen pro zálohování konfigurací. Do dohledového systému rConfig byla připojena databáze zákazníků, která byla zálohována z předchozího dohledového systému. Databázi bylo potřeba upravit a aktualizovat seznam reálných zařízení. V nástroji rConfig byly upraveny kategorie zařízení a nastaveny na každodenní zálohování konfigurací. Dále byly do systému implementovány nové funkce jako například zobrazení pozice zařízení na mapě, automatické připojení terminálu systému Windows, přidání zobrazení hesla, zobrazení zařízení all / online / offline, automatické generování sériového čísla, diagnostika chybovosti BER, DSL parametry Upstream / Downstream a kopírování názvu do prompt řádku zařízení. Tyto funkce byly otestovány v reálném provozu. Na závěr jsem provedl zhodnocení mé práce.

Klíčová slova

řízené služby, outsourcování, počítačový cloud, poskytovatel řízených služeb, dohoda o úrovni poskytovaných služeb (SLA), monitorování, dohledové systémy, Cacti, Nagios, rConfig, SCOM, Zabbix, nastavení dohledového systému rConfig a implementace.

Abstract

The thesis deals with managed services problematics and suitable monitoring system choice for managed services development. Features and properties of the most popular monitoring systems like Cacti, Dude, Nagios, rConfig, System Center Operation Manager 2012 and Zabbix are described. Suitable monitoring system was chosen according to the managed services demand. Real server Godot with pre-installed Linux CentOS Red HAT operating system was available. Real installation of rConfig monitoring system was performed on this server. rConfig tool is primarily designed for configurations backup creation. Costumer database, which was backed up from previous monitoring system, was added to the rConfig system. Costumer needed to edit the database and update real devices list. In the rConfig tool devices categories was edited and automatic everyday backup was set up. New functions were implemented to the system like display device position on map, automatic Windows terminal connection, serial number automatic generation, Bit error rate diagnostics, DSL downstream and upstream parameters, and name to prompt line copying. These functions were tested under real conditions. In conclusion, I have evaluated my work.

Keywords

Managed services, outsourcing, cloud computing, MSP, SLA, monitoring systems, Cacti, Nagios, rConfig, SCOM, Zabbix, setting the monitoring system rConfig and implementation.

Obsah

Seznam obrázků

Seznam tabulek

Seznam zkratek

ADSL	Asymmetric Digital Subscriber line
ATM	Asynchronous Transfer Mode
CAN	Campus Area Network
CDP	Cisco Discovery protokol!!
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Server
DSLAM	Digital Subscriber Line Multiplexor
FDDI	Fiber Distributed Data Interface
FTP	File Transfer Protocol
GPL	General Public License
GPL	General Public License
HDD	Hard Disk Drive
HTTP	HyperText Transfer Protocol
HW	HardWare
ICMP	Internet Control Message Protocol
ICT	Information and Communication Technologies
IIS	Internet Information Services
IM	Instant Messaging
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IT	Informační Technologie
LAN	Local Area Network
MAN	Metropolitan Area Network
MIB	Managent Information Base
MMS	Multimedia Messaging Service
MSP	Managed Services Provider
MSP	Managed Services Provider
NTP	Network Time Protocol
OID	Object Identifier
OS	Operační Systém
PAN	Personal Area Network
PC	Personal Computer
PoE	Power over Ethernet
POP	Post Office Protocol
QoS	Quality of Service
RAM	Random Access Memory
RFC	Request For Comments
RRDTool	Round-Robin Database Tool
RSS	Really Simple Syndication
SAN	Storage Area Network
SCCM	Systém Center Configuration Manager
SCOM	System Center Operations Manager
SMI	Structure of Management Information
SMS	Short Message Service
SMTP	Simple Message Transfer Protocol
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSH	Secure SHell
SW	SoftWare
TCP	Transmission Control Protocol
TELNET	TELecommunication NETwork
UDP	User Datagram Protocol
VDSL	Very-high-data-rate Digital Subscriber line
VLAN	Virtual Local Area Network
WAN	Wide Area Network

1 Úvod

Každým rokem se díky novým síťovým technologiím zlepšuje kvalita poskytovaných služeb. Uživatelé si přejí používat ty nejspolehlivější a nejkvalitnější služby. Konkurence v tomto odvětví je opravdu veliká, protože zákazník důvěřuje jen těm nejlepším a nejkvalitnějším poskytovatelům služeb na trhu. Počítačové vybavení odstraňuje časově náročnou administrativu a usnadňuje pracovní procesy. V dnešní době již není skoro žádná firma, která by počítače nepoužívala.

Tato práce je zaměřena na odvětví síťových služeb a monitorování zákazníků. Důležitým pojmem v oboru síťových služeb je pojem „managed services“, který můžeme přeložit do češtiny jako „řízené služby“, dále budu používat pouze pojem řízené služby. Tyto služby přišly s příchodem prvních počítačových strojů, kdy byla potřeba kontrola činnosti, zda stroje, počítače nebo systémy pracují správně. Myšlenka kontroly služby pro zákazníky zde byla mnohem dříve. Služby začaly vznikat v druhé polovině dvacátého století pro uspokojení zákazníků a dnes je nalezneme téměř všude. Příkladem začátku dohledových služeb je připojení zákazníků na ADSL lince, kde operátor poskytuje zákazníkovi konektivitu a zároveň má povinnost v případě výpadku tuto službu co nejrychleji zprovoznit. Proces kontroly může být mimořádně náročný na vybavení techniky a kvalifikovaných lidí. Právě proto je ideální, pokud existuje dohledový systém s pravidelným hlášením kontroly služeb zákazníků.

Cílem této práce je zálohovat původní databázi zákazníků, vybrat vhodný open source dohledový systém a implementovat nové funkce do zálohovacího systému. Toto téma bylo vybráno s ohledem na autorův dlouhodobý zájem o počítačové sítě a díky možnosti tento projekt realizovat prakticky na reálných zařízeních. Tento projekt byl realizován ve společnosti ICZ a.s., kde autor působí na praxi - rozvíjí dohledový systém rConfig. Nástroj rConfig je zde využíváme pro správu některých zákazníků v rámci poskytování řízených služeb. Odborným konzultantem této práce je autorův vedoucí z ICZ a.s. Ing. Jan Carva.

Dohledový systém by měl umět pravidelně stahovat konfiguraci na server, kontrolovat funkčnost zařízení, kontrolovat připojení, kvalitu připojení a mnoho dalších věcí. Cílem mé práce bude nainstalovat dohledový systém na server, kde pomocí otevřeného kódu zprovozním dohledové funkce, které následně otestuji na reálných zařízeních.

2 Seznámení s problematikou

2.1 Počítače a počítačové sítě

Počítače a výpočetní technika jsou zařízení, které zpracovávají data pomocí programů. Počítačovou síť můžeme definovat jako spojení dvou a více počítačů, nebo jiných zařízeních. Tyto zařízení komunikují nejčastěji pomocí technologie Ethernet, pomocí TCP/IP protokolu. Komunikace mezi počítači, je sdílení zdrojů jako informací, softwaru, zařízení, a mnoho dalších. Počítačovou síť můžeme nazývat jako datovou síť, která se skládá z množiny více prvků (počítačů) a dohromady tvoří infrastrukturu potřebnou pro přenos dat.

2.1.1 Rozdělení datových - počítačových sítí

Datové sítě můžeme rozdělit z hlediska typu přenášeného signálu:

- **Analogový** signál je dán spojitou funkcí v čase (spojitý signál). Má nekonečno stavů.
- **Digitální** signál je takový, který má omezený počet stavů. Nabývá dvou úrovní 0 a 1. Zajímá nás, jestli hodnota spadá do jednoho nebo do druhého intervalu.

Dále můžeme rozdělit datovou síť podle technologie přepojování:

- **Přepojování okruhů** je starší technologie, která se používá typicky na přepojování okruhů u telefonní sítě.
- **Přepojování paketů** je novější technologie. V paketové síti jsou přepojovány pakety, typické u počítačových sítí. Nyní je využívána tato technologie.

2.1.2 Technologie připojení

Nyní si popíšeme několik základních technologií připojení počítačových sítí.

Token Ring je starší technologie lokální sítě LAN. Tato technologie byla vytlačena Ethernetem. Princip sítě komunikace v kruhu je takový, že komunikuje pouze ten, co má „peška“ pomocí speciálního rámce. Tento rámec si předávají zařízení mezi sebou.

Frame Relay je starší technologie používaná u větších sítí typu WAN. Tato technologie nahradila technologii X.25 k propojování např. sítí typu LAN. Používá se mezi routrem zákazníka a přepínačem, kde vytvoří virtuální okruh.

Ethernet je dnes nejpoužívanější standart pro přístup ke sdílenému mediu pomocí metody CSMA/CD pro sítě typu LAN. Metoda CSMA/CD v Ethernetu byla opuštěna od 10 Gbit/s rychlosti, kde tato verze pracuje plně duplexně. Medium pro přenos je koaxiální kabel, kroucená dvoulinka nebo optické vlákno. [3][4]

FDDI (Fiber Distributed Data Interface) je technologie z 90. let. V této době to byla jediná možnost vybudování jednoduché rychlé sítě. Tato technologie umí rychlost 100 Mb/s. Technologie FDDI byla využívána pro sítě typu MAN.

ATM (Asynchronous Transfer Mode). Této technologii byla přisuzována velká budoucnost. Měla ovládnout svět telekomunikací a datových přenosů pod stejným mechanismem. ATM se využívá u páteřních sítí telekomunikačních operátorů, kde ATM poskytuje dobrou garanci kvality svých služeb v oblasti propojování DSLAM (Digital Subscriber Line Multiplexor) u DSL technologií, nejčastěji je to ADSL / VDSL. a další. [1][2]

2.1.3 Rozloha sítí

Počítačové sítě můžeme také rozdělit podle geografické rozlohy (účelu). Rozdělení podle geografického rozložení počítáme podle vzorce:

$$\boldsymbol{R = \frac{D_{avg}}{8B} * v_p}$$ [7]

$\boldsymbol{D_{avg}}$ - průměrné zpoždění přenosu

$\boldsymbol{v_p}$ - průměrná přenosová rychlost

$\boldsymbol{B}$ - počet bajtů v datové jednotce

Local Area Network (LAN) je lokální síť s rozsahem stovek metrů (v rámci budov).

Metropolitan Area Network (MAN) je metropolitní síť s rozsahem několika km (v rámci města nebo měst).

Wide Area Network (WAN) jsou největší sítě, co se týká rozlohy. Spojuje LAN a MAN sítě. Tyto sítě jsou na úrovni velikosti států nebo celých kontinentů. [3][4]

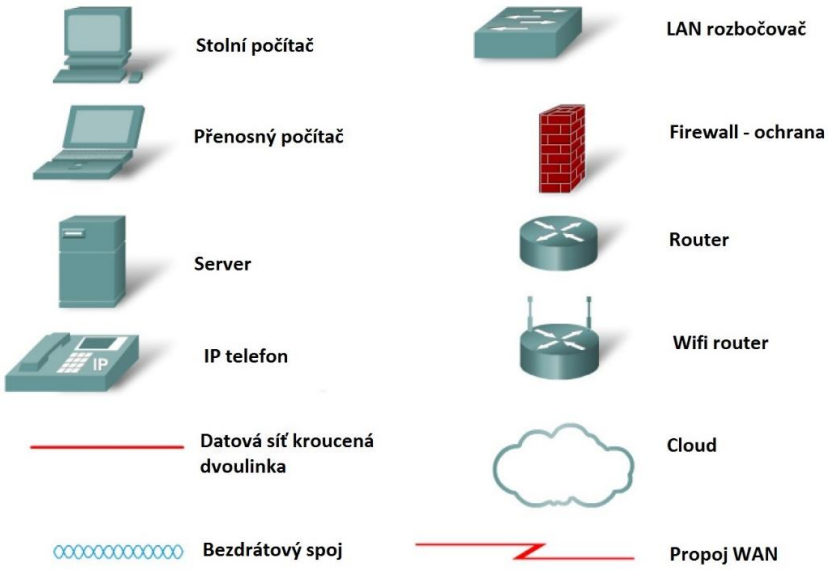
Virtual Private Network (VPN) je veřejná privátní síť, která dovoluje připojení uživatele v internetu do LAN konkrétní organizace. VPN je levné řešení pro rychlé připojení např. do firemní sítě. Vytváří šifrovaný tunel mezi dvěma body (řeší důvěryhodnost, autentizaci, neporušenost komunikace atd.).

Storage Area Network (SAN) je počítačová síť, která propojuje počítače (servery nebo PC) se zařízeními pro ukládání dat, jako jsou disková pole. Pro dosažení vysoké rychlosti jsou propoje dělány zpravidla optickými vlákny. Technologie SAN má výhody zejména vyšší spolehlivost, odstranění limitů lokálně připojených disků a zvýšený výkon.

Pro realizaci všech těchto sítí můžeme použít různé technologie připojení např. drátové sítě (wired) realizovanou kroucenou dvoulinkou (twisted-pair), koaxiálním kabelem, optickým vláknem nebo bezdrátově (wireless). [5] [6]

2.2 Klíčové komponenty a datové sítě

Na obrázku 1. vidíme základní zařízení používané v počítačové síti. Tyto prvky můžeme v dohledovém systému monitorovat.



Obr. 1 Klíčové prvky datové sítě [7]

2.3 Definice dohledového systému

Málokteré odvětví se dnes obejde bez počítačů, počítačových sítí a připojení k internetu. Tyto prvky je nezbytné u rozsáhlejších sítí kontrolovat a sledovat správnou funkčnost, kterou má na starost dohledový systém. Důležité je spojení mezi technologií dohledového systému a správou společnosti (podniku). Propojení podnikání, IT technologií zlepšujeme služby společnosti a zvyšujeme zisk bez nákladů na lidské zdroje. Další výhodou dohledových systémů je rychlost řešení problémů, výpadků a také rychlost zavádění nových služeb.

Dohledový systém je možné definovat jako nějaký prvek přidaný do počítačové sítě, který monitoruje síť a sleduje události. Tento prvek periodicky sbírá informace a sleduje dostupnost nebo stav ostatních prvků. Systém nemusí být fyzický prvek, ale může to být software nebo aplikace nainstalovaná na serveru, který má stejnou funkci jako samostatný prvek. V případě nedostupnosti nebo nesplnění definované podmínky systém informuje administrátora zprávou. Pokročilejší dohledové systémy umí řídit síťovou komunikaci. Určují cestu, kudy potečou data a některé umí do určité míry konfigurovat zákazníkovu síť. Dohledové systémy můžeme rozdělit na několik typů podle jejich možností. Existuje celá řada dohledových systémů, jak od komerčních (placených) tak po ty, které jsou open source (zdarma).

Obr. 2 Zapojení dohledového systému [29]

3 Základní pojmy

Pro orientaci v síťových službách na začátek vysvětlil několik základních pojmů, které je třeba znát v problematice poskytování služeb zákazníkovi.

3.1 Služba

Pojem služba je často využívaný termín a existuje mnoho definic tohoto pojmu. Služba není chápána pouze klasicky jako standardní poskytnutí činnosti, ale objevují se i nové pojmy jako informační či webová služba. V nejjednodušším případě můžeme pojem služba definovat jako činnost námi vykonaná k určitému uspokojení potřeby zákazníka za určitý finanční poplatek. [8]

3.2 ICT systém

Information and Communication Technologies ve zkratce ICT je výraz pod kterým si představíme komunikaci a práci s informacemi. Výraz ICT je spojován s počítačovou technologií. Starší definice ICT je pouze IT (Information Technologies), ale když mezi sebou počítače začaly komunikovat, tak ke zkratce informačních technologií IT byl přidán prvek komunikace, tedy ICT. Pokud se bavíme o ICT prvcích, nebavíme se pouze o fyzickém hardwarovém vybavení (HW), jako jsou počítače, notebook, servery a další, ale bavíme se i softwarovém vybavení (SW). Mezi SW vybavení patří operační systémy, internetové vyhledávače, různé program, síťové protokoly a další. Pojem ICT systém nahrazuje starší výpočetní techniku a informatiku, protože nyní bez informací a komunikace mezi zařízeními nelze postupovat. [9]

3.3 Outsourcing

Pro tento pojem neexistuje přesný překlad z angličtiny. Pokud bychom ho ale přeci jen chtěli přeložit, mohli bychom jej rozložit na dvě části, „outside“ a „resourcing“. V překladu by tedy znamenal zajišťování či získávání zdrojů z venku. Překlad slova outsource lze přeložit i jako „vytěsnění“ a využívání některých aktivit nebo služeb mimo podnik. Tato metoda vede ke zvýšení konkurenceschopnosti podniků, zlepšení procesů a zásadní proměny struktury podniku. V praxi to funguje tak, že firma svěří správu jiné externí firmě, tedy rozdělí práci interních záležitostí.

Typicky se tato služba používá pro úklid, správu počítačů, ochranku a mnoho dalších. Outsourcing firma provádí z důvodu, jednoduchosti, nižších finančních nákladů a zaměření na konkrétní činnost, kterou chce provozovat. [10][11]

3.4 Insourcing

Je opačná činnost outsourcingu, kdy služby které jsou outsourcovány externí firmou jsou převedeny zpět do interních záležitostí firmy. Definice je: „převzetí a začlenění služby nebo výroby produktu poskytované původně smluvně (dodavatelsky) do podnikatelských činností organizace.“ V praxi se tato metoda používá k nižším transakčním nákladům, jednoduší koordinaci, lepší bezpečnosti, či strategických důvodů a mnoho dalším. [10][11]

3.5 Managed services

Řízené služby dnes představují moderní pokrytí služeb v oblasti ICT např. počítačových sítí, komplexního outsourcingu, datových služeb, ochrany firemních informacích, hlasové komunikace, zálohy dat a mnohých dalších. Řízené služby mohou pomoci v každodenních pracovních procesech. Poskytovatelé zařizují zákazníkům jednotlivé služby na profesionální úrovni a bez nákupu drahých komponentů infrastruktury. V rámci zmiňované služby poskytovatel pronajme zákazníkovi drahá komunikační zařízení (routery, telefony, ústředny atd.). Zákazník neinvestuje do svých komponentů infrastruktury, pouze platí za službu předem definovaný tarifní poplatek. Zároveň zákazník získává i servis zařízení. V případě nefunkčnosti nemusí řešit technické zabezpečení, aniž by musel investovat do vlastních zdrojů (HW, SW či svého zaměstnance IT). Řízené služby nemají dlouhodobou strategii pro zákazníka, jako tomu je u outsourcingu. Neovlivňují tolik dlouhodobou strategii, jako je tomu například u outsourcingu, kdy outsourcovanou činnost provádějí zkušenější pracovníci, kteří v tomto oboru pracují řadu let a neustále zlepšují svou práci. Zkušenější pracovníci mají v plánu dlouhodobou strategii firmy. Zaměří se na konkrétní místa a ty dlouhodobě zlepšují. Celkový výsledek je takový, že dochází k dlouhodobému zlepšování procesů a zákazník se může plně věnovat své práci.

Jde tu o požadavky zákazníka a dostupné služby na trhu jak cenově tak kvalitou služby, proto se řízené služby mění dle dostupných technologií trendu trhu. Nejčastěji řízené služby usnadňují a zlepšují využití komunikačních prostředků mezi zaměstnanci, zákazníky a externími firmami. [10] [11]

3.6 Monitoring

Toto slovo pochází z angličtiny a v překladu znamená sledování nebo dohlížení. Pokud se bavíme o monitoringu zákazníka nebo služby, nejde o klasické sledování toho, co přesně zákazník dělá či kde se současně nachází.

Je to služba, která hlídá funkčnost jiné službu nebo zařízení, zda bez problému správně pracují. V případě výpadku informuje zákazníka nebo poskytovatele služeb o změně stavu.

Dále se můžeme setkat s pojmem dohledový systém. Tento systém je monitorovací nástroj, který sleduje konkrétní služby. Dohledový systém či podobná služba sleduje stav zákazníkových zařízení a sbírá různé informace. Služba dovoluje sledovat stav hardwaru, systémové informace a stav běžících činností. Další důležitou funkcí dohledového systému je sběr dat v určitých časových intervalech.

3.7 MSP – Managed Services Provider

MSP můžeme přeložit jako poskytovatel řízených služeb. Může to být jakákoliv společnost, která poskytuje řízené služby pro jiné společnosti. Klienti MSP, tedy zástupci společností, komunikují pomocí interního webového rozhraní, kde si objednají služby a získávají potřebné informace. MSP může zákazníkům nabídnout pronájem infrastruktury, databázové servery, aplikace jako email nebo dohled a zálohování dat.

Další typ komunikace se serverem MSP je přes ASP (Active Server Pages). ASP je skriptovací platforma společnosti Microsoft, která funguje na dynamickém zpracování webových. Příkladem MSP může být helpdesk. [10] [11]

3.8 Cloud computing

Jde o nový model založený na sdíleném používání počítačových technologií přes internet. Nový druh poskytování služeb, kdy MSP poskytuje data na severech a internetu uživatelům přes webové rozhraní. Uživatel se připojuje svým zařízením do své placené soukromé sítě. Výhoda uživatele je, že se může ke svým datům připojit odkudkoliv kde má připojení. Mezi další výhody cloud computingu je sdílení mezi uživateli, okamžitá změna výkonosti hw, automatické aktualizace všech uživatelů a mnoho dalších. Mezi cloudové služby patří např. gmail.com, rapidshare.com, seznam.cz. a mnoho dalších.

Cloud computing znamená v překladu „mrak“. Je to způsob pronajmutí hardware nebo software formou služeb zákazníkovi. Zákazník se nemusí starat o „zbytečné“ formality pro funkčnost, pouze obdrží funkční službu, kterou si nastaví podle vlastních potřeb. O vše ostatní se stará poskytovatel. Je to jeden z nejoblíbenějších trendů společnosti v odvětví informačních systémů. Výhodou služby je, že zákazník platí pouze za to, co opravdu využívá. Nepatří do toho aplikace, sw, výpočetní výkon, datové úložiště a mnoho dalších věcí, které jsou potřeba k realizaci. Model Cloudu je jedna forma outsourcingu – jde o dodávku zdrojů informačních technologií.

Prvotní myšlenka Cloud Computingu je realizována poměrně dlouho. Přišla v podobě používání e-mailu či hostingových služeb. Myšlenka je taková, že zákazník si zaplatí službu, na kterou přestupuje někam do Cloudu „mraku“(zákazník neví, kde fyzicky hw leží). Od služeb e-mailu a hostingu se Cloud Computing liší zejména ve škálovatosti služeb, kde lze navýšit výpočetní výkon nebo datovou kapacitu. Další výhoda Cloud Computingu je úspora provozních zdrojů zákazníka na provoz provozované služby (např. systém).

První pokus a zprovoznění Cloud Computingu byl v roce 2002 společností Amazon, jmenovala se Amazon Web Services (AWS). Společnosti nevyhovovalo, že se využívá pouze 10% její kapacit a proto se služba Cloud Computingu ještě na pár let odložila. Další pokus měla společnost Amazon v roce 2006 pod názvem Amazon Elastic Compute Cloud (EC2). K masivnímu používání techlonologie Cloud Computingu bylo možné z důvodu lepších technologií a hlavně lepší konektivity

k zákazníkům. Ze strany poskytovatelů služeb MSP se začalo virtualizovat. To pomohlo k lepšímu rozložení výkonu, protože na jednom serveru mohlo být víc virtuálních strojů = zákazníků. [10][11]

Cloud Computing se skládá z těch tří kategorií:

1) SaaS (Software as a Service) – aplikační služby
2) PaaS (Platform as a Service) – platformní služby
3) IaaS (Infrastructure as a Service) – infrastrukturní služby

Výhody Cloud Computingu:

- Škálovatelnost služeb
  - zákazník platí, co opravdu používá
  - dynamická změna rychlosti/kapacity
- Nízké investice
  - není údržba vlastního SW či HW
- Úspora IT pracovníků
  - není nutné mít vlastní IT specialistu
- Minimální požadavky pro službu
  - nezávislost z jakého místa je služba používána
  - na čase spouštění a jakou technologii používám

Nevýhody Cloud Computingu:

- Data nemá zákazník uložena u sebe
- Data se mohou nacházet v jiné zeměpisné lokalitě (několik kopií v jiných zemích)
- Závislost na poskytovateli služeb (přechod služby - komplikovaný)
- Přechod od klasického modelu zákazníka do cloudových služeb může být náročný
- Cloudový model má vyšší požadavky na konektivitu zákazníka

3.9 SLA – Service Level Agreement

Service level agreement (SLA) je pojem, který bychom mohli přeložit jako dohoda o úrovni poskytovaných služeb. Určuje úroveň a intenzitu poskytovaných služeb zákazníkovi. Nejčastěji se používá v podmínkách outsourcingu, kde definuje jak rychle a kvalitně bude reagovat externí firma při řešení problému. Každá ze stran, jak poskytovatel/dodavatel outsourcingu, tak odběratel, má jasně definované

povinnosti služby. Tyto povinnosti by měly vést ke dlouhodobé spolupráci mezi stranami a efektivnější komunikaci.

