VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

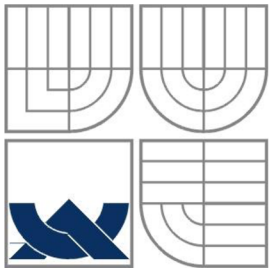
MONITOROVACÍ A DOHLEDOVÝ SYSTÉM POČÍTAČOVÉ SÍTĚ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

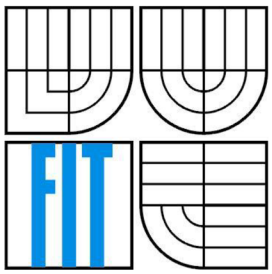
AUTOR PRÁCE
AUTHOR

BC. PAVEL MÍČA

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MONITOROVACÍ A DOHLEDOVÝ SYSTÉM POČÍTAČOVÉ SÍTĚ

SYSTEM FOR MONITORING AND SUPERVISORY OF COMPUTER NETWORK

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. PAVEL MÍČA

VEDOUCÍ PRÁCE
SUPERVISOR

MGR. ROMAN TRCHALÍK

BRNO 2008

Abstrakt

Práce rozebírá problematiku monitoringu počítačové sítě a na jejím základě definuje řešení tvorby dohledového a monitorovacího systému pro takovou síť. Praktický základ práce tvoří skutečná počítačová síť poskytovatele bezdrátového internetového připojení, založená na routovacím systému Mikrotik. První část práce se zabývá rozborem a kritickým zhodnocením uvedeného systému Mikrotik. Práce pokračuje analýzou požadavků na vytvářený systém a stanovením metodologie návrhu a následné implementace. Funkční struktura systému z pohledu uživatele je definována vytvořeným use case modelem, reprezentovaným grafickou i textovou formou. Navazující kapitola informuje o konkrétních přístupech k realizaci definovaných funkčních částí systému a nakonec také stanovuje použité vývojové prostředky a nástroje. Závěr shrnuje výsledky celé práce a naznačuje další možná rozšíření do budoucna.

Klíčová slova

monitorování LAN, Mikrotik, RouterOS, Internet Service Provider, SNMP, RRDtool, SSH

Abstract

This diploma thesis deals with development of the system for monitoring and supervisory of computer network, which can monitor and control computer network. This information system is based on real experiences from the real running of network, which is based on specialized routing operation system Mikrotik. The analysis phase describes problems of monitoring and control in the real world and in the real network. Main kernel of this thesis is focused to analysis of requirements to system for monitoring and supervisory of computer network based on Mikrotik and makes the acquaintance of design and implementation methods. Functional system structure is described by graphical and textual use case model. The next chapter describes realization of key system parts and determines used development technologies. Last chapter summarize whole thesis and shows possible future improvements.

Keywords

LAN monitoring, Mikrotik, RouterOS, Internet Service Provider, SNMP, RRDtool, SSH

Citace

Míča Pavel: Monitorovací a dohledový systém počítačové sítě. Brno, 2008, diplomová práce, FIT VUT v Brně.

Monitorovací a dohledový systém počítačové sítě

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Mgr. Romana Trchalíka.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Pavel Míča
10.5.2008

Poděkování

Rád bych tímto poděkoval Mgr. Romanu Trchalíkovi za poskytnuté rady a odborné vedení celé práce.

Obsah

1 Úvod

Komunikace tvoří důležitý základ každého společenství živých bytostí, neboť dovoluje sdělovat informace, myšlenky či pocity a tím umožňuje rozvoj takového společenství. Lidská komunikace je založena především na jazyku a řeči, které jsou součástí tzv. verbální komunikace.

Překotný vývoj výpočetní techniky v několika posledních desetiletích předznamenal nové možnosti v lidské komunikaci a vytyčil cestu, po které se bude vyvíjet v nedaleké budoucnosti. Již v počátcích této informační revoluce se začali objevovat požadavky na vzájemnou komunikaci mezi vytvářenými datovými centry se sálovými počítači. v 60 letech minulého století se tak začínají objevovat první pokusy o vytvoření komunikačních kanálů mezi jednotlivými počítači, za již vysloveným účelem výměny a tedy i sdílení informací. v následujících letech strmě stoupali požadavky na takovéto kanály, především z pohledu rychlosti a spolehlivosti. v rámci zdokonalování a standardizace vzniklo velké množství technologií a protokolů, z nichž se celá řada používá dodnes. Příkladem může být rodina protokolů TCP/IP vzniklá na přelomu 70. a 80. let, která je dnes používána k realizaci naprosté většiny síťových spojení. Uvedené řešení bylo výsledkem snahy vytvořit robustnější model síťové komunikace schopný do určité míry odolávat výpadkům částí sítě. Podstatou celého řešení je myšlenka neposílat data jako souvislý proud informací (stream), ale po malých částech (paketech), které mohou být do cíle dopraveny různými cestami. Uvedený postup dovoluje jednoduché řešení problému ztráty takových paketů. Dané řešení se nazývá přepínání paketů (paket switching) a díky své jednoduchosti a flexibilitě se stalo dominantním řešením v oblasti přenosu digitální informace. v praxi je myšlenka paket switchingu realizována sadou protokolů rozdělených z důvodů zjednodušení celého procesu do několika úrovní (vrstev). Nejčastěji uváděným modelem je referenční model ISO/OSI vytvořený organizací ISO, který byl roku 1984 přijat jako mezinárodní norma pod označením ISO 7498. Samotný model nedefinuje žádné protokoly a způsoby své implementace, ale pouze rozděluje celý proces přenosu informace do sedmi vrstev. Účelem tohoto rozdělení je snaha izolovat jednotlivé úlohy, jejichž řešením je úspěšná komunikace dvou systémů. Každá z vrstev tak vymezuje skupinu přesně definovaných funkcí nutných pro komunikaci. Funkce ke své činnosti přitom využívají služeb vrstvy nižší třídy a poskytují své služby vrstvě třídy vyšší. Samotné protokoly implementují tyto funkce a stanovují tím již konkrétní přístupy pro řešení daných úloh v rámci své vrstvy. Komunikující systémy tak musí podporovat na určité konkrétní vrstvě stejné protokoly.

Předchozí odstavec definoval komplexní řešení komunikace dvou systémů. Ve skutečnosti však v drtivé většině případů vzájemně komunikuje nespočet takových systémů (počítačů). Jejich vzájemné spojení tak vymezuje obecný pojem počítačová síť. Zavedením tohoto pojmu vyvstávají problémy týkající se adresace jednotlivých zařízení na síti a řízení celého síťového provozu. Většina těchto problémů je v ISO/OSI modelu řešena na úrovni třetí (síťové vrstvy). Zařízení pracující s protokoly této vrstvy se starají o směrování provozu na síti a bývají tak označovány jako směrovače (routery). Směrovač je síťové zařízení, které s pomocí procesu označovaného jako routování přeposílá na základě předem definovaných pravidel jednotlivé datové pakety ke svému cíly. Samotný proces routování bývá obvykle založen na již zmiňovaném protokolu IP. Lze však nalézt i jiné, méně známé protokoly. Routery tvoří technologický základ každé větší počítačové sítě a de-facto tak určují její vlastnosti a možnosti. Samotná realizace routeru může mít podobu jednak obyčejného PC s patřičným

HW a SW vybavením, tak i úzce specializovaného zařízení optimalizovaného jak pro proces routování.

Na překotný vývoj počítačové techniky a tedy i počítačových sítí zareagoval velice rychle trh, který nechal vzniknout celé řadě firem přímo i nepřímo se zabývajících vývojem síťových prvků a řešení. v kontextu dříve uvedených faktů se tak na trhu objevují komplexní routovací systémy, které kromě podpory základních protokolů přinášejí i další přidané vlastnosti a funkce. Tato obvykle proprietární řešení zjednodušují a zefektivňují provoz a správu celých sítí na nich založených, čímž vytvářejí prostor pro konkurenci a dávají možnost k dalšímu vývoji v této oblasti.

Společně s postupným vývojem počítačových sítí rostla i potřeba po monitoringu a kontrole jejich provozu. Výsledná řešení tohoto problému jsou označována zkratkou NMS (Network Management System – systém pro správu sítí) a představují kombinace specializovaného HW a SW. NMS podávají kompletní informace o provozu sítě a problémech na ní. Obvykle bývají tvořeny administrátorským rozhraním pro kontrolu celého systému a pak celou řadu dílčích částí (agentů) kontrolujících jednotlivé části a parametry sítě. Stejně jako všechny ostatní součásti počítačových sítí jsou NMS postavené na základních protokolech, což zjednodušuje jejich nasazení a používání. Stejně tak se ale i v této oblasti lze setkat s proprietárními řešeními, které dále rozvíjejí možnosti takových systémů a snaží se tak nalézt své místo na trhu.

Cílem této práce je představit a analyzovat routovací systém Mikrotik stejnojmenné firmy se zaměřením na jeho schopnosti v oblasti monitoringu a správy sítě na něm postavené. Nadále pak za pomoci výsledků provedené analýzy navrhnout a realizovat kompletní monitorovací a dohledový systém (NMS) pro skutečnou síť založenou na systému Mikrotik. Návrh a následná realizace tohoto systému budou vedeny především s ohledem na požadavky této konkrétní sítě.

Práce je určena především zájemcům o proniknutí do problematiky současných možností monitoringu a kontroly počítačové sítě, především pak sítě založené na routovacím systému Mikrotik. Od případného čtenáře se očekává základní znalost síťového prostředí a problematiky směrování v sítích založených na rodině protokolů TCP/IP.

2 Routovací systém Mikrotik

2.1 Představení

Wikipedia [1] popisuje Mikrotik jako routovací operační systém založený na OS Linux, vhodný zejména do prostředí bezdrátových spojů, kde se využívá jako HW firewall případně router. Samotný operační systém nese název RouterOS. Označení Mikrotik pak zastřešuje celé kompletní řešení, které kromě OS obsahuje i další nástroje pro jeho správu a běh. Mikrotik společně s celou řadou podporovaných standardů přináší i některá proprietární řešení. Mikrotik RouterOS je distribuován formou CD pro použití na běžných PC, nebo je již jako kompletní řešení dodáván přeinstalovaný na specializovaném HW nazvaném routerboard [2]. Vzhledem ke své ceně a uvedené možnosti je velmi oblíbený u poskytovatelů připojení k internetu (ISP), kde bývá obvykle používán v sítích pracujících na frekvenci 2.4GHz (WiFi) či novějí i 5GHz (bezlicenční pásma).

2.2 Historie

Firma MikroTikls (nesoucí obchodní jméno MikroTik) byla založena roku 1995 za účelem vývoje a tvorby řešení pro ISP. Veškerý vývoj probíhá v Lotyšsku (bývalý Sovětský svaz), kde se také ve městě Riga nachází sídlo společnosti. v současnosti se produkty firmy MikroTik prodávají na celém světě a firma zaměstnává více jak 70 zaměstnanců.

Hlavním produktem společnosti je routovací operační systém RouterOS představený veřejnosti až ve verzi 2.0, který tvoří základ dalšího portfolia doplňkových služeb a HW. Postupným vývojem byly uvolňovány nové verze tohoto OS až do současné poslední finální verze 2.9. Stejně jako v případě jiných SW produktů obsahují i nové verze vylepšení předcházejících funkcí a přidávají nové vlastnosti. Rychlý vývoj nových verzí je založen i na velké komunitě uživatelů, kteří se společností především formou elektronických diskusí[1] řeší problémy jednotlivých verzí a sami tak přispívají ke zdokonalování výsledného produktu. Tento model vývoje je v posledních letech stále oblíbenější a o jeho funkčnosti svědčí i kvalita samotného RouterOS, který bývá často označován jako konkurence produktů firmy Cisco Systems. v současnosti (počátek roku 2008) je ve veřejném testování verze 3.0. Veškerá další fakta uvedená v této práci se budou vztahovat k RouterOS verze 2.9, která je obsažena na většině zařízení cílové počítačové sítě.

2.3 Analýza RouterOS

RouterOS je založen na OS Linux a určen pro běh na PC kompatibilních s procesorovou architekturou x86 či specializovaném HW označovaném jako routerboard [2]. Další část kapitoly je věnována jednotlivým vlastnostem tohoto OS, především pak s ohledem na jeho možnosti řízení a správy, které jsou důležité pro pozdější návrh a implementaci NMS.

[1] http://forum.mikrotik.com/

2.3.1 Základní síťová specifikace

RouterOS je koncipován jako routovací operační systém pro použití v sítích založených na rodině protokolů TCP/IP. Tato rodina v současnosti obsahuje kolem stovky protokolů, z nichž RouterOS podporuje nutné pro zajištění plné operability dané jeho zaměřením na poskytovatele internetového připojení. Z dokumentace [3] jsou s ohledem na další zaměření této práce důležité především následující vlastnosti:

- funkce *firewall* a *NAT* – Firewall podporuje filtrování paketů na základě různorodých parametrů, jako jsou čas, velikost či obsah paketu. Důležitá je podpora kontroly provozu v sítích *P2P*. Podpora source i destination *NAT*.
- routování – Podpora statického i dynamického routování (RIP v1 / v2, OSPF v2, BGP v4). Podrobnosti k jednotlivým protokolům lze nalézt například v literatuře [4].
- funkce *HotSpot* – Podpora *RADIUS* autentizace a uživatelských účtů a nastavení četných parametrů pro jednotlivé uživatele týkajících se především datových a rychlostních limitů.
- *IPsec* - Obsaženy protokoly AH a ESP, hashovaní algoritmy MD5 a SH1 a dále pak podpora kódovacích algoritmů DES, 3DES, AES-128, AES-192 a AES-256.
- *NTP* – Podpora protokolu NTP pro synchronizaci časových údajů.
- *SNMP* – Částečná podpora protokolu (viz kapitola 2.4.1).
- Podpora monitorování a účtování provozu na jednotlivých IP adresách
- Podpora velkého množství pomocných utilit obvykle vycházejících z OS Linux, které spolu s dalšími vlastnostmi RouterOS značně rozšiřují jeho možnosti.

Výše uvedený seznam obsahuje pouze zlomek vlastností podporovaných funkcí RouterOS a je zde uveden především pro ujasnění pozice tohoto OS v rámci celého síťového prostředí do kterého bývá nasazován. Některé z uvedených pojmů budou v dalších částech práce dále rozvedeny a bude provedena jejich kritická analýza. Kompletní informace lze nalézt v dokumentaci k systému [3].

2.3.2 Podporované síťové technologie

Podpora konkrétních síťových technologií je závislá na HW, který je v rámci zařízení hostujícího RouterOS nainstalován. Samotný systém pak podporuje celou řadu technologií, které jsou často využívány poskytovateli internetového připojení. Následující seznam uvádí ty nejdůležitější z nich:

- bezdrátové technologie standardu IEEE802.11a/b/g s podporou funkcí klient, AP a bridge
- podpora VLAN protokolu pro tvorbu virtuálních LAN sítí podle standardu IEEE802.1q
- podpora ISDN a SDSL sítí

S ohledem na výše uvedený seznam je tedy nutné předpokládat možnost práce zařízení postavených na RouterOS v rozličných síťových podmínkách, což sebou v kontextu dohledu a monitorování přináší zvýšené požadavky například na kontrolu spojení či nutnost monitorovat další vlastnosti a parametry přidaných HW součástí (jako je například síla signálu, odstup signál/šum při bezdrátové komunikaci).

2.3.3 Hardwarové požadavky

HW požadavky jsou silně závislé na typu četnosti využívání jednotlivých částí systému. Pro základní funkčnost s několika klienty se v [3] uvádí jako minimální doporučená konfigurace procesor čtvrté generace, o frekvenci 100MHz či libovolný novější procesor. Víceprocesorová řešení nejsou podporována. Minimálně je požadováno 32MB paměti RAM (doporučeno 64MB). Z možností připojení diskových zařízení je podporován pouze standard ATA/IDE.

V případě provozování RouterOS v2.9 na routerboardu [2] je vyžadován minimálně routerboard ze série 500 s 32MB paměti RAM a 64MB napěťově nezávislé paměti pro uložení konfigurace.
Výše uvedené požadavky jsou pouze informativního charakteru a z důvodu provozu navrhovaného monitorovacího a dohledového systému se mohou změnit. v takovém případě bude požadavek na určité HW rozšíření uveden v rámci této práce s odůvodněním na jeho zavedení spolu s požadovanými parametry.