SLA je právní dokument, který definuje přesný rozsah a úroveň poskytované služby. Také zároveň definuje postihy při nedodržení dohody. Tato služba je výhodou, protože definuje jasnou úroveň služby dodavatelem outsourcingu a zákazníkem. Měla by sloužit k jako ochrana proti větším finančním škodám, které by mohly být zákazníkovi způsobeny.

Pojem SLA obsahuje tři klíčové oblasti. První je „záruka infrastruktury“, jedná se o vybavenost hardwaru (používané přístroje a konektivita). Druhá oblast je „procesní záruky“, jde o mapování procesů, kde se řeší jejich změny (přidání uživatele, nový účet, jiný proces atd..). Poslední třetí oblastí je „vzrůstající záruka“, tato oblast má zlepšit záruky a jistoty poskytované služby. Záruka služeb je uváděna v procentech. Tyto záruky zákazníkovi uvádí, nakolik procent bude jeho služba dostupná. Někteří poskytovatelé uvádí číslo 99,9% v kterém nejsou zahrnuty různé výluky jako je údržba systémů, upgrade a kvalita zákazníkova hardwaru, takže jde vlastně a marketingový trik. Časové rozpětí SLA je nejčastěji stanoven na dobu jednoho měsíce, kdy je služba monitorována a většinou zákazník platí jednou měsíčně platbu za poskytované služby. Kapacita SLA je někdy přehlížena, protože je těžké ji určit. Musí se vzít v potaz všechny výkonnostní prvky výpočetní techniky. Na trhu se ale objevil nový trend kdy dodavatelé outsourcingových služeb při rozhodování zákazníkovy mohou doporučit hardware počítačů, operačních systémů a celkově ICT zařízení pro komunikaci. Pokud se zamyslíme nad dobou odezvy, tak poskytovatelé nemohou zaručit okamžitou odezvu, protože pokud se uživatel připojuje do databáze na druhé straně zeměkoule, je jasné, že odezva může být pár vteřin a podle toho je potřeba upravit i SLA. U SLA můžou také nastat nečekané události, u kterých 80% problémů bude vyřešeno okamžitě a efektivně on-line. Tyto problémy jsou specifikovány jako problémy zákazníka nebo neznalosti uživatele. Ostatních 20% mohou být závažnější chyby v systému, na které je potřeba delšího časového intervalu pro vyřešení. V pravidelně časových intervalech zasílá poskytovatel zákazníkovi reporty o stavu, nejčastěji to bývá s pravidelnou platbou, tedy jednou za měsíc.

Oba partneři, jak poskytovatel, tak zákazník by měli být motivováni, k vzájemné podpoře a rozvoji. Společně by měli těžit ze vzájemné spolupráce a strategicky dosahovat větší produktivity. Můžeme říci, že jasná definice SLA patří mezi nejdůležitější body outsourcingové smlouvy. [12]

4 Historie a charakteristika managed services

4.1 Definice

Managed services můžeme do češtiny přeložit jako řízené nebo spravovatelné služby. Někdy se používá i pojem managované služby. Tento pojem se u poskytovatelů služeb v oblasti ICT většinou nepřekládá.

Ve svých pravidlech datového provozu při poskytování služby přístupu k síti Český telekomunikační úřad definuje řízené služby jako: „služby elektronických komunikací, které umožňují přístup ke konkrétnímu obsahu, aplikacím nebo službám, případně jejich kombinaci, a jejichž technické vlastnosti jsou při jejich poskytování mezi koncovými body řízeny nebo, které umožňují odesílat či přijímat data jen mezi určitým počtem subjektů nebo koncových bodů. Zároveň nejsou nabízeny ani běžně používány jako náhrada za službu přístupu k síti internetu (např. IPTV).“

Poskytovatelé řízených služeb se často řídí definicí: *“Managed services is the practice of outsourcing day-to-day management responsibilities as a strategic method for improving operations and cutting expenses.”* V překladu bychom mohli definovat následovně: „Řízené služby jsou procvičování outsourcingu s každodenní odpovědností za řízení jako strategická metoda pro zlepšení fungování a snižování nákladů“.

Mnoho poskytovatelů řízené služby definuje jako: *“Managed services is the practice of outsourcing day-to-day management responsibilities as a strategic method for improving operations and cutting expenses.”* V překladu bychom mohli definovat následovně: „Řízené služby jsou procvičování outsourcingu s každodenní odpovědností za řízení jako strategická metoda pro zlepšení fungování a snižování nákladů“. [10][11][12][13]

Úřad ČTU stanovil následující pravidla a doporučení pro řízení datového provozu:

1) Svoboda výběru služby přístupu k síti internet a její kvality.
2) Nepřípustnost diskriminace, blokování nebo degradace jednotlivých datových toků.
3) Možné výjimky z pravidel č. 1 a 2.
4) Transparentnost informací
5) Nabídka služby přístupu k síti internet (služba bez omezení).
6) Nabídka služeb s limitem datového objemu.
7) Nabídka řízených služeb.

Více specifikací nalezneme v prohlášení ČTU na stránkách pro obecná pravidla řízených služeb.

U řízených služeb definujeme zákazníka nebo klienta, to je tedy ten, který odebírá řízené služby. Dále máme u řízených služeb poskytovatele, ten je definován jako podnik, který poskytuje službu zákazníkovi. V řízených službách poskytovatele jmenujeme MSP (Manager Service Provider).

Řízené služby se objevují v mnoha odvětví a můžeme říct, že jsou téměř všude:

- stavebnictví,
- webová bezpečnost,
- nákladní přeprava,
- poštovní a kurýrní služby,
- zálohování a obnova dat,
- bezpečnost,
- aplikační podpora,
- vzdálená podpora a dohled služeb,
- infrastruktura.

Řízené služby v těchto oblastech pomáhají rozvíjet a urychlovat procesy. Právě proto jsou řízené služby stále populárnější a dostávají se téměř všude.

4.2 Charakteristika

V ICT představují řízené služby systém služeb a funkcí, které zajišťují spolehlivý provoz řešení. Cílem není dodávka jednotlivých komponent, ale komplexního řešení formou řízené služby, která splní veškeré požadované potřeby. Kombinací provozní a technické podpory včetně sledování celkové výkonnosti řešení získává zákazník vysoce efektivní a přitom ekonomicky přijatelné služby. Pokud jsou tyto služby navíc koncipovány jako plně flexibilní a škálovatelné, mohou přinést zajímavý obchodní a provozní potenciál pro obě smluvní strany.

Přenesení každodenních zodpovědností spojených s provozem ICT na jiný subjekt má za cíl především zvýšení efektivity vlastního podnikání. Parametry a kvalita služeb je pevně dána SLA smlouvou. V této smlouvě jsou definovány parametry dodávky služby včetně toho, jakým způsobem bude měřena. Rovněž zde bývají zakotveny sankce pro případ nedodržení předepsané kvality služby. [10][11][12][13]

5 Dohledové systémy

Při budování počítačových sítí se prováděla i digitalizace a modernizace telekomunikační infrastruktury. Po této modernizaci nebyl až tak velký rozdíl mezi datovou a telekomunikační sítí, proto mohlo dojít ke sloučení obou sítí. Největší důraz se klade na rychlost a spolehlivost těchto služeb. Dnes si asi málokdo umí představit život bez televize, internetu, telefonu nebo sociálních sítí. Stále více lidí i firem je závislých na existenci internetu. Pokud dojde k výpadku internetu nebo části infrastruktury, nastává zde závažný problém. Ohledně větších firem, obchodů na internetu nebo e-shopů může dojít k velkým finančním ztrátám, za každou hodinu či minutu výpadku. Tyto organizace si však tuto hrozbu uvědomují a proto vznikly systémy na monitorování infrastruktury. Tyto systémy mají nelehký úkol. Sledovat infrastrukturu a v případě přicházejícího problému upozornit uživatele na poruchu.

Monitorování má na starosti dohledový systém, který může využívat řadu protokolů a technologií. Dohledový systém můžeme v první řadě rozdělit na bezplatné nebo placené řešení. Pokud si nezakoupíme komerční řešení, budeme spoléhat na bezplatnou aplikaci, kde budeme vytvářet vlastní skripty či programy. V tomto případě budeme potřebovat lepší znalosti pro nastavení systému, jako jsou skriptování, html kód, databáze, síťové protokoly a další technologie. Takto vytvořit dohledový systém bude časově náročné, ale poté bude 100% funkční. Velikou výhodou je komplexní prostředí (včetně konfigurace, dashboard, grafické zpracování, grafy). Dále můžeme vytvořit šablony, které bychom do budoucna mohli používat.

Naopak pokud zvolíme placené komerční řešení, tak můžeme očekávat program, který se nainstaluje během pár minut s minimální námahou pro rozchození systému. Stačí zadat do systému pár IP adres zařízení a údaje, jaké chceme monitorovat. Komerční řešení má k dispozici mnoho šablon, které nám zjednoduší práce, ale zároveň nás mohou omezovat v možnostech používání dohledového systému. [9]

5.1 Bezpečnost

Dobré zabezpečení dohledového systému, sledování sítě a provozu v síti by mělo být hlavní prioritou. Pokud nasadíme dohledový systém, tak si musíme uvědomit, že centralizujeme informace o síťovém provozu na jedno místo. To může být nebezpečné, pokud by se útočník náhodou dostal do dohledového systému. V tomto případě by útočník mohl odposlouchávat veškerý provoz, zároveň by se dostal k přihlašovacím údajům pro připojení zařízení a mohl je plně ovládat.

Jak zabezpečit dohledový systém, je výborná otázka. Nejlepším způsob je omezit počet uživatelů na co nejmenší a počet míst, podle IP adres, odkud se připojit. Ideální je mít dohledový server (dohledový systém) umístěný za firewallem počítačové sítě. Pokud bych se chtěl přihlásit na server, musím se autorizovat přes certifikát s heslem, např. pomocí VPN klienta, do sítě kde leží server. Tento přístup je řízený šifrovanou technologií a je bezpečný.

Bezpečné připojení mezi serverem a zařízením by mělo být provedeno opět šifrovaně pomocí např. SSH tunelu. Další zabezpečení komunikace server-zařízení je konfigurovat na zařízení ACL (access list), kde definujeme IP adresy, které se mohou přihlásit na zařízení pomocí SSH. Takto nastavené zařízení můžeme považovat za zabezpečené. [7][9][14]

5.2 Přehled sledovaných portů

Pro navázání spojení využíváme protokoly rodiny IP. Pro navázání spojení musíme znát typ protokolu, kde musí být definovaná IP adresa odesílatele, příjemce a samozřejmě číslo portu odesílatele i příjemce. Síťový provoz je realizován pomocí protokolů TCP/UDP a slouží k rozlišení služeb podle portů. Rozlišení čísla portu lze identifikovat dvoubajtovým číslem, které adresujeme porty v rozsahu 0 - 65535. Byl zřízen oficiální seznam čísel portů TCP a UDP, který stanovuje jednotlivým službám standartní číslo portu. Čísla portů přidělovala organizace IANA (Internet Assigned Numbers Authority) a od 21. března 2001 je touto funkcí pověřena organizace ICANN (Internet Corporation for Assigned Names and Numbers). [24]

Porty jsou rozděleny do tří základních skupin, dle rozsahů a specifikací:

1. Porty 0 – 1023 Well known ports – Známé porty
2. Porty 1024 – 49151 Registered ports – Registrované porty
3. Porty 49152 – 65535 Dynamic and private ports – Dynamické a privátní porty

Číslo	Protokol		Služba	Popis
4		udp	NTP	Network Time Protocol - protokol pro synchronizaci času
7	tcp	udp	ECHO protocol	Echo server poslouchá TCP nebo UDP
20	tcp	udp	FTP (data)	Network Time Protocol - protokol pro synchronizaci času
21	tcp	udp	FTP (příkazy)	File Transfer Protocol - režijní přenos
22	tcp	udp	SSH	File Transfer Protocol - režijní přenos
23	tcp	udp	Telnet	File Transfer Protocol - režijní přenos
25	tcp	udp	SMTP	Simple Mail Transfer Protocol - protokol pro přenos elektronické pošty
53	tcp	udp	DNS	Domain Name System - protokol sloužící pro výměnu informací o převodu doménových jmen a IP adres
69	tcp	udp	TFTP	Trivial File Transfer Protocol - protkol pro přenos souborů, základní funkce FTP
80	tcp	udp	HTTP	HyperText Transfer Protocol - protokol pro přenos hypertextových dokumentů
109	tcp	udp	POP2	Post Office Protocol - protokol používaný pro stahování elektronické pošty
110	tcp	udp	POP3	Post Office Protocol - protokol používaný pro stahování elektronické pošty
115	tcp		SFTP	SSH File Transfer Protocol - protokol pro zabezpečený přenos dat
123	tcp	udp	NTP	Network Time Protocol - protokol pro synchronizaci času
143	tcp	udp	IMAP	Internet Message Access Protocol - protokol pro vzdálenou správu emailové schránky
161	tcp	udp	SNMP	Simple Network Management Protocol - protokol pro získávání informací o síťových prvcích
443	tcp	udp	HTTPS	Hypertext Transfer Protocol Secure -nadstavba protokolu HTTP umožňující zabezpečit spojení
993	tcp		IMAPS, SSL	Protokol pro vzdálenou správu emailové schránky, SSL-Secure Sockets Layer - vložená vrstva k zabezpečení komunikace
995	tcp		POP3S, SSL	Protokol používaný pro stahování elektronické pošty, SSL-Secure Sockets Layer - vložená vrstva k zabezpečení komunikace
1433	tcp		MSSQL	MSSQL - Microsoft SQL Server
3306	tcp		MySQL	MySQL - databázový systém
3389	tcp		RDP	Vzdálená plocha v Microsoft Windows
5060	tcp	udp	SIP	Session Initiation Protocol - protokol pro inicializaci relací, určení pro přenos v internetové telfonii

Tabulka 1 Nejznámější porty v dohledových systémech [24]

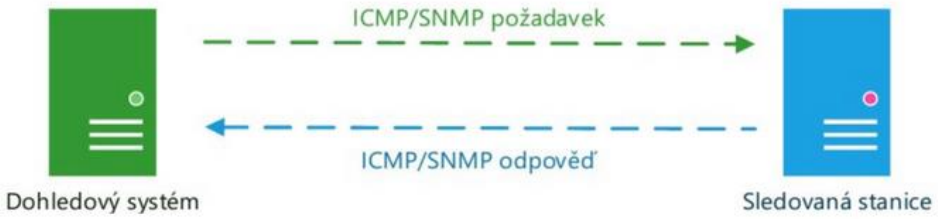
5.3 Způsoby dohledu

Důležitou funkcí systémů pro monitoring je zasílání informací ze sledovaných zařízení. Pokud sledujeme prvek bez pomoci lokálního agenta, využívá se jednoduché dotazování standartních síťových protokolů. Tyto protokoly jsou např. TCP/IP, ICMP, Telnet, SSH, SNMP, LDAP, FTP, IMAP, POP, DHCP,WMI atd.

V případě že disponujeme serverem, na kterém je naistalována MIB databáze dohledového systému, můžeme využít lokálního agenta k získání mnohem více informací, jako je např. vytížení CPU, zaplnění paměti RAM, stav zaplnění disků a mnoho dalších informací. Lokální agent může proaktivně spravovat dohledový systém, tedy zasahovat do běžného provozu systému a provádět potřebné operace. V případě výpadku služeb může např. restartovat služby a podobně. [9][19] [20]

5.3.1 Sledování pomocí síťových protokolů

Základní metodou je dotazovaní na stav zařízení pomocí standardizovaných protokolů jako je ICMP, TCP, apod. Tento způsob kontroly zařízení nepožaduje žádné konfigurační změny zařízení, pouze povolení výjimek firewallu pro komunikaci s daným zařízením. Dochází k cyklickému dotazování zařízení pomocí ICMP na druhé straně. Tato metoda přináší jednoduchost a univerzální kontrolu zařízení všech výrobců. Dokáže zjistit dostupnost zařízení okamžitě. Nevýhodou jsou omezené funkce dohledového systému. [9][19] [20]



Obr. 3 Příklad sledování pomocí síťových protokolů [9]

5.3.2 Sledování pomocí SNMP trap

Lepší možnosti využívá dohledový systém, který využívá SNMP oznámení, tzv. SNMP trap. Na rozdíl od prvního případu, kdy jsme cyklicky zasílali informace, jestli dané zařízení „žije“ a je v pořádku, zde nezasíláme cyklicky informace. Zařízení samo odešle dohledovému systému zprávu v případě nějaké nestandartní činnosti (chyby, výpadku

portu, výpadku internetu, zaplnění RAM či procesoru, vysoké vytíženosti sítě, atd.). Na základě získaných informací, pak dohledový systém reaguje a provádí další operace. Tento způsob se často využívá s kombinací cyklického zasílání dotazů. Kombinací zvyšujeme efektivitu dohledového systému. [9][19] [20]

Obr. 4 Příklad sledování pomocí SNMP trap [9]

5.3.3 Sledování zařízení pomocí instalovaného agenta

Třetí možnost získání dat, je nasazení speciálního agenta, který pracuje na koncovém zařízení jako aplikace a sbírá informace o daném zařízení. Komunikace mezi zařízeními funguje na principu klient-server. Pro přenos mezi zařízeními se používá standartní protokol TCP/IP. Velikou výhodou je přístup k velmi detailním informacím o sledovaném systému. V případě proaktivního dohledového systému je možnost sledované zařízení ovládat a konfigurovat nastavení sítě. Tento model sledování zařízení se využívá typicky pro monitoring serverů. [9][19] [20]

Obr. 5 Příklad pomocí nainstalovaného agenta [9]

5.4 Používané protokoly dohledových systémů

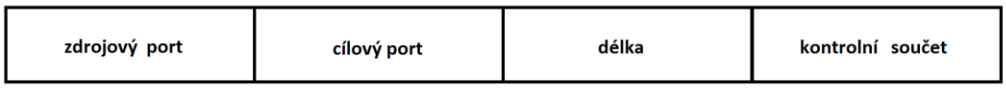
Dohledové systémy používají pro svoji činnost mnoho síťových protokolů. V následující kapitole se budu snažit nejpoužívanější protokoly základně popsat.

5.4.1 Protokoly transportní vrstvy

Síťová vrstva zajišťuje adresaci a směrování datagramů, zároveň ale nezajišťuje doručení datagramů na cílovou adresu nebo čas i pořadí doručení datagramů. Služba nabízená vyšším vrstvám je spolehlivá nebo nespolehlivá (TCP/UDP). V TCP/IP modelu je známá jako transportní vrstva a odpovídá transportní vrstvě u OSI modelu (poskytuje komunikaci pro koncový přenos dat mezi dvěma stanicemi). Transportní službu nabízí se spojením nebo bez spojení pomocí protokolů TCP nebo UDP protokolu.

5.4.2 UDP

Je velmi jednoduchý protokol transportní vrstvy. Některé protokoly upřednostňují rychlost doručení paketů a na spolehlivosti jim tolik nezáleží. Takový je přesně protokol UDP (User Datagram Protokol), který nabízí nespolehlivou transportní službu bez spojení pro aplikace, které nepotřebují zabezpečení protokolu od transportní vrstvy v tak velikém rozsahu jako má protokol TCP. Protokol UDP je využívám řadou služeb v DNS, DHCP, SNMP a RIP. Aplikace, kde využíváme protokol UDP je např. VoIP, real-time onlinové hry, internetová radia, a další. Protokol UDP je pro tyto aplikace vhodnější, protože nemění kvalitu přenosových služeb. Protokol UDP je bezstavový a zdrojový port nemusí být vyplněn, může být definován jako 0. Ke službám protokolu IP přidává jen kontrolní součty a port konkrétní aplikace. Kontrolní součet můžeme vynechat. Největší výhodou toho to protokolu je rychlost a efektivity. [14][15]



Obr. 6 Hlavičku protokolu UDP

5.4.3 TCP

Většinou tento protokol známe ve spojení TCP/IP (kde TCP je Transmission Control Protocol a IP je Internet Protocol. Technologickým základem komunikace dnešního internetu jsou tyto dva protokoly TCP/IP. TCP je základním síťovým protokolem, který je popsán v dokumentu RFC 793 (Information Sciences Institute University of Southern California, 1981). TCP je plně transparentní protokol. TCP je spojově orientovaný protokol pracující na transportní vrstvě ISO/OSI modelu. Jeho úkolem je zajistit potvrzené doručení paketů, které navíc budou ve správném pořadí. Tato vlastnost s sebou přináší značnou nevýhodu v nízké rychlosti procesu doručení dat.

Na transportní vrstvě existují dva druhy protokolů TCP a UDP (User Datagram Protocol). Jednotlivé služby si mohou vybrat, jestli budou používat (rychlejší), ale nespolehlivý protokol UDP nebo budou používat spolehlivý (pomalejší) protokol TCP. Úkol protokolu TCP je poměrně náročný, protože musí zajistit spolehlivé služby, i když má nespolehlivé prostředky. Pro každý přenesený paket je vyžadována značná režie na potvrzení nebo seřazení paketů a další. TCP se snaží nabízet své vyšší vrstvě přenos po jednotlivých bytech a ne jako celek, jak je tomu u UDP. Proto není TCP protokol vhodný pro všechny typy přenosů jako je video, hlas, terminálové relace a další. TCP ke své práci využívá protokoly vyžadující spolehlivou transportní vrstvu jako je FTP (File Transfer Protocol) nebo TELNET (TELecommunication NETwork). Protokol TCP je úzce spjatý s protokolem IP. Tyto dva protokoly spolu tvoří základní komunikační sadu v datových sítích. [19]

5.4.4 IP

Početná rodina protokolů TCP/IP má také svůj základ, na kterém staví a to je protokol IP (Internet Protocol). Protokol IP je specifikován a popsán v dokumentu RFC 791 vydaném v roce 1981. Jde o jeden z nejstarších sítových protokolů, které se dnes a denně využívá. Protokol IP je nejdůležitější protokol síťové vrstvy. Je to základní protokol pro komunikaci počítačových sítí. Má za úkol zajistit přenos paketů z jednoho uzlu sítě do druhého. Zároveň protokol IP negarantuje doručení paketů a ani nezaručuje správné pořadí doručení paketů. Protokol IP zabraňuje nestandartním situacím, jako je zacyklení. Pro realizaci přenosu dat využívá protokol tzv. IP adresu, která slouží jako jednoznačný identifikátor pro každý uzel dané sítě. IP adresu v privátní počítačové síti rozdělujeme na veřejné a neveřejné adresy, podle RFC 1918 a RFC 4193.

Protokol IP řídí své vysílání formou datagramů na základě síťových adres obsažených v jejich záhlavích a síťovou službu poskytuje bez spojení. Datagram je samostatná datová jednotka, která obsahuje základní informace o adresátovi a odesílateli. Pořadí datagramů je nahodilé a nezaručuje doručení.

Ve veřejné počítačové síti musí být tento identifikátor jedinečný. Využívají se současně dvou verzí IP protokolů. Verze 4 (IPv4) a verze 6 (IPv6). Díky velkému množství připojovaných zařízení a k identifikaci veřejné adresy každého zařízení muselo dojít k revitalizaci protokolu na novější IPv6.

Obr. 7 TPC/IP model a vedle něho model ISO OSI

Protokol IP můžeme definovat jako nadstavbu nad přenosovou technologií jako je Ethernet, Token Ring, FDDi, ATM a další. Tyto odlišné technologie využívá protokol IP, proto, abychom všude mohli poskytovat jednotné služby, kvalit i vlastností.

IP adresa zařízení se navíc mění v každé podsíti, ale jen za předpokladu, že tato adresa není veřejná. Veřejná adresa je unikátní napříč všemi podsítěmi, a proto se nemusí překládat u přechodu z jedné podsítě do druhé. [9][19]

5.4.5 IPv4

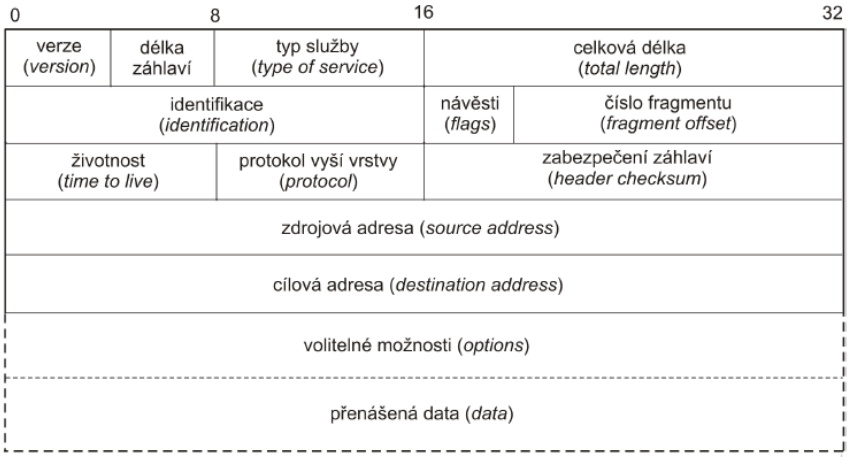
Starší verze protokolu, ale stále nejrozšířenější protokol. IPv4 je čtvrtá verze protokolu navržená na konci 70. let 20. století. Tato verze pracuje s 32-bitovou maskou, kde můžeme přiřadit až 2^32 adres. To odpovídá asi 4 miliardám IP adres. Příklady IP adres můžeme vidět v tabulce 2.

Označení RFC1918	Rozsah IP adres	Počet adres	Největší CIDR blok (maska podsítě)	Pro síťové rozhraní
24-bitový blok	10.0.0.0 - 10.255.255.255	16,777,216	10.0.0.0/8(255.0.0.0)	24 bitů
20-bitový blok	172.16.0.0 - 172.31.255.255	1,048,576	172.16.0.0/12 (255.240.0.0)	20 bitů
16-bitový blok	192.168.0.0 - 192.168.255.255	65,536	192.168.0.0/16 (255.255.0.0)	16 bitů

Tabulka 2 Znázorňuje normu RFC 1918 s rozsahem IP adres

IP adresa obsahuje dvě základní části, řídící část (záhlaví datagramu) a datovou část. Do datagramu se v případě vlastního přenosu vkládají data do části rámce, s touto částí dále pracuje nižší vrstva. Záhlaví je maximálně 60 oktetů dlouhé s minimální délkou 20 oktetů. V záhlaví jsou uloženy důležité data pro doručení.

IPv4 má záhlaví IP datagramu dlouhé 4 bity, binární hodnota je rovna 4. To odpovídá délce záhlaví jako počet 32 bitových slov, tedy 60 oktetů a tím je definována maximální délka záhlaví u IPv4. Formát datagramu protokolu IPv4 můžeme vidět na obrázku. [9]



Obr. 8 Formát datagramu v protokolu IPv4

5.4.6 IPv6

Novější verze protokolu IPv6 byla vypracována v roce 1995 z důsledku docházejících IP adres. Většina IP adres byla rozprodána, nebo zarezervována a do budoucna by hrozilo, že nebude žádná adresa „volná“. Vzrůstající spotřeba IP adres souvisí s rozvojem nových počítačových sítí, připojování k internetu a k velkému počtu připojovaných inteligentních bezdrátových koncových stanic. Tak jako IPv4, tak i IPv6 poskytuje datagramovou službu. Odlišení je v adresaci a formátu datagramu. IPv6 dále potřebuje novější verzi protokolů, např. ICMPv6 (Internet Control Message Protocol), také směrovací protokoly a změnu i v aplikační vrstvě.

Největší výhodou IPv6 je obrovský adresní prostor. IPv6 adresa má 128 bitů a je o 4 bity větší než IPv4 adresa. Rozsah IPv6 adres potom pokrývá prostor cca $6{,}67*10^{23}$ adres na jeden čtvereční metr povrchu země nebo 36 165 koncových sítí na jeden čtvereční metr povrchu země, kde v každé sítí může být až $1{,}84*10^{19}$ koncových

zařízení. Uvedené hodnoty jsou pouze teoretické, protože dle specifikace oprávněná organizace bude používat uvedený prefix 48 bitů (/48). Menší prefix 64 bitů lze je ve speciálních případech.

IPv6 obsahuje mechanismy automatické konfigurace IP adres pro koncová zařízení. Po připojení počítačové stanice do sítě, pomocí ICMP zpráv od routeru zjistí prefix sítě, výchozí bránu a potřebné nastavení sítě. Stanice ihned obdrží svoji unikátní veřejnou IP adresu a může se připojit do internetu. Pouze zde chybí doladění DHCPv6, kterou si musí ohlídat, neboli nakonfigurovat správce dané počítačové sítě.

IPv6 adresa je každá adresa unikátní a tedy veřejná adresa. Zápis IPv6 adresy byl zvolen v šestnáctkové soustavě, 128 bitů (16 bytů) je rozděleno po dvou bytech dvojtečkou. Ukázka zápisu globální individuální adresy. (obdoba IPv4 veřejné adresy).

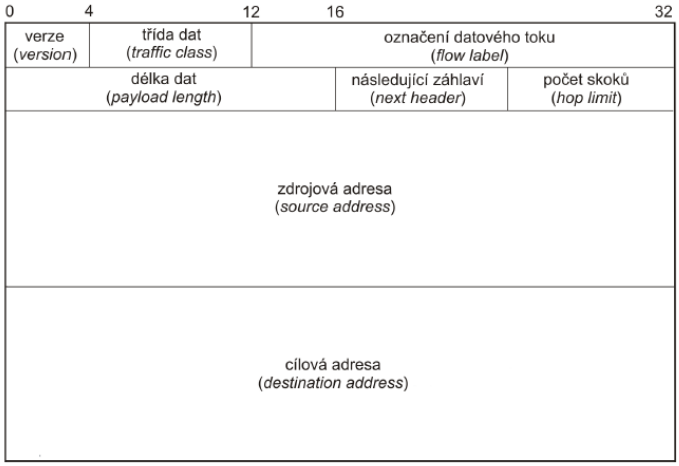
Příklad IPv6: 1101:0de1:db00:00d3:0000:0000:0000:0006

Existuje několik pravidel pro lepší zápis IPv6 adresy. Jde především a zkrácení a úpravou adresy:

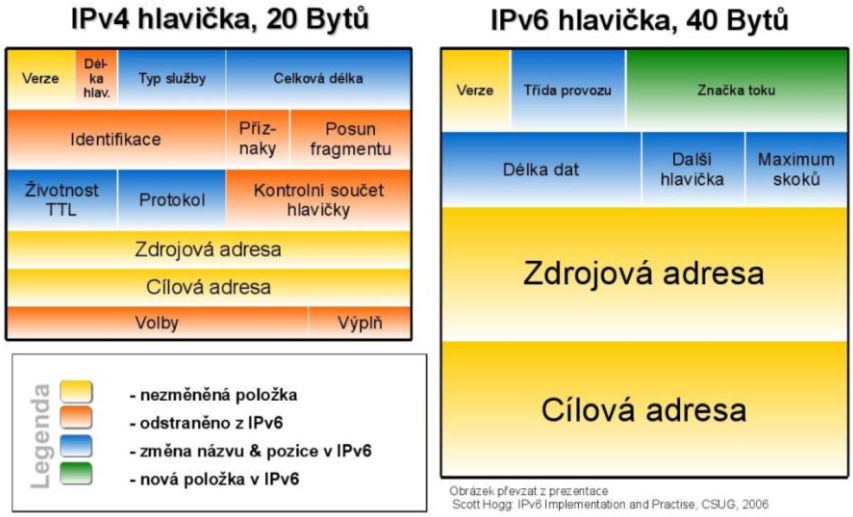
1) Počáteční nuly v každé dvojici bytů lze vynechat: 2001:1488:0:3:0:0:0:2

2) Sousedící nulu lze nahradit dvojitou dvojtečkou :: (příklad: 2001:1488:0:3::2)

Datagram IPv6 efektivně přesouvá informace do volitelných rozšířených záhlaví. IPv6 má povinné záhlaví fixní délky 40 oktetů, ale obsahuje pouze 8 polí. Za povinnými záhlaví mohou následovat další volitelná záhlaví, které jsou proměnné délky a pracovávají se až v koncových uzlech nebo na směrovači. [9][18]



Obr. 9 Formát datagramu protokolu IPv6



Obr. 10 Porovnání protokolů IPv4 a IPv6 [18]

5.4.7 ICMP

Protokol ICMP (Internet Control Message Protocol) je nejdůležitější protokolem pro funkci internetu. Standardně neslouží k přenášení dat, ale využívá se k zasílání chybových nebo inicializačních zpráv (např. u operačního systému, zřízení komunikace v síti a podobně). Specifikace toho to protokolu je popsána pod RFC 792 (Postel,1981). ICMP zprávy se generují např. při chybových událostech v počítačové síti, jako je vypršení doby TTL (time to live), nedostupnosti koncové stanice a podobně. Tyto ICMP zprávy jsou velmi důležité pro aktivní prvky sítě, protože podle těchto zpráv mohou přizpůsobovat komunikaci sítě. ICMP zpráva putuje v síti pouze v jednom datagramu a to celou komunikaci zrychluje.

Nejčastěji používané ICMP zprávy:

Echo Request	- požadavek na odpověď (známe jako ping)
Echo Reply	- odpověď na požadavek
Destination Unreachable	- informace o nedostupnosti cíle
Redirect	- přesměrování, pokud vede k cíli lepší cesta než přes bránu
Time Exceeded	- nedostupnost, vypršel časový limit

Tento protokol má hodně utilit, které jsou velikou výhodou pro administraci počítačových sítí (např. využití má v dohledových systémech). Velice známý příkaz, který využívá ICMP zprávy je např „ping“ a za ním IP adresa. Tento příkaz vysílá zprávy „echo request“. Utilitka „traceroute“ testuje počet směrovačů k cíli, tedy ke každému posílá zprávy „echo request“ s postupným inkrementováním doby života TTL. Pokud se doba TTL rovná 0, znamená to, že na výchozí stanici zašlou ICMP zprávu o nedoručitelném paketu. [9][14]

Obr. 11 Test ICMP zpráv pomocí utilitky „tracert“

5.4.8 SNMP

SNMP je protokol navržený speciálně pro správu a dohled síťových zařízení. Tento protokol je popsán v dokumentech RFC 3411 a RFC 3418. Jde o standardizovaný jednoduchý protokol aplikační vrstvy z pohledu ISO/OSI, který má za úkol periodicky zasílat data, informace o stavu počítačové sítě. SNMP (Simple Network Management Protocol) je protokol na kterém je dnes založena většina nástrojů pro správy počítačové sítě. Jeho úkol je získávání nebo nastavování hodnot na určitém zařízení. Tyto úlohy se

provádí vzdáleně v počítačové síti. Podporu SNMP má mnoho zařízení např. aktivní síťové prvky, počítačová čidla, tiskárny, přístupové body nebo speciální pomocný SW, který mají počítače či servery. Nasbírané hodnoty za určitý časový interval se nasbírají do databáze a následně se mnou vykreslit pomocí grafu. Tímto způsobem se dá přehledně zobrazit vytížení procesoru, teplota, datový tok upload či download nebo vytíženost dané počítačové sítě.

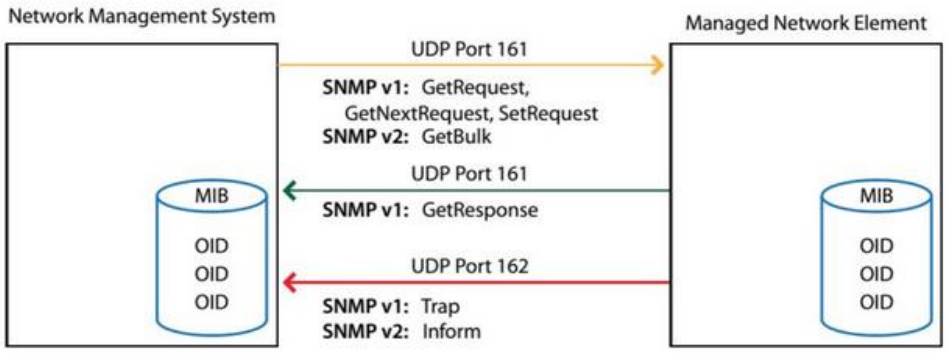
SNMP funguje na bázi dvou stran, vlastně jako client-server, ale jedna strana musí být správce (manager - server) a druhá musí být client (agent). SNMP pracuje ve dvou režimech.

1) Správce posílá dotazy agentovi a přijímá odpovědi (může být i několik správců a mohou se ptát kdykoliv).

2) Agent zasílá oznámení (trapy) na adresu správce (v pravidelných intervalech nebo ve zvláštních případech – nestandartní situace).

Agent je realizovaný aplikací „démonem“ snmd, který naslouchá na portu UDP 161. Na tomto prvku agent sleduje zařízení počítačové sítě, jako jsou servery, switche, počítače, tiskárny a další aktivní prvky sítě.

Manager je taktéž realizován démonem „smtptrapd“, který naslouchá na portu UDP 162 a běží na monitorovaném serveru. Server zasílá žádosti z dynamického portu a agent pak na ně odpovídá. To znamená, že pro každý požadavek může běžet jiný port komunikace. Díky využívání protokolu UDP v SNMP je komunikace velmi rychlá, ale zároveň nespolehlivá kvůli ztrátě dat a vlastnosti protokolu UDP. Právě z toho to důvodu je od verze SNMPv2 implementována kontrola doručení.



Obr. 12 Proces komunikace podle protokolu SNMP [9]

První standard protokolu SNMP pochází již z roku 1988. Od té doby prošel několika změnami a současně existují 3 verze.

5.4.8.1 SNMPv1

První verze se moc neujala díky špatnému zabezpečení. Heslo (community string) zde bylo, ale zasílalo se v nešifrovaném tvaru jako „plain text“.

5.4.8.2 SNMPv2

Navazuje na verzi 1, kde podle RFC 1441 a RFC 1452 se snaží zlepšit stabilitu, výkon a bezpečnost. Došlo zde k razantnímu zlepšení. Byl přidán „GetBulkRequest“, který dovolí managerovi získat více dat od agenta a mít větší kontrolu, co se posílá agentovi. Dále přidán „InformResponse“, to je vlastně Trap s potvrzením (past na události – nahlášení významné události). Dále vznikají další odvození od SNMPv2 jako je SNMPv2c a SNMPv2u, které poskytují další zlepšení protokolu. Tyto myšlenky byly dále implementovány v další verzi SNMPv3.

5.4.8.3 SNMPv3

Dochází k dalšímu významnému zlepšení protokolu v oblasti bezpečnosti, kde SNMPv3 přináší možnost ověřování jménem, heslem a zároveň je možné celou komunikaci šifrovat. Šifrování přenášených dat má na výběr mezi DES a AES. Autentizace je dělána pomocí šifrování hesla MD5 nebo SHA1. [9][20]

SNMP operace

Posílá manager

- GetRequest – posílá manager, úvodní zpráva komunikace (specifikuje vyžadované info.)
- SetRequest – posílá manager, určuje agentovi, jakou hodnotu má nastavit
- GetNextRequest – posílá manager, vyžaduje následující záznam ze seznamu
- GetBulkRequest – posílá manager, umí získat více hodnot jedním dotazem (od ver. 2)
- InformRequest – posílá manager, je to výměnu informací mezi managery (od ver. 2)

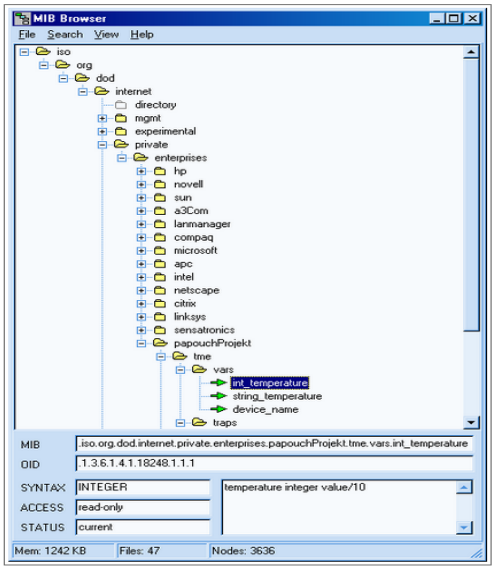
Posílá agent

- Response – posílá agent, odpověď na dotazy managera,
- Trap – posílá agent (speciální typ) zasílá se při vyskytnutí specifických událostí, není třeba dotaz managera.

5.4.9 MIB databáze

„MIB (managent information base) je hierarchicky strukturovaná databáze, která se skládá z objektů, které definují vlastnosti zařízení.“[21] MIB databáze má význam hierarchické datové struktury, která charakterizuje strom. Databáze popisuje sadu objektů, které jsou předmětem správy. Konkrétní zařízení může být připojeno k jedné nebo více MIB, v závislosti na jeho funkci. MIB dovoluje jednoznačně identifikovat systém správy pomocí datového identifikátoru OID (Object Identifier) a předat informace pro SNMP managera a agenta. Pro předání informací je nutnost znalosti struktury MIB. Pravidla MIB jsou popsány podle SMI (Structure of Management Information) dle specifikací dokumentů RFC1155, RFC 1212 a RFC1215.

Začátek této struktury je nepojmenovaný kořen, označován tečkou, pod kterým se nacházejí další uzly. Tyto uzly nebo-li kořeny tvoří celou stromovitou strukturu MIB databáze. Identifikátor OID, zvaný uzel je složen z posloupností čísel, která jsou oddělena tečkou. Hodnota OID vznikne tak, že OID nadřazeného prvku, doplní se tečka a aktuální číslo. Takto uložená celá stromová struktura je v MIB databázi. MIB databáze obsahuje navíc popisy a jména jednotlivých hodnot OID. V rámci stromu MIB jsou označení unikátní, protože slouží k směrování SNMP dorazů.



Obr. 13 Ukázka MIB stromu v aplikaci „MIB Browser“[20]

Příkladem OID může být třeba hodnota 1.3.6.1.2.1.2.2.1.6.1, které odpovídá textová verze z MIB databáze:

iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifPhysAddress.

MIB Browser

Pro prohlížení databáze MIB můžeme využít šikovný software „MIB Browser“, který umožnuje předávání MIB souborů a dotazů na zařízení.

Základní rozdělení MIB do pěti oblastí

- ***Configuration Management***
  Oblast obsahuje jména všech zařízení na síti, jejich charakteristiky a aktuální status. Umožňuje administrátorovi uvidět celkové fyzické rozložení sítě.
- ***Performance Management***
  Určuje efektivní užití sítě a poskytuje informace požadované pro výkonnostní analýzu. Umožňuje administrátorovi monitorovat dostupnost, čas odezev, průchodnost a užití jednotlivých prostředků.
- ***Fault Management***
  Detekuje, isoluje a případně opravuje vzniklé problémy.
- ***Security Management***
  Řídí a chrání dostupnost informací na síti.
- ***Accounting***
  Umožňuje měření využití jednotlivých komponent.
  [20][23]

5.5 Využití dohledových systémů

Monitoring a dohled IT systému je nezbytností, ale většinou chybí jejich řádné propojení. Propojením těchto dvou bloků, monitoringu a dohledu IT získáme rychlejší spolupráci a zvýšení zisků společnosti. Nyní provedeme rozdělení dohledu.

5.5.1 Technický dohled

Je nejpoužívanější typ dohledového systému, který umožňuje sledovat funkčnost určitého hardwaru. Sleduje zařízení, jestli je zapnuté a pracuje správně. Nejčastěji to jsou síťové komponenty (přepínače, switche, firewally a další). Dále to mohou být čistě hardwarové komponenty (servery, tiskárny, sítové úložiště a další). Tyto informace zaznamenává dohledový systém a dále podle nastavení reaguje a informuje správce dohledového systému.

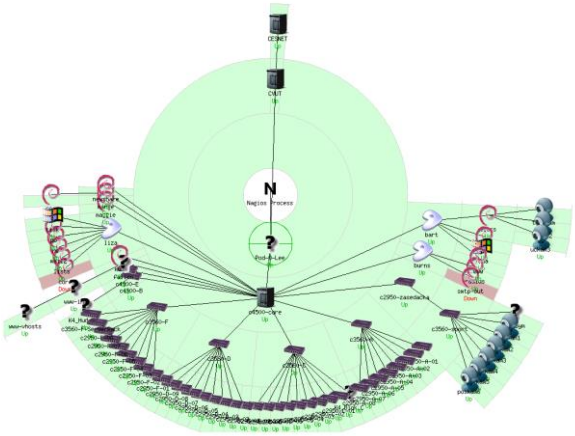
5.5.2 Dohled služby

Umožňuje pokročilý dohled služby pro sledování určitých služeb v zákazníkovi síti. Dokáže pomocí servisního modelu mapovat výpadky technických komponent na síťových službách a business službách. Tento dohled může sledovat a testovat služby např. zákazníkova e-shopu. Testováním se myslí provádění nakupování a odesíláním dat.

5.5.3 Dohled transakcí

Na nejvyšší úrovni dohledu sledujeme časově kritické aplikace, kde sledujeme doby trvání časů transakcí a následně poskytujeme provozovateli data, kterými ověřujeme jejich dobu trvání. Tento dohled používáme u zpracování objednávek např. u e-shopu. Vytvoříme objednávku, dáme odeslat a pak čekáme na zpětnou vazbu, zda naše objednávka byla provedena správně.

5.5.4 Servisní model



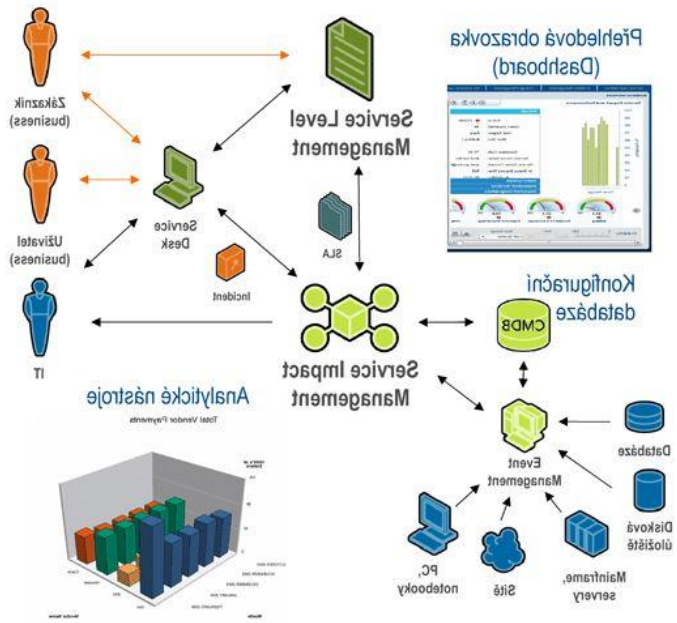
Obr. 14 Model sítě v systému Nagios

Tento model popisuje vzájemnou vazbu jednotlivých prvků sítě. Mapuje všechny prvky a rozděluje je do určitých skupin. Např. do skupin hardware, software, síťové prvky, služby a další. Při mapování těchto skupin servisní model popisuje jejich funkci a vliv na zákazníkovi služby, kde definuje jednotlivé procesy v podniku. Proto můžeme lehce rozpoznat, v jaké skupině se stala chyba a jak tento problém začít řešit. Servisní model umí sbírat data a může plánovat, jaké změny v servisním modelu nebudou dělat problémy. Na základě analýzy těchto nasbíraných dat, provede zodpovědná osoba (správce systému) patřičné opatření. Takto navržený

servisní model dodává dobře zpracované informace pro všechny články (lidi) na úrovni řízení systému, jako jsou technici, správci, operátoři, obchodníky až po ředitele společnosti.[22]

5.6 Funkčnost systému

Dohledový systém se nachází na jednom místě, kde zachytává zasílané informace. Funguje jako centrální uzel. Zde se sbírají veškerá data. Jednotlivé informace se filtrují a došlé události jsou zaslány jednotlivým IT týmům. Řešitelským týmům říkáme helpdesk nebo service desk. Jednotlivé události nebo-li „events“ proudí k administrátorům helpdesku. Události se v systému čítá opravdu hodně a operátoři je nestíhají zpracovávat. Zpravidla administrátoři nemají čas na prohlídnutí všech událostí. Pouze pokud nastane nějaký závažnější incident (výpadek, chyba), tak se administrátoři podívají do logů a situaci řeší. Každý incident má definovanou dobu SLA, časového limitu do kdy musí být problém vyřešen. [22]



Obr. 15 Schéma dohledového systému svázaného s monitoringem a SLA[22]

5.7 Dělení dohledových systémů

Další rozdělení dohledových systémů provedeme na základě hloubky monitoringu infrastruktury. Dohledové systémy můžeme rozdělit na základní, pokročilé, proaktivní a sledující datový tok.