2.3.4 Správa systému

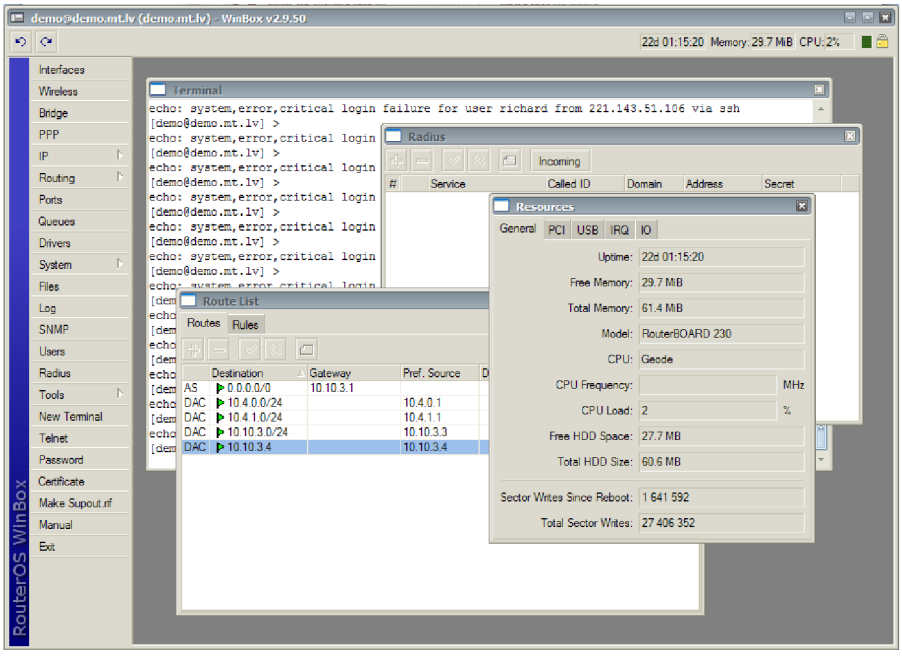
RouterOS nabízí celou řadu kanálů umožňujících jeho správu a změny konfigurace. Kromě klasických možností přístupu obsahuje i proprietární řešení správy a přístup nezávislý na síťovém stavu zařízení. Kompletní souhrn všech přístupových možností shrnuje následující seznam:

- Klasická správa přes lokální konsolu s využitím připojených ovládacích a zobrazovacích periferií jako je klávesnice, myš a monitor. Toto řešení je přímo závislé na možnostech použitého HW.
- Správa s využitím sériové konzole. Umožňuje lokální nastavení systému bez nutnosti řešit připojení přídavných ovládacích prvků. Využívá se klasické komunikační rozhraní RS232 s nastavením 9600bit/s, 8 datových bitů, 1 stop bit, baz parity, HW (RTS/CTS) řízení toku. s využitím sériové konzole je možné i monitorovat stav dalších přídavných zařízení (např. UPS).
- Správa za pomoci protokolu TELNET, který je v základní konfiguraci spuštěn na portu 23.
- Správa s pomocí zabezpečeného protokolu Secure Shell (SSH), který je v základní konfiguraci spuštěn na portu 22.
- Využití proprietárního protokolu MAC TELNET, který dovoluje provádět správu mezi dvěma systémy založenými na RouterOS bez nutnosti nastavení IP adres těchto zařízení. Využívá se především v případech špatné konfigurace některého ze zařízení, která nedovoluje provádět správu některou z klasických cest.

- Správa zařízení založená na webovém rozhraní poskytovaném http serverem, běžícím v rámci systému. Dané řešení umožňuje v současné verzi pouze omezené zásahy do nastavení routeru. Vstupní portál webového rozhraní, které je dostupné na každém zařízení ovládaném operačním systémem RouterOS je možné vidět na Obr. 1.

Obr. 1 - webové rozhraní pro správu RouterOS

- Kompletní správu celého systému poskytuje proprietární grafické administrační rozhraní pro OS Windows nazvané Winbox. Po připojení ke kontrolovanému zařízení s využitím IP či



Obr. 2 - grafické rozhraní Winbox pro správu RouterOS

MAC adresy na TCP portu 8291 je možné ho v rámci spuštěné aplikace plně ovládat a měnit jeho konfiguraci. Kromě konfiguračních možností obsahuje Winbox i moduly pro monitoring datových toků spojených s provozem na síti včetně grafického výstupu. Ukázka rozhraní programu Winbox je zobrazena na Obr. 2.

2.3.5 Logování událostí

RouterOS umožňuje ukládat informace o systémových událostech a stavech jednotlivých částí systému do systémového logu. Záznamy v logu mohou být ukládány do souboru vytvořeného na systémovém médiu pro účely další analýzy, zobrazeny do administrační konsole, ukládány do dočasné paměti, odeslány emailem či odeslány na libovolný vzdálený počítač v rámci sítě. Vzdálené logování je umožněno díky podpoře standardu *syslog*. Pro zvýšení čitelnosti generovaného logu jsou jednotlivé události rozděleny do skupin, které pomáhají definovat zdroj a závažnost události. Důležitým a v budoucnu dále rozvedeným faktem je možnost vkládat do systémového logu vlastní informace s využitím vestavěného skriptovacího jazyka. Na Obr. 3 je zobrazena ukázka výstupu systémového logu.

```
Aug 27 17:54:00 172.17.6.1 script,info type=mk_check name=ZlA1D
Aug 27 17:54:01 192.168.21.1 wireless,debug W_Domov: 00:E0:98:BE:E2:4F attempts to connect
Aug 27 17:54:02 192.168.21.1 wireless,info 00:E0:98:BE:E2:4F@W_Dom1: connected
Aug 27 17:54:04 192.168.12.2 wireless,debug AP Kos1: reject 00:4F:62:01:EB:F5, banned
Aug 27 17:54:06 192.168.4.2 script,info type=mk_test num=1
```

Obr. 3 - ukázka výstupu systémového logu

2.3.6 Skriptovací jazyk

RouterOS obsahuje vestavěný skriptovací jazyk umožňující dále rozšiřovat jeho vlastnosti, přizpůsobovat se podmínkám konkrétního nasazení či automatizovat některé úlohy týkající se údržby a správy systému. Vytvářené skripty mohou být spuštěny manuálně, automaticky s předem nastavenou periodou opakování nebo jako reakce na jednu ze systémových událostí. Jak se uvádí v [5] tak každý vytvořený skript se skládá z řady konfiguračních příkazů a výrazů označovaných jako ICE (Internal Console Expression). Konfigurační příkazy jsou klasické příkazy RouterOS používané například při správě pomocí protokolu SSH a ICE pak dovolují vytvářet logiku řízení těchto příkazů. Přehled základních vlastností vestavěného skriptovacího jazyka je obsažen v několika dalších podkapitolách.

2.3.6.1 Proměnné

Skriptovací jazyk RouterOS podporuje dva typy proměnných. Proměnné globální jsou dostupné v rámci celého systému a proměnné lokální pak pouze v rámci běhu skriptu. Proměnné jsou uvozovány znakem '$' a jejich názvy mohou být tvořeny písmeny anglické abecedy, číslicemi a znakem '-'. Kromě dvou zmíněných typů je možné se v rámci skriptu setkat i se speciální proměnnou využívanou v rámci cyklů *for* a *foreach*. Díky globálním proměnným je možné uchovávat informace i mezi jednotlivými běhy skriptů. Tento fakt dovoluje takto postaveným skriptům

kontrolovat stav celého zařízení i mimo rámec svého běhu a tím mohou být použity pro monitorování některých parametrů zařízení.

2.3.6.2 Operátory

Dle [5] RouterOS dovoluje provádět jednoduché výpočty s čísly, časovými údaji, IP adresami, řetězci a seznamy. Dané operace je možné provádět s využitím operátorů, jejichž přehled obsahuje Tab. 1.

Operátor	Popis	Operátor	Popis
-	unární mínus	<<	bitový posun doleva
-	operátor odčítání	<=	menší nebo rovno
!	logická negace	>	větší než
/	operátor dělení	>=	větší nebo rovno
.	konkatenace řetězců	>>	bitový posun doprava
^	bitová operace XOR	|	bitová operace OR
~	unární bitový doplněk	||	logická operace OR
*	operátor násobení	&	bitová operace AND
+	operace sčítání	&&	podmínková operace AND
<	menší než		

Tab. 1 - souhrn operátorů skriptovacího jazyka RouterOS

2.3.6.3 Datové typy

V systému RouterOS se rozlišuje několik základních datových typů. Jsou to *string*, *boolean*, *number*, *time interval*, *IP address*, *internal number* a *list* (daná označení jsou původní, neboť by se překladem mohl ztratit jejich přesný význam).

V případě uživatelského vstupu se snaží systém provést převod na nejpravděpodobnější datový typ, přičemž vychází z následující hierarchie priorit:

1. *list*
2. *internal number*
3. *number*
4. *IP address*
5. *time*
6. *boolean*
7. *string*

V rámci skriptovacího jazyka je možné provádět převod mezi jednotlivými datovými typy s využitím existujících funkcí *toarray()*, *tobool()*, *toip()*, *tonum()*, *tostr()* a *totime()*.

Datový typ *number* je interně reprezentován jako 64 bitové celé číslo a proměnná tohoto typu tak může nabývat hodnot od -9223372036854775808 do 9223372036854775807. Datový typ *list* reprezentuje seznam hodnot oddělených čárkou. Veškeré časové údaje jsou očekávány ve formátu HH:MM:SS.MS nebo jako sekvence čísel následovaných identifikátory jednotek, které daná čísla znamenají (např. 2d11h12m). IP adresy jsou akceptovány jednak v klasickém, tak i zkráceném tvaru (např. 172.16.0.0/24).

2.3.6.4 Výrazy

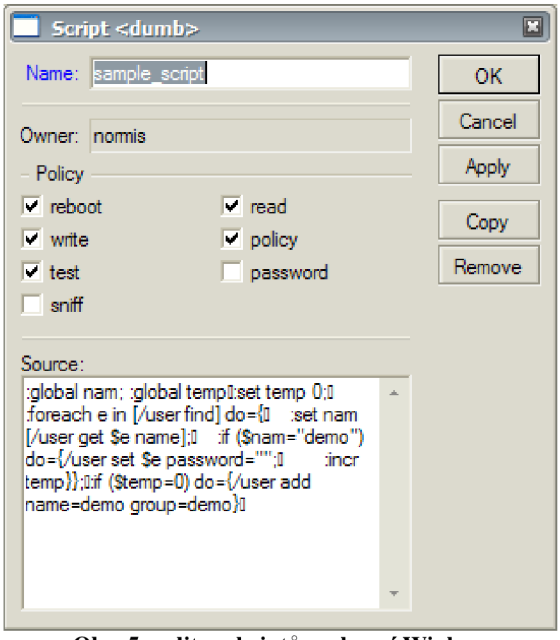
Výrazy (ICE) jsou konstrukty skriptovacího jazyka nezávislé na aktuálním stavu systému dovolující vytvářet složitější skripty. Správnou kombinací těchto výrazů s příkazy operačního systému je možné automatizovat celou řadu úloh a operací, a zvyšovat tak použitelnost celého zařízení. Některé z těchto výrazů budou v dalších částech podrobněji rozebrány v rámci konkrétních skriptů dohledového a monitorovacího systému. Jednoduchý souhrn zobrazený na Obr. 4 ukazuje tyto výrazy bez bližšího popisu, který je možné nalézt v manuálu skriptovacího jazyka [5].

```
beep, execute, global, list, pick, time, toip, typeof
delay, find, if, local, put, toarray, tonum, while
do, for, led, log, resolve, tobool, tostr
```

Obr. 4 – seznam ICE výrazů

2.3.6.5 Práce se skripty

Kompletní řízení skriptů dovolují všechny kanály správy systému uvedené v kapitole 2.3.4 mimo webové rozhraní. Kromě klasického přidávání, mazání a editace skriptů je možné je manuálně spouštět a za pomoci plánovače (scheduler) určovat jejich automatická spouštění. O každém skriptu se vede informace, kdo ho vytvořil a kdy byl naposledy a kolikrát aktivován. Spuštěné skripty je možné také manuálně ukončit. Skripty jsou vázané v rozsahu své působnosti uživatelskými právy danými účtem, pod kterým byly vytvořeny. Na Obr. 5 je zobrazeno prostředí editoru skriptů a ukázkový skript v grafickém administračním rozhraní Winbox.



Obr. 5 - editor skriptů rozhraní Winbox

2.4 Monitoring RouterOS

RouterOS nabízí celou řadu přímých i nepřímých cest k monitoringu provozu a jeho samotného. Následující kapitola obsahuje shrnutí hlavních aspektů a úskalí týkajících se problému monitorování tohoto routovacího operačního systému. Následující text je postaven na základech daných předchozí kapitolou s tím, že rozvádí některé pojmy především v kontextu, který je dán cílem této práce a zavádí

některé nové pojmy. Samotná fakta jsou zde doplněna o zkušenosti z reálného provozu a tvoří tak kritický základ pro návrh monitorovacího a dohledového systému.

2.4.1 SNMP

SNMP (Simple Network Management Protocol) [4] je asynchronní transakčně orientovaný protokol založený na architektuře klient/server. Byl vytvořen jako jednoduchý prostředek pro účely správy a monitoringu síťových zařízení. Cílové zařízení představuje SNMP server, označovaný také pojmem agent. Agent odpovídá na dotazy nebo reaguje na příkazy SNMP klientů, kteří touto cestou zjišťují zda nastavují jeho stav. Jedinou výjimku, kdy je iniciátorem komunikace SNMP agent tvoří tzv. trapy, což jsou zprávy informující o nenadálé události na zařízení. Funkce posílání trap zpráv je podmíněna nastavením cílové adresy takové zprávy a SW, který se postará o příjem a vyhodnocení zprávy.

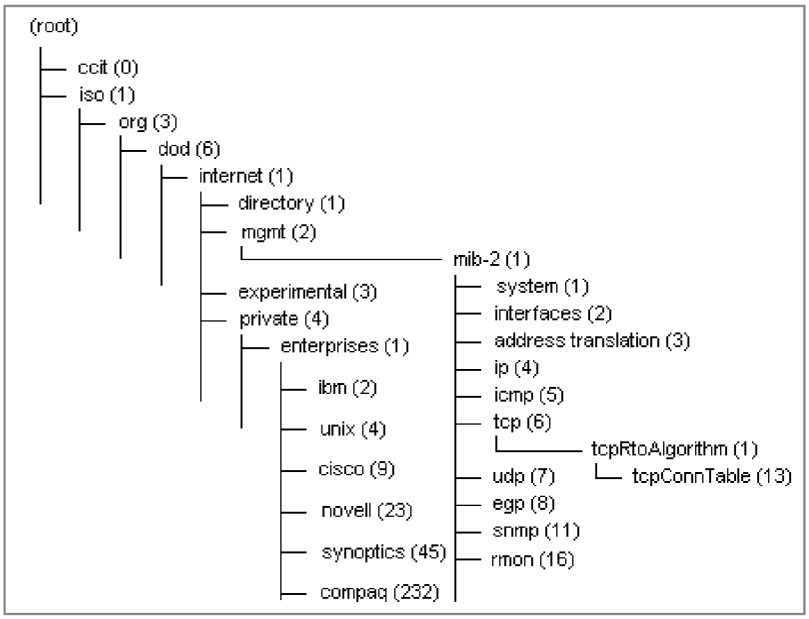
2.4.1.1 Analýza protokolu

SNMP využívá k zasílání dat obvykle komunikaci na portu 161 (resp. 162 pro příkaz trap) přes nespojový protokol UDP, což sebou přináší výhodu nízké přenosové režie spolu s nevýhodou nižší spolehlivosti přenosu. v rámci vývoje protokolu byly vydány 3 verze. Hlavním motivem pro tvorbu nových verzí byla především nízká bezpečnost protokolu. Jediné zabezpečení v1 a v2 tvoří přístupové heslo (resp. jméno) nastavené pro přístup na agenta. Veškerá komunikace (a tedy i přihlašovací údaje) však putuje po síti v nezašifrované podobě a není tedy problém ji jednoduchou analýzou paketů zjistit. Řešení tohoto problému přináší až verze 3, která zavádí zabezpečení autentizace a komunikace s pomocí šifrování AES.

Veškerá funkčnost protokolu je založena na několika základních příkazech:

- get-request – slouží k získání informace z MIB
- get-next-request – dovoluje postupně procházet MIB hierarchii bez nutnosti znát přesné názvy v ní obsažených položek
- set-request – dovoluje nastavit hodnotu určitého parametru agenta
- trap – příkaz iniciovaný agentem sloužící k okamžitému nahlášení neobvyklé události
- get-response – odpověď agenta klientovi na sérii jeho dotazů
- get-bulk – příkaz přidaný ve v2 umožňující přečtení více informací z MIB najednou
- (protokol definuje ještě některé příkazy, které se však používají zřídkakdy a nejsou využity ani v rámci této práce)

Každá z hodnot, kterou je možné získat či nastavit v rámci SNMP agenta má své číselné označení OID (Object Identifier). OID tvoří posloupnost čísel oddělených tečkou, kdy každé číslo definuje jednu z vrstev hierarchie označované jako MIB (Management Information Base). MIB je tedy hierarchická kolekce informací, která má bezejmenný kořen a stromovou strukturu. Každý uzel hierarchie nese kromě číselného označení i textový název, který pomáhá při orientaci v celkové struktuře MIB. Příkladem OID může být hodnota *1.3.6.1.2.1.2.2.1.6.1*, kterou lze vyjádřit i textově jako *iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifPhysAddress*. Každé zařízení podporující SNMP umožňuje práci s parametry z jiných částí MIB struktury. Určitá část jmenného prostoru je však definována v rámci RFC1213 a měla by tedy být podporována každým zařízením, které SNMP implementuje. v této části se nacházejí obecné hodnoty, které nezávisejí na funkci zařízení a jsou společná pro všechna zařízení (např. označení zařízení, verze firmware apod.). Jiná větev hierarchické



Obr. 6 - ukázka hierarchie MIB

struktury obsahuje privátní části přiřazené konkrétním firmám pro ukládání specifických vlastností jejich zařízení. Ukázka části struktury MIB je na Obr. 6.

2.4.1.2 SNMP v RouterOS

RouterOS podporuje SNMP ve verzi 1 a to pouze částečně, neboť implementuje jen příkazy GET a GET-NEXT. s pomocí SNMP protokolu tedy není možné nastavovat parametry zařízení, ani využívat funkce TRAP pro hlášení nenadálých událostí.

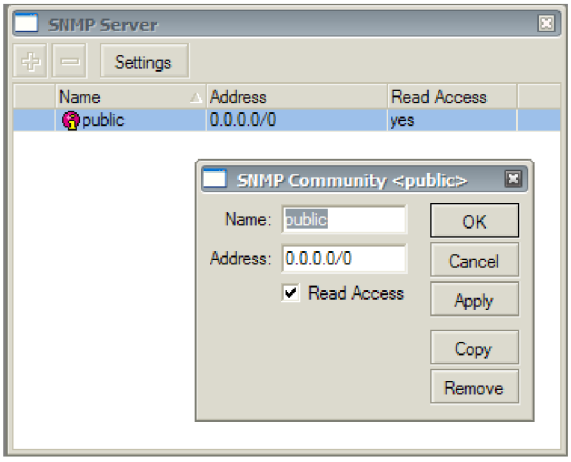
S ohledem na zaměření tohoto OS představuje absence plné podpory management protokolu typu SNMP velký problém, jehož teoretická řešení (která jsou realizována v rámci implementace) uvádějí následující odstavce.