5.7.1 Základní dohledové systémy

Tyto systémy vyhodnocují periodicky stav dostupných zařízení na základě dotazů. Vyhodnocují pouze stav dostupný/ nedostupný a také informaci o odezvě zařízení. Základní dohledové systémy využívají protokol ICMP. Tento systém je omezený svými funkcemi a je vhodný pro menší sítě typu LAN, nebo u sítí, kde zařízení není schopné předat více informací. Administrátoři sítě kontrolují pouze dostupnost zařízení.

5.7.2 Pokročilé dohledové systémy

Pro počítačové sítě většího rozsahu LAN nebo WAN je lepší pro správu zařízení použít pokročilejší dohledový systém. Tento pokročilejší dohledový systém využívá propracovanějších síťových protokolů, jako jsou např. SNMP, CDP, SSH a dalších. Pro představu pokročilejší dohledové systémy sledují téměř všechno dění v počítačové síti. Sledují dostupnost zařízení, vytížení sítě, stav spuštěných služeb, aktuální datový tok a další komunikaci. Pokud máme nainstalovaný dohledový systém na serveru, tak využíváme programů, tzv. agentů, kteří shromažďují data síťových protokolů. Z těchto informací vyhodnocuje dohledový systém nestandartní situace v síti a ve většině případů upozorní správce sítě na budoucí problém, který by mohl nastat v síti. Administrátoři tedy vědí o útoku dříve než by nastal. Díky pokročilejšímu dohledovému systému má správce sítě maximum možných informací o dění v počítačové síti.

5.7.3 Proaktivní dohledové systémy

Mají stejnou funkci jako pokročilé dohledové systémy, pouze umí navíc zařízení v síti dálkově ovládat. Tento typ systémů využíváme ve vysoce automatizovaném prostředí, jako jsou např. rozsáhlé počítačové sítě, cloudová řešení, datacentra, vysoce dostupné clustery a další. Systém můžeme specifikovat pro používání v nejvyšších vrstvách sítě a u nejrozsáhlejších počítačových sítí. V posledních letech s vývojem cloudových služeb se tyto systémy začínají hodně vyskytovat. Výrobci se tomuto trendu snaží přizpůsobovat. Velká výhoda je především implementace předem nadefinovaného scénáře, kdy systém automaticky reaguje na daný scénář. Dochází zde k velké úspoře času, tím i nákladů a k celkovému zlepšení kvality poskytovaných služeb.

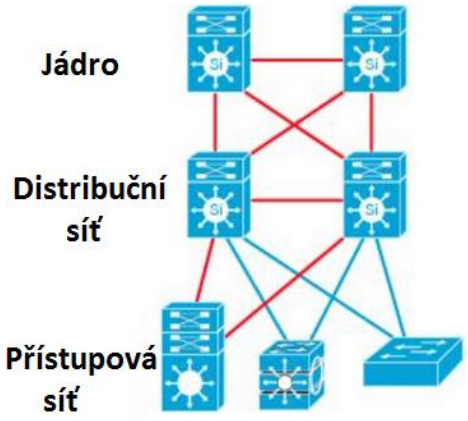
5.7.4 Systémy sledující datový tok

Tato kategorie systémů pracuje na bázi odposlechu v celé počítačové síti. Jde čistě o síťový monitoring. Tento typ dohledu je značně náročný na výkon a zejména zázemí. Odposlech není možné provádět z jednoho místa, ale musí být několik bodů odkud monitorovat síť. Pokud se používá tato technologie, jsou využívány speciální „inteligentní přepínače“, které zrcadlí veškerý datový tok na další port, který je připojen přímo na přenosové medium. Přenosové jednotky po určitých částech přeposílají data do centrálního prvku, který se stará o uložení, analýzu a grafickou prezentaci uživatelům. Uživatelé si mohou tyto data prohlížet. [9]

5.8 Topologie dohledových systémů

Každá datová sít může být v podstatě unikátní, i přes fakt, že by měl existovat nějaký obecný popis, jakou danou architekturu datové sítě použít. Musíme vzít ohled na robustnost, rychlost, spolehlivost přenosu dat a taky cenu navrhované datové počítačové sítě.

Důležité jsou základy velké sítě, proto je dobré zavedení třívrstvé hierarchické sítě, která je uvedená na obrázku 35. Zavedení hierarchie a členění sítě do vrstev umožňuje a usnadňuje její budoucí růst, významně napomáhá k jednoduššímu směrování, adresaci a samostatnosti jednotlivých částí a bloků sítě.



Obr. 16 Hierarchická představa WAN sítě [9]

5.8.1 Jádro

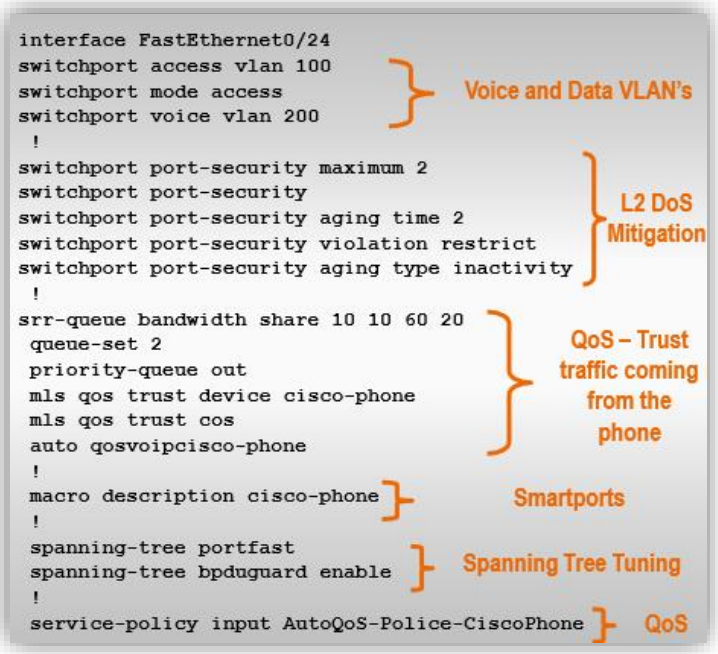
Jedná se o nejvyšší vrstvu - páteřní síť. Vrstva má za úkol co nejrychleji směrovat pakety a dosáhnout nejvyššího možného výkonu. Tato vrstva nemá velké nároky na konfiguraci z důvodu rychlosti, kdy požadavek na vrstvu není analýza datového toku, ale pouze přenos co nejrychlejšího doručení paketu. Rozšířitelnost této vrstvy je méně nákladná, protože zde není povinnost propojení s distribuční sítí. Pokud dojde k výpadku sítě, má to kritický dopad. Dochází k přerušení přenosu dat a ztrátu spojení.

5.8.2 Distribuční síť

Prostřední vrstva odděluje jádro od přístupové sítě. Jejím hlavním úkolem je oddělení-izolace, agregace sítí, řízení a omezení toky atd. Vrstva využívá směrování, ale není zde kladen důraz na konvergenci sítě.

5.8.3 Přístupová síť

Nejnižší vrstva je přístupová. Zde dochází do styku uživatele v síti. Tuto vrstvu je potřeba maximálně zabezpečit a omezit přístup uživatelům, kteří nemají oprávnění do sítě. Tuto část sítě spravuje nejčastěji IT specialista, který uživatele rozdělí do příslušných VLAN, nastaví QoS (Quality of Service) a celkově přizpůsobí síť uživatelům. Tyto operace se provádí na úrovni L2, ale čím dá tím častěji i úrovni L3, díky L3 přepínačům.



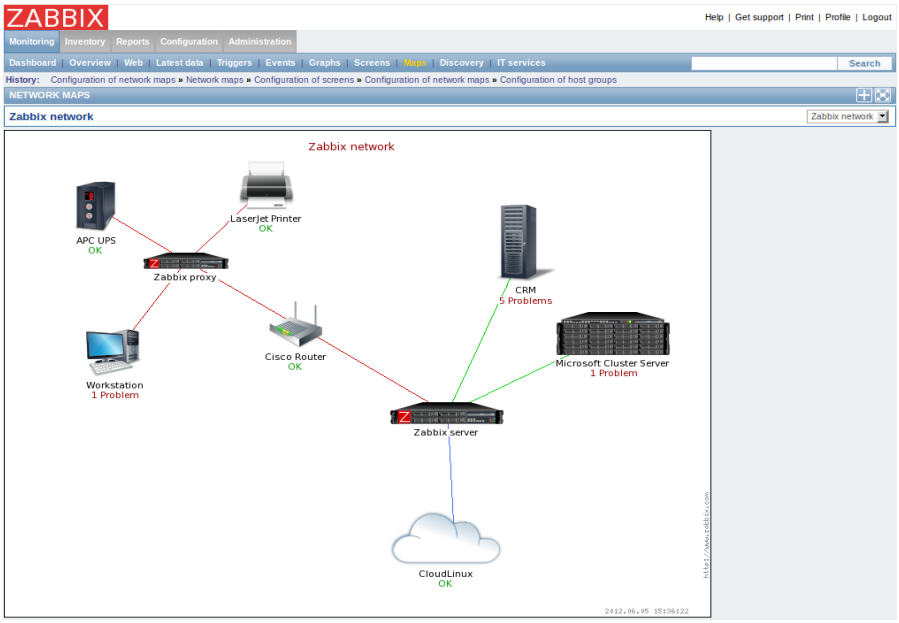
Obr. 17 Nastavení switche v přístupové síti [25]

6 Typy dohledových systémů

Zaměřil jsem se na výběr nejpopulárnějších dohledových systémů. Mým úkolem je prozkoumat možnosti systémů, jak pracují a co všechno umí. V krátkosti se je pokusím popsat. Podle toho to průzkumu zvolím nejvhodnější dohledový systém.

6.1 Zabbix

Mezi opensource nástroje patří dohledový systém Zabbix. Tento dohledový systém je šířen pod GPL (GNU General Public License) licencí. Zabbix se stává populární mezi uživateli i ve větších firmách, protože je jednoduchý a přehledný. Alexei Vladishev je autorem toho to dohledového systému. Na systému začal pracovat od roku 1998 a v roce 2001 byl Zabbix vyvíjen pod licencí GPL. Dále v roce 2004 byla vypuštěna stabilní verze 1.0 a nyní se využívá verze 2.4.6 (rok 2015). Systém je překvapivě výkonným a flexibilním řešením. Samozřejmě je možné si dokoupit placenou podporu pro Zabbix.



Obr. 18 Schéma síťových prvků v dohledovém systému Zabbix[26]

Dohledový systém Zabbix slouží k monitorování aktivních síťových prvků, jako jsou počítače, UPS, servery, routery, switche, kamery, modemy a další zařízení, které jsou připojeny k počítačové síti. Zabbix umí monitorovat zařízení, která mají IP adresu. Pro správu systému využíváme k dispozici webové rozhraní. Webové

rozhraní není na první pohled úplně přehledné, vyžaduje to důkladný průzkum a znalost systému. To samé platí i o vygenerovaných grafech. Zabbix umožňuje implementace SLA, sledování IT infrastruktury a mnoho dalších funkcí.

Metody sledování, které umožňuje Zabbix:

ICMP echo request (ping)
SNMP (Simple Network Management Protocol)
IPMI (Intelligent Platform Management Interface)
JMX (Java Management Extensions)

Dále můžeme využít připojení vzdáleného přístup přes SSH, Telnet nebo použít agenta, který je přístupný pro většinu operačních systémů. Agent umí monitorovat stav hardware (operační paměť, procesor, úložné zařízení,...). Do prostředí můžeme aplikovat svoje vlastní externí skripty, nebo můžeme využít API (application programming interface). API je programovatelní interface, kde můžeme naprogramovat aplikaci.

Podle tvůrců, Zabbix podporuje monitoring přes 100 000 hostů a provádí přes 1000 000 vyhodnocení/minutu. Tyto údaje jsou samozřejmě závislé na hardwarovém vybavení serveru, na kterém Zabbix běží. Zabbix umí pracovat distribuovaně, to znamená, že data ve vzdálených lokalitách jsou v proxy režimu a poté se přenáší do hlavního uzlu. Výstupem Zabbixu je webové rozhraní, kde se provádí správa a vyhodnocování.

Výhody systému:

- rychlost
- otevřenost (licence GPL)
- přehledné prostředí
- intuitivní ovládání
- živý a rychlý vývoj
- mnohostrannost (možné sledovat prakticky cokoliv)
- podpora (fórum, dokumentace, ...)

Server

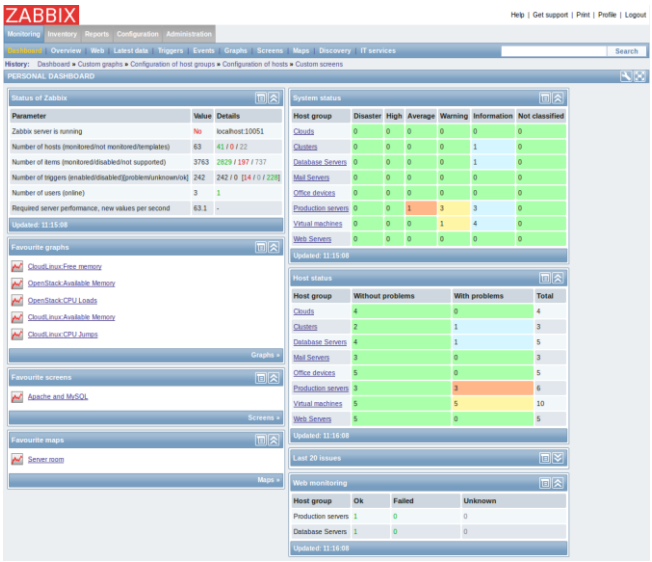
Řídí hlavní proces, který zajišťuje logiku celého systému a veškeré činnosti. Je psaný v jazyku C, pro svůj běh potřebuje malé množství systémových prostředků. Slouží k provádění akcí / pokynů od agenta.

Agent

Jde o klientský proces, který sbírá data a předává je serveru. Je psaný v jazyku C.

Proxy

Zabbix v režimu proxy, plní funkci serveru. Chvíli sbírá data, jako server a poté je přeposílá na server. Nastavuje se v případech, kdy máme hodně lokalit a chceme odlehčit serveru. V tomto případě můžeme mluvit o distribuovaném monitoringu.[26]



Obr. 19 Ukázka uživatelského rozhraní systému Zabbix [26]

6.2 Nagios

Patří do skupin open source dohledových systémů pro automatizované sledování stavu počítačových sítí a poskytovaných služeb. Základní jádro Nagiosu je distribuováno zdarma. Nagios (Nagios Enterprises) je jedním z nejpopulárnějších dohledových systémů na světě a využívá ho tisíce uživatelů. Dříve byl zvaný pod pojmem NetSaint. Vznikl v roce 1999 a v současnosti je využíván mnoha společnostmi. Nagios je linuxový standard, který se opět používá pro hromadné sledování sítových prvků. Lze hromadně monitorovat servery, switche, tiskárny, atd.

téměř vše na co existuje řada pluginů, které lze doinstalovat. Nagios je primárně určen pro Linux, nebo jiný unixový systém. Pokud doinstalujeme program plugin Agenta, tak můžeme sledovat i nesíťové zařízení. Díky desítkám pluginů, které se mohou doinstalovat, můžeme sledovat téměř všechno. Z důvodu složitosti implementací pluginů, může být prvotní nasazení zdlouhavé a obtížné.

Existuje také placená verze Nagiosu XI, kterou také má podporu ze strany vývojářů. Výhodou placené verze je ucelený balík s implementovanými funkcemi. Zároveň je zachovaná i rozšiřitelnost síťového i aplikačního řešení. Se zakoupenou verzí může společnost využívat podporu vývojářů při řešení problémů. Pro firemní řešení je toto řešení nejrozumnější.[28]

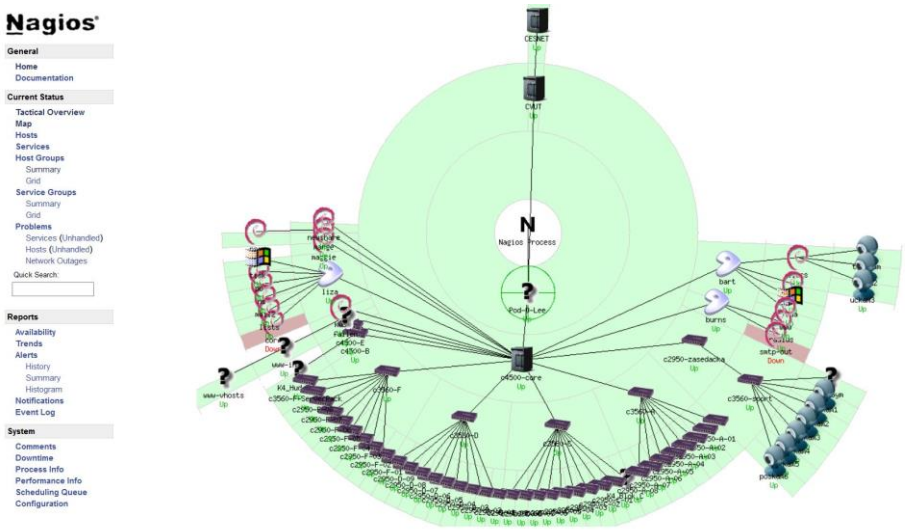
Základní terminologie Nagiosu:

Host - zařízení počítačové sítě, které budeme monitorovat (počítač, server, router, switch..).

Host group – je to skupina hostů (počítače, servery, routery, switche..). Zařízení musejí být steného typu.

Service – služba poskytnutá hostu v režimu monitorování.

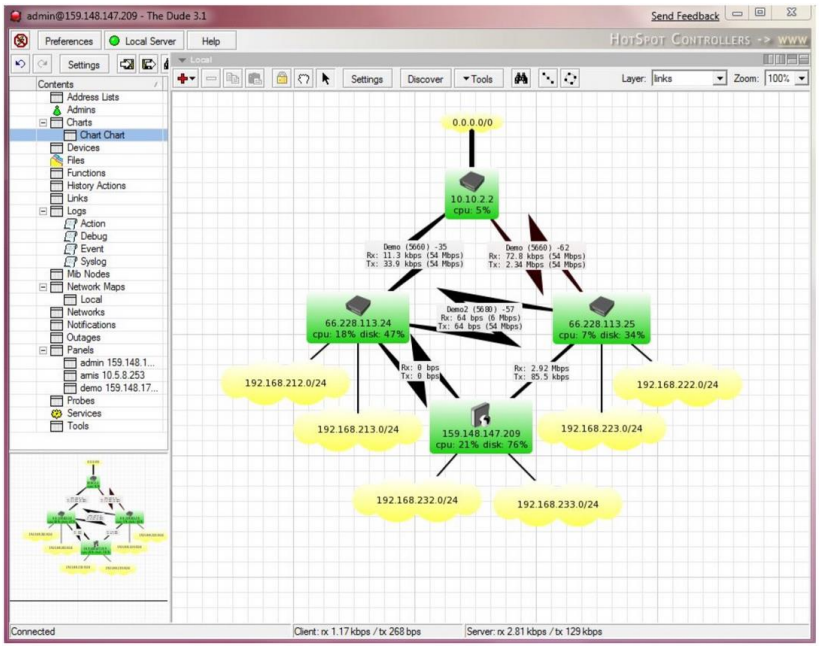
Service group – služba poskytovaná skupině hostů.



Obr. 20 Ukázka pracovního prostředí v systému Nagios

6.3 Dude

Je dohledový, monitorovací systém pro Mikrotiky. Dude je zástupce open source (volně šiřitelných) produktů. Tento systém vznikl jako vedlejší produkt ve společnosti Mikrotik při vývoji routerů. Dohledový systém Dude je určen pro platformu Windows, Linux či MacOS. Systém pracuje na modulární architektuře. Konfigurace probíhá výhradně přes grafické prostředí. Umožňuje režim automatického vyhledání prvků, vykreslení a zobrazení zařízení do síťové mapy. Systém Dude je hlavně určen pro dohled zařízení Mikrotik, ale zvládá i všechny prvky ostatních výrobců, je-li k nim k dispozici odpovídající MIB. Z hlediska topologie je systém Dude orientován, jak na centralizovaný, tak i na federativní způsob dohledu. Dovoluje hromadné upgradování systému RouterOs. Mezi podporované protokoly patří ICMP, TCP, UDP, DNS a SNMP. Systém Dude je šířen zcela zdarma. [9]

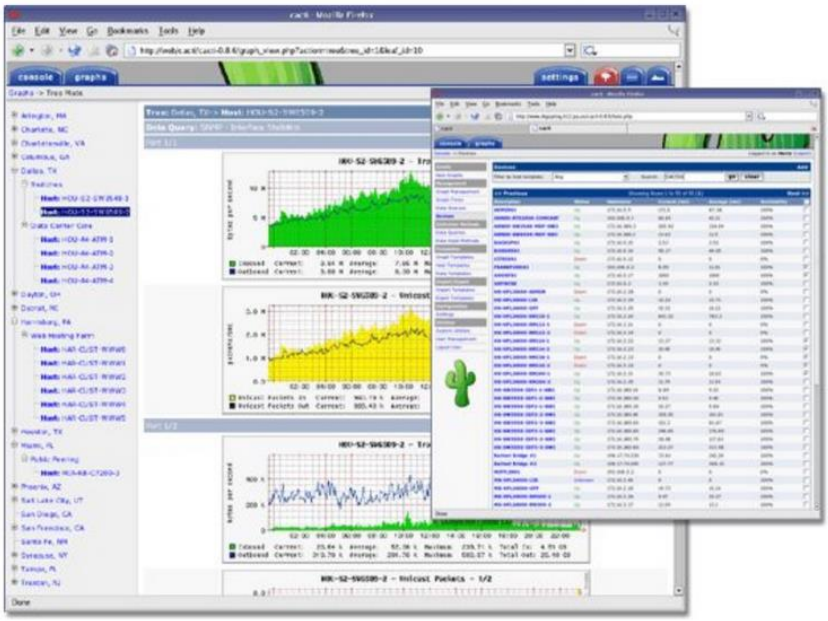


Obr. 21 Ukázka uživatelského rozhraní Dude[9]

6.4 Cacti

Dohledový systém Cacti je open source postavený na nástroji RRDTool. Tento nástroj byl vyvinut jako intuitivní, jednoduchý, dobře škálovatelný, speciálně pro sběr a vizualizaci dat. Jeho využití pro dohledový systém je optimální. Systém Cacti se snaží pracovat primárně jako systém pro sběr a následnou analýzu dat. Umožňuje také sledování v reálném čase, které má zde sekundární účel. Upozornění na

výpadky je uživateli prezentováno v řádech několika minut. U menších počítačových sítí, kde hodně záleží na dostupnosti, může velká odezva způsobovat. Nasazení dohledového systému pro menší datové sítě není vhodné. Systém Cacti je komunitně vyvíjený systém a to přináší i jistá úskalí v podobě zdlouhavého nastavení pro konkrétní sledovaná zařízení, kdy je občas nutné složitě dohledávat a nastavovat parametry pro úlohy, zajišťující sběr dat. Z hlediska topologie je systém orientován na centralizovaný způsob dohledu. Pro sledování aplikačních serverů se využívá lokálních agentů, kteří slouží ke generování SNMP odpovědí na přijímané dotazy. Mezi podporované protokoly patří ICMP, TCP, UDP a SNMP. [9]



Obr. 22 Ukázka uživatelského rozhraní Cacti [9]

6.5 Systém Center Operations Manager SCOM

SCOM 2012 je z rodiny produktů společnosti Microsoft v oblasti správy, řízení a proaktivního monitoringu IT infrastruktury. Tento monitorovací systém se souhrnně nazývá System Center2012. Historicky byl SCOM zaměřen výhradně na velká enterprise prostředí. Současným rozvojem sítí s příchodem verze 2012 dává smysl ho implementovat i v menších prostředích. Systém je navržen jako škálovatelný, ale na začátku je třeba navrhnout design SCCM (Systém Center Configuration Manager) pro danou síť. Systém se prodává jako komerční placené řešení a platí se počet připojených serverů i dalších balíčků. Dohledový systém pracuje se standartními protokoly jako je SMTP, TCP, ICMP,SNMP, Telnet, SSH, POP, IMAP a mnoho

dalších. Samozřejmostí je možnost upozornění emailem, SMS a komunikačním klientem.