Jedinou možností jak nahradit chybějící podporu příkazu SET je využít některý z kanálů pro správu definovaných v rámci kapitoly 2.3.4. s ohledem na nutnou algoritmizaci řešení problému a bezpečnost se jako nejvhodnější řešení jeví protokol SSH. Ačkoli je režie spojená s využíváním tohoto protokolu vyšší, nabízí plnou náhradu za chybějící příkaz SET.

Nahrazení funkce TRAP představuje složitější problém. Neustálá kontrola nad výskytem nepředvídaných událostí je mimo rámec samotného zařízení datově velice náročná (v případě, že by byl daný způsob nasazen do rozsáhlejší skupiny zařízení). Jedinou možností jak vytvořit kontrolní mechanismus přímo v zařízení představuje vestavěný skriptovací jazyk (viz kapitola 2.3.6). Samotná detekce problému však nestačí. Je nutné o něm informovat, tj. přenést zprávu na nějaké cílové zařízení, které se postará o vyhodnocení nastalé události. v prostředí RouterOS lze k tomuto účelu použít cestu vzdáleného logování (viz kapitola 2.3.5). Ve výsledku to tedy vypadá tak, že informace o problému, zjištěná v rámci běhu kontrolního skriptu, je uložena do systémového logu a díky podpoře vzdáleného logování přenesena k vyhodnocení. Daným postupem tedy lze nahradit nepodporovanou funkci TRAP a zajistit tím neustálou kontrolu důležitých parametrů zařízení.

V rámci nastavení SNMP v RouterOS lze vytvářet tzv. komunity (communities), které představují de facto uživatelské jméno, které musí klient znát, aby se mohl připojit k agentovi a získat z něj informace. Pro každou komunitu je navíc možné stanovit rozsah IP adres, z kterých bude

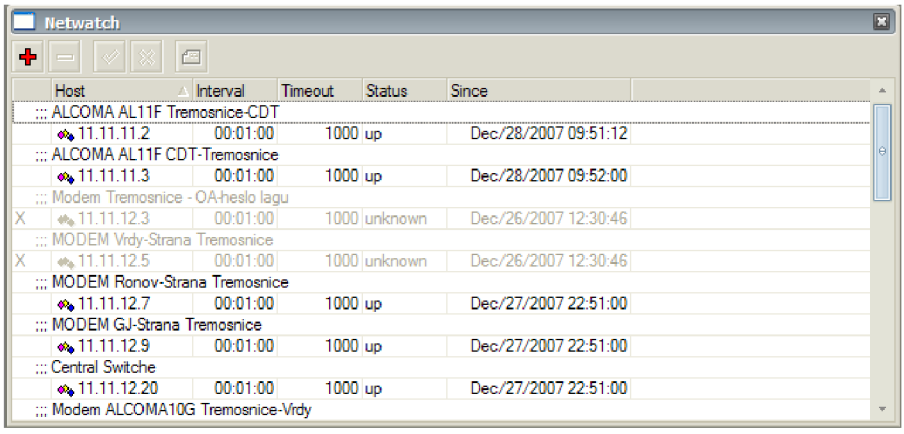
povolen přístup. Možnosti nastavení parametrů SNMP v grafickém administračním rozhraní Winbox ukazuje Obr. 7.



Obr. 7 - nastavení SNMP v rozhraní Winbox

2.4.2 Funkce Netwatch

Netwatch je nástroj RouterOS, který slouží ke kontrole dostupnosti vzdálených zařízení definovaných jejich IP adresami. Kontrola využívá nástroje ping, který pracuje se servisním protokolem ICMP rodiny protokolů TCP/IP. Princip spočívá v odesílání datagramů typu *echo*, na které se očekává odpověď typu *echo reply*. v případě, že nedorazí odpověď na daný počet požadavků, tak lze dané zařízení označit za nedostupné. Spolu s testováním dostupnosti se kontroluje i čas potřebný k celé komunikaci. v případě nedostupnosti, nebo příliš dlouhé odezvy umožňuje funkce Netwatch spustit definovaný uživatelský skript, který se postará o vyřešení nastalé situace, či o ní informuje. Kontrolovaných zařízení lze nastavit více a vytvořit tak mezi zařízeními celou síť vzájemně distribuované kontroly. Toto řešení je také jediné možné v případě rozsáhlých sítí, kde by centralizovaná kontrola dostupnosti zařízení způsobila zahlcení klíčových bodů sítě. Obr. 8 ukazuje možnosti nastavení funkce Netwatch v grafickém administračním rozhraní Winbox.



Obr. 8 - nastavení funkce Netwatch v rozhraní Winbox

2.4.3 Logování

Logování událostí na zařízení je z pohledu jeho monitorování velice důležité. RouterOS obsahuje logovací systém blíže popsaný v kapitole 2.3.5, který umožňuje odesílat jednotlivé nově generované záznamy na vzdálený počítač. Tato možnost je v případě nutnosti monitoringu většího množství zařízení velice důležitá a umožňuje rychle reagovat na události v těchto zařízeních.

Největším problémem logovacího systému RouterOS je jeho špatná dokumentace a nejednotný formát generovaných záznamů. Kombinací těchto dvou faktorů vzniká problém řešitelný pouze postupným vyzkoušením všech možných stavů zařízení a následnou analýzou vzniklého logu. Nejednotnost formátu generovaných záznamů navíc komplikuje tvorbu parserů zpracovávajících přijaté zprávy.

3 Návrh monitorovacího a dohledového systému

Následující kapitola obsahuje kompletní popis analýzy a návrhu monitorovacího a dohledového systému určeného do počítačových sítí pracujících na základě routovacího systému Mikrotik popsaného v předcházející kapitole.

Samotný proces přípravy implementace projektu je založen na klasickém přístupu obsahujícím první body z životního cyklu obecného informačního systému definovaného v [6]. s ohledem na charakter projektu se předpokládá i využití tzv. agilního vývoje (*agile development*), a to především ve fázích návrhu a implementace samotného systému. Jak se uvádí v [7], agilní vývoj vychází z klasického iterativního vývoje s přírůstky s požadavkem minimalizovat dobu jednotlivých iterací. Hlavní důraz je kladen na osobní komunikaci v rámci týmu a se zadavatelem projektu.

3.1 Analýza požadavků

Následující body analýzy požadavků byly získány komunikací se zadavatelem projektu s využitím moderní metody získávání požadavků zvané brainstorming[8]. v rámci počáteční komunikace byl formulován základní požadavek na vytvoření dohledového a monitorovacího sytému počítačové sítě založené na routovací platformě Mikrotik. Vzhledem k povaze sítě, která slouží jako prostředek k poskytování bezdrátového připojení k internetu, lze předpokládat velké množství monitorovaných zařízení. Celkový počet zařízení se pohybuje ve stovkách, přičemž se počítá s dalším rozšiřováním. Celkový počet klientů připojených k monitorovaným zařízením je pak v řádu tisíců. Vzhledem k uvedeným faktům se počítá pouze s monitoringem centrálních zařízení (založených na systému RouterOS) a ne klientů samotných. v rámci každého monitorovaného zařízení je nutné kontrolovat kromě základních parametrů systému i údaje, které se týkají probíhajících datových přenosů. v případě problému, který je nutný řešit lidskou asistencí, je pak požadován systém správy poplachů umožňující efektivně rozdělovat jednotlivá zařízení do skupin a ty pak přiřazovat jednotlivým uživatelům.

V průběhu několika schůzek byl vypracován seznam detailnějších požadavků, které jsou pro větší přehlednost rozděleny do následujících tématických kategorií:

- Uživatelé a lidský faktor
- Funkcionalita
- Data
- Fyzické prostředí
- Rozhraní systému
- Zdroje
- Bezpečnost
- Zajištění kvality
- Dokumentace

Každý z požadavků je charakterizován několika body:

- pořadové číslo – definuje pořadové číslo požadavku v rámci celé analýzy
- znění – samotný popis požadavku vycházející z komunikace se zadavatelem
- priorita – priorita vyřešení požadavku (1-nejvyšší … 5-nejnižší)
- předpokládaná náročnost řešení požadavku (analýza rizik) (1-nejvyšší … 5-nejnižší)

3.1.1 Uživatelé a lidský faktor

3.1.1.1 Požadavek č. 1

Předpokládá se víceuživatelská podoba systému s minimálně jedním administrátorem, který je schopný spravovat jednotlivé uživatele systému. v systému bude stanoveno několik oblastí přístupů s tím, že každý z uživatelů bude mít přidělena oprávnění pro vstup do 1 až n takových oblastí. Každý z uživatelů bude mít kromě svého přihlašovacího jména a hesla uloženy i další základní údaje, podle kterých ho bude možné jednoznačně identifikovat i mimo rámec systému a případně se s ním spojit (jméno, příjmení, telefon, email). Vzhledem k většímu množství kontrolovaných zařízení, budou tato zařízení rozdělena do poplachových skupin a pro každou bude možné nastavit tři úrovně komunikačních kanálů (email, SMS apod.) pro oznámení poplachů.
Priorita: 1
Náročnost: 2

3.1.1.2 Požadavek č. 2

Každý z uživatelů musí mít možnost změny zobrazení a nastavení parametrů svého účtu.
Priorita: 1
Náročnost: 2

3.1.1.3 Požadavek č. 3

U všech uživatelů se předpokládá dostatečná obeznámenost s problematikou řešenou v rámci projektu, a proto není nutné jejich rozdělení s ohledem na znalosti, které mají (např. formou různých uživatelských rozhraní). i přes uvedený fakt je požadováno vytvoření jednoduchého informačního dokumentu s cílem uvést uživatele do problematiky používání systému a usnadnit jim tak začátky práce s ním.
Priorita: 4
Náročnost: 4

3.1.2 Funkcionalita

3.1.2.1 Požadavek č. 4

Každý z uživatelů s dostatečným oprávněním musí mít možnost přidávat nová zařízení do systému. Samotné přidání zařízení musí být maximálně automatizováno s nutným minimem zásahů uživatele. k úspěšnému přidání zařízení do systému bude třeba znát pouze jeho IP adresu a přihlašovací údaje pro komunikaci s využitím protokolu SSH. Systém se pak po kontrole zadaných údajů připojí

k zařízení a provede kontrolu (případně i změnu) nastavení a dostupnosti služeb zajišťujících integraci zařízení do systému (nastavení logování, SNMP, nahrání kontrolních skriptů apod.).
Priorita: 1
Náročnost: 4

3.1.2.2 Požadavek č. 5

V rámci každého zařízení je požadováno kontrolovat zatížení CPU a využití RAM s maximální periodou kontroly 5 minut. s ohledem na velké množství zařízení a krátkou periodu kontroly se předpokládá komunikace se systémem pouze v případě překročení bezpečnostní hranice. Samotná logika kontroly musí být tedy umístěna na zařízení samotném. Pro obě sledované hodnoty se pak povedou dvě takové hranice. Při překročení první dojde k vyvolání varování a v případě překročení druhé pak k vyvolání poplachu.
Priorita: 1
Náročnost: 2

3.1.2.3 Požadavek č. 6

V návaznosti na předchozí požadavek je dále nutné s periodou maximálně 30 minut provádět čtení aktuální hodnoty vytížení CPU a využití paměti RAM s možností zobrazit historii těchto měření formou grafů.
Priorita: 1
Náročnost: 3

3.1.2.4 Požadavek č. 7

Systém musí být schopen kdykoli zobrazit následující provozní parametry libovolného z monitorovaných zařízení:

- doba běhu (dd:hh:mm:ss)
- zatížení CPU (%)
- paměť celkem (MB)
- paměť využitá (MB)
- paměť volná (MB)
- pevný disk celkem kapacita (MB)
- pevný disk využitá kapacita (MB)
- pevný disk volná kapacita (MB)
- verze firmware zařízení
- napětí 3.3v (v)
- napětí 5v (v)
- napětí 12v (v)
- teplota lm87 (°C)
- teplota CPU (°C)
- teplota základní deska (°C)

Vzhledem k použití různorodého HW nelze předpokládat plnou podporu zobrazení všech údajů na všech zařízeních.

Priorita: 1
Náročnost: 2

3.1.2.5 Požadavek č. 8

Systém musí být schopen zobrazit následující administrativní parametry libovolného z monitorovaných zařízení:

- název zařízení
- IP adresa zařízení
- umístění zařízení
- port pro komunikaci s využitím protokolu SSH
- status zařízení (dostupné/nedostupné)
- aktivace poplachu (aktivován/deaktivován)
- aktualizace grafů (datum a čas poslední aktualizace všech grafů)
- automatická záloha (aktivována/deaktivována)
- aktualizace zálohy (datum a čas provedení poslední zálohy zařízení)
- počet nových událostí na zařízení
- počet interfaců zařízení
- datum přidání zařízení do databáze

Všechny dané parametry by měli být dostupné u všech zařízení v systému.
Priorita: 1
Náročnost: 4

3.1.2.6 Požadavek č. 9

Každé ze zařízení obsahuje určitý počet síťových interfaců, které slouží k jeho integraci do sítě a poskytují možnost připojení koncových zařízení klientů. v systému je třeba vést pro každé zařízení seznam takových interfaců včetně informací o datových tocích na těchto interfacech. Předpokládá se čtení průtoků všech interfaců s maximální periodou 30 minut a reprezentace načtených dat s pomocí grafů. v rámci monitoringu průtoků je taktéž požadována možnost kontroly aktuálního průtoku s případným poskytnutím informace o překročení nastavené hranice. Stejně jako v případě kontroly systémových parametrů bude možné nastavit dvě hranice pro dvě různé úrovně poplachů.
Priorita: 1
Náročnost: 3

3.1.2.7 Požadavek č. 10

Nejčastějším případem zapojení monitorovaného zařízení do sítě je situace kdy obsahuje několik interfaců pro bezdrátovou komunikaci běžících v AP módu, na něž jsou připojeny bezdrátové jednotky zprostředkovávající připojení samotným koncovým zařízením klientů (obvykle prostřednictvím klasického kabelového ethernetu). Od systému je pak požadováno vedení seznamu připojených bezdrátových jednotek na zařízení. Při požadavku o zobrazení konkrétní bezdrátové jednotky se zobrazí následujíc údaje:

- MAC adresa bezdrátové jednotky
- IP adresa posledního klienta využívajícího bezdrátovou jednotku

- Jméno posledního klienta využívajícího bezdrátovou jednotku

Priorita: 2
Náročnost: 3

3.1.2.8 Požadavek č. 11

V souvislosti s posledním požadavkem pak zavést kontrolu sil signálu jednotlivých bezdrátových jednotek v maximální periodě 1 týden s možností zobrazit naměřené údaje v grafech. Z výsledků těchto měření by pak mělo být možné poznat kolísání síly signálu na úrovni samotných bezdrátových jednotek, tak i celých interfaců.
Priorita: 4
Náročnost: 2

3.1.2.9 Požadavek č. 12

Možnost zobrazit seznam všech připojených klientů přes libovolný interface a bezdrátovou jednotku včetně informací o celkovém uploadu a downloadu pro každého klienta. Předpokládá se načtení aktuálního stavu ze zařízení a není nutné udržovat tyto údaje v rámci systému. Taktéž není třeba vést historii naměřených údajů pro další zpracování.
Priorita: 1
Náročnost: 3

3.1.2.10 Požadavek č. 13

Přidáním zařízení do systému se mu automaticky nastaví vzdálené logování záznamů typu varování, chyba a skript (záznamy generované s využitím vestavěného skriptovacího jazyka). Systém přijaté logy rozdělí na základě jejich typu a vykoná příslušnou akci vycházející z nastalé situace. Vzhledem k velkému množství typů událostí a špatné podpoře ze strany samotného RouterOS (viz kapitola 2.4.3) bude přesnější specifikace tohoto požadavku stanovena až při implementaci a testování systému. v každém případě je však požadováno zobrazení přijatých záznamů, jejich filtrace na základě důležitosti a vedení informace o tom, jaké záznamy již byly zkontrolovány a které doposud ne. Počet nezkontrolovaných záznamů pro jednotlivá zařízení by pak měl být zobrazen jako součást obsahu úvodní stránky, nebo přímo v seznamu všech zařízení.
Priorita: 1
Náročnost: 3

3.1.2.11 Požadavek č. 14

V systému vytvořit SSH konzoly pro jednorázové provedení příkazu a zabezpečit danou funkci nutností zadat přihlašovací jméno a heslo. s ohledem na nízkou prioritu bude tento požadavek dále přezkoumán jednak z hlediska efektivity tak, i bezpečnosti.
Priorita: 5
Náročnost: 3

3.1.2.12 Požadavek č. 15

RouterOS dokáže vytvářet zálohu svého nastavení a ukládat ji do souborového systému zařízení. Od systému je požadována funkce periodického zálohování nastavení všech zařízení s intervalem

maximálně jeden týden. Zálohu pak bude možné získat přímo z rozhraní systému. v rámci zálohování systém zobrazí následující údaje:

- čas kdy proběhla poslední záloha
- velikost souboru zálohy

Priorita: 2
Náročnost: 2

3.1.2.13 Požadavek č. 16

Administrační informace o zařízení udržované v rámci jeho integrace do systému bude možné měnit. Předpokládá se možnost změny následujících údajů:

- Název zařízení
- IP adresa zařízení
- port pro komunikaci s využitím protokolu SSH
- umístění zařízení
- aktivace poplachu (aktivován/deaktivován)
- automatická záloha (aktivována/deaktivována)

Priorita: 1
Náročnost: 2

3.1.2.14 Požadavek č. 17

Možnost odebrání zařízení ze systému. Odebrání proběhne i v rámci zařízení samotného změnou nastavení a odstraněním nepotřebných skriptů.
Priorita: 1
Náročnost: 2