Využívá se zde hierarchický model několika typů „site“, kde může být jeden či více serverů. Na každý „site server“ se instalují určité systémové role, pokud máme menší síť je zde pouze jedna „site“ a obsahuje více serverů.

Typy „site“:

- **Central administration site** - volitelná „site“ (velké instalace), může spravovat více primary site (nepracuje přímo s klienty)
- **Primary site** - povinná „site“, která spravuje klienty
- **Secondary site** - volitelná „site“, pro správu klientů na pobočkách nebo vzdálených lokalitách, kde je omezená konektivita

Systémové požadavky SCOM 2012:

- 1024 MB volného místa na disku
- Operační systém Windows Server 2008, Windows Server 2008 R2, Windows Server 2008 R2 SP1, Windows Server 2008 Service Pack 2, Windows Server 2012, Windows Server 2012 R2
- 64 bitový procesor
- Pro server musí být povolena vzdálená správa
- .NET Framework 4

Z hlediska infrastruktury je možné použít jak centralizované, tak necentralizované řešení dohledového systému. Dohledový systém je poměrně dost náročný na HW a placené licence na všech stanicích. [9][30]

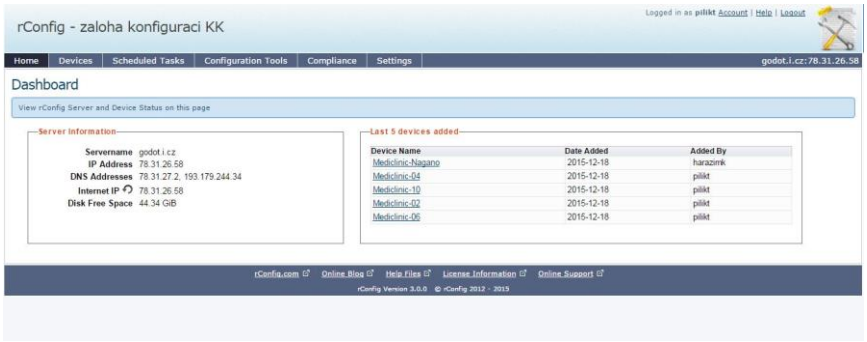
6.6 rConfig

rConfig je zdarma open source dohledový systéme pro zálohu konfigurace zařízení a nástroj pro síťové inženýry. Je to unikátní dohledový systém, kde si můžete sami nakonfigurovat příkazy, které chcete spustit na definovaných zařízeních. Tyto zařízení mohou mít různé výrobce.

Systém nainstalujeme na síťový server s Linuxem CentOS, na kterém doinstalujeme programy (PHP, MySQL a Apache) a nakonfigurujte dohledový systém. Na rConfig můžeme konfigurovat seznamem příkazů, které chcete aplikovat na určitou kategorii zařízení. Další zařízení můžete přidávat do této kategorie, nebo vytvořit jinou kategorii. Můžeme vytvářet a plánovat úlohy. Systém rConfig se podle zadaných konfiguračních příkazů postará o zbytek.

Nyní je ke stažení už třetí verze rConfigu, která má nyní „Configuration Management Compliance“ nástroje, které umožnují sledovat zařízení, zálohovat konfigurace zařízení a v případě odpojení zařízení nahlásí výpadek administrátorovi.

rConfig je zcela open source a je zde možnost přizpůsobit a přidat funkce, jak je libo. rConfig je napsán v rodném PHP. Pro práci v rConfigu nebudete muset instalovat žádné další moduly PHP. Z hlediska topologie je systém orientován na centralizovaný způsob dohledu. [27]



Obr. 23 Systém rConfig úvodní obrazovka

Funkce:

- Zdarma a open source
- Konfigurace pomocí příkazu na zařízeních
- Configuration management Compliance
- Hromadné nasazení konfigurace
- One-Click stahování konfigurací
- Nativní implementace PHP
- Extrémně rychlé nastavení výstupu + porovnávací a vyhledávací funkce
- Vestavěný plánovač založený na CRON
- Vestavěné hlášení
- Funkce zálohování systému
- Telnet a podpora SSHv2

Požadavky:

- CentOS 6.3 nebo vyšší (RedHat verze pracovní i)
- PHP 5.3.3 nebo vyšší
- MySQL 1.5.61 nebo vyšší
- Apache 2.2.15 nebo vyšší
- Browser IE7 +, Firefox3.5 +, Chrome11 +, Safari3 +, Opera 9.4+

7 Výběr dohledového systému

Při své praxi ve společnosti ICZ a.s. jsem dostal zajímavou možnost na rozvoji nástroje rConfig. Dostal jsem za úkol od svého vedoucího rozvinout stávající dohledový systém Nagios, novým systémem pro projekty řízených služeb na trhu (týká se projektu A a projektu B). Můj úkol byl vybrat vhodný dohledový systém pro dané projekty, zprovoznit tyto služby na serveru a implementovat nové funkce. Stávající databáze má okolo 700 záznamů, to odpovídá 700 zákazníkům této služby. Původní databázi bylo potřeba zálohovat a aktualizovat.

7.1 Porovnání a výběr

Požadavky byly následující. Jednoduchý systém umožňující přístup, kontrolu zařízení zákazníků a zálohu zařízení. Pokud možno multiplatformní přístup nejlépe přes webové rozhraní, jak pro OS Linux, tak pro OS Windows. V systému musí být databáze všech zákazníků určených projektů s aktuální kontrolou dostupnosti zařízení. Jednoduše řečeno, po přihlášení zde uživatel lehce nalezne zákazníka, lokalitu, jestli je zařízení „online“ a starší konfigurace daného zařízení, která zde bude uložena v případě, že zařízení odejde a je potřeba okamžitě vyměnit. Další požadavek je hlášení stavu zařízení. Pravidelně by měl chodit report hlášení o dostupných a hlavně nedostupných zařízeních, ke kterým se vztahuje SLA dané služby.

V základu tato služba poskytuje router firmy Cisco řady 8XX nebo router Mikrotik RB 7XX,9XX. Služby mohou být dále doplněny dalšími prvky síťové infrastruktury, jako jsou IP telefony, switche, wifi AP, kamerové systémy a další zařízení.

Nejčastěji používané routery mají konektivitu ADSL/VDSL a výjimečně i jinou konektivitu (např. optickou, ethernetovou, nebo bezdrátovou technologii). Na tyto zařízení se vztahuje SLA dané služby a proto je potřeba tyto zařízení monitorovat, zálohovat a v případě nutnosti vyměnit.

Po důkladném zvážení potřeb obchodních zástupců a techniků z řad naší firmy, kteří potřebují do toho to systému přistupovat, bylo rozhodnuto pro instalaci dohledového systému rConfig. Tento systém splňuje požadavky open source systému, kde mohu časem dodělávat různé potřebné skripty. Systém je poměrně jednoduchý pro přístup uživatelů a administraci celého systému.

7.2 Stručný popis rConfigu

rConfig je open source dohledový systém, nástroj pro správu konfigurací, který používáme jako databázi zákazníků na vybraných projektech pro stahování konfigurací z hraničních zařízení (Cisco routery, ASA prvky). rConfig se na každý prvek připojuje přes SSH v definovaných intervalech.

Dohledový systém rConfig je zcela open source, což znamená, že si ho můžete přizpůsobit podle vlastních potřeb a přidávat funkce, jak je libo. Celý program rConfig je napsaný v jazyce PHP. Není třeba instalovat další moduly PHP pro práci v rConfigu.

rConfig běží na doméně godot.i.cz (IP adresa 78.31.x.x). Přístup na server je možný pouze z vnitřní sítě, vzdálený přistup je možný pouze přes VPN klienta.

Oficiální web pro rConfig je na http://www.rconfig.com , kde je potřeba projít registrací a počkat na schválení účtu administrátorem. Potom je dostupná sekce stahování a podpory (guides, knowledge base a forum).

7.3 Záloha původní databáze v systému Nagios

Můj firemní počítač, s kterým pracuji, má operační systém Windows 7, proto vše co jsem dělal, popisuji přes OS Windows. Nejdříve jsem začal s zálohováním původní databáze.

Databáze MySQL umí exportovat do MS Office Excelu svůj obsah. Umí vytvořit XML Spreadsheetu (*.xml) s kterým dobře pracuje MS Office Excel. Excel umí tabulku uložit v různých formátech jako je XLSX a XLS, data jsou buď zašifrovaná, nebo zkomprimovaná. Excel také umí data exportovat v čitelné podobě

ve formátu XML Spreadsheet (XMLSS). XML Spreadsheet je podporován od verze Microsoft® Excel 2002.

Zálohu dokumentu z MySQL databáze jsem provedl jednoduše. Se získanými údaji pro přihlášení na virtuální server jsem se přihlásil na server. K přihlášení jsem použil program „WinSCP“, který umožňuje použít SFTP a FTP klienta. Nyní jsem se proklikal do umístění databáze MySQL na serveru a odtud jsem stáhnul databázi ve formátu XML, s kterou budu dále pracovat.

Dále byl do nového serveru Godot nainstalován software Apache server a program phpMyAdmin. Nyní se mohu připojit k serveru Godot, kde budu moci zkopírovat upravená data databáze.

7.4 Instalace systému rConfig

K dispozici mám virtuální server jménem Godot, který má nainstalovaný CentOS verze Red Hat 4.4.7-11 a disponuje těmito hardwarovými komponenty: procesorem Intel® Xeon ® CPU E5-2640 @ 2,5 GHz (cache 16MB), 2GB RAM a pevný disk 50 GB.

V tomto serveru disponuji nainstalovanou verzí PHP verze 5.5.3, MySQL verze 5.4.6 a Apache HTTP Serverem verze 2.4.5 . Virtuální server je nově nainstalovaný, prázdný a určený pouze pro funkci rConfigu.

Dále mám k dispozici exportovanou zálohu databáze zákazníků ze starého dohledového systému Nagios. Na serveru Godot je nyní puštěn program Apache HTTP server, který je nastaven, aby nám vytvořil webové rozhraní pod veřejnou IP adresou našeho serveru Godot (IP: 78.31.x.x).

Minimální hardwarové požadavky rConfigu:

- Samostatný nebo specifikovaný server (fyzický nebo virtuální)
- 100 GB volného místa na pevném disku
- 1 GB RAM
- Instrukční sada x86_64 CPU

Minimální softwarové požadavky rConfigu:

- CentOS minimálně verze 6.3 a vyšší (nebo verzi RedHat)
- PHP verze 5.3.3

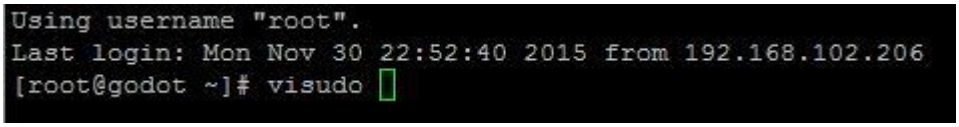
- Apache 2.2.15
- MySQL 5.1.61
- CRON

Virtuální server splňuje požadavky open source dohledového systému rConfig. Proto mohu začít s instalací. K instalaci jsem použil SSH klienta „mRemote“ a připojil se s ním na virtuální server Godot, kde jsem začal instalaci podle následujících bodů.

7.4.1 Aktualizace konfiguračních souborů CentOs

Nejdříve jsem musel aktualizovat některé konfigurační soubory Linuxu CentOs, aby instalace rConfigu neselhala nebo by se mohlo stát, že by některé funkce rConfigu nemusely fungovat.

1. Aktualizujte soubor sudoers, pomocí visudo.



Obr. 24 Aktualizace systému Cent OS

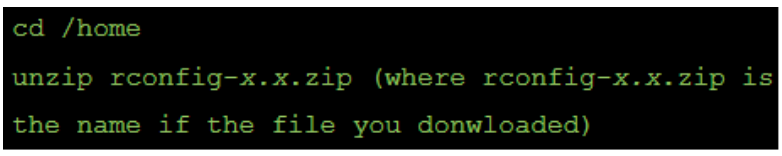
2. Přidávám následující řádky do souboru sudoers.

Obr. 25 Přidání souborů podle bodu 2.

7.4.2 Instalace rConfig do CentOs

Již máme aktualizované konfigurační soubory a jsme připraveni na server nahrát instalace dohledového systému rConfig.

1. Nyní jsem opět přes program WinSCP překopíroval do složky home zabalený program rConfig verze 2. Program rConfig jsem stáhnul z adresy www.rconfig.com.
2. Dále jsem provedl rozzipování balíčku rconfig-x-x.zip.



Obr. 26 Rozbalení balíčku

3. Poté je potřeba přiřadit oprávnění rConfigu.

```
chown -R apache /home/rconfig
```

Obr. 27 Přidělení oprávnění

4. Teď už jen zbývá aktualizovat soubor httpd.conf v základním adresáři z přednastaveného souboru httpd.conf, který je zabalený v balíčku rConfigu.

```
mv /etc/httpd/conf/httpd.conf /etc/httpd
/conf/httpd.conf.original
cp /home/rconfig/www/install/httpd.conf.new
/etc/httpd/conf/httpd.conf
service httpd restart
```

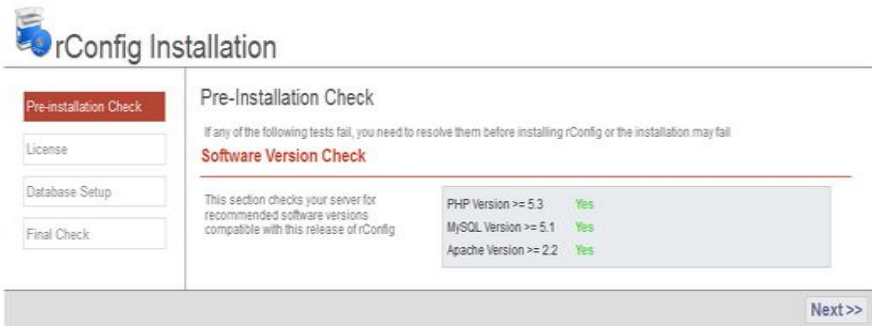
Obr. 28 Aktualizace souboru httpd.conf

5. Jsme připraveni a můžeme zahájit instalaci ve spuštěném webovém prohlížeči přes stránku „http://rconfig.hostname/install/“ pro dokončení instalace. V našem případě jde o příkaz s konkrétní IP adresou serveru Godot „78.31.x.x/install/“.

7.4.3 Webové dokončení instalace rConfig

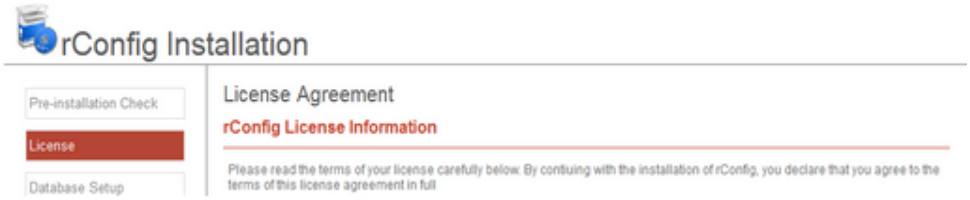
Po spuštění „78.31.x.x/install/“ v okně prohlížeče počítače uvidíme dialogové okno instalace, kde vyplníme všechna pole.

1. Nejdříve se objeví položka „Pre-installation Check“, kde je kontrolní seznam všech bodů, které musíme splnit, než budeme pokračovat s instalací. Program rConfig zde kontroluje aktualizaci verzí PHP, MySQL a Apache http serveru.



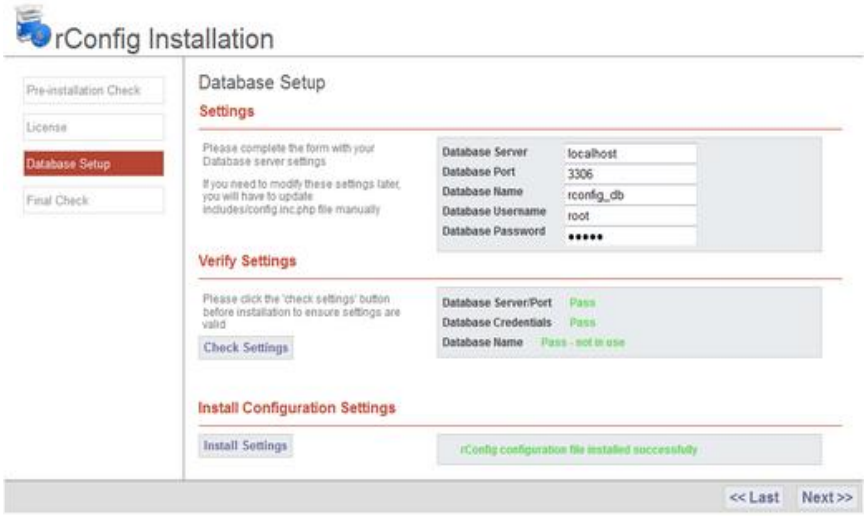
Obr. 29 Postup instalace část 1.

2. Další položka je Licence, kde si musíme přečíst licenci rConfigu, která je open source a pokračujeme s instalací.



Obr. 30 Postup instalace část 2.

3. Na této obrazovce zadáme nastavení databáze pro rConfig. Je doporučeno, aby se vytvořil databázový účet root pro databázi rConfigu. Pokud je databáze nainstalovaná na stejném serveru jako je rConfig, pak můžeme zadat „loocalhost“ jako „databázový server“. Využil jsem lokální databázový server MySQL a PHP, kde byla vložena záloha. Zadal jsem přihlašovací údaje a zkontroloval propojení. Propojení se zdařilo úspěšně.



Obr. 31 Postup instalace část 3.

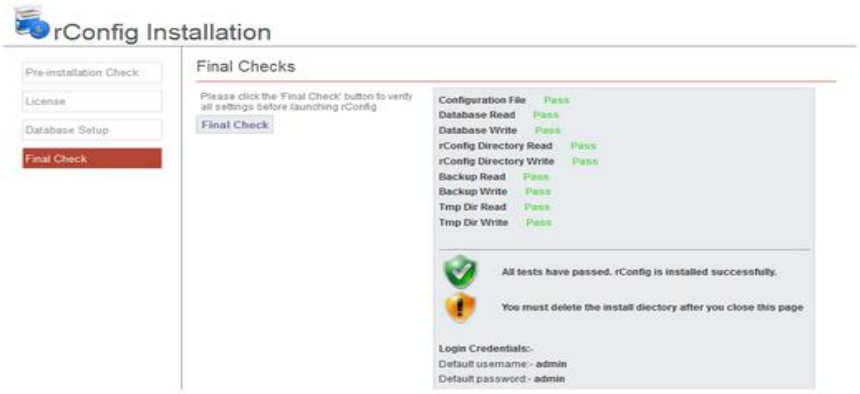
4. Poslední bod webové instalace mi potvrdil, že se celková instalace rConfigu podařila. Stačí kliknout na tlačítko „Final Check“, kde vidíme kontrolu veškerých parametrů a zda se jejich instalace vydařila. Při prvním pokusu instalace se mi vyskytla chyba v načtení databázového serveru a musel jsem celý proces instalace opakovat. Po důkladném opakování procesu instalace (a nastavení) se vše povedlo bez chyby a instalace proběhla v pořádku.

   Dále je potřeba smazat instalační adresář „install“ ve složce „home / rconfig / www /“ z bezpečnostních důvodů (kvůli defaultnímu heslu admin/admin).

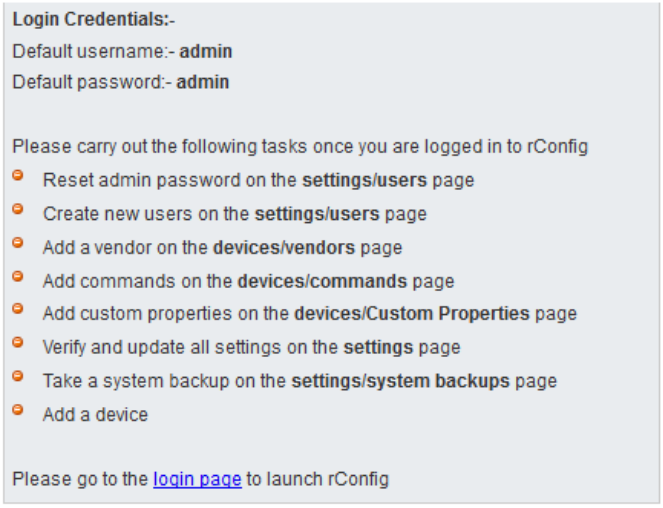
```
rm -fr /home/rconfig/www/install/
```

Obr. 32 Smazání souboru

5. V dolní části obrazovky se mi zobrazil seznam položek doporučení, které je potřeba udělat při prvním spuštění rConfigu. Doporučuje se provést tento seznam, aby se předešlo chybám v systému.

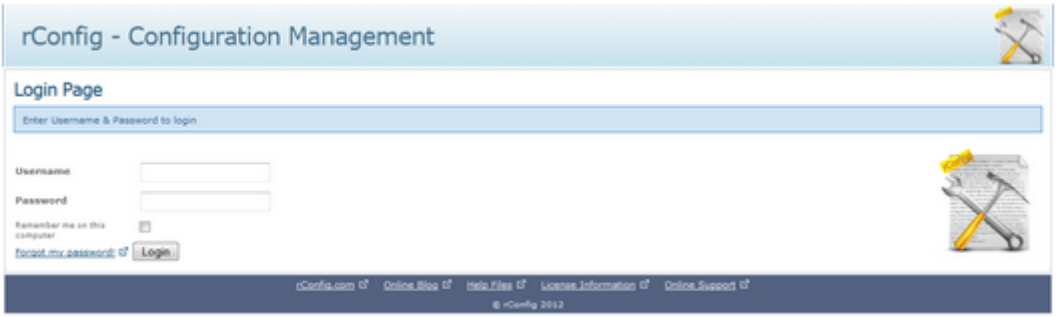


Obr. 33 Dokončení instalace



Obr. 34 Doporučené pokyny po instalaci

6. Nyní když zadáme do prohlížeče IP adresu našeho serveru Godot IP 78.31.x.x, nebo adresu www.godot.i.cz, tak se mi objeví úvodní obrazovka, kde jsem se mohl přihlásit defaultním uživatelským jménem a heslem admin/admin.



Obr. 35 První přihlášení do systému rConfig

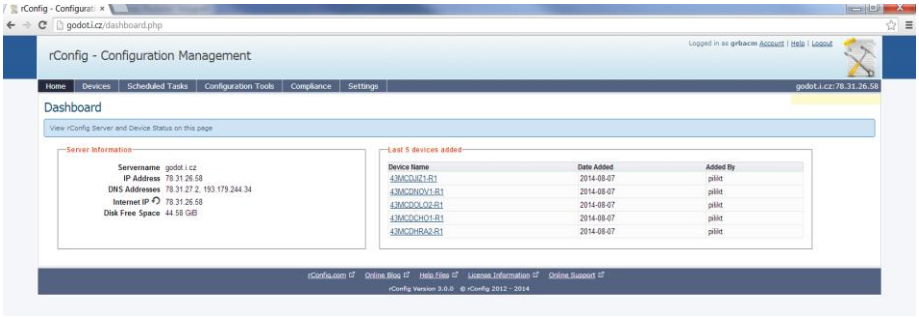
7. Po nainstalování byl systém prázdný a bylo potřeba vše nastavit, nakonfigurovat a přizpůsobit uživatelům a technikům.

7.5 Úvodní stránka (Home)

Záložka home je úvodní obrazovka ihned po přihlášení. Na úvodní desce (Dashboard) vidíme dvě informativní okna.