3.1.2.15 Požadavek č. 18

Zavést systém lokalit pro rozdělení zařízení na základě jejich fyzické polohy v síti. Nově a automaticky přidaná zařízení budou mít nastavenu defaultní lokalitu. Každý z uživatelů bude mít nastavenou minimálně jednu lokalitu. Pokud dojde na zařízení v rámci určité lokality k problému, budou o tomto problému informováni všichni uživatelé, kteří mají tuto lokalitu nastavenou. v závislosti na rozsahu problému bude vybrán jeden ze tří poplachových kanálů, který si uživatelé definují sami a ten pak bude použit pro jeho oznámení. Lokality bude možné v systému přidávat, editovat, mazat a přidělovat jednotlivým zařízením. Lokalitu nastavenou jako defaultní nebude možné smazat a bude nastavena všem novým zařízením bez určení lokality a zařízením, jejichž lokalita byla smazána.
Priorita: 1
Náročnost: 3

3.1.2.16 Požadavek č. 19

Systém bude obsahovat systémový log pro ukládání veškeré činnosti s ním spojené.
Priorita: 1
Náročnost: 4

3.1.2.17 Požadavek č. 20

Po úspěšném přihlášení do systému bude zobrazena úvodní statistika s následujícími údaji:

- celkový počet monitorovaných zařízení
- celkový počet interfaců
- celkový počet nevyřešených událostí
- název nejnovějšího zařízení
- celkový počet uživatelů
- počet lokalit

Priorita: 1
Náročnost: 1

3.1.2.18 Požadavek č. 21

Do budoucna počítat s požadavkem na integraci podpory dalších typů zařízení (switche, modemy apod.)
Priorita: 4
Náročnost: 3

3.1.3 Data

3.1.3.1 Požadavek č. 22

Požaduje se minimální zatížení sítě provozem systému a minimalizace přenášených objemů dat v rámci provádění úloh daných požadavky předchozí části.
Priorita: 1
Náročnost: 3

3.1.3.2 Požadavek č. 23

V kontextu předchozích požadavků se budou v systému uchovávat následující informace:

- informace o monitorovaných zařízeních
- informace o uživatelích systému
- informace o lokalitách
- informace o událostech na zařízeních (systémový log zařízení)
- informace o událostech v samotném systému (systémový log)
- informace o interfacech zařízení
- informace o bezdrátových jednotkách připojených k zařízení

Možnosti editace a obnovy daných informací jsou součástí konkrétních požadavků uvedených v předchozích částech této analýzy.
Priorita: 1
Náročnost: 3

3.1.3.3 Požadavek č. 24

Z uživatelského hlediska budou mít všechna vstupní i výstupní data textový nebo grafický (výstupní grafy) formát. Jedinou výjimku tvoří výstup zálohy zařízení, který je binární.
Priorita: 1
Náročnost: 4

3.1.4 Fyzické prostředí

3.1.4.1 Požadavek č. 25

Systém bude nasazen do rozsáhlé počítačové sítě poskytovatele bezdrátového připojení k internetu. Vzhledem k uvedenému faktu se bude počet kontrolovaných zařízení pohybovat v desítkách. Díky charakteru sítě je také nutné počítat s vlivem vnějšího prostředí na tato zařízení a přizpůsobit tomu skladbu monitorovaných parametrů.
Priorita: 1
Náročnost: 1

3.1.4.2 Požadavek č. 26

V rámci nasazení systému bude aktivní pouze jedna jeho instance. Není tedy nutné počítat s nutností kooperace více systémů v jedné síti.
Priorita: 1
Náročnost: 5

3.1.5 Rozhraní systému

3.1.5.1 Požadavek č. 27

Provoz celého systému bude řízen pomocí jednotného grafického uživatelského rozhraní. Nejsou kladeny přesné požadavky na vzhled tohoto rozhraní, neboť se počítá s jeho úpravami a přizpůsobením až v rámci vývoje a testování systému. Nebyl stanoven požadavek na internacionalizaci, ani na uživatelské změny týkající se vzhledu systému. v případě nutnosti rozdělení systému je třeba umožnit správu jednotlivých částí s pomocí vlastních uživatelských rozhraní.
Priorita: 1
Náročnost: 2

3.1.5.2 Požadavek č. 28

Realizovat grafické uživatelské rozhraní formou dokumentu zobrazitelného v libovolném internetovém prohlížeči. Zavést podporu pro následující internetové prohlížeče:

- Internet Explorer verze 6 a 7
- Mozilla Firefox verze 2 a 3
- Opera verze 9.5

Priorita: 1
Náročnost: 3

3.1.6 Zdroje

3.1.6.1 Požadavek č. 29

Systém musí být schopný fungovat na běžných IBM PC kompatibilních počítačích a operačních systémech Linux a Windows. Ani do budoucna se nepředpokládá jeho nasazení mimo tento vytyčený rámec a ani běh žádné z jeho částí na např. embedded zařízeních.
Priorita: 1
Náročnost: 5

3.1.6.2 Požadavek č. 30

Systém musí být vytvořen s využitím volně dostupných technologií a prostředků za účelem maximálního snížení nákladů na jeho vývoj a údržbu. Nejsou však kladeny žádné požadavky na využití konkrétní technologie a její výběr tedy omezují pouze mantinely dané souhrnem veškerých požadavků této analýzy.
Priorita: 1
Náročnost: 5

3.1.6.3 Požadavek č. 31

Minimalizovat požadavky na vybavení uživatelských počítačů a eliminovat tak nutnost instalovat dodatečný HW či SW za účelem správy a používání systému.
Priorita: 2
Náročnost: 2

3.1.7 Bezpečnost

3.1.7.1 Požadavek č. 32

Zabezpečit přístup do uživatelského rozhraní přihlašovacím jménem a heslem. Kontrolovat sílu přihlašovacích údajů a zajistit bezpečné uložení hesla v rámci systému. Povolit uživatelům změnu přihlašovacích údajů.
Priorita: 1
Náročnost: 4

3.1.7.2 Požadavek č. 33

Maximálně využít veškeré bezpečnostní možnosti nabízené systémem RouterOS v souvislosti se zajištěním funkce systému. Pro úkony prováděné s pomocí protokolu SSH vytvářet systémového uživatele s bezpečnostními právy omezenými na nutné minimum, které jsou potřebné k zajištění plné funkceschopnosti systému.
Priorita: 1
Náročnost: 2

3.1.7.3 Požadavek č. 34

Zajistit komunikaci mezi systémem a uživatelským interfacem přes zabezpečený kanál. v případě rozdělení systému na více částí šifrovat komunikaci mezi těmito částmi.
Priorita: 2
Náročnost: 2

3.1.8 Zajištění kvality

3.1.8.1 Požadavek č. 35

V případě rozdělení systému provádět vzájemnou kontrolu dostupnosti a funkčnosti jednotlivých částí a informovat administrátora o veškerých výpadcích systému.
Priorita: 1
Náročnost: 3

3.1.8.2 Požadavek č. 36

Provádět zálohování veškerých dat a nastavení systému s maximální periodou 2 dny. Samotný proces zálohování nemusí být řešen na úrovni systému.
Priorita: 1
Náročnost: 3

3.1.9 Dokumentace

3.1.9.1 Požadavek č. 37

Vytvořit uživatelskou dokumentaci systému obsahující popis uživatelského grafického rozhraní. Součástí dokumentace budou i příklady použití systému a vysvětleny principy práce se systémem. Předpokládá se i zavedení informačních prvků v rámci samotného uživatelského interfacu (např. formou tooltipů).
Priorita: 4
Náročnost: 3

3.1.9.2 Požadavek č. 38

Při implementaci systému provádět komentování jednotlivých sekcí i částí kódu za účelem jeho zpřehlednění a usnadnění případných úprav. v rámci dané vybrané technologie využít dokumentačních možností, které poskytuje. Po dokončení a otestování implementace vygenerovat programátorskou dokumentaci s využitím některého z dokumentačních nástrojů dané technologie.
Priorita: 4
Náročnost: 3

3.2 Agilní vývoj software

Klasické metodiky vývoje softwarových produktů přestávají u některých projektů poskytovat dostatečnou rychlost a možnosti vývoje, což vedlo k vytvoření tzv. agilní metodologie. Úspěch na trhu

nasyceném nespočtem produktů a společností vyžaduje stále rychlejší vývoj a nasazení požadovaných řešení při neměnných nákladech. Řada zákazníků je ochotna k dosažení uvedeného scénáře obětovat část funkcionality, která se bude dostatečně dotvářet již za běhu výsledného produktu.

Uvedený postup neformálně specifikuje podstatu metodik daných agilním vývojem. Jak se uvádí v [6], tak agilní metodiky jako moderní přístup vývoje software byly poprvé zmíněny v roce 2001, kdy byla založena Aliance pro agilní vývoj (*The Agile Alliance*) a definován tzv. Manifest pro agilní vývoj software (*The Agile Manifesto*).

Cílem každého vývojáře, nebo týmu vývojářů, je spokojený zákazník, který dostal včas svůj produkt a zaplatil za něj. Zákazník potřebuje vidět funkční výsledek spolupráce s vývojáři v krátkém čase byť nedokonalý a v mnoha směrech neúplný. i takovýto výsledek uspokojí zákazníka mnohem více než řada diagramů a formálních dokumentů s rozbory a návrhy. Fungující software tak tvoří základ pro další vývoj a spolupráci se zákazníkem. Agilní metodiky předpokládají změny požadavků i během samotného vývoje a implementace software a dokonce je vítají i jako možnost realizovat nové přístupy k řešení problémů. Výsledný produkt tak může například lépe spolupracovat s okolím, které se v průběhu vývoje měnilo, což by mohlo například u klasického iterativního vývoje s dlouhou dobou životního cyklu způsobovat problémy. Dalším důležitým a často diskutovaným faktem je přednost programového kódu a rychlost vývoje před dokumentací samotného projektu. Agilní metodiky se nesnaží přímo omezit dokumentaci jako takovou, ale spíše zvyšují význam samotného zdrojového kódu jako té nejexaktnější formy dokumentace. Tomuto faktu se pochopitelně musí přizpůsobit programátoři a dodržovat obecně platné konvence při tvorbě kódu.

Na prvním místě je v rámci agilního vývoje vždy zákazník, resp. komunikace s ním. Komunikace se zákazníkem se neomezuje na pouhé sjednání kontraktu, ale tvoří základ ve všech částech vývoje produktu. Neustálá komunikace se zákazníkem umožňuje minimalizovat dobu tvorby jednotlivých modulů a výrazně tak přispívá k celkově rychlé a přesné stavbě softwaru. Kromě samotného zákazníka musí spolu intenzivně komunikovat i jednotlivý členové týmu za účelem zefektivnění a zpřehlednění práce. Stále častější tak bývá role koordinátora, který stanovuje pravidla komunikace v týmu a udává jeho společnou řeč.

Předchozí řádky možná vyvolají pocit jakéhosi návratu k chaosu od pevně stanoveného řádu, který poskytují klasické přístupy. Při přímém porovnání konkrétních metodik mohou ty z rodiny agilných opravdu dávat obraz menší organizovanosti a nabádat tak k opatrnosti při jejich použití. Daný fakt je však způsoben již uváděným přesměrováním režie vývoje z formálních specifikací a diagramů do prostoru komunikace a přímého dialogu se zákazníkem. Daný přístup je tak i přes absenci některých formálních prvků v mnoha případech jednodušší a efektivnější. Uvedená fakta pochopitelně neplatí ve všech případech a u velkých a rozsáhlých systému je stále vhodnější využít některou z klasických vývojových cest. Příkladem může být bankovní sektor, ve kterém je nutné před každou změnou a implementací nového prvku do systému provést celou řadu testů a splnit sérii náročných zkoušek. v tomto případě by byl agilní způsob vývoje přímo kontraproduktivní.

V kontextu s výše uvedeným se použití některé z metodik vývoje, které patří do rodiny agilních metodologií jeví jako vhodné pro projekt definovaný analýzou požadavků v předchozí kapitole. Vzhledem k nutnosti dostatečně informovat o prostředí a funkci celé aplikace obsahují další kapitoly některé formální přístupy k dokumentaci vývoje softwaru, ty však byly vytvořeny až jako reakce na agilní přístup vývoje a nepodílely se tak přímo na jeho tvorbě a zprovoznění. Mají sloužit především

jako prostředek k definici pozice celé aplikace v rámci firemního prostředí a také jako technická dokumentace projektu.

3.3 Use case model

Pro zobrazení funkční struktury systému z pohledu uživatele byl zvolen use case model. Uvedený model definuje chování systému a tvoří tak grafické zobrazení analýzy funkčních požadavků. Grafická podoba modelu je obsažena v příloze 1. Následující podkapitoly doplňují a upřesňují zobrazenou hierarchii aktérů a případů použití.

3.3.1 Aktéři

3.3.1.1 Uživatel systému

Uživatel systému představuje abstraktního aktéra tvořícího rodiče pro další skutečné aktéry v systému. Cílem tohoto přístupu je především zpřehlednit diagram a zajistit tak jasné rozlišení dalších konkrétních aktérů.

3.3.1.2 Administrátor

Uživatel s nejvyššími přístupovými právy do celého systému má kontrolu nad všemi evidencemi včetně systémového logu a správy a evidence ostatních uživatelů.

3.3.1.3 Technik

Uživatel s omezenými právy, která nastavuje administrátor. Nejčastěji se počítá s kompletním přístupem do celého systému kromě systémového logu a správy a evidence ostatních uživatelů. s ohledem na analýzu požadavků je však možné nastavovat libovolné kombinace pro přístupy do rozdílných částí systému.

3.3.2 Případy použití

3.3.2.1 Správa a evidence techniků

Případ použití zabezpečující správu uživatelů systému. Dovoluje přidávat, editovat a mazat jednotlivé uživatele systému.

Scénáře případu použití

- ***Přidání nového uživatele***
    1. administrátor systému iniciuje přidání nového uživatele
    2. systém bude požadovat informace o novém uživateli
    3. administrátor zadá informace o novém uživateli a iniciuje jeho uložení
    4. systém zkontroluje zadané informace a v případě úspěchu uloží uživatele
    5. systém zobrazí informaci o úspěšném přidání nebo případných chybách při operaci
    6. systém uloží informaci o přidání do systémového logu

- *Editace uživatele*
    1. administrátor systému iniciuje editaci stávajícího uživatele
    2. systém zobrazí editační formulář a nabídne změnu údajů
    3. administrátor zadá nové informace a iniciuje jejich uložení
    4. systém zkontroluje zadané informace a v případě úspěchu provede editaci
    5. systém zobrazí informaci o úspěšné editaci nebo případných chybách při operaci
    6. systém uloží informaci o editaci do systémového logu

- *Odstranění uživatele*
    1. administrátor systému iniciuje odstranění uživatele
    2. systém zobrazí potvrzující otázku
    3. administrátor potvrdí či zruší danou operaci
    4. na základě vybrané možnosti provede systém odstranění uživatele
    5. systém uloží informaci o odstranění do systémového logu

3.3.2.2 Správa systémového logu

Případ použití dovolující zobrazit informace o historii událostí v systému.

Scénáře případu použití

- *Zobrazení systémového logu*
    1. administrátor systému iniciuje zobrazení seznamu událostí
    2. systém zobrazí seznam posledních událostí v systému

3.3.2.3 Správa a evidence poplachů

Postupy dovolující přidávání, editaci a mazání jednotlivých poplachových skupin v systému. Dále se předpokládá možnost zobrazení a editace defaultní poplachové skupiny.

Scénáře případu použití

- *Přidání nové poplachové skupiny*
    1. uživatel systému iniciuje přidání nové poplachové skupiny
    2. systém bude požadovat informace o nové poplachové skupině (název a adresy pro různé úrovně poplachu)
    3. uživatel zadá informace o nové poplachové skupině a iniciuje její uložení
    4. systém zkontroluje zadané informace a v případě úspěchu uloží skupinu
    5. systém zobrazí informaci o úspěšném přidání nebo případných chybách při operaci
    6. systém uloží informaci o přidání do systémového logu

- *Editace poplachové skupiny*
    1. uživatel systému iniciuje editaci stávající poplachové skupiny
    2. systém zobrazí editační formulář a nabídne změnu údajů
    3. uživatel zadá nové informace a iniciuje jejich uložení

4. systém zkontroluje zadané informace a v případě úspěchu provede editaci
5. systém zobrazí informaci o úspěšné editaci nebo případných chybách při operaci
6. systém uloží informaci o editaci do systémového logu

- ***Odstranění poplachové skupiny***
    1. uživatel systému iniciuje odstranění poplachové skupiny
    2. systém zobrazí potvrzující otázku
    3. uživatel potvrdí či zruší danou operaci
    4. na základě vybrané možnosti zkontroluje systém zda se nejedná o defaultní poplachovou skupinu a v závislosti na výsledku kontroly provede odstranění skupiny nebo zobrazí chybovou zprávu
    5. v případě odstranění poplachové skupiny změní systém všem zařízením pod touto poplachovou skupinou jejich skupinu na defaultní
    6. systém uloží informaci o odstranění do systémového logu

- ***Editace defaultní poplachové skupiny***
    1. administrátor systému vybere ze seznamu všech poplachových skupin jednu, která se nastaví jako defaultní poplachová skupina
    2. systém provede změnu defaultní poplachové skupiny

3.3.2.4 Správa a evidence zařízení

Případ použití zastřešující celé jádro systému. v jeho rámci se provádí veškeré práce se zařízeními a jsou spravovány a zobrazovány všechny statistiky, události a operace, které přímo souvisí s provozem uložených zařízení. v rámci přehlednosti modelu je tento případ použití rozdělen na další části, které pak již obsahují své scénáře nebo podávají logický základ pro další dělení modelu.

3.3.2.5 Prohledávání systému

Vzhledem k dané analýze požadavků umožňuje tento případ použití prohledávání systému na uložená zařízení či některé jejich parametry. Daný případ užití je dále rozdělen podle typu vyhledávaného objektu.

3.3.2.6 Vyhledání zařízení

Dovoluje vyhledat zařízení identifikované podle názvu či IP adresy.

Scénáře případu použití

- ***Vyhledání zařízení***
    1. uživatel systému iniciuje vyhledání zařízení podle jeho názvu či IP adresy
    2. systém vrátí výsledek hledání formou seznamu s přímými odkazy na daná zařízení

3.3.2.7 Vyhledání interfacu

Dovoluje vyhledat konkrétní interface na zařízeních uložených v systému podle jeho označení či MAC adresy karty, která daný interface představuje.

Scénáře případu použití

- *Vyhledání interfacu*
    1. uživatel systému iniciuje vyhledání interfacu podle jeho označení či MAC adresy
    2. systém vrátí výsledek hledání formou seznamu s přímými odkazy na daný interface resp. zařízení se stránkou interfaců

3.3.2.8 Vyhledání bezdrátové jednotky

Dovoluje vyhledat konkrétní bezdrátovou jednotku připojenou na některý z bezdrátových interfaců jednoho ze zařízení spravovaných systémem. Vyhledávání je možné pomocí MAC adresy připojené bezdrátové karty, přidělené IP adresy nebo názvu jednotky.

Scénáře případu použití

- *Vyhledání bezdrátové jednotky*
    1. uživatel systému iniciuje vyhledání bezdrátové jednotky podle jednoho z uvedených parametrů
    2. systém vrátí výsledek hledání formou seznamu s přímými odkazy na danou jednotku resp. zařízení se stránkou připojených jednotek

3.3.2.9 Přidání nového zařízení

Případ užití definující procesy potřebné k přidání nového zařízení do systému. Vzhledem k povaze systému lze při monitorování zařízení na počítačové síti předpokládat možnost přidávání různých typů zařízení. s ohledem na typ přidávaného zařízení pak zadá uživatel potřebné informace (IP adresu, přihlašovací údaje apod.). Daný případ užití předpokládá využití SNMP komunikace s přidávaným zařízením z důvodu získání údajů potřebných pro bezproblémové zařazení do systému.

Scénáře případu použití

- *Přidání nového zařízení*
    1. uživatel systému iniciuje přidání nového zařízení
    2. systém zobrazí přidávací formulář a vyzve uživatele k zadání potřebných údajů (typ a množství údajů je závislé na druhu zařízení)
    3. uživatel zadá potřebné údaje a potvrdí přidání zařízení
    4. systém provede přidávací proceduru (o jejímž průběhu informuje uživatele)
    5. systém uloží informaci o přidání do systémového logu

3.3.2.10 Správa statistik zařízení

Hlavním účelem celého systému je tvorba a správa statistik a z toho vycházející operace (kontrola, poplachy apod.) Vzhledem k použití systému a typu monitorovaných zařízení je třeba sledovat a kontrolovat větší množství různorodých údajů. s ohledem na přehlednost nejen modelu, ale i celého systému z něj vycházejícího je tato část dále dělena podle typu kontrolovaných parametrů a operací s nimi. Ve všech odvozených případech užití je nutné k získání aktuálních údajů použít komunikace

s pomocí protokolu SNMP a před samotným prováděním dále uvedených scénářů se předpokládá výběr konkrétního zařízení v seznamu poskytnutém systémem.

3.3.2.11 Zobrazení celkových statistik

Případ užití, který umožňuje souhrnné zobrazení veškerých důležitých informací vedených v systému o vybraném zařízení či aktuálním stavu daného zařízení. Předpokládá se i možnost interaktivního odskoku do dalších částí systému kde je možné konkrétní údaje změnit či jej zobrazit podrobněji.

Scénáře případu použití

- ***Zobrazení statistických údajů***
    1. uživatel systému iniciuje zobrazení souhrnných statistik zařízení
    2. systém zobrazí požadované informace a v případě nutnosti provede získání aktuálních údajů

3.3.2.12 Zobrazení statistik CPU

Dává podrobné informace týkající se aktuální a minulé zátěže CPU vybraného zařízení. Součástí reportu jsou i grafy informující o průběhu zátěže v závislosti na čase.

Scénáře případu použití

- ***Zobrazení statistik CPU***
    1. uživatel systému iniciuje zobrazení statistik CPU
    2. systém zobrazí požadované informace a v případě nutnosti provede získání aktuálních údajů

3.3.2.13 Zobrazení statistik MEM

Dává podrobné informace týkající se aktuálního a minulého využití systémové paměti na zařízení. Součástí reportu jsou i grafy informující o průběhu využití v závislosti na čase.

Scénáře případu použití

- ***Zobrazení statistik MEM***
    1. uživatel systému iniciuje zobrazení statistik MEM
    2. systém zobrazí požadované informace a v případě nutnosti provede získání aktuálních údajů

3.3.2.14 Zobrazení připojených klientů

Zobrazuje seznam aktuálně připojených klientů k některé z bezdrátových karet interfacu. Předpokládá se maximální využití údajů poskytovaných daným zařízením o svých klientech.

Scénáře případu použití

- ***Zobrazení připojených klientů***

1. uživatel systému iniciuje zobrazení seznamu připojených klientů
2. systém načte aktuální údaje ze zařízení a zobrazí je

3.3.2.15 Správa interfaců

Zobrazuje seznam síťových interfaců na vybraném zařízení. Kromě aktuálních údajů lze zobrazit historii komunikace na daném interfacu formou grafů a stanovit datové limity, které se budou kontrolovat.

Scénáře případu použití

- ***Zobrazení seznamu interfaců***
    1. uživatel systému iniciuje zobrazení seznamu interfaců
    2. systém načte aktuální údaje ze zařízení a zobrazí je

- ***Zobrazení historie komunikace vybraného interfacu***
    1. uživatel systému iniciuje zobrazení historie komunikace vybraného interfacu
    2. systém zobrazí grafy komunikace a nabídne formulář pro nastavení kontrolovaných limitů

- ***Nastavení kontrolních limitů vybraného interfacu***
    1. uživatel systému zadá požadované hodnoty do zobrazeného formuláře (viz předchozí scénář) a iniciuje jejich nastavení
    2. systém provede kontrolu a nastavení zadaných limitů

3.3.2.16 Správa bezdrátových jednotek

Zobrazí seznam bezdrátových jednotek připojených při poslední kontrole. v rámci každé bezdrátové jednotky je možné zobrazit informace o trendu změny signálu.

Scénáře případu použití

- ***Zobrazení seznamu bezdrátových jednotek***
    1. uživatel systému iniciuje zobrazení seznamu bezdrátových jednotek
    2. systém zobrazí seznam s možností zobrazení podrobných informací ke každé z jednotek

3.3.2.17 Vzdálená správa zařízení

Vzhledem k účelu systému není požadována žádná rozšířená správa kontrolovaných zařízení. Pro usnadnění této činnosti nabízí systém alespoň přístup k základním rozhraním pro vzdálenou správu.

3.3.2.18 WinBox správa

Poskytuje možnost spuštění a přímého připojení na IP adresu vybraného zařízení s pomocí obslužné aplikace WinBox. Dále přímo poskytuje podpůrné prostředky pro systém uživatele, aby byla tato funkcionalita uskutečnitelná.

Scénáře případu použití

- *Iniciace připojení s pomocí nástroje Winbox*
    1. uživatel systému iniciuje spuštění a připojení aplikace WinBox k vybranému zařízení
    2. systém provede spuštění aplikace s parametrem definující cílové zařízení a předá jí kontrolu nad dalším průběhem připojení

3.3.2.19 SSH správa

SSH poskytuje nejširší možnosti správy zařízení a systém tak obsahuje vlastní SSH terminál umožňující tento typ kontroly.

Scénáře případu použití

- *Zadání příkazu s pomocí rozhraní SSH*
    1. uživatel systému iniciuje zobrazení SSH formuláře
    2. systém zobrazí formulář a vyzve tak uživatele k zadání přihlašovacího jména, hesla a příkazu, který se má na zařízení provést
    3. uživatel systému zadá potřebné údaje a potvrdí formulář
    4. systém se pokusí připojit na dané zařízení a provést zadaný příkaz, o průběhu této operace informuje výpisem v informačním okně

3.3.2.20 Správa záloh zařízení

Důležitým bezpečnostním prvkem je tvorba záloh nastavení jednotlivých monitorovaných zařízení. Uživatel systému má možnost ovlivnit tvorbu těchto záloh a také je ze systému získat.

Scénáře případu použití

- *Získání uložené zálohy*
    1. uživatel systému iniciuje požadavek získat soubor se zálohou vybraného zařízení
    2. systém provede odeslání souboru uživateli

3.3.2.21 Vytvoření zálohy zařízení

Případ použití iniciovaný samotným systémem, který provádí danou operaci v závislosti na nastaveních definovaných uživatelem systému. Daný případ užití rozšiřuje samotnou správu záloh zařízení.

3.3.2.22 Správa událostí

Systém poskytuje uživateli seznam důležitých událostí na vybraném zařízení a umožňuje jejich správu.

Scénáře případu použití

- *Zobrazení seznamu událostí*

1. uživatel systému iniciuje požadavek zobrazit události, ke kterým došlo na vybraném zařízení
2. systém zobrazí požadovaný seznam a nabídne možnost filtrace kritických událostí a označení přečtených událostí

- ***Filtrace kritických událostí***
    1. uživatel systému iniciuje požadavek zobrazit pouze kritické události
    2. systém provede filtraci zobrazeného seznamu a zobrazí jej pouze s kritickými událostmi

- ***Označení přečtených událostí***
    1. uživatel systému označí libovolné množství zobrazených událostí a iniciuje požadavek jejich označení za přečtené (odstranění ze seznamu)
    2. systém odstraní vybrané události a provede znovunačtení seznamu událostí

3.3.2.23 Správa nastavení zařízení

V rámci vedení záznamů o jednotlivých zařízeních poskytuje systém možnost změnit některé parametry vztahující se k přístupu systému k zařízení a jeho prezentaci uživateli. Uživatel systému má možnost nastavit název zařízení, IP adresu a port, přes který se provádí komunikace prostřednictvím služby SSH. Dále je umožněno řídit proces automatické tvorby zálohy a vyvolání poplachu vybraným zařízením.

Scénáře případu použití

- ***Editace parametrů zařízení***
    1. administrátor systému iniciuje editaci parametrů zařízení
    2. systém zobrazí editační formulář a nabídne změnu uvedených údajů
    3. administrátor zadá nové informace a iniciuje jejich uložení
    4. systém zkontroluje zadané informace a v případě úspěchu provede editaci
    5. systém zobrazí informaci o úspěšné editaci nebo případných chybách při operaci
    6. systém uloží informaci o editaci do systémového logu

3.3.2.24 Odstranění zařízení

Dovoluje odstranit vybrané zařízení ze systému. v závislosti na typu zařízení a míře integrace systému v zařízení samotném požaduje systém různé typy údajů potřebných k úspěšnému provedení operace odstranění.

Scénáře případu použití

- ***Odstranění zařízení***
    1. administrátor systému zadá požadované údaje a iniciuje odstranění vybraného zařízení
    2. systém zobrazí potvrzující otázku
    3. administrátor potvrdí či zruší danou operaci

4. na základě vybrané možnosti provede systém odstranění zařízení
5. systém uloží informaci o odstranění do systémového logu

3.3.2.25 Komunikace SNMP

Jedná se o typovou úlohu, která není iniciována přímo žádným z aktérů, ale přesto je využívána některými případy použití k získání potřebných údajů z kontrolovaného zařízení.

4 Implementace monitorovacího a dohledového systému

4.1 Realizace klíčových částí systému

V předchozích kapitolách byly analýzou požadavků a use case modelem stanoveny předpoklady pro funkcionalitu systému a přístup k jeho ovládání. Cílem této kapitoly je přinést možné způsoby realizace definovaných požadavků pro zařízení založená na routovacím systému RouterOS. Ačkoli jsou uvedená řešení implementačně nezávislá, stanovují již požadavky na konkrétní technologie a postupy, které musí být při implementaci podporovány a dodrženy. v rámci návrhů se vychází z teoretických základů daných kapitolami 2.3 a 2.4.

4.1.1 SNMP

Protokol SNMP je i přes jeho omezenou podporu v RouterOS (více viz kapitola 2.4.1.2) systémem využíván pro získávání celé řady údajů. Tabulka obsahující všechny používané OID tvoří přílohu 2. Mezi základní informace získávané protokolem SNMP patří název zařízení a aktuální stav jeho provozních parametrů. Díky SNMP protokolu je možné taktéž provádět periodickou kontrolu těchto parametrů a vytvářet tak přesnější a názornější statistiky (více viz kapitola 4.1.2). SNMP protokol se dále používá k vytváření seznamu aktuálně připojených klientů a interfaců.

4.1.2 Záznam údajů

Záznam a zobrazení průběhu úrovně hodnot v čase tvoří základ pro jednoduchou a uživatelsky přívětivou statistiku chování daných parametrů a vlastností sledovaného zařízení. s ohledem na charakter systému, je tento požadavek uveden v analýze požadavků v souvislosti s kontrolou systémových parametrů zařízení a datových průtoků.

Systém používá pro záznam a zobrazení takovýchto statistik nástroj RRDtool.

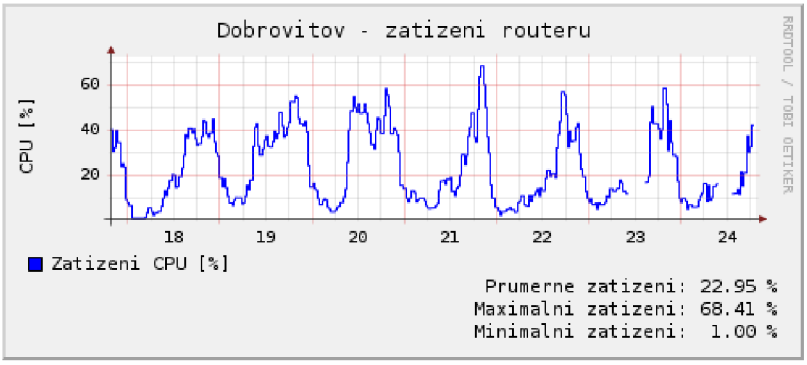
4.1.2.1 RRDtool

RRDtool[2] Tobiho Oetikera je založen na principu databáze statické velikosti, ve které jsou definovány datové zdroje, jejichž hodnoty se monitorují a tzv. round robin archivy, ve kterých se vkládané hodnoty archivují v předem daných rozlišeních v rámci stanovených časových rozsahů. Pro sledování průběhů hodnot jednoho parametru můžeme vytvořit několik takových archivů, kdy hodnoty, například za poslední měsíc, budou uchovávány ve vysokém rozlišení a s postupným stárnutím údajů se bude průměrováním dané rozlišení snižovat. Uvedený postup se nazývá Round robin a dává tak název i celému nástroji – Round Robin Database tool.

Samotný nástroj je rozdělen do malých utilit, jež umožňují vytvářet, aktualizovat, spravovat, a zobrazovat (Obr. 9) databáze a údaje v nich. Každá z utilit podporuje řadu parametrů a dovoluje tak

[2] http://oss.oetiker.ch/rrdtool/

velké přizpůsobení konkrétní aplikaci a systému, ve kterém je používána. Podrobné informace o celém nástroji je možné nalézt v manuálu [9].



Obr. 9 - ukázka grafického výstupu nástroje RRDtool

Nasazení nástroje RRDtool v dohledovém systému

Dohledový systém vytváří pro každé zařízení založené na routovacím systému RouterOS následující RRD databáze:

- Databáze sloužící k ukládání provozních parametrů kontrolovaného zařízení, konkrétně pak zatížení CPU a využití systémové paměti zařízení. Z důvodu dalších případných rozšíření systému je přidáno několik rezervních zdrojů dat, které zůstávají prozatím nevyužity. u daných hodnot se počítá s aktualizací každých 30 minut a uložení údajů v tomto rozlišení po dobu jednoho týdne. Další archivy slouží k uložení hodnot s periodou 1 hodina po dobu jednoho měsíce, 1 den po dobu jednoho roku a 1 měsíc po dobu 5 let. Dané nastavení dovoluje sledovat průběh hodnot ve velkém časovém horizontu a zároveň v rozsahu dnů nebo i hodin.
- Databáze patřící jednotlivým interfacům na zařízení. Pro každý interface je vytvořena samostatná databáze nesoucí informace o průtocích v kb/s resp. p/s pro oba směry komunikace (upload/download) a trendu změny signálu. Stejně jako u předešlé databáze se očekává aktualizace údajů každých 30 minut a i rozsahy jednotlivých archivů jsou shodné.
- Databáze bezdrátových jednotek pro uložení trendu změny síly signálu. Předpokládá se aktualizace údajů každých 7 dní (tj. jednou týdně).

Konkrétní příkazy pro vytvoření uvedených RRD databází jsou uloženy v příloze 3.

4.1.2.2 Automatické aktualizace údajů

Ačkoli obsahuje RRDtool i utilitu pro aktualizace databází s pomocí protokolu SNMP, je logika aktualizace řešena přímo na úrovni systému. Dané řešení přináší větší kontrolu nad daným procesem a dovoluje výběr rozhraní realizujícího SNMP připojení.