V prvním okně vlevo vidíme základní informace o serveru, kolonka „Server information“. Jako první položka v této kategorii je „Servername“ které je godot.i.cz , dále je IP adresa daného zařízení a pod tím veřejná IP adresa DNS serverů, kde probíhá překlad IP adres tzv. DNS. Pod údajem DNS serverů je veřejná IP adresy, která je stejná 78.31.x.x. Jako poslední údaj v první tabulce o serveru je volné místo na disku.

Na druhé straně vpravo obrazovky vidím okno, kde jsou zaznamenány poslední změny zařízení. Vidíme zde např., na jakém zařízení došlo k určité úpravě informací a jaký uživatel tuto změnu provedl, samozřejmě s časovým údajem konkrétního dne.



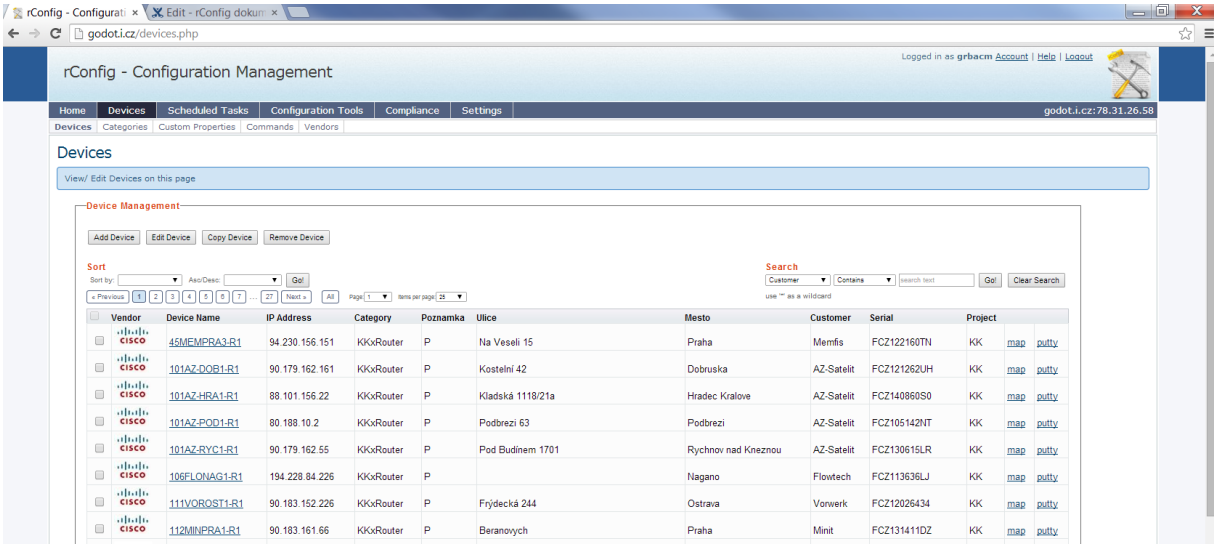
Obr. 36 Webové rozhraní rConfigu

7.6 Zařízení (Devices)

Zde je možné přidávat, editovat, kopírovat a mazat zařízení. Dále je zde zobrazena databáze zařízení. Je možné hledat zařízení na základě jména zákazníka (Customer), jména zařízení (Device Name), kategorie (Category), IP adresy (IP Address) a sériového čísla zařízení (Serial Number).

Pro zobrazení aktuální konfigurace zařízení stačí kliknout na název zařízení, zobrazí se stránka devicemgmt.php, na které jsou v levé části zobrazeny detailní informace o zařízení. Důležitá je informace o dostupnosti - Status Available/Unavailable. V pravé části obrazovky jsou uloženy konfigurace zařízení, kde po rozkliknutí hierarchického menu rok / měsíc / den je umístěna konfigurace, která má v názvu vykonaný konfigurační příkaz, čas a kdy byla operace provedena (např. shrun-2300.txt tedy ve 23:00). Po kliknutí na odkaz se otevře nové okno s konfigurací nebo je možné konfiguraci stáhnout v souboru txt.

Přidání nového zařízení se snadno provede tlačítkem „Add Device“ nebo okopírováním stávajícího tlačítkem Copy Device. Zobrazí se nabídka s několika údaji, které je potřeba doplnit nebo upravit.



Obr. 37 Položka Devices s dostupnými údaji

7.6.1 Detaily zařízení (Device Details):

První skupinou jsou detaily zařízení, kde definujeme základní popis zařízení.

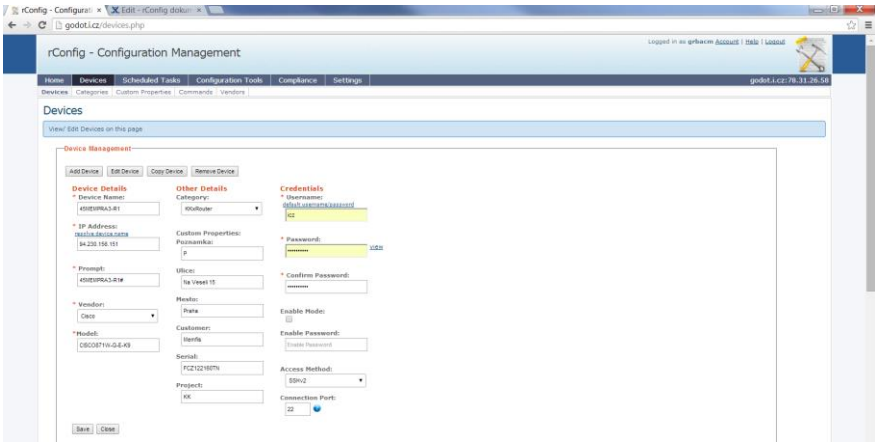
- Device Name – hostname prvku (např. 45MEMPRA3-R1)
- IP address – IP adresa zákazníka/zařízení
- Prompt – řetězec znaků, které skript očekává, než vloží příkaz (např.: 45MEMPRA3-R1#)
- Vendor – lze definovat v Devices/Vendors i s obrázky
- Model – používáme Product ID prvku (např. CISCO871W-G-E-K9)

Další skupinou jsou Other Details (pro snadné rozšíření umožňuje rConfig vytvořit vlastní pole do databáze (Devices/Custom Properties):

- Category – definované v Devices/Categories
- Custom Properties – definované pole uživatelem
- Poznámka – většinou uvádíme P – pobočka nebo C – centrála
- Ulice – např. Na Veseli 15
- Město – např. Praha
- Customer – název zákazníka, např. Memfis
- Serial – Serial Number prvku (např. FCZ122160TN)
- Project – XX pro řízené služby

Poslední skupinou jsou Credentials:

- Username – pro vložení defaultních přihlašovacích údajů (definovaných v Settings) stačí kliknout na default username/password a údaje se vloží
- Password
- Confirm Password
- Enable Mode – zaškrtnout pokud je potřeba zadat enable heslo
- Enable Password
- Access Method – telnet nebo sshv2
- Connection Port – většinou port 22 pro ssh



Obr. 38 Vytvoření nového zařízení s dostupnými údaji v položce „Devices“

Každé zařízení je umístěné do jedné kategorie (Devices/Categories). Pro každou kategorii jsou definované konfigurační příkazy (Devices/Commands), na základě kterých se potom spustí časově plánovaná úloha (Scheduled Task).

7.6.2 Zařízení/kategorie (Devices/Categories)

Zde se definují kategorie zařízení. Jako název kategorie používáme vždy zkratku pro projekt oddělenou písmenem x a název kategorie, např. KKxRouter, KKxAAA atd. Je to z toho důvodu, protože rConfig neumožňuje zadat názvy s mezerou nebo s pomlčkou.

Zákazníky o větším počtu poboček jsem umístil do samostatné kategorie, např. AAA a BBB.

Kategorie	**Popis**
KKxRouter	KK obecný routery
KKxFW	KK Cisco ASA
KKxAAA	KK AAA routery (Cisco)
KKxBBB	KK BBB (Cisco)
KKxSSS	KK SSS routery (Mikrotik)
KKxNoBackup	KK nedostupne routery
CIPxRouter	Cisco routery
CIPxFW	Cisco ASA

Tabulka 3 Zobrazení výpisu kategorie

7.6.3 Zařízení/vlastnosti zákazníka (Devices/Custom Properties)

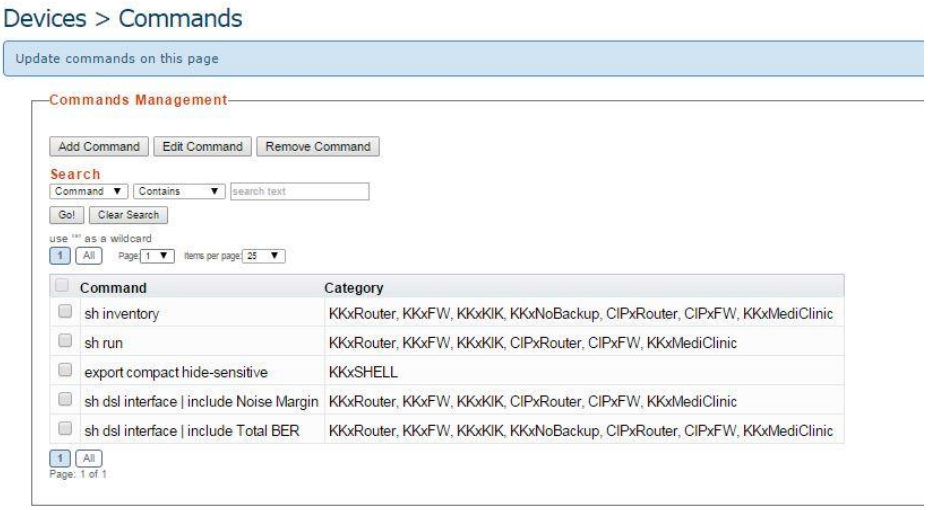
Tato položka je propojená s databází MySQL. Můžeme tedy v určité míře upravovat databázi z rConfigu. Je zde možné přidat pole definované uživatelem pro přidání dalších informací jednotlivých zařízení do databáze.

Custom Properties	**Popis**
Poznámka	P - pobočka, C - centrála
Ulice	Ulice a č. p.
Město	Město
Customer	Název zákazníka
Serial	SN (Serial Number) zařízení
Project	KK - Kompletní kancelář a CIP - Cipísek

Tabulka 4 Sloupce definované v databázi MySQL

7.6.4 Zařízení/příkazy (Devices/Commands)

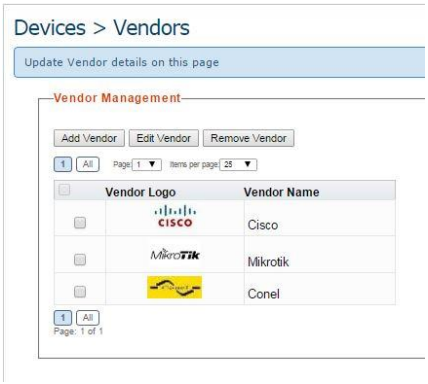
V této kategorii jsou definované konfigurační příkazy. Příkazy můžeme spustit pro každou kategorii, na které se vykonají v určitém čas. Příklad použitých příkazů vidíme na *Obr. 39*.



Obr. 39 Výpis příkazů konfigurace pro každou kategorii

7.6.5 Zařízení/výrobce (Devices/Vendors)

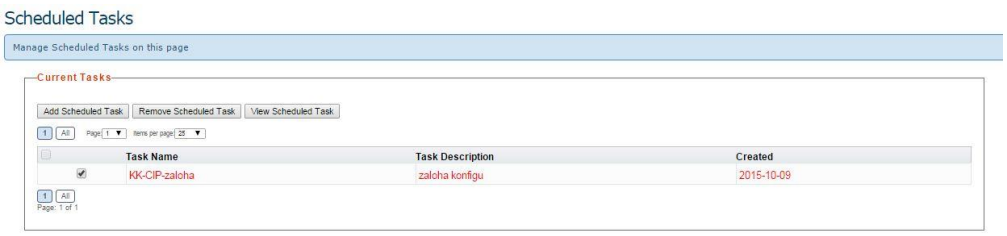
Pro definování výrobců zařízení v systému volím položku Vendor. Zde si můžu vytvořit výrobce a přidat konkrétně jméno a logo. Nejčastěji se dostávám do kontaktu se zařízením od firmy Cisco, Mikrotik nebo Conel. Zatím jsem definoval pouze tyto.



Obr. 40 Použití výrobci v systém rConfig

7.7 Naplánované úlohy/Scheduled Tasks

V kategorii „Naplánování úloh“ jsem definoval opakované spouštění úloh pro všechny zařízení. Stahování konfigurace a ostatních příkazů se provádí každý den okolo půlnoci. V tuto dobu je nejméně vytížená síť. Samozřejmě můžu úlohy pustit manuálně na konkrétním zařízení pro ověření funkčnosti. Každá kategorie může být nastavena na určitý čas.



Obr. 41 Definované úlohy pro zálohu konfigurací.

Pro zálohu konfigurací je třeba definovat následujíc položky:

- Select a Task Type: Download Configurations – výběr operace.
- Task Name – název úlohy.
- Task Description – popis úlohy (např. záloha konfigurací hraničních zařízení).
- Email report – po skončení plánované úlohy zašle email.
- Email Errors Only - po skončení plánované úlohy zašle email.
- Select Devices – možnost vybrat určitá zařízení, pro která poběží plánovaná úloha.
- Select Categories – možnost vybrat celou kategorii, pro kterou poběží plánovaná úloha.
- Enter Schedule Details – interval, se kterým se plánovaná úloha bude opakovaně spouštět (např. Once a day).

Pro posílání emailů je nutné mít nastavený SMTP server a příchozí emailové adresy v kategorii Settings následovně:

Local SMTP Server	vidle.i.cz
From Address	RCONFIG@i.cz
Email Recipients	xxx@i.cz; xxx@i.cz

Tabulka 5 Nastavení SMTP serveru

7.8 Konfigurační nástroje

Pro řešení problémů s připojením na zařízení jsou zde k dispozici logy. Dále je zde možné porovnávat konfigurace a full textově prohledávat zálohované konfigurace. Tato položka má zajímavou funkci „Compare“, která umí porovnat konfigurace a vypíše, jestli došlo k změně konfigurace na zařízení. Pokud došlo ke změně konfigurace, tak vypíše na jakém řádku. Dále je zde vyhledávání stringu v konfiguraci. Pak zde můžeme najít hlášení a logy. V současné době, kdy je v databázi okolo 700 zařízení a zálohují se každý den, trvá poměrně dlouho, než se načtou všechny záznamy. Tuto sekci konfigurační nástrojů bych rád do budoucna urychlil, aby se s ní dalo lépe pracovat. Nyní se položka „Configuration Tools“ moc nepoužívá.

7.9 Nastavení

V nastavení je k dispozici přehled využití paměti a zaplnění disku, nastavení časové zóny, výchozí přihlašovací údaje a nastavení SMTP serveru pro odesílání emailů. Dále je zde možné zapnout/vypnout debugování systému. Proces debugování jsem vypnul, není u této služby zatím potřeba.

V submenu Users je možné přidat či upravit uživatelské účty a nastavit mu práva. Uživatelé mohou dostat práva buď jako User nebo Admin. Zde jsem vytvořil kolegům účty pro přístup.

8 Implementace nových funkcí do rConfigu

Naše centrální databáze všech zákazníků, která se jmenuje „db_rconfig00“ se nachází na databázovém serveru Godot. Jako databázového klienta pro práci s databází jsem použil program myphpadmin, který byl nainstalovaný na serveru. Dále jsem pro úpravu databáze použil program MySQL Workbench, který zobrazuje lepší vizuální návrh databáze a umožňuje lepší orientaci. Databázový klient zjednodušuje práci uživatelů s databází na serveru. Přímou komunikaci se serverem využívají spíše programátoři, kvůli složitosti systému by to bylo pro uživatele příliš abstraktní a nepohodlné. Na serveru Godot se vyskytli problémy s prací v databázi přes program myphpadmin a proto jsem používal i tuto druhou variantu přímé komunikace přes příkazový řádek.

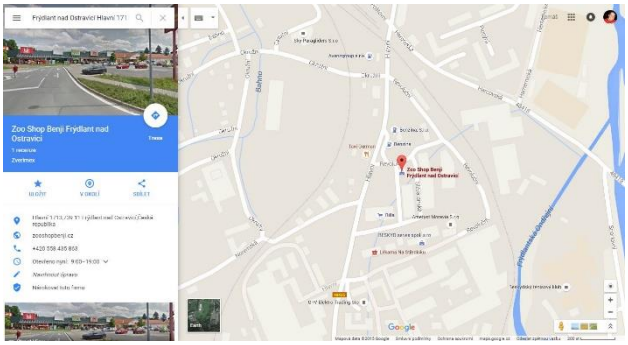
Pokud implementuji do systému rConfig nové funkce, musím počítat s tím, že systém nesmím automaticky aktualizovat, jinak bych mohl přijít o funkčnost všech úprav v systému. Ve webovém okně dohledového systému jsem proto odstranil položku automatické aktualizace v kategorii home/devices, aby ji náhodou nějaký uživatel nespustil chybným kliknutím.

Databáze systému rConfig funguje na server Godot asi jeden rok. Za tuto dobu zjišťujeme a stále zkoušíme zdokonalovat funkce toho to systému. Primárně je systém určen pro řízené službu menších projektů. Současně je připojeno na rConfig okolo stovek zařízení.

8.1 Zobrazení pozice zařízení na mapě

Na serveru Godot, jsem chtěl implementovat funkci do webového rozhraní, která určí polohu zákazníka. Tato funkce má zobrazovat konkrétní adresu zákazníka pro lepší orientaci pomocí odkazu v Google mapách. Vyhledávání mapy bylo zvoleno pomocí asociativní mapy, kde mapuji vyhledávání adresy zákazníka. Adresa zákazníka je uložena přímo v databázi, kam jsem ji při vytvoření zařízení zadával. Po zmáčknutí na pozici mapy konkrétního zákazníka skript sáhne do databáze, kde vytáhne data a zkopíruje je do předem připraveného funkce, která otevře nové okno s polohou zákazníka.

Obr. 42 Zobrazení zákazníka pro přesné určení pozice v mapách



Obr. 43 Zobrazení na mapě po kliknutí na zákazníka

Určení polohy jsem programoval v položce rconfig/www/devices.inc.php kde jsem dodělal odkaz přes Google mapy. Tento skript se přihlásí do databáze, kde vytáhne potřebná data. Jde přesně o položku „custom_Mesto“ a „custom_Ulice“. Po kliknutí na odkaz načte v nově otevřeném okně polohu konkrétního zákazníka.

```
<td >
        <a href="http://maps.google.com/maps?q=
        <?php echo $rows['custom_Mesto']?>
        <?php echo $rows['custom_Ulice']?>"
        title="google map">map
        </td>
<td >
```

Obr. 44 Ukázka skriptu, který vytahuje data z databáze

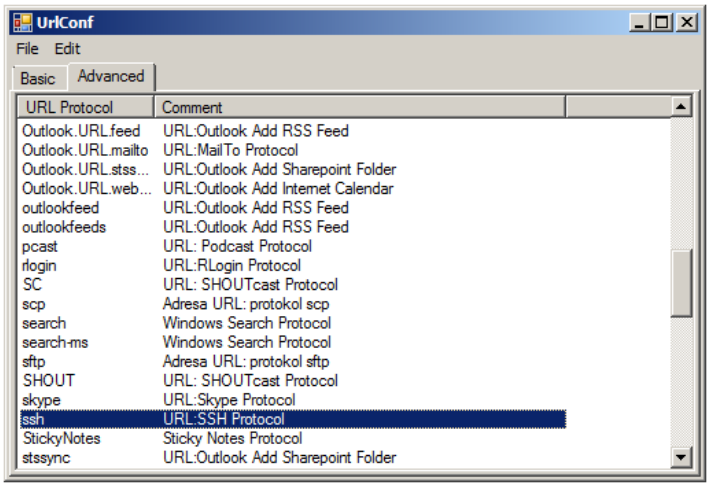
8.2 Multiplatformní připojení terminálu přes Windows a Linux

Pro zobrazení aktuálního zákazníka jsem dodělal funkci, která urychluje připojení k zákazníkovi přes terminál. Vytvořil multiplatformní připojení, jak pro Windows, tak i pro Linux.

8.2.1 Připojení terminálu přes Windows

V operačním systému Windows jsem zvolil pro přístup terminálu program „Putty“. Tento program lze bez problému legálně a zdarma stáhnout na internetu.

Pro použití PuTTY z prohlížeče je potřeba nainstalovat aplikaci UrlConf. Postup instalace je popsán v poznámce v sekci **Devices** ve spodní části stránky.

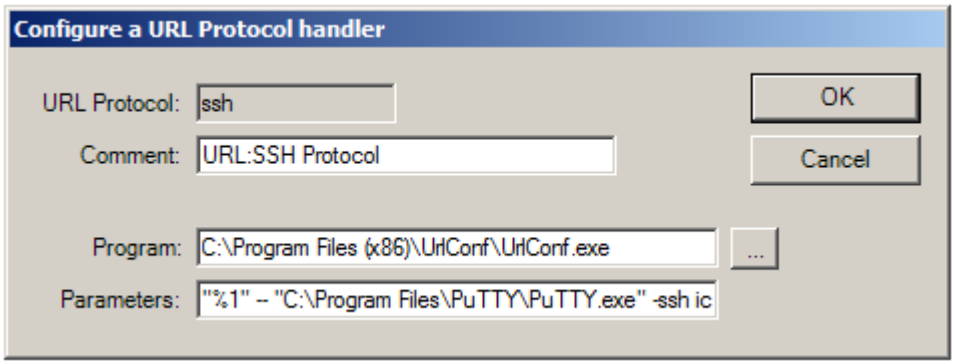


Obr. 45 Nastavení programu UrlConf

Odkaz *putty* lze nastavit pro přímé zalogování do zařízení s defaultními přihlašovacími údaji pro Cisco routery.

Je potřeba spustit aplikaci UrlConf a změnit Parameters:

"%1" -- "C:\Program Files\PuTTY\PuTTY.exe" -ssh icz@$h -pw password



Obr. 46 Přesné nastavení ssh přes program Putty

8.2.2 Připojení terminálu přes Linux

U operačního systému Windows jsem použil terminál přes program „Putty“, ale u operačního systému Linux, žádný takový program být nemusí, protože v Linuxu se dá jednoduše připojit na zařízení přes systémovou konzoli.

Příklad připojení ssh:

```
ssh bfu@ssh.linuxsoft.cz
```

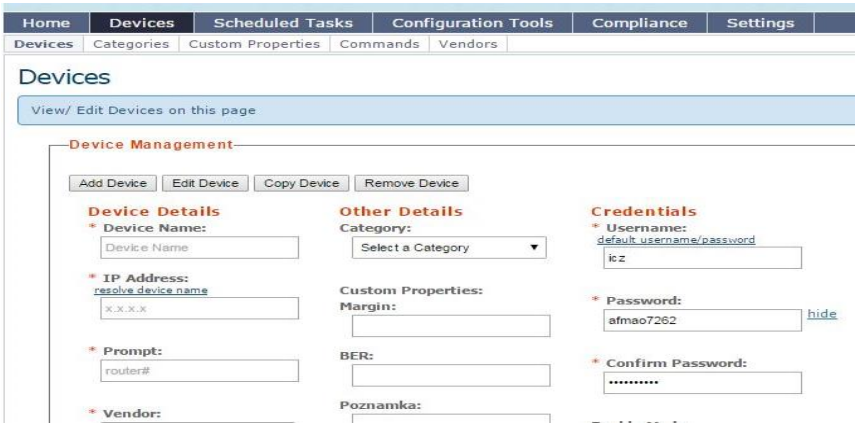
Moje myšlenka připojení přes terminál OS Linux je taková, že vybraný řádek (zákazník) s určitým ID, by po kliknutí na tlačítko SSH měl být překopírován do schránky, paměti počítače na určitou dobu (např. 15s). Ve schránce by se měl automaticky vygenerovat příkaz pro připojení na dané zařízení, uživatelské heslo a jméno.

Obr. 47 Webové rozhraní, kde je označení SSH terminálu

V systému jsem vytvořil tlačítko s odkazem, ale bohužel se mi tuto funkci zatím nepovedlo zprovoznit. Určitě ji budu chtít v budoucnu zprovoznit.