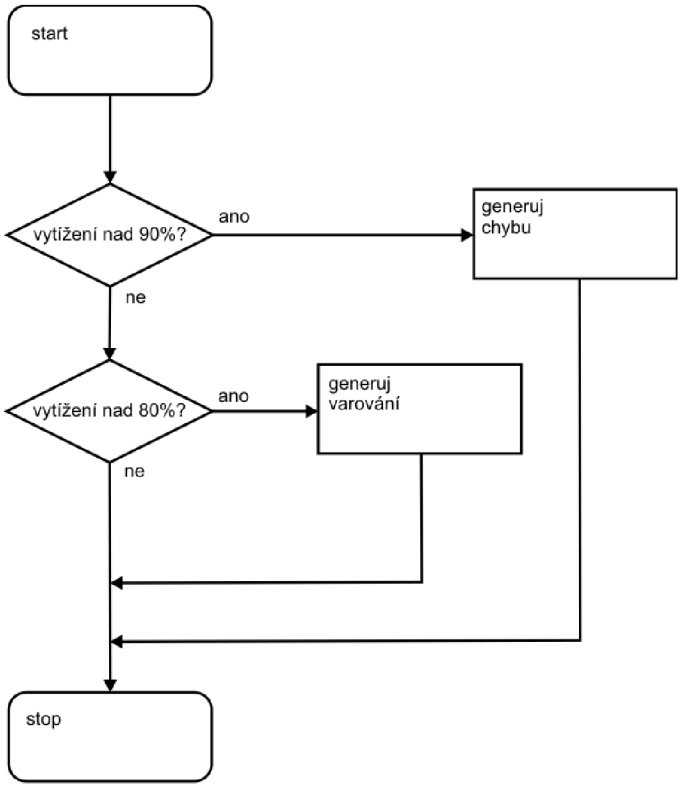
S ohledem na určitou časovou náročnost aktualizací jednotlivých zařízení a zvolenou periodu 30 minut provádí systém proces aktualizace každou minutu a rozděluje tak všechna zařízení do jednotlivých běhů. Každému zařízení je při přidání přiřazeno číslo v rozsahu 1 – 30, které definuje běh, v němž bude provedena aktualizace údajů zařízení. Po přidělení všech volných běhů se proces přidělování opakuje (součástí jednoho běhu tedy může být aktualizace více zařízení). Maximální počet zařízení, které je možné aktualizovat v průběhu jednoho běhu, je omezen dobou aktualizace jednoho zařízení (doba běhu je omezena na 1 minutu).

4.1.3 Kontrola provozních parametrů

Minulou kapitolou definované provozní parametry jsou pro účely tvorby statistik snímány každých 30 minut. s ohledem na nutnost síťové komunikace při zjišťování těchto parametrů a předpokládaný počet zařízení je daná perioda stále dostatečně krátká pro tvorbu podrobných statistik v rámci požadavků na daný systém. Pro kontrolu těchto hodnot je však nutné zvolit periodu kratší, aby bylo v případě problému možné dostatečně rychle reagovat. Z důvodu krátké periody není z již zmíněných důvodů možné zavést logiku této kontroly do systému samotného, ale přímo do zařízení. Pro tento účel je vhodné využít skriptovacích možností, které RouterOS nabízí (podrobnosti viz kapitola 2.3.6). Kombinací vzdáleného logováním (více v kapitole 2.4.3) a kontrolního skriptu lze vytvořit řešení problému dostatečně efektivní kontroly provozních parametrů zařízení.

4.1.3.1 Řešení kontroly provozních parametrů

Kontrola pracuje na principu periodicky spouštěného skriptu (perioda stanovena na 4 minuty), který porovnává aktuální hodnoty parametrů s hodnotami hraničními a v případě jejich překročení uloží do systémového logu záznam o nastalé situaci. Díky nastavenému vzdálenému logování se daná informace přenese i do dohledového systému a ten v závislosti na nastavené poplachové skupině provede příslušná oznámení. RouterOS vrací aktuální stav parametru formou čísla od 0 do 100, které reprezentuje jeho aktuální procentuelní vytížení. Dohledový systém stanovuje hranice 80% pro generování varování a 90% pro generování chyby. Vývojový diagram kontrolního skriptu je zobrazen na Obr. 10. Konkrétní podoba kontrolního skriptu je pak obsažena v příloze 4.



Obr. 10 - vývojový diagram kontrolního skriptu

4.1.4 Správa bezdrátových jednotek

Bezdrátové jednotky představují přímo klientská zařízení nebo bezdrátová rozhraní nasazená jako výchozí body pro další šíření sítě. Každá bezdrátová jednotka je připojena k některému z interfaců monitorovaných systémem. Důležitým parametrem je síla signálu dané bezdrátové jednotky, z které lze usuzovat kvalitu spojení a trend změny síly signálu. Tato informace může pomoci odhalit blížící se problém jak jednotky samotné, tak i celého interfacu. Routovací systém RouterOS neposkytuje dostatečné množství informací o připojených bezdrátových jednotkách prostřednictvím protokolu SNMP. Je proto nutné provádět kontroly pomocí SSH připojení, kontrolního skriptu bezdrátových jednotek a vzdáleného logování. Proces aktualizace informací o bezdrátových jednotkách je následující:

1. Systém provede připojení pomocí protokolu SSH a uživatelského účtu vytvořeného při přidání zařízení a iniciuje spuštění skriptu pro generování informací o bezdrátových jednotkách do systémového logu.
2. Spuštěný skript postupně projde všechny bezdrátové jednotky a pro každou vygeneruje zprávu do systémového logu obsahující MAC adresu jednotky, sílu signálu jednotky, MAC adresu interfacu, na který je jednotka připojena a čas poslední aktivity jednotky. Některé jednotky po delší nečinnosti přecházejí do režimu snížené spotřeby energie, při kterém se sníží jejich vysílací výkon a nemohou tak být zahrnuty do statistiky.
3. Díky vzdálenému logování jsou informace přeneseny do systému, který parsováním jednotlivých zpráv získá odeslané parametry jednotek a aktualizuje příslušné informace v databázích.
4. Znalost MAC adresy interfacu, na který je bezdrátová jednotka připojena, dává možnost sledovat trend změny síly signálu v rámci všech jednotek připojených na stejný interface. Získaný průběh změny síly signálu je možné využít pro včasné odhalení problému s interfacem či kabelovým systémem a anténou, které interface využívá.

Uvedený postup se opakuje s periodou jeden týden. Konkrétní podobu skriptu odesílaného do zařízení obsahuje příloha 5.

4.1.5 Tvorba záloh zařízení

Zálohování nastavení kontrolovaných zařízení je realizováno formou SFTP připojení na zařízení a stažení souboru se zálohou. Před samotným stažením je ještě nutné iniciovat (s využitím SSH správy) vytvoření zálohy na zařízení. Veškeré uvedené kroky jsou na zařízení prováděny pod uživatelským účtem vytvořeným při procesu přidání zařízení. Diagram komunikace mezi systémem a zařízením obsahuje příloha 6. Pro každé zařízení ukládá systém jednu zálohu, která se aktualizuje jednou týdně.

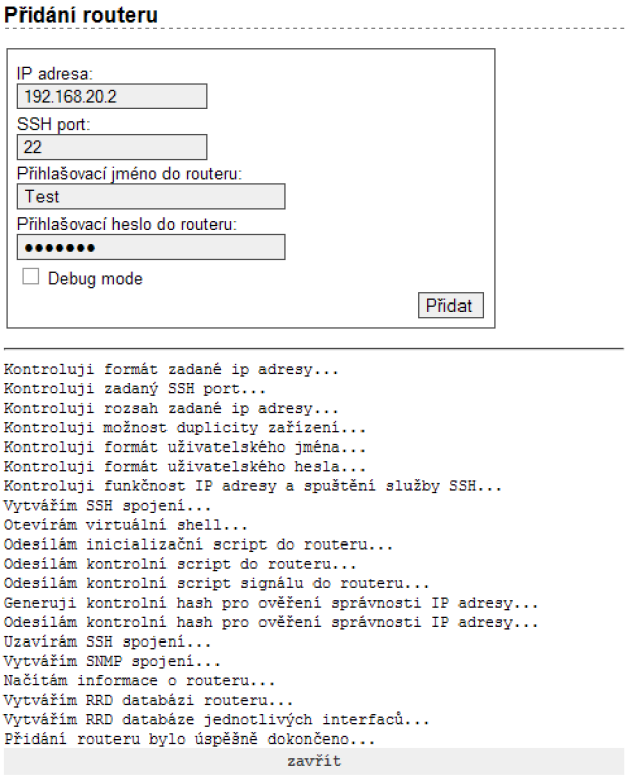
4.1.6 Přidání zařízení do systému

Přidání nového routeru do systému představuje proces složený z několika kroků, jež postupně připraví jak systém, tak i samotné zařízení pro spolupráci vedoucí k splnění dohledových a kontrolních požadavků definovaných předchozí analýzou. Uživatel je o každém kroku procesu přidání průběžně

informován, což zjednodušuje odhalení případného problému. Celý průběh lze shrnout do následující sekvence kroků:

1. Uživatel iniciuje přidání nového zařízení založeného na routovacím systému RouterOS, zadá IP adresu příslušného zařízení, port a přihlašovací údaje, s pomocí kterých je možné uskutečnit vzdálenou správu zařízení, založenou na protokolu SSH.
2. Systém zkontroluje funkčnost zadané IP adresy a pokusí se vytvořit SSH připojení.
   Po úspěšném připojení se provede kolekce příkazů, které zařízení připraví na kontrolu a spolupráci se systémem:
   - nahraje se skript pro kontrolu provozních parametrů (příloha 4.)
   - nahraje se skript pro aktualizace informací o bezdrátových jednotkách (příloha 5.)
   - nastaví se automatické spouštění kontrolního skriptu
   - nastaví se SNMP a povolí získávání údajů z IP adresy systému
   - vytvoří se nový uživatel, který bude využit pro tvorbu záloh a další SSH komunikaci se zařízením
   - aktivuje se a nastaví vzdálené logování
3. Systém prostřednictvím SSH připojení vyvolá vygenerování nového záznamu do systémového logu zařízení s unikátním identifikátorem procesu přidávání zařízení. Vzhledem k faktu, že je již aktivováno vzdálené logování, očekává systém návrat tohoto identifikátoru. Pokud se identifikátor vrátí ze zadané IP adresy, znamená to, že byl proces nastavení zařízení úspěšný a zařízení komunikuje se systémem ze správné adresy (vzhledem k možnosti instalace většího počtu síťových interfaců s různými IP adresami může být zařízení ze systému dostupné pod více adresami, ale směrem k systému bude komunikovat pouze z jedné). Odesílaný příkaz je součástí přílohy 7.
4. S využitím protokolu SNMP se načte název zařízení a informace o jeho interfacech.
5. Zařízení a zjištěné interfacy se uloží do databáze systému a uživatel je informován o úspěchu proběhlého procesu.

Ukázka výstupu informujícího o průběhu přidávání zařízení je zobrazena na Obr. 11 a diagram komunikace mezi uživatelem, systémem a zařízením pak obsahuje příloha 8. Konkrétní podobu celého sledu uvedených příkazů obsahuje příloha 9.



Obr. 11 - informační výpis přidání nového routeru

4.1.7 Odstranění zařízení ze systému

Při ukončení kontroly a monitoringu zařízení systémem je nutné kromě informací v samotném systému odstranit i nastavení provedená v zařízení (daná nastavení popisuje kapitola 4.1.6).

Vzhledem k nutnosti změn nastavení zařízení je nutné i při procesu odstranění provést SSH připojení a realizovat sadu příkazů, které uvedou nastavení zařízení do podoby, ve které bylo před jeho integrací do systému. Provádějí se následující změny:

- vypnutí vzdáleného logování
- vypnutí SNMP
- vypnutí automatického spouštění kontrolního skriptu
- odstranění kontrolního skriptu a skriptu pro aktualizaci informací o bezdrátových jednotkách
- odstranění systémového uživatele

Konkrétní sada vykonávaných příkazů je součástí přílohy 10.

4.2 Implementace a použité technologie

4.2.1 Programovací jazyk

S ohledem na požadavek dostupnosti rozhraní systému z libovolného internetového prohlížeče byl jako hlavní programovací prostředek zvolen skriptovací jazyk PHP[3] ve verzi 5. i přes určité nedostatky

[3] http://www.php.net/

zvoleného řešení daného především agilním přístupem k jeho vývoji poskytuje PHP ve své páté verzi dostatečně robustní základ pro tvorbu systému daného uvedenou analýzou požadavků. Velká vývojářská komunita je pak zárukou dostatečné dokumentace a množství rozšíření, která dovolují implementovat všechny části systému tak, jak byly definovány v kapitole 4.1. Následující seznam udává použitá rozšíření a technologie realizující klíčové části systému:

- **SSH2**[4] – rozšíření, které dovoluje využívat některých funkcí protokolu SSH verze 2 pro vzdálenou správu zařízení. Součástí rozšíření je i podpora pro zabezpečené kopírování souborů, která se využívá při získávání souborů záloh ze zařízení. Implementačně je celé rozšíření zapouzdřeno ve třídě ***sshConnect***.
- **SNMP** – rozšíření implementující práci s protokolem SNMP. Rozšíření zapouzdřuje třída ***snmpConnect***, díky čemuž lze v případě například nedostatečného výkonu daného rozšíření provést jeho výměnu bez nutnosti četných zásahů do celého systému.
- **RRDtool** – Ačkoli je k nástroji blíže popsanému v kapitole 4.1.2.1 dostupné rozšíření pro PHP, je v systému zvolen přímý přístup k jednotlivým utilitám. Dané řešení dovoluje kompletní využití nabízených funkcí a zvyšuje výkon při tvorbě výstupních grafů. Nástroj RRDtool zapouzdřuje třída ***rrdConnect***.

Převážná část systému je realizována formou objektově orientovaného programování, k čemuž využívá objektový model představený v páté verzi PHP. Výstupem práce skriptů jsou dokumenty založené na značkovacím jazyce XHTML[5]. Formu výstupním dokumentům dodávají styly CSS[6]. Ukázky uživatelského rozhraní jsou součástí obrazových příloh 15 a 16. PHP skripty jsou použity u částí systému, které nemají žádný uživatelský výstup. Jedná se především o automatizované aktualizace provozních parametrů zařízení (viz kapitola 4.1.2), tvorba záloh (viz kapitola 4.1.5) a aktualizace informací o bezdrátových jednotkách (viz kapitola 4.1.4). O periodické spouštění skriptů pro uvedené operace se stará démon Cron[7] spuštěný na stejném serveru, kde běží dohledový systém. Konfigurace Cron démona je součástí přílohy 11.

V PHP je realizován i skript sloužící k parsování zpráv vzdáleného logování. O samotný příjem zpráv a jejich transport samotnému parsovacímu skriptu se stará démon vytvořený ve skriptovacím jazyce Perl[8], který je součástí přílohy 12.

Konkrétní podobu implementace jednotlivých tříd reprezentuje diagram tříd obsažený v příloze 13 a kompletní programátorská příručka. Příručka byla vytvořena s pomocí dokumentačních komentářů nástrojem phpDocumentor[9] a je dostupná na přiloženém CD.

[4] http://pecl.php.net/package/ssh2
[5] http://cs.wikipedia.org/wiki/XHTML
[6] http://cs.wikipedia.org/wiki/Cascading_Style_Sheets
[7] http://en.wikipedia.org/wiki/Cron
[8] http://cs.wikipedia.org/wiki/Perl
[9] http://www.phpdoc.org/

4.2.2 Databázový systém

Pro účely ukládání a správy dat byl zvolen databázový systém (SŘBD) MySQL[10] ve verzi 5. MySQL byl zvolen především z důvodu bezproblémové spolupráce se skriptovacím jazykem PHP a nízkým nárokům na jeho provoz a správu.

V systému je veškerá práce s databází realizována prostřednictvím vložené databázové vrstvy (database layer), která zajišťuje nezávislost na použitém databázovém rozšíření PHP a databázi samotné. Z důvodu kompatibility s předchozími verzemi MySQL, případně jinými SŘBD, nejsou použity žádné procedury ani jiné pokročilé přístupy pro práci s daty. Data jsou skladována v úložišti dat (storage engine) typu *MyISAM*.

Kompletní struktura databáze je zobrazena v datovém diagramu, který obsahuje příloha 14.

4.2.3 Struktura projektu

Celá implementace je tvořena sadou skriptů PHP, RRD databází, souborů obsahujících příkazy a skripty nahrávané do přidávaných zařízení a pomocných souborů pro tvorbu uživatelského rozhraní, jako jsou obrázky a styly CSS. Celý projekt je rozdělen do následující hierarchie adresářů:

- **/** – kořenový adresář, obsahuje skripty pro přímou komunikaci s prohlížeči uživatelů
- **/class** – adresář se soubory jednotlivých tříd projektu
- **/cron_update** – adresář se soubory automaticky spouštěných skriptů démonem Cron
- **/download** – adresář obsahující soubory pro konfiguraci a správu zařízení utilitou Winbox
- **/img** – adresář s obrázky pro GUI systému
- **/log_analyzer** – adresář s výchozím skriptem pro zpracování zpráv vzdáleného logovaní
- **/rrd_storage** – adresář s veškerými databázemi RRD (každá databáze je realizována jedním souborem)
- **/ssh_control** – adresář se skripty a sadami příkazů pro nastavení kontrolovaných routerů
- **/styles** – adresář se soubory stylů CSS a jejich pomocnými soubory

[10] http://www.mysql.com/

5 Závěr

Ve svém úvodu definuje tato práce nutnost monitorovat a kontrolovat provoz a parametry moderních počítačových sítí. v reakci na to je pak stanoven samotný cíl práce, kterým je vytvořit monitorovací a dohledový systém pro skutečnou počítačovou síť.

V rámci druhé kapitoly byl představen routovací systém Mikrotik tvořící základ počítačové sítě poskytovatele bezdrátového připojení k internetu, který je následně analyzován z pohledu možností jeho kontroly a monitorování. Analýza dále zmiňuje zkušenosti z reálného provozu přímo ovlivňující schopnosti monitorovat zařízení postavená na uvedeném řešení.

Uvedená analýza a teoretické podklady se staly základem pro definici samotného monitorovacího a dohledového systému, jehož podrobná analýza požadavků tvoří druhou část této práce.

Následuje kapitola definující metodologie vývoje použité při návrhu a implementaci samotného dohledového a monitorovacího systému. Na uvedenou analýzu požadavků navazuje use case model systému s podrobným popisem jednotlivých aktérů, případů použití a jejich scénářů.

Čtvrtá kapitola uvádí řešení realizace klíčových částí navrženého systému a odkazuje již na konkrétní implementace popsaných postupů. Kapitolu uzavírá část věnující se konkrétní implementaci systému, použitému programovacímu prostředku a rozvržení celého systému. Uvedené informace doplňuje velké množství textových i obrazových příloh, na něž text často odkazuje.

Algoritmizací postupů uvedených v první části čtvrté kapitoly vznikl monitorovací a dohledový systém pro počítačovou síť založenou na routovací platformě Mikrotik definovaný v úvodu této práce. Návrhem a následnou implementací byly splněny všechny požadavky kladené na tento systém a výsledné řešení bylo nasazeno do provozu na reálné počítačové síti.

Do budoucna lze předpokládat další rozvoj systému především v podobě podpory dalších typů zařízení a jeho případné decentralizaci do více částí sítě z důvodu snížení zátěže systému a zvýšení jeho funkčních možností. v rámci uvedených vylepšení je možné předpokládat i novou implementaci, která zajistí zvýšení výkonu systému a podporu monitoringu většího počtu zařízení. Teoretické postupy a analýzy provedené v rámci této práce jsou však implementačně nezávislé a využitelné pro většinu dnes dostupných vývojových technologií.

Literatura

[1] Wikipedie: Otevřená encyklopedie: Mikrotik [online]. ©2007 [citováno 29. 12. 2007]. Dostupný z WWW: <http://cs.wikipedia.org/w/index.php?title=Mikrotik>

[2] MikroTik. Routerboard [online]. [2007] [cit. 2007-12-29]. Dostupný z WWW: <http://www.routerboard.com/>

[3] Dokumentace Mikrotik v2.9 : specifikace systému [online]. © Copyright 1999-2007 [cit. 2007-12-29]. Dostupný z WWW:<http://www.mikrotik.com/testdocs/ros/2.9/guide/specs.php>

[4] SHINDER, Debra Littlejohn. *Počítačové sítě : Nepostradatelná příručka k pochopení síťové teorie, implementace a vnitřních funkcí.* COMPUTER NETWORKING ESSENTIALS. [s.l.] : SoftPress, 2003. 752 s. ISBN 80-86497-55-0

[5] Dokumentace Mikrotik v2.9 : skriptovací jazyk [online]. © Copyright 1999-2007 [cit. 2007-12-29]. Dostupný z WWW: <http://www.mikrotik.com/testdocs/ros/2.9/system/scripting.php>

[6] KRÁL, J. *Informační systémy (specifikace, realizace, provoz).* 1. vyd. Science Veletiny 1998. ISBN 80-86083-00-4

[7] KADLEC, Václav. *Agilní programování : Metodiky efektivního vývoje softwaru.* 1. vyd. Brno: Computer Press, 2004. 278 s. ISBN 80-251-0342-0

[8] Wikipedie: Otevřená encyklopedie: *Brainstorming* [online]. ©2007 [citováno 30. 12. 2007]. Dostupný z WWW: <http://cs.wikipedia.org/w/index.php?title=Brainstorming>

[9] *Dokumentace nástroje RRDtool* [online]. Tobias Oetiker, 2005 , 18.4.2005 [cit. 2008-05-24]. Dostupný z WWW: <http://oss.oetiker.ch/rrdtool/doc/index.en.html>

[10] BARRETT, D.J, SILVERMAN, R.E. *SSH : Kompletní průvodce.* Martin Blažík. 1. vyd. Brno : Computer Press, 2003. 556 s. ISBN 80-7226-852-X

[11] MRÁZEK, Oldřich. *SNMP protokol a jeho využití* [online]. Redakce HW serveru, © Copyright 1997-2005 [cit. 2008-05-24]. Dostupný z WWW: <http://hw.cz/Produkty/ART957-SNMP-protokol-a-jeho-vyuziti.html>

[12] ULLMAN, Larry E. *PHP - pokročilé programování pro World Wide Web.* Praha : SoftPress, 2003. 512 s. ISBN 80-86497-36-4

[13] DUBOIS, Paul. *MySQL profesionálně : Komplexní průvodce použitím, programováním a správou MySQL.* 1. vyd. Praha : Mobil Media, 2003. 1071 s. IDnes internet i knihy. ISBN 80-86593-41-X

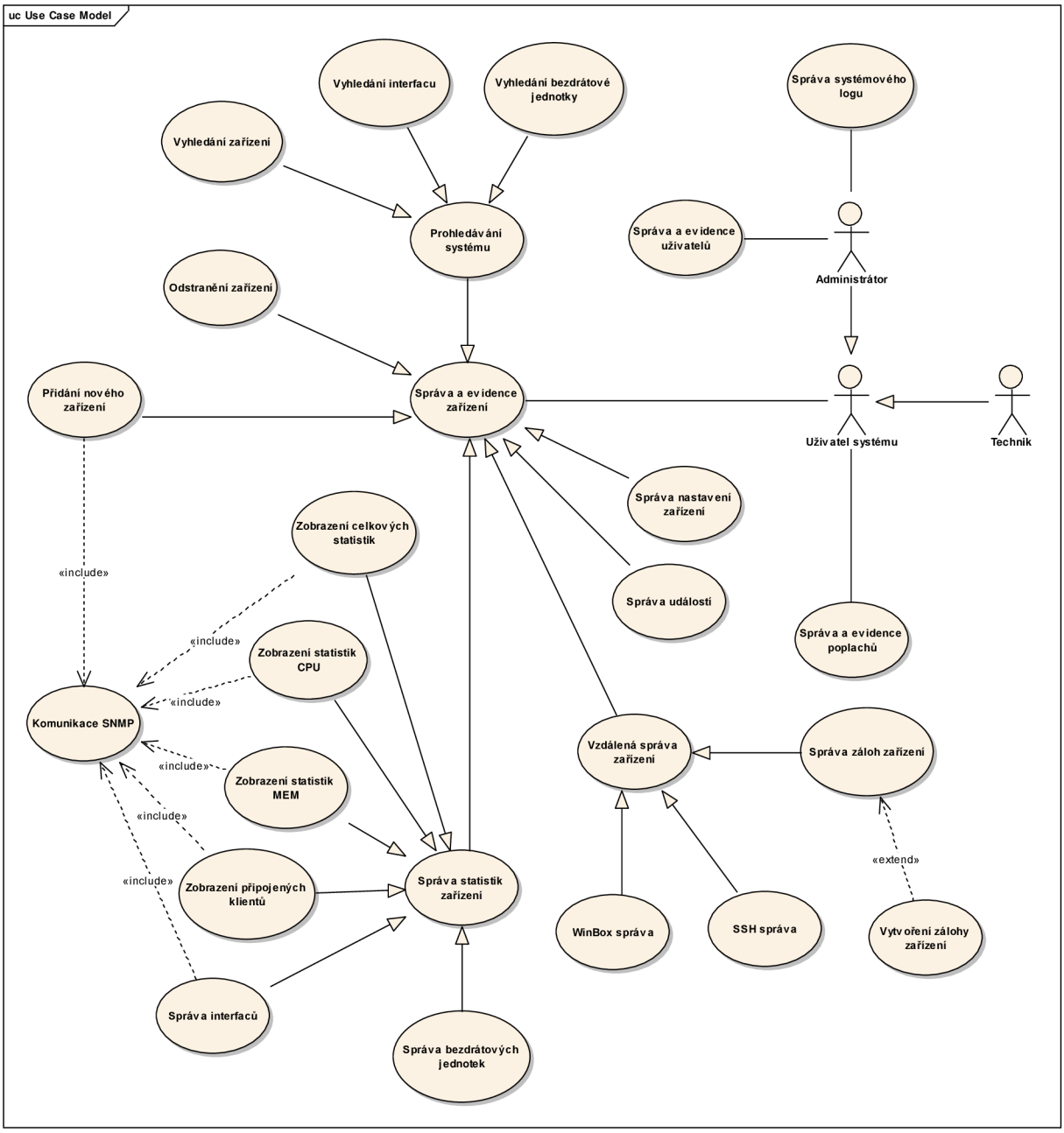
[14] CONVERSE, Tim, PARK, Joyce, MORGAN, Clark. *PHP 5 and MySQL bible.* Indianapolis USA : Wiley Publishing, Inc., 2004. 1042 s. ISBN 0-7645-5746-7

[15] STANÍČEK, Petr. *CSS Kaskádové styly : Kompletní průvodce.* 1. vyd. Praha : Computer Press, 2003. 178 s. ISBN 80-7226-872-4

[16] PUŽMANOVÁ, R. *Routing and Switching: Time of Convergence?* Pearson Education Limited, 2002, ISBN 0-201-39861-3

Seznam příloh

Příloha 1. Use case model



Příloha 2. OID používaná v systému pro monitoring parametrů routerů na bázi RouterOS

Označení	OID
čas běhu	1.3.6.1.2.1.1.3.0
využití CPU	1.3.6.1.2.1.25.3.3.1.2.1
využití paměti	1.3.6.1.2.1.25.2.3.1.6.2
celkové množství paměti	1.3.6.1.2.1.25.2.3.1.5.2
využití kapacity HDD	1.3.6.1.2.1.25.2.3.1.6.1
celková kapacita HDD	1.3.6.1.2.1.25.2.3.1.5.1
název zařízení	1.3.6.1.2.1.1.5.0
název interfacu	1.3.6.1.2.1.2.2.1.2
MAC adresa interfacu	1.3.6.1.2.1.2.2.1.6
stav interfacu	1.3.6.1.2.1.2.2.1.8
download (bajty)	1.3.6.1.2.1.2.2.1.10
upload (bajty)	1.3.6.1.2.1.2.2.1.16
download (pakety)	1.3.6.1.2.1.2.2.1.11
upload (pakety)	1.3.6.1.2.1.2.2.1.17
verze firmware	1.3.6.1.4.1.14988.1.1.4.4.0
napětí 3V	1.3.6.1.4.1.14988.1.1.3.2.0
napětí 5V	1.3.6.1.4.1.14988.1.1.3.3.0
napětí 12V	1.3.6.1.4.1.14988.1.1.3.4.0
teplota case	1.3.6.1.4.1.14988.1.1.3.5.0
teplota CPU	1.3.6.1.4.1.14988.1.1.3.6.0
teplota základní desky	1.3.6.1.4.1.14988.1.1.3.7.0
jméno klienta	1.3.6.1.4.1.14988.1.1.2.1.1.2 (SUBOID)
IP adresa klienta	1.3.6.1.4.1.14988.1.1.2.1.1.3 (SUBOID)
download klienta (bajty)	1.3.6.1.4.1.14988.1.1.2.1.1.8 (SUBOID)
upload klienta (bajty)	1.3.6.1.4.1.14988.1.1.2.1.1.9 (SUBOID)
download klienta (pakety)	1.3.6.1.4.1.14988.1.1.2.1.1.10 (SUBOID)
upload klienta (pakety)	1.3.6.1.4.1.14988.1.1.2.1.1.11 (SUBOID)

Příloha 3 . Příkazy vytvoření RRD databáze pro router, interfacy a bezdrátové jednotky

```
//router
rrdtool create /database_name.rrd
   --start 0                 //zaciname na 0 (timestamp)
   --step 1800               //update se ocekava kazdych 30m
   DS:cpu:GAUGE:2400:0:100   //data cpu minimum je 0 a maximum 100
   DS:mem:GAUGE:2400:0:100   //data pameti
   DS:rsa:GAUGE:2400:U:U     //rezervni prostor a
   DS:rsb:GAUGE:2400:U:U     //rezervni prostor b
   DS:rsc:COUNTER:2400:U:U   //rezervni prostor c (typu counter)
   DS:rsd:COUNTER:2400:U:U   //rezervni prostor d (typu counter)
   RRA:AVERAGE:0.5:1:336     //ulozeni 30m rozliseni po dobu jednoho tydne (2 * 24h * 7d)
   RRA:AVERAGE:0.5:2:744     //ulozeni 1h rozliseni po dobu jednoho mesice
   RRA:AVERAGE:0.5:48:365    //ulozeni 1d rozliseni po dobu jednoho roku
   RRA:AVERAGE:0.5:288:31    //ulozeni 1m rozliseni po dobu peti let

//interface
rrdtool create /database_name.rrd
   --start 0                        //zaciname na 0 (timestamp)
   --step 1800                      //update se ocekava kazdych 30m
   DS:trend:GAUGE:2400:U:U          //tred sily signalu
   DS:bytesin:COUNTER:2400:U:U      //dosla data
   DS:bytesout:COUNTER:2400:U:U     //odeslana data
   DS:packetsin:COUNTER:2400:U:U    //dosle packety
   DS:packetsout:COUNTER:2400:U:U   //odeslane packety
   RRA:AVERAGE:0.5:1:336            //ulozeni 30m rozliseni po dobu jednoho tydne
   RRA:AVERAGE:0.5:2:744            //ulozeni 1h rozliseni po dobu jednoho mesice
   RRA:AVERAGE:0.5:48:365           //ulozeni 1d rozliseni po dobu jednoho roku
   RRA:AVERAGE:0.5:288:31           //ulozeni 1m rozliseni po dobu peti let

//bezdratova jednotka
rrdtool create /database_name.rrd
   --start 0                  //zaciname na 0 (timestamp)
   --step 604800              //update jednou tydne
   DS:trend:GAUGE:604800:U:U  //trend zmeny sily silgnalu
   RRA:AVERAGE:0.5:1:250      //ulozeni 1t rozliseni po dobu cca peti let
```

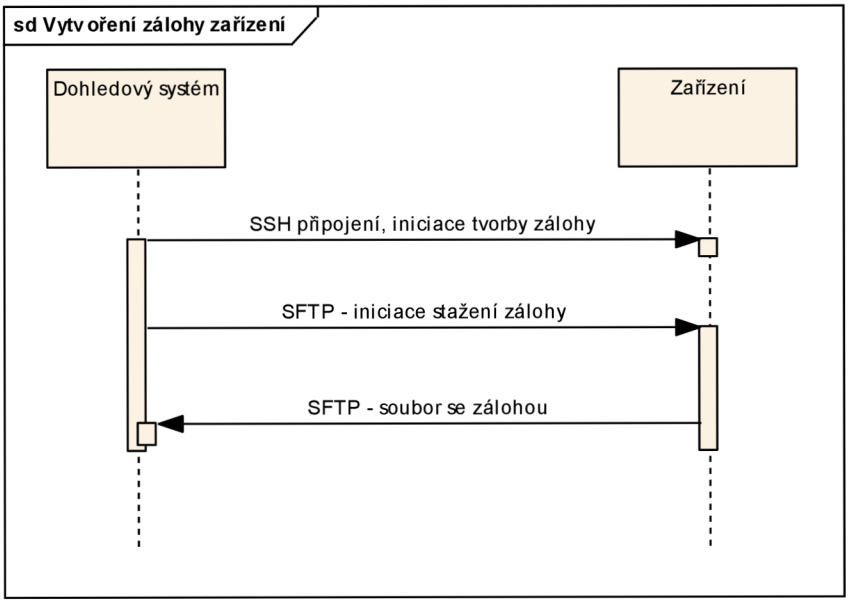
Příloha 4. Skript kontroly provozních parametrů zařízení

```
##############################################################################
# MikrotikMon
# Pavel Mica (xmicap01)
# Skript kontroly provoznich parametru zarizeni
##############################################################################

#nastaveni a ulozeni kontrolniho scriptu
:if ((":" . [/system script find name=dohled] . ":") != "..") do={/system script remove [find
name=dohled]}
/system script add name="dohled" source={

#deklarujeme lokalni promenne
:local cpu 0
:local mem 0

#v cyklu provedeme 10 mereni po 6 sekundach (celkova delka je tedy 1 minuta)
:for i from=1 to=10  do={
#zjisteni a ulozeni zatizeni CPU
  :set cpu ($cpu + [/system resource get cpu-load])
#prevedeni na procentualni vyuziti a ulozeni pameti
  :set mem ($mem + ((([/system resource get total-memory] - [/system resource get free-memory]) *
100) / [/system resource get total-memory]))
  :delay 6
}

#vyhodnotime vysledky a pripadne odesleme zravu
#kazda zprava se odesila dvakrat se spozdenim 2s aby byla vetsi jistota ze se neztrati

#cpu
:if ($cpu>800) do={
  :if($cpu>900) do={
    :log info ("type=cpu_error value=" . ($cpu/10))
    :delay 2
    :log info ("type=cpu_error value=" . ($cpu/10))
  }else={
    :log info ("type=cpu_warning value=" . ($cpu/10))
    :delay 2
    :log info ("type=cpu_warning value=" . ($cpu/10))
  }
}


#pamet
:if ($mem>800) do={
  :if($mem>900) do={
    :log info ("type=mem_error value=" . ($mem/10))
    :delay 2
    :log info ("type=mem_error value=" . ($mem/10))
  }else={
    :log info ("type=mem_warning value=" . ($mem/10))
    :delay 2
    :log info ("type=mem_warning value=" . ($mem/10))
  }
```

Příloha 5. Skript aktualizace informací o bezdrátových jednotkách

```
##############################################################################
# MikrotikMon
# Pavel Mica (xmicap01)
# Skript aktualizace informaci o bezdratovych jednotkach
##############################################################################

#nastaveni a ulozeni kontrolniho scriptu
:if ((":" . [/system script find name=dohled_signal] . ":") != "..") do={/system script remove [find
name=dohled_signal]}
/system script add name="dohled_signal" source={
##############################################################################################################
############
# Script kontroly sily signalu, odesila silu signalu pomoci logu - spousten externe pres ssh jednou
tydne
##############################################################################################################
############

#nastavime pracovni adresar
/interface wireless registration-table

#postupne projdeme celou tabulku wireless klientu
:foreach i in=[/interface wireless registration-table find] do {
:set mac [get $i mac-address]
:set signal [get $i signal-strength]
:set interface [get $i interface]
:set activity [get $i last-activity]
:set last-ip [get $i last-ip]
#zjistime MAC adresu interfacu pres ktery je dany klient pripojen
/interface wireless
:set ifacemac [get [find name=$interface] mac-address]
/interface wireless registration-table
:set client "NaN"
:if ((":" . [/queue simple find target-address=$last-ip] . ":") != "..") do={:set client [/queue
simple get [find target-address=$last-ip] name]}
#odesleme ziskane informace pomoci log cesty do systemu k dalsimu zpracovani
:log info ("type=mk_signal_test_run mac=" . $mac . " signal=" . $signal  . " interface=" . $ifacemac .
" activity=" . $activity . " client=" . $client . " ip=" . $last-ip)
#pockame sekundu
:delay 1
```