8.3 Zobrazení hesla

Dále jsem přidal funkci, která dokáže zobrazit defaultně zadané heslo pro kontrolu. Pro zobrazování hesla, které je předem definované jsem použil opět funkci JavaScriptu „getElementById()“. Tato metoda umí přistupovat k dokumentu pomocí ID. Po kliknutí na zobrazení změní hodnotu Boolean na logickou 1. zobrazí se heslo (hide). Po změně hodnoty Boolean na logickou 0 se zakryje heslo (view). Funkce je vložena v rconfig/includes/head.inc.php.



Obr. 48 Zobrazení hesla pro kontrolu

8.4 Nastavení zobrazení všech zařízení

Protože zálohovací systém má v sobě více než stovky zařízení a pro přehlednost bylo potřeba, aby se na stránce zobrazily všechny zařízení a né 10 jak to bylo nastaveno defaultně. V kategorii rConfig/classes/paginator.class.php jsem přepsal skript z 10 záznamů, na zobrazení 1000 záznamů. Pokud chceme zobrazit menší množství zařízení, můžeme tuto volbu nahoře v liště změnit např. 25,50,100 nebo všechny záznamy.

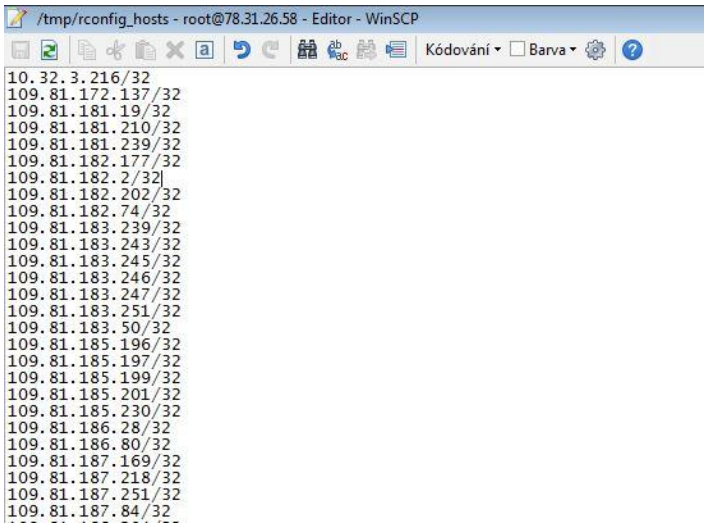
8.5 Program NMAP

Nmap je velmi populární nástroj v oblasti GNU/Linuxu pro skenování sítí. Slouží k otestování firewallu a ověření jaké informace mohou mít případní útočníci o počítačové síti. Nmap umí skenovat porty, také zvládne skenovat sítě (hledat zapnuté počítače), detekovat nejenom běžící služby, ale i verze příslušných démonů a verzi použitého operačního systému.

Detekování cizích sítí bez souhlasu správce sítě či majitele, může být považováno za útok na počítačovou síť.

Instalaci softwaru Nmap jsem provedl pomocí příkazu „`yum install nmap`“. Nmap jsem použil ke skenování pingem všech zařízení (IP adres), které jsem získal z databáze. Programu Nmap jsem nejdříve přidělit práva v systému.

Napsal jsem skript, který se jmenuje „myscan.php“ a má za úkol otestovat celou databázi IP adres. Skript zjistí počet zařízení v online nebo offline stavu.



Obr. 49 Výpis všech IP adres s maskou z databáze

Ve skriptu se pomocí údajů připojím do mysql databáze rConfigu, kde si vyberu potřebná data. Pomocí funkce getHoustlist z databáze načtu jednotlivé záznamy ve formě IP adres s maskou sítě. Tyto záznamy uložím do TMP/rconfig_houst.

getHoustlist

Funkce getHoustlist uloží výsledek testování do pomocného souboru TMP/rconfig_houst. Zde se uloží všechny IP adresy, které chceme otestovat.

scanhosts

Další funkce ve skriptu otestuje všechny IP adresy z souboru rconfig_houst. Test probíhá funkcí pomocí programu Nmap, kde příkazem „nmap -sP -iL“ otestuji všechny IP adresy ze souboru a filtrovaný výstup uložím do soboru TMP/rconfig_report. Uložené IP adresy jsou pouze těch zařízení, která jsou online.

8.6 Spouštění skriptu pomocí Cronu

Cron je aplikace v systému Linux (softwarový démon), který automatizovaně spouští v určitý čas nějaký proces (skript, program apod.). Jde o specializovaný systémový proces, který v operačním systému Linux Cent Os použiju pro spuštění naplánovaných úloh. Tyto úlohy umožnuje Cron spouštět pravidelně.

Příklad použití příkazu v Cronu se zapisuje takto:

```
# ┌──────────────── min (0 - 59)
# │ ┌────────────── hour (0 - 23)
# │ │ ┌──────────── day of month (1 - 31)
# │ │ │ ┌────────── month (1 - 12)
# │ │ │ │ ┌──────── day of week (0 - 6) (0 to 6 are Sunday to
Saturday, or use names; 7 is Sunday, the same as 0)
# │ │ │ │ │
# │ │ │ │ │
# * * * * *  command to execute

* * * * *  php /home/rconfig/lib/myscan.php
```

Obr. 50 Nastavení pro spouštění naplánované úlohy v Cronu

Ve složce etc je soubor crontab. Tento soubor jsem otevřel v editoru vim, kde jsem nastavil podle předlohy dvě naplánované úlohy pro spuštění skriptu myscan.php a updateNodes.php.

8.7 Aktuální počet všech zařízení

Pro zjištění aktuálního počtu všech zařízení jsem použil skript rConfig/lib/myscan.php, který se připojí do databáze rConfigu a vypíše seznam všech IP adres do souboru TMP/rconfig_houst. S tímto souborem dále pracujeme.

Obr. 51 Zobrazení celkového počtu zařízení

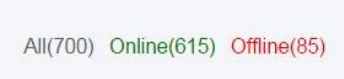
Ve skriptu myscan.php je funkce getOnlineHostsCount(), která jednoduše pomocí cyklu spočítá počet záznamů v souboru a tím určí kolik zařízení je celkem v databázi. Tuto hodnotu skript vrátí na stránku devices.php, kde v pravém horním rohu vidíme celkový počet zařízení. V soubor devices.php tyto zařízení počítá proměnná „stateSum“. Aktualizace tabulky je naplánovaná v programu Crone každou minutu.

8.8 Aktuální stav všech online / offline zařízení

V souboru device.php jsem definoval proměnou „stateSum“, která v zápise počítá všechny zařízení v prvním řádku.

V druhém řádku zjišťuje počet všech zařízení, která jsou online. Do proměnné „stateSun“ se uloží počet přes funkci „getOnlineHostCount()“, která vrátí počet všech online zařízení.

A ve třetím řádku v proměnné „stateSum“ od sebe tyto dvě hodnoty odečtu a získám počet všech zařízení, která nenavázali spojení a jsou offline.

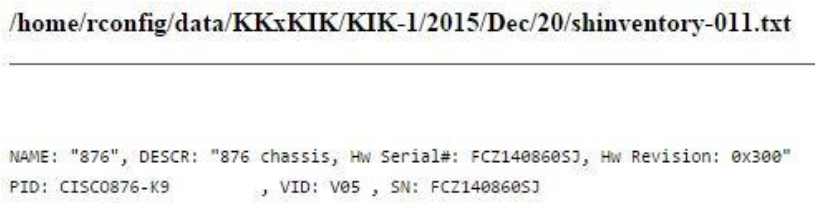


Obr. 52 Zobrazení počtu All / Online / Offline zařízení

8.9 Automatické generování sériového čísla

Aplikace rConfig má primárně sloužit k záloze konfigurací a zjišťování aktuální dostupnosti stavu zařízení. V případě kdy zjistíme, že nějaký router u zákazníka odejde a je potřeba vyměnit. Tak do nového routeru nahraji konfiguraci, kterou jsem stáhnul ze systému rConfig, ale neaktualizuje se sériové číslo zařízení, které je pro nás, v rámci výměny a evidence hardwaru, důležité.

V systému rConfig jsem zvolil položku „Devices“ a dále „Commands“ kde jsem konfiguroval další naplánovanou úlohu, která bude spouštět 1x denně příkaz „sh inventory“. Tento příkaz na zařízeních Cisco vypíše základní informace o zařízení.



Obr. 53 Výpis uložení po zadání příkazu „sh inventory“

Pro spočítání aktuálního stavu počtu zařízení jsem vytvořil skript ve složce rConfig/updateNodes.php. Tento skript bere zdrojová data ze složky

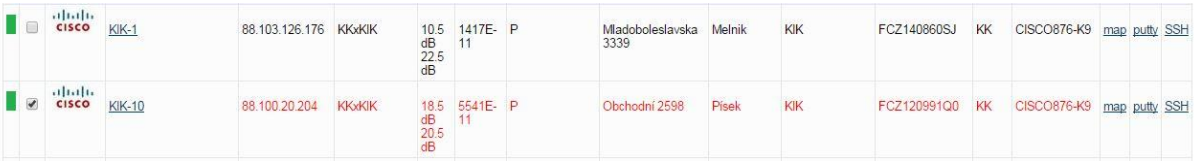
home/rconfig/data/. Dále synchronizuji světový čas podle „UTC“. Pro přístup do databáze musím definovat přístupové údaje. Pomocí sql dotazu zjistím název kategorie a název zařízení pod určitým ID záznamem.

getPathForNode

Cesta k aktuálnímu zařízení je definovaná ve funkci getPathForNode podle čísla ID z databáze. Pokud znám z jaké kategorie je zařízení, dále jeho hostname, tak vytvořím podle posledního časového údaje cestu do aktuálního adresáře zařízení s nejaktuálnějším datem. Dostane spojené položky. Na základě časové zóny uhádne soubor aktuálního data. Jestli v souboru nejsou data, tak se dotazuje na starší záznam.

getSerialfornode

Tato funkce getSerialfornode přijímá cestu kategorie/zařízení atd. Zjistí kde je soubor „shinventory-*.txt“ je uložený. Když soubor neexistuje tak skončí a nemáme data. Proměnná Content načte vše, co rozdělím po mezerách a vypíše poslední údaj z textového souboru a to je sériové číslo. Tuto hodnotu vrátí a aktualizuje položku sériového čísla.



		cisco	KIK-1	88.103.126.176	KKxKIK	10.5 dB 22.5 dB	1417E-11	P	Mladoboleslavska 3339	Melnik	KIK	FCZ140860SJ	KK	CISCO876-K9	map	putty SSH
		cisco	KIK-10	88.100.20.204	KKxKIK	18.5 dB 20.5 dB	5541E-11	P	Obchodní 2598	Písek	KIK	FCZ120991Q0	KK	CISCO876-K9	map	putty SSH

Obr. 54 Automaticky vygenerované sériové číslo

8.10 Diagnostika chybovosti BER na zařízení

Pro zjištění chybovosti BER jsem definoval automatický příkaz v rConfigu, opět položka Devices\Commands, kde jsem použil příkaz „sh dsl iterface | include Total BER“.

getBerForNode

Stále pokračuji ve skriptu updateNodes.php, kde je funkce getBerForNode. Opět přijímá cestu kategorie/zařízení atd. Zjistí kde je soubor „shdslinterface|includeTotalBER-*.txt“ je uložený. Opět rozdělíme podle mezer a funkce vrací poslední hodnotu. Vrácená hodnota se uloží do databáze.

Obr. 55 Vyfiltrovaná chybovost BER

Tento skript ukáže v rConfigu u zařízení jaká je chybovost linky. Pokud je chybovost menší než hodnota XX*10^{-12} může na lince docházet k častým výpadkům linky.

8.11 Diagnostika parametrů linky DSL Upstream/Downstream

Poslední údaj zjišťuje parametry DSL připojení tzv. Margin. Pro zjištění Marginu jsem definoval automatický příkaz v rConfigu. Položka Devices\Commands, kde jsem použil příkaz „sh dsl iterface | include Noise Margin“. Ukazatel Noise Margin u DSL linky nám ukazuje šumovou rezervu a hodnota by neměla překročit 26 dBm, jinak může dojít k omezenému spojení.

getMarginForNode

Další funkcí ve skriptu updateNodes.php je funkce getMarginForNode. Opět přijímá cestu kategorie/zařízení atd. Zjistí kde je soubor „shdslinterface|includeNoiseMargin-*.txt“ je uložený. Nyní rozdělí údaj podle dvojtečky a pravou část funkce vrací zpátky. Vrácená hodnota se uloží do databáze v podobě dvou údajů První číslo Margin pro upstream a druhý je pro downstream

Obr. 56 Výpis Margin vlevo upstream a vpravo downstream

updatehosts

Nejprve si na začátku vytáhnu seznam všechny aktivních zařízení příkazem sql. Každý řádek procházím zvlášť. Řádek je definovaný unikátním ID toho záznamu, kombinace cesty kategorie/zařízení. Do proměnné data ukládám do pomocného úložiště (pole) hodnotu. Pro každé pole si zachovám serial, ber a margin.

foreach

Tímto cyklem procházím celé pole a aktualizuji samostatné položky. Aktualizuji tabulku všech zařízení. Podle ID nastavíme každému záznamu správnou hodnotu, jakou má mít a na konci spouštíme funkci „updatehost“ která to všechno spustí.

Skript updateNodes.php je nastaven v Cronu, aby se spouštěl jednou denně v 23:30.

8.12 Zkopírování názvu zařízení do prompt

Pro urychlení zapsání nového zařízení do rConfigu, kde řádek deviceName ručně opisuji do řádku devicePrompt, jsem přidal funkci, které kopíruje řádek deviceName a není ho potřeba psát znovu. V řádku devicePrompt je akorát navíc znak hash, který slouží jako poznávací znak. Tuto funkci jsem implementoval do souboru rconfig\www\includes\head.inc.php.

9 Závěr

Mým úkolem bylo zálohovat databázi MySQL ze starého dohledového systému Nagios a použít ji k rozvoji nového dohledového systému. Dále jsem měl provést předvýběr vhodného systému, který by zálohoval a kontroloval připojení části zařízení, použitých u projektů řízených služeb. Do vybraného dohledového systému následně naimportovat aktualizovanou databázi zákazníků a zprovoznit její základní funkce. V neposlední řadě bylo mým úkolem do dohledového systému implementovat vhodné skripty k zajištění nových funkcionalit. Tento projekt jsem si vybral za svoji diplomovou práci ve společnosti ICZ a.s., kde jsem na praxi pod vedením odborného vedoucího Ing. Jana Carvy.

Po podrobném prozkoumání nejznámějších dohledových systémů, zhodnocení jejich funkcionalit, byl po rozhodnutí vedení sekce vybrán mnou doporučený dohledový systém rConfig, který splňoval veškeré požadavky.

Při jeho implementaci jsem postupoval tak, že jsem nejprve vytvořil zálohu databáze MySQL, kterou jsem si zkopíroval na externí disk. Tuto operaci jsem provedl pomocí FTP serveru a dostupných přihlašovacích údajů. Databázi bylo potřeba upravit a aktualizovat. Následně jsem hledal na internetu vhodný dohledový systém, který by uměl automaticky a pravidelně zálohovat konfigurace zařízení. Musel jsem vzít v potaz požadavky řízených služeb a náročnost dohledového systému.

Na aplikaci nového dohledového systému jsem dostal předem nainstalovaný server s Linuxem CentOS, na kterém jsem doinstaloval Apache server, MySQL server, PHP server a zkopíroval zálohovanou databázi. Dále jsem začal s instalací dohledového systému rConfig na server Godot. Instalaci jsem prováděl v příkazovém řádku v systému Linux. Nejdříve jsem aktualizoval systém a pak začal s instalací rConfigu. Na první pokus se mi nepovedlo dohledový systém zprovoznit. Vyskytla se zde chyba při instalaci systému. Následně jsem proces opakoval a rConfig se povedlo rozběhnout na stránce www.godot.i.cz. V poslední fázi instalace jsem zvolil připojení existující databáze zákazníků. Tím jsem propojil databázi MySQL a systém rConfig. Databázi bylo potřeba upravit a aktualizovat data, která jsou pro nás potřebná. Úpravy databáze jsem dělal pomocí programů MS Office Excel a MySQL Workbench. Po úspěšně importované databázi jsem začal konfigurovat v prostředí rConfigu. Začal jsem podle manuálu nastavovat stahování konfigurací a plánování úloh.

Do systému rConfig jsem úspěšně implementoval nové funkce. První funkce je zobrazení pozice zařízení na mapě. Tato funkce pomůže rychlému nalezení aktuálního zákazníka. Další užitečnou funkcí je automatické připojení SSH terminálu přes Putty (Windows). Při otevření rConfigu v OS Windows se po kliknutí na vybrané zařízení, automatické připojím přes Putty (SSH). Dále při zadání defaultního jména / hesla zařízení jsem přidal funkci zobrazení / skrytí defaultního hesla. Touto funkcí si mohu ověřit, zda je heslo zadané správně. Další potřebnou funkcí bylo zobrazení celkového počtu zařízení v databázi, z které filtruji všechny zařízení (funkce All / Online / Offline). K ověření dostupnosti zařízení jsem přidal do webového rozhraní kolonku status, která barevně monitoruje dostupnost zařízení. Při servisu zařízení a následné výměně zařízení jsem přidal funkci automatické aktualizace SN. Pokud je HW vyměněn, automaticky se přepíše SN v systému. Skript pravidelně jednou za den kontroluje SN u všech zařízení. Původní SN je uloženo ve starších složkách zálohy konfigurace a je možné jej využít například pro potřeby inventarizace. Další funkce vypisuje chybovosti BER dané linky. Pokud se chybovost zvyšuje, tak na lince můžeme pozorovat časté výpadky a linka bude třeba opravit. Poslední funkce vypisuje parametry DSL linky Noise Margin pro Upstream / Downstream. Tyto hodnoty popisují kvalitu signálu.

Funkce stahování konfigurací bez problému funguje na zařízeních Cisco, kde je poznávací znaménko pro zadání příkazu „hash“, nebo-li „#”. Po připojení přes SSH na zařízení, rConfig provede plánované konfigurační příkazy. Bohužel na Mikrotiku se mi zálohování konfigurací zařízení nepodařilo zprovoznit. Rozdíl mezi Mikrotikem a Cisco routerem z hlediska přístup je jiný prompt, Cisco má hash a Mikrotik využívá „>“. Po změně znaménka prompt je stále stejný problém, zařízení je sice připojeno, ale neprovádí se u něj záloha konfigurace. Myslím si, že tento problém je díky velké odezvě výpisu konfigurace na Mikrotiku (testoval jsem na zařízení RB751U-2HnD). Zatím tento problém řeším. Dále se při generování chybovosti BER a Noise Marginu nepravidelně stává, že se hodnota nevygeneruje. Tato chyba je pravděpodobně způsobena tím, že pro dané zařízení se parametr zjistí jiným příkazem, než je definováno v konfiguračním příkazu rConfigu. Tento problém se dá vyřešit tak, že kategorie budou rozdělené podle typu zařízení a pro každou kategorii se spustí jiný skript. Na řízených službách se převážně využívají Cisco 8XX u kterých tato funkce funguje. Dohledový nástroj rConfig je velkým přínosem v řízených službách. Zde pomáhá nejvíce při zálohování konfigurace

zařízení a zjišťování dostupnosti aktivních zařízeních. Tento systém by mohl být nasazen na více projektech.

Výhledově bych rád implementoval do rConfigu funkční přístup terminálu z OS Linux a zprovoznil zálohování konfigurace pro zařízení Mikrotik. Myslím si, že rConfig splnil své očekávání ohledně kontroly stavu a zálohování konfigurací zařízení. Dále bych rád do budoucna rozšířil uvedené funkcionality o sběr statistik ze všech zařízení, nejlépe formou přehledných grafů. Závěrem však mohu konstatovat, že tyto funkce není vhodné implementovat přes dohledový systém rConfig. Pro sběr údajů, vytváření statistik a grafů je vhodnější zvolit dohledový systém Cacti nebo Nagios. Uvedené systémy nabízí tyto funkce již v základním módu.

Na úplný závěr mohu konstatovat, že práce v rConfigu byla často poměrně náročná z důvodu nesystematičnosti otevřeného kódu. Námětem pro zlepšení tohoto systému by tedy bylo kód rozdělit na méně obsáhlé moduly. Tento systém se jeví jako velice perspektivní a vhodný pro implementaci dalších funkcí. V současné době je systém využíván nejen technickými specialisty, ale rovněž projektovými manažery, kteří využívají nově implementované funkcionality.

10 Seznam použité literatury

[1] KLAŠKA, Petr, © 1999. FDDI – osvědčený standard [online]. [cit. 2015-11-1]. Dostupné z: http://www.svetsiti.cz/clanek.asp?cid=FDDI-osvedceny-standard-1891999

[2] PETERKA, Jiří, © 2002. ATM – technologie která nezvítězila [online]. [cit. 2015-11-1]. Dostupné z: http://www.earchiv.cz/b02/b0300023.php3

[3] BOUŠKA, Petr, © 2008. Počítačová síť [online]. [cit. 2015-11-1]. Dostupné z: http://www.samuraj-cz.com/clanek/pocitacove-site-a-jejich-typy/

[4] BOUŠKA, Petr, © 2008. Ethernet – CSMA/CD, kolizní doména, duplex [online]. [cit. 2015-11-1]. Dostupné z: http://www.samuraj-cz.com/clanek/ethernet-csmacd-kolizni-domena-duplex/

[5] BOUŠKA, Petr, © 2011. VPN 1 – Ipsec VPN a Cisco [online]. [cit. 2015-11-1]. Dostupné z: http://www.samuraj-cz.com/clanek/vpn-1-ipsec-vpn-a-cisco/

[6] STOPKA, Marek, © 2010. Storage Area Network [online]. [cit. 2015-11-1]. Dostupné z: http://www.abclinuxu.cz/clanky/storage-area-network-1-uvod

[7] VANĚK, Tomáš, © 2014. Prezentace ZST – Základy síťových technolgií [online]. [cit. 2015-11-1]. Dostupné z: http://www.kme.moodle.cvut.cz

[8] BURKOŇ, Lukáš, © 2009. Unifikovaná definice služby [online]. [cit. 2009-3-1]. Dostupné z: http://www.cssi.cz/cssi/unifikovana-definice-sluzby

[9] BEZPALEC, Pavel, © 2015. Management ICT systémů [online]. [cit. 2015-10-26]. Dostupné z: https://publi.cz/books/242/01.html

[10] CARVA, Jan, © 2011. *Outsourcing v ICT*. Praha. Bakalářská práce (Bc.). České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra ekonomiky, manažerství a humanitních věd. Vedoucí práce Pavel Náplava.

[11] CARVA, Jan, © 2014. *Možnosti využití Managed Services v odvětví služeb* Praha. Diplomová práce (Ing.). České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra ekonomiky, manažerství a humanitních věd. Vedoucí práce Pavel Náplava.

[12] UČEŇ, Pavel, © 2015. Service Level Agreement aplikačních služeb [online]. [cit. 2015-9-21]. Dostupné z: http://www.systemonline.cz/clanky/service-level-agreement-aplikacnich-sluzeb.htm

[13] Obecná pravidla a doporučení pro využívání řízení datového provozu při poskytování služby přístupu k síti internet, © 2015 [online]. [cit. 2015-8-29]. Dostupné z: http://www.ctu.cz/cs/download/datovy_provoz/rizeni_datoveho_provozu_obecna_pravidla-doporuceni_19_12_2013.pdf

[14] BOUŠKA Petr, © 2009. Začínáme s monitoringem sítě [online]. [cit. 2015-11-1]. Dostupné z: http://www.samuraj-cz.com/clanek/zaciname-s-monitoringem-site/

[15] SOCHOR, Tomáš, © 2009. *Počítačové sítě II , skripta pro distanční studium. Vydala Ostravská univerzita v Ostravě, Přírodovědecká fakulta, Ostrava 2009*

[16] PETERKA, Jiří, © 2002. Protokol TCP [online]. [cit. 2015-11-1]. Dostupné z: http://www.earchiv.cz/a93/a305c110.php3
[17] PETERKA, Jiří, © 2012. Protokol IPv4 [online]. [cit. 2015-11-11]. Dostupné z: http://www.earchiv.cz/b12/b0917001.php3
[18] EMANUEL, Petr a Ondřej Surý, © 2011. Computerworld – IPV6 [online]. [cit. 2015-11-11]. Dostupné z: https://www.nic.cz/files/nic/doc/Computerworld_IPv6_122010.pdf
[19] BOUŠKA, Petr, © 2009. Začínáme s monitoringem sítě [online]. [cit. 2015-11-1]. Dostupné z: http://www.samuraj-cz.com/clanek/snmp-simple-network-management-protocol/
[20] BOUŠKA, Petr, © 2006. SNMP – Simple Network Management Protokol [online]. [cit. 2015-11-1]. Dostupné z: http://www.businessit.cz/cz/kdy-a-jak-prejit-z-ipv4-na-ipv6-rady-tipy.php
[21] *Multiagent System Implementation for Network.* **M., Néstor D. Duque, a další.** Berlin : Springer-Verlag Berlin Heidelberg, 2009.
[22] BĚLINA, Radek, © 2008. Proč a jak využívat dohledový systém [online]. [cit. 2015-10-1]. Dostupné z: http://www.systemonline.cz/sprava-it/proc-a-jak-vyuzivat-dohledovy-system.htm
[23] KLAŠKA, LUBOŠ, @2000. SNMP objekty a MIB [online]. [cit. 2015-10-2]. Dostupné z: http://www.svetsiti.cz/clanek.asp?cid=SNMP-objekty-a-MIB-1362000
[24] TOUCH, Joe; Eliot Lear, Allison Mankin, Markku Kojo, Kumiko Ono, Martin Stiemerling@2015. Service Name and Transport Protocol Port Number Registry [online]. [cit. 2015-12-10]. Dostupné z:
[24] http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml
[25] BOCH, Radek, @2012. Co vše umí dnešní přístupová vrstva sítě [online]. [cit. 2015-12-12]. Dostupné z: http://www.cisco.com/web/CZ/expo2012/pdf/T-NET1_NET2_DnesniPristupovaVrstva_Radek_Boch.pdf
[26] KOLÍSEK, Antonín, @2013. Dohledový systém Zabbix - představení I. sítě [online]. [cit. 2015-12-14]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=1963
[27] RCONFIG@2015. Dohledový systém rConfig [online]. [cit. 2015-12-12]. Dostupné z: http://www.rconfig.com/
[28] PAŘÍK, Tadeáš, @2013. Nagios [online]. [cit. 2015-11-19]. Dostupné z: http://wiki.ubuntu.cz/nagios
[29] KINDERNAY, David, © 2015. Obrázek – Počítačová síť [online]. [cit. 2015-11-1]. Dostupné z: http://pocitacovyzachranar.cz/wp-content/uploads/2013/06/it.jpg
[30] BOUŠKA, Petr, © 2012. System Center Configuration Manager 2012 [online]. [cit. 2015-12-5. Dostupné z: http://www.samuraj-cz.com/clanek/system-center-configuration-manager-2012-instalace/

11 Přílohy

Výpis skriptu myscan.php:

```
<?php

define('HOSTS_FILENAME', '/tmp/rconfig_hosts');
define('REPORT_FILENAME', '/tmp/rconfig_report');

$mysql_config = array(
        'hostname' => ':/tmp/mysql.sock',
        'username' => 'root',
        'password' => 'XXXXXXXX',
        'database' => 'db_rconfig00'
);

/**
 * Funkce vytahne z databaze seznam IP hostu
 *
 */
function getHostList($conf)
{
        $fw = fopen(HOSTS_FILENAME, 'w');

        $conn = mysql_connect($conf['hostname'], $conf['username'], $conf['password']);
        mysql_select_db($conf['database'], $conn);

        $query = "SELECT DISTINCT deviceIpAddr FROM nodes WHERE status = 1 ORDER BY deviceIpAddr";
        $result = mysql_query($query);

        if (!$result)
                die("MySQL Query Error: " . mysql_error() . "\n");

        while ($row = mysql_fetch_row($result))
        {
                if (empty($row))
                        continue;

                $ip = trim($row[0]);

                // IP obsahuje masku - jake nevime skenovat
                if (strpos($ip, '/') !== false)
                        continue;

                // není  platnou ip adresou
                if (filter_var($ip, FILTER_VALIDATE_IP) === false)
                        continue;

                // jinak zapiseme host do souboru
                fwrite($fw, $ip."/32\n");
        }

        fclose($fw);
}

/**
 * Funkce zjisti, jestli hosty jsou online
 * Seznam IP je ulozen v soubore s definovanou konstantou HOSTS_FILENAME
 *
 * Vyzaduje dostupnou funkci PHP: system() - default nepovolena konfiguraci PHP
 * Vyzaduje program nmap
 */
function scanHosts()
{
        // nmap
        // -sP  -> skenuj pingom
        // -iL  -> zoznam zo suboru
        $fname = HOSTS_FILENAME;
        $report = REPORT_FILENAME;
        $report_tmp = REPORT_FILENAME . "tmp";
        $command = "nmap -sP -n -iL $fname | grep -i 'nmap scan report' | rev | cut -d' ' -f1 | rev > $report_tmp";
```

```
	system($command);
	system("mv $report_tmp $report");
}


/**
 * Funkce overi, jestli je host online
 *
 * @return bool		true kdyz je online, false v jinom pripade (offline nebo nevime)
 */
function checkHost($ip)
{
	$report = REPORT_FILENAME;
	$fr = fopen($report, "r");

	while (($line = fgets($fr)) !== false)
	{
		$line = trim($line);

		if ($line == $ip)
			return true;
	}

	fclose($fr);

	return false;
}

/**
 * Funkce spocte, kolik hostu je online
 *
 */
function getOnlineHostsCount()
{
	$report = REPORT_FILENAME;
	$fr = fopen($report, "r");

	$i = 0;

	while (($line = fgets($fr)) !== false)
		$i++;

	fclose($fr);

	return $i;
}

// jestli je skript spusteny primo v prikazovom radku s argumentem scan,
// oskenuju vsechny zarizeni
//
// Nastavene automaticke spusteni skriptu v pravidelnych
// intervaloch v daemonu "cron"
//
// do /etc/crontab byl pridan radek
// */5 * * * * root  php /cesta/k/skriptu/myscan.php scan
if ( count($argv) == 2 && $argv[1] == 'scan' )
{
	// nacteme si seznam stroju
	getHostList($mysql_config);
	// oskenujeme stroje
	scanHosts();
}
```

Výpis skriptu updateNodes.php:

```
<?php

define('DATA_PATH', '/home/rconfig/data/');
date_default_timezone_set('UTC');

$mysql_config = array(
		'hostname' => ':/tmp/mysql.sock',
		'username' => 'root',
		'password' => 'XXXXXXXX',
		'database' => 'db_rconfig00'
);


$sql = "select distinct n.id, concat(c.categoryName, '/', n.deviceName) as path from nodes n inner join categories c on c.id = n.nodeCatId where
n.status = 1 and c.status = 1";

function getPathForNode($nodeDataPath)
{
		$path = DATA_PATH . $nodeDataPath;
		$path .= '/' . date('Y/M/d/');

		if ( ! is_dir($path) )
			return "";

		return $path;
}

function getSerialForNode($nodeDataPath)
{
		if ( ($path = getPathForNode($nodeDataPath)) == "" ) return "";

		$path = glob( $path . "shinventory-*.txt" );

		if ( empty($path) )
			return "";

		$path = $path[0];

		$content = file_get_contents($path);
		$tokens = explode(' ', $content);
		$serial = $tokens[ count($tokens)-1 ];

		return $serial;
}

function getBerForNode($nodeDataPath)
{
		if ( ($path = getPathForNode($nodeDataPath)) == "" ) return "";

		$path = glob( $path . "shdslinterface|includeTotalBER-*.txt" );

		if ( empty($path) )
			return "";

		$path = $path[0];

		$content = file_get_contents($path);
		$tokens = explode(' ', $content);
		$ber = $tokens[ count($tokens)-1 ];

		return $ber;
}


function getMarginForNode($nodeDataPath)
{
		if ( ($path = getPathForNode($nodeDataPath)) == "" ) return "";

		$path = glob( $path . "shdslinterface|includeNoiseMargin-*.txt" );

		if ( empty($path) )
```

```
                return "";

        $path = $path[0];

        $content = file_get_contents($path);
        $tokens = explode(':', $content);
        unset($tokens[0]);
        $margin = join(' ', $tokens);

        return $margin;
}


function updateHosts($conf)
{
        $conn = mysql_connect($conf['hostname'], $conf['username'], $conf['password']);
        mysql_select_db($conf['database'], $conn);

        global $sql;
        $result = mysql_query($sql);

        if (!$result)
                die("MySQL Query Error: " . mysql_error() . "\n");

        $data = array();

        while ($row = mysql_fetch_row($result))
        {
                if (empty($row))
                        continue;


                $data[] = array(
                        'id' => intval($row[0]),
                        'serial' => getSerialForNode($row[1]),
                        'ber' => getBerForNode($row[1]),
                        'margin' => getMarginForNode($row[1]),
                );
        }

        foreach ($data as $node)
        {
                $serial = $node['serial'];
                $ber = $node['ber'];
                $id = $node['id'];
                $margin = $node['margin'];

                if ( !empty($serial) )
                {
                        $update_sql = "UPDATE nodes SET custom_Serial = '$serial' WHERE id = $id";
                        if(! mysql_query($update_sql) )
                                echo "MySQL Error: " . mysql_error(). "\n";
                }

                if ( !empty($ber) )
                {
                        $update_sql = "UPDATE nodes SET custom_BER = '$ber' WHERE id = $id";
        if(! mysql_query($update_sql) )
            echo "MySQL Error: " . mysql_error(). "\n";

                }

        if ( !empty($margin) )
        {
            $update_sql = "UPDATE nodes SET custom_Margin = '$margin' WHERE id = $id";
            if(! mysql_query($update_sql) )
                echo "MySQL Error: " . mysql_error(). "\n";
        }
            }
}

updateHosts($mysql_config);
```

Výpis skriptu device.php:

```
<?php include("includes/head.inc.php"); ?>
<?php

$_db = new db();
$_rs          = $_db->q('SELECT COUNT(*) AS total FROM nodes WHERE status = 1');
$_row         = mysql_fetch_row($_rs);
$_result["total"] = $_row[0];

include_once("../lib/myscan.php"); // contains getOnlineHostsCount();

$stateSum = new stdClass();
$stateSum->all = intval($_result["total"]);
$stateSum->online = getOnlineHostsCount();
$stateSum->offline = $stateSum->all - $stateSum->online;

?>

<body>
<!-- Headwrap Include -->
<?php include("includes/masthead.inc.php"); ?>
<div id="mainwrap">
        <!-- TopNav Include -->
        <?php include("includes/topnav.inc.php"); ?>
        <div id="main">
        <?php
        /* Custom Devices Custom Form Functions */
        require_once("lib/crud/devices.frm.func.php");
        ?>
                <!-- Breadcrumb Include -->
                <?php include("includes/breadcrumb.inc.php"); ?>
                <!-- Announcement Include -->
                <?php include("includes/announcement.inc.php"); ?>
                <div id="content">
                        <?php
                                if(isset($_SESSION['errors'])){ $errors = $_SESSION['errors']; }
                                // below are populated if errors are sent back from CRUD script to re-populate from

                                if(isset($_SESSION['deviceName'])){     $deviceName     =     $_SESSION['deviceName'];
unset($_SESSION['deviceName']); }
                                if(isset($_SESSION['deviceIpAddr'])){   $deviceIpAddr   =   $_SESSION['deviceIpAddr'];
unset($_SESSION['deviceIpAddr']); }
                                if(isset($_SESSION['devicePrompt'])){   $devicePrompt   =   $_SESSION['devicePrompt'];
unset($_SESSION['devicePrompt']); }
                                if(isset($_SESSION['vendorId'])){       $vendorId       =       $_SESSION['vendorId'];
unset($_SESSION['vendorId']); }
                                if(isset($_SESSION['deviceModel'])){    $deviceModel    =   $_SESSION['deviceModel'];
unset($_SESSION['deviceModel']); }
                                if(isset($_SESSION['deviceUsername'])){ $deviceUsername = $_SESSION['deviceUsername'];
unset($_SESSION['deviceUsername']); }
                                if(isset($_SESSION['devicePassword'])){ $devicePassword = $_SESSION['devicePassword'];
unset($_SESSION['devicePassword']); }
                                if(isset($_SESSION['devicePassConf'])){ $devicePassConf = $_SESSION['devicePassConf'];
unset($_SESSION['devicePassConf']); }
                                if(isset($_SESSION['deviceEnableMode'])){               $deviceEnableMode               =
$_SESSION['deviceEnableMode']; unset($_SESSION['deviceEnableMode']); }
                                if(isset($_SESSION['deviceEnablePassword'])){           $deviceEnablePassword           =
$_SESSION['deviceEnablePassword']; unset($_SESSION['deviceEnablePassword']); }
                                if(isset($_SESSION['catId'])){ $catId = $_SESSION['catId']; unset($_SESSION['catId']); }
                                if(isset($_SESSION['deviceAccessMethodId'])){           $deviceAccessMethodId           =
$_SESSION['deviceAccessMethodId']; unset($_SESSION['deviceAccessMethodId']); }
                                if(isset($_SESSION['connPort'])){       $connPort       =       $_SESSION['connPort'];
unset($_SESSION['connPort']); }


                                /* "Do NOT unset the whole $_SESSION with unset($_SESSION) as this will disable the registering
of session variables through the $_SESSION superglobal." */
                                $_SESSION['errors'] = array();
                        ?>
                        <fieldset id="tableFieldset">
                                <legend>Device Management</legend>
                                <?php
```

```
                                        if(isset($errors['Success'])){echo "<span class=\"error\">".$errors['Success']."</span><br/>";}
                                        if(isset($errors['Fail'])){echo "<span class=\"error\">".$errors['Fail']."</span><br/>";}
                                        if(isset($errors['username'])){echo "<span class=\"error\">".$errors['username']."</span><br/>";}
                                ?>
                                <div id="toolbar">
                                        <div id="toolbarButtons" style="float: left;">
                                                <button class="show_hide">Add Device</button>
                                                <button onclick="editDevice()">Edit Device</button>
                                                <button onclick="copyDevice()">Copy Device</button>
                                                <button onclick="delDevice()">Remove Device</button>
                                                <!-- <button class="show_import"><img src="images/icon_import_dark.gif"/>  Bulk Import</button> -->
                                        </div>
                                        <!-- end toolbarButtons -->
                                        <div id="statusInfo" style="float: right;">
                                                <span>All(<?php echo $stateSum->all; ?>)</span>
                                                <span style="color: green;">Online(<?php echo $stateSum->online; ?>)</span>
                                                <span style="color: red;">Offline(<?php echo $stateSum->offline; ?>)</span>
                                        </div>
                                </div>
                                        <!-- <div class="bulkImportDiv">
                                        <img src="images/helpIcon16.png" /><b> Upload CSV file:</b> <input type="file" >
                                </div> -->
                                <!-- begin devices form -->
                                <div class="mainformDiv">
                                        <form method="post" action="lib/crud/devices.crud.php" class="myform stylizedForm stylized">
                                                <div id="left">
                                                        <legend>Device Details</legend>
                                                        <label for="deviceName"><font color="red">*</font> Device Name:</label>
                                                        <input name="deviceName" id="deviceName" placeholder="Device Name" tabindex='1' style="width:150px;" value="<?php if(isset($deviceName)) echo $deviceName;?>" onkeyup="copy_data(this)">
                                                        <div class="spacer"></div>
                                                        <?php if(isset($errors['deviceName'])){echo "<span class=\"error\">".$errors['deviceName']."</span>";}?>

                                                        <label><font color="red">*</font> IP Address:</label>
                                                        <span class="small"><a href="javascript:void(0)" onclick="resolveDevice(document.getElementById('deviceName').value);">resolve device name</a></span>
                                                        <input name="deviceIpAddr" id="deviceIpAddr" placeholder="x.x.x.x" tabindex='2' style="width:150px;" value="<?php if(isset($deviceIpAddr)) echo $deviceIpAddr;?>">
                                                        <div class="spacer"></div>
                                                        <?php if(isset($errors['deviceIpAddr'])){echo "<span class=\"error\">".$errors['deviceIpAddr']."</span>";}?> <br/>

                                                        <label><font color="red">*</font> Prompt:</label>
                                                        <input name="devicePrompt" id="devicePrompt" placeholder="router#" tabindex='2' style="width:150px;" value="<?php if(isset($devicePrompt)) echo $devicePrompt;?>">
                                                        <div class="spacer"></div>
                                                        <?php if(isset($errors['devicePrompt'])){echo "<span class=\"error\">".$errors['devicePrompt']."</span>";}?> <br/>


                                                        <label><font color="red">*</font> Vendor:</label>
                                                        <select name="vendorId" id="vendorId" tabindex='3' style="width:155px;">
                                                                <?php
                                                                        if(isset($vendorId)) {
                                                                                vendorId($vendorId);
                                                                        } else {
                                                                                vendorId();
                                                                        }
                                                                        /* taken from devices.frm.func.php */
                                                                ?>
                                                        </select>
                                                        <div class="spacer"></div>
```

```
<?php if(isset($errors['vendorId'])){echo "<span class=\"error\">".$errors['vendorId']."</span>";}?>

<label><font color="red">*</font>Model:</label>
<input name="deviceModel" id="deviceModel" placeholder="Model" tabindex='4' style="width:150px;" value="<?php if(isset($deviceModel)) echo $deviceModel;?>">
<div class="spacer"></div>
<?php if(isset($errors['deviceModel'])){echo "<span class=\"error\">".$errors['deviceModel']."</span>";}?>

</div>

<div id="right">
<legend>Other Details</legend>
<label>Category:</label>
<select name="catId" id="catId" style="width:155px;" tabindex='5' value="<?php if(isset($catId)) echo $catId;?>">

<?php
if(isset($catId)) {
categories($catId);
} else {
categories();
}
/* taken from devices.frm.func.php */
?>
</select>
<div class="spacer"></div>
<?php if(isset($errors['catId'])){echo "<span class=\"error\">".$errors['catId']."</span>";}?> <br/>

<label>Custom Properties:</label>
<?php customProp(); /* taken from devices.frm.func.php */ ?>
<br/>
<br/>
</div>

<div id="left">
<legend>Credentials</legend>

<label><font color="red">*</font> Username:</label>
<span class="small"><a href="javascript:void(0)" onclick="getDefaultUserPass();">default username/password</a></span>
<input name="deviceUsername" id="deviceUsername" placeholder="username" tabindex='6' style="width:150px;" value="<?php if(isset($deviceUsername))echo $deviceUsername;?>" autocomplete="off">
<div class="spacer"></div>
<?php if(isset($errors['deviceUsername'])){echo "<span class=\"error\">".$errors['deviceUsername']."</span>";}?>
<br/>

<label><font color="red">*</font> Password:</label>
<input type="password" name="devicePassword" id="devicePassword" placeholder="password" tabindex='6' style="width:150px;" value="<?php if(isset($devicePassword)) echo $devicePassword;?>" autocomplete="off"> <a href="#" id="passView" onclick="viewpass();">view</a>
<div class="spacer"></div>
<?php if(isset($errors['devicePassword'])){echo "<span class=\"error\">".$errors['devicePassword']."</span>";}?> <br/>

<label><font color="red">*</font> Confirm Password:</label>
<input type="password" name="devicePassConf" id="devicePassConf" placeholder="password" tabindex='7' style="width:150px;" value="<?php if(isset($devicePassConf)) echo $devicePassConf;?>" autocomplete="off">
<div class="spacer"></div>
<?php if(isset($errors['devicePassConf'])){echo "<span class=\"error\">".$errors['devicePassConf']."</span>";}?> <br/>

<label>Enable Mode:</label>
<input type="checkbox" name="deviceEnableMode" id="deviceEnableMode" tabindex='8' style="width:15px;" <?php if(isset($deviceEnableMode) && $deviceEnableMode !== 'off') echo 'checked';?>>
<br/>

<label>Enable Password:</label>
```

```
                                                <input type="password" name="deviceEnablePassword" id="deviceEnablePassword" placeholder="Enable Password" tabindex='9' style="width:150px;" value="<?php if(isset($deviceEnablePassword)) echo $deviceEnablePassword;?>" autocomplete="off">
                                                <div class="spacer"></div>
                                                <?php if(isset($errors['deviceEnableMode'])){echo "<span class=\"error\">".$errors['deviceEnableMode']."</span>";}?> <br/>

                                                <label> Access Method:</label>
                                                <select name="deviceAccessMethodId" id="accessMeth" tabindex='10' style="width:155px;" onchange="updatePort(this.value);">
                                                        <?php
                                                                if(isset($deviceAccessMethodId)) {

        accessMethod($deviceAccessMethodId);
                                                                }else {
                                                                accessMethod();
                                                                }
                                                                /* taken from devices.frm.func.php */
                                                        ?>
                                                </select>

                                                <label for="connPort">Connection Port:</label>
                                                <input name="connPort" id="connPort" title="'TCP Port Number for connection" size="10" maxlength="5" value="22" tabindex='11' style="width:40px;" value="<?php if(isset($connPort)) echo $connPort;?>">
                                                <div style="float:left;">
                                                <img id="helpIcon" src="images/helpIcon16.png" style="margin-top:5px;margin-left:5px;" alt="Please select your TCP port number for the access method i.e. telnet = 23" title="Please select your TCP port number for the access method i.e. telnet = 23"/>
                                                </div>
                                        </div>
                                        <input type="hidden" id="add" name="add" value="add">
                                        <input type="hidden" id="username" name="username" value="<?php echo $session->username; ?>">
                                        <input type="hidden" id="editid" name="editid" value="">
                                        <!-- end devices form -->
                                        <div id="saveButtons">
                                                <button class="" tabindex='13' id="submit" type="submit">Save</button>
                                                <button class="show_hide" type="button" tabindex='14'>Close</button>
                                        </div>
                                </form>
                        </div>
                        <!-- end mainformDiv -->
                        <div id="devicesTable">
                                <?php include("devices.inc.php");?>
                        </div>
<br />
Pozn. Pro pouziti Putty z prohlizece je potreba nainstalovat aplikaci URLconf (ke stazeni <a href="downloads/UrlConf.msi">zde</a>), spustit ji jako locadmin a zaskrtnout pouziti ssh (viz. <a href="downloads/urlconfobr.png">obr</a>). Dale je potreba mit nainstalovany Putty v adresari C:\Program Files\PuTTY\PuTTY.exe
<br />
                                </fieldset>
                        </div>
                        <!-- End Content -->
                        <div style="clear:both;"></div>
                        <!-- JS script Include -->
                        <script type="text/JavaScript" src="js/devices.js"></script>
        </div>
        <!-- End Main -->
        <!-- Footer Include -->
        <?php include("includes/footer.inc.php"); ?>
</div>
<!-- End Mainwrap -->
</body>
</html>
```

Výpis části skriptu devices.inc.php

Určení pozice na mapě:

```
<td align="left"><?php echo $rows['model'] ?></td>
<td >
    <a href="http://maps.google.com/maps?q=<?php echo $rows['custom_Mesto']?> <?php echo $rows['custom_Ulice']?>" title="google map">map
    </td>
<td >
    <a href="ssh://<?php echo $rows['deviceIpAddr']?>" title="putty">putty
    </td>

<td >
```

Změna grafického statusu online/offline:

```
<td align="center">
            <?php $barva = (checkHost($rows['deviceIpAddr'])) ? "zelena" : "cervena"; ?>
            <img src="/images/status/<?php echo $barva; ?>.jpg"/>
  </td>
```

Výpis skriptu head.inc.php pro zobrazení hesla a kopírovaní deviceName do prompt:

```
<script type="text/javascript">

            function copy_data(val){
            var a = document.getElementById(val.id).value + "#"
            document.getElementById("devicePrompt").value=a
            }

            var bool=false;
            function viewpass(){
            if(bool==false){
            document.getElementById("devicePassword").type="text";
            document.getElementById("passView").innerHTML="hide";
            bool=true;
            }
            else {
            document.getElementById("devicePassword").type="password";
            document.getElementById("passView").innerHTML="view";
            bool=false;
            }

            }
</script>
```

Použitý skript z linuxového daemona Cron:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

* * * * * root  php /home/rconfig/lib/myscan.php scan
30 23 * * * root php /home/rconfig/updateNodes.php
```