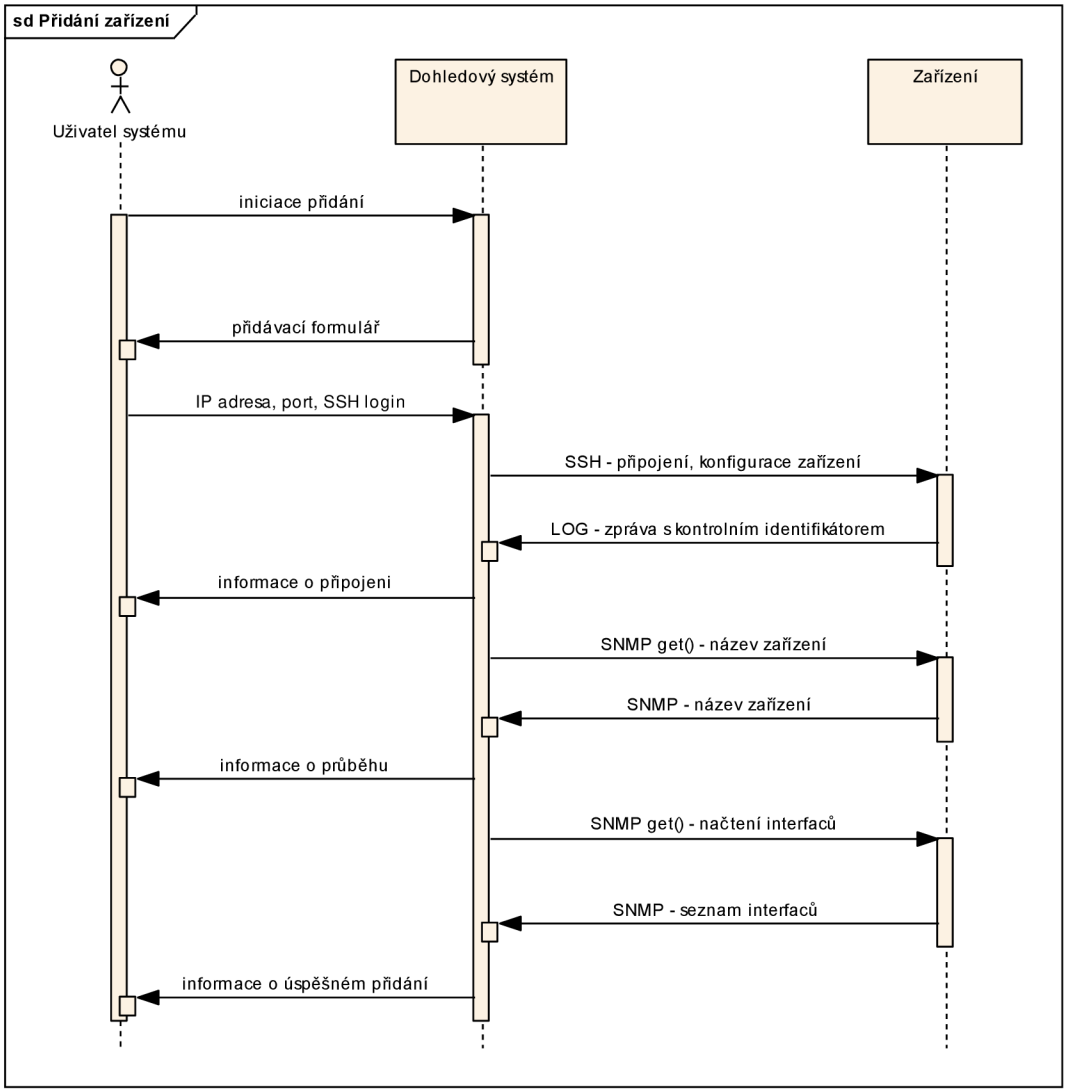
Příloha 6. Diagram komunikace při procesu tvorby zálohy zařízení



Příloha 7. Příkaz odeslání kontrolního identifikátoru při přidávání zařízení

```
###############################################################################
# MikrotikMon
# Pavel Mica (xmicap01)
# Odeslani kontrolniho identifikatoru
###############################################################################

#Prikaz pro odeslani kontrolniho hashe vygenerovaneho pri procesu pridavani routeru
#Pokud bude IP adresa routeru z ktere prijde zprava s timto hashem schodna s IP adresou ktera se
pouzila pri pridavani
#tak se jedna o spravnou IP a router bude ulozen pod touto IP adresou
:log info ("type=check_hash hash=%CHECK_HASH%")
```

Příloha 8. Diagram komunikace při procesu přidávání nového routeru



Příloha 9. Sled příkazů pro nastavení nově přidávaného routeru

```
##############################################################################
# MikrotikMon
# Pavel Mica (xmicap01)
# Inicializacni sled prikazu noveho routeru
##############################################################################

#--
#Nastaveni SNMP
#--
#odstraneni mozne defaultni community public
:if (("." . [/snmp community find name=public] . ".") != "..") do={/snmp community remove [find
name=public]}
#zapnuti snmp
/snmp set enabled=yes
#pridani community pro dohled s omezenim na danou IP adresu
:if (("." . [/snmp community find name=dohled] . ".") != "..") do={/snmp community remove [find
name=dohled]}
/snmp community add name=dohled address=%IP% read-access=yes

#--
#nastaveni a zapnuti scheduleru pro kontrolni script
#--
:if (("." . [/system scheduler find name=dohled] . ".") != "..") do={/system scheduler remove [find
name=dohled]}
/system scheduler add name=dohled interval=4m on-event=dohled start-time=00:00:00

#--
#vytvoreni uzivatele pro potreby zalohovani a cteni informaci z routeru (pozor na adresu ze ktere bude
pristup)
#--
#smazeme usera dohled ktery vyuziva skupinu dohled a pote smazeme i skupinu dohled
:if (("." . [/user find name=dohled] . ".") != "..") do={/user remove [find name=dohled]}
:if (("." . [/user group find name=dohled] . ".") != "..") do={/user group remove [find name=dohled]}
#vytvorime skupinu dohled a usera dohled ktery ji vyuziva
/user group add name=dohled policy=ssh,test,ftp,policy,read
/user add name=dohled password=%PASS% group=dohled address=%IP%

#--
#Nastaveni a zapnuti logovani (pozor na adresu kam se bude log posilat), logovani se zapina jako
posledni aby do systemu neslo info o pridavanych nastavenich v ramci procedury pridani
#--
#pokud existuje logovaci akce dohled tak se smazou vsechna logovani ktera tuto akci vyuzivaji a pak se
smaze samotna akce
:if (("." . [/system logging action find name=dohled] . ".") != "..") do={
  :if (("." . [/system logging find action=dohled] . ".") != "..") do={/system logging remove [find
action=dohled]}
  /system logging action remove [find name=dohled]
}
#vytvori se nova logovaci akce dohled a priradi se k ni dana logovani
/system logging action add name=dohled target=remote remote=%LOGIP%:5140
/system logging add topics=script action=dohled
/system logging add topics=system action=dohled
```

Příloha 10. Sled příkazů vykonávaný při odstranění zařízení

```
##############################################################################
# MikrotikMon
# Pavel Mica (xmicap01)
# Sled prikazu k odstraneni nastaveni z routeru
##############################################################################

#--
#Vypnuti logovani (logovani se vypina jako prvni aby info o remove jiz neslo do systemu)
#--
#pokud existuje logovaci akce dohled tak se smazou vsechna logovani ktera tuto akci vyuzivaji a pak
se smaze samotna akce
:if (("." . [/system logging action find name=dohled] . ".") != "..") do={
  :if (("." . [/system logging find action=dohled] . ".") != "..") do={/system logging remove [find
action=dohled]}
  /system logging action remove [find name=dohled]
}

#--
#Nastaveni SNMP
#--
#vypnuti snmp
/snmp set enabled=no
#odebrani community
:if (("." . [/snmp community find name=dohled] . ".") != "..") do={/snmp community remove [find
name=dohled]}

#--
#Vypnuti scheduleru pro kontrolni script
#--
:if (("." . [/system scheduler find name=dohled] . ".") != "..") do={/system scheduler remove [find
name=dohled]}

#--
#Odstraneni servisniho uzivatele
#--
#smazeme usera dohled ktery vyuziva skupinu dohled a pote smazeme i skupinu dohled
:if (("." . [/user find name=dohled] . ".") != "..") do={/user remove [find name=dohled]}
:if (("." . [/user group find name=dohled] . ".") != "..") do={/user group remove [find name=dohled]}

#--
#Odstraneni kontrolniho scriptu
#--
:if (("." . [/system script find name=dohled] . ".") != "..") do={/system script remove [find
name=dohled]}

#--
#Odstraneni  scriptu aktualizace bezdrátových jednotek
#--
:if (("." . [/system script find name=dohled_signal] . ".") != "..") do={/system script remove [find
name=dohled_signal]}
```

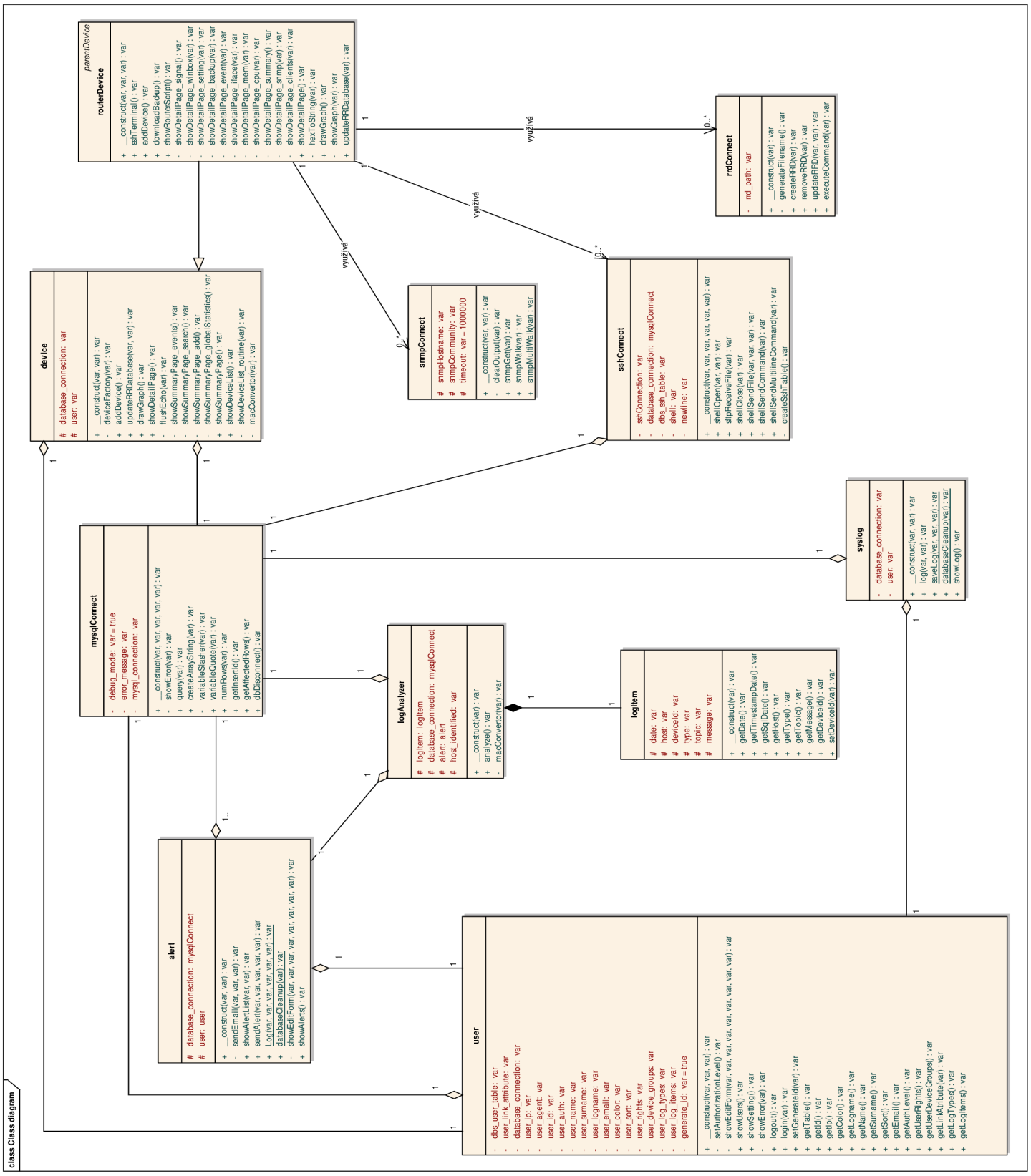
Příloha 11. Konfigurace Cron démona

```
####################
# MikrotikMon
# Pavel Mica (xmicap01)
# cron daemon nastaveni
####################

#aktualizacni skript provoznich parametru zarizeni a provozu na interfacech
*/1 * * * * /usr/bin/php /var/www/crempa.net/diplomka/cron_update/device_rrd_update.php

#aktualizacni skript tvorby zaloh zarizeni
*/10 * * * * /usr/bin/php /var/www/crempa.net/diplomka/cron_update/router_backup_update.php

#aktualizacni skript parametru bezdratovych jednotek
*/5 * * * * /usr/bin/php /var/www/crempa.net/diplomka/cron_update/router_signal_update.php

#aktualizacni skript cisteni databaze
@daily usr/bin/php /var/www/crempa.net/diplomka/cron_update/database_cleanup.php
```

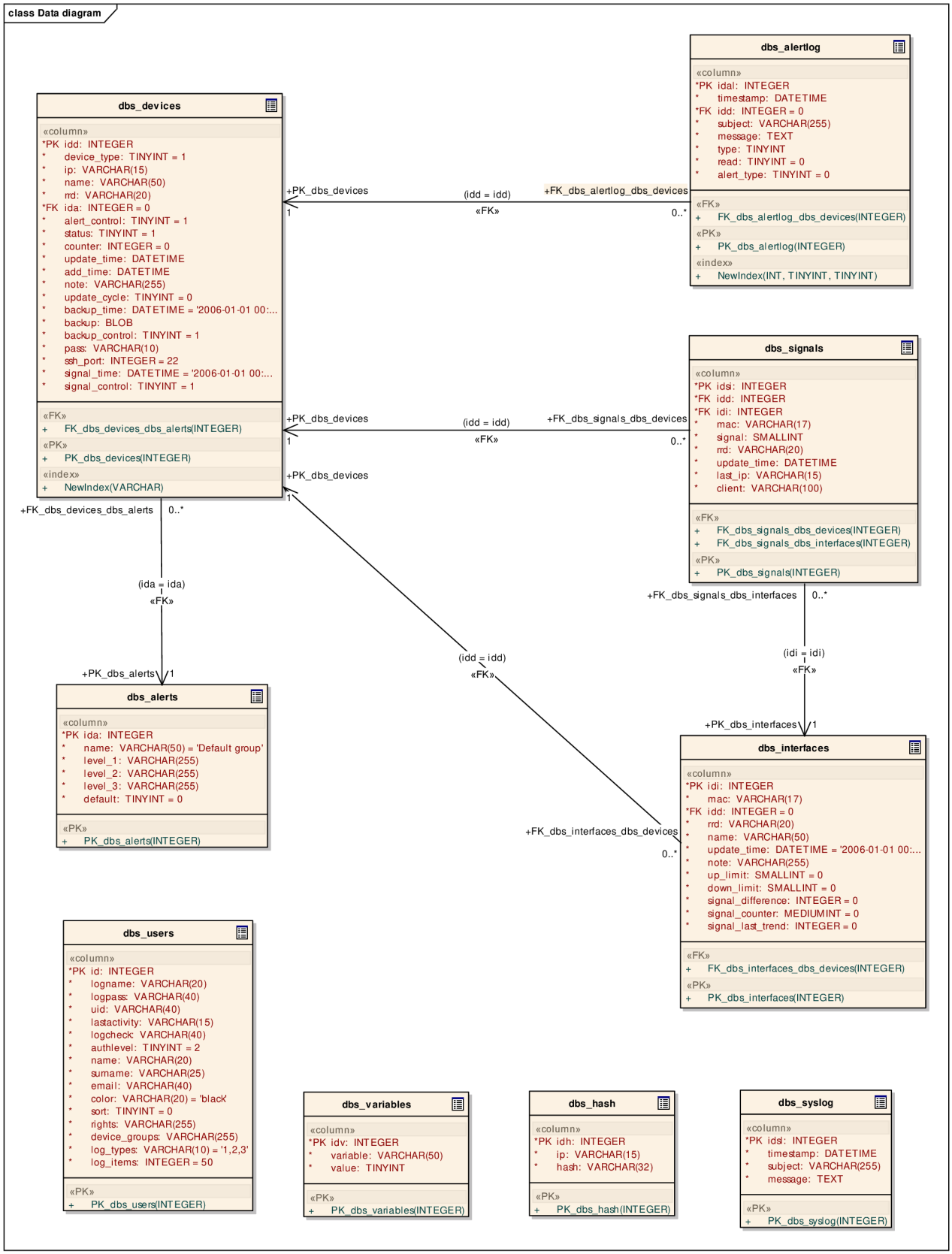
Příloha 12. Perl démon pro příjem zpráv vzdáleného logování

```
#!/usr/bin/perl
while(defined($_ = ))
{
    chomp;
    s/\'//g;
    s/\//;
    system ("su ftp -c \"echo '$_' |php /var/www/crempa.net/diplomka/log_analyzer/analyze.php\" &");
}
```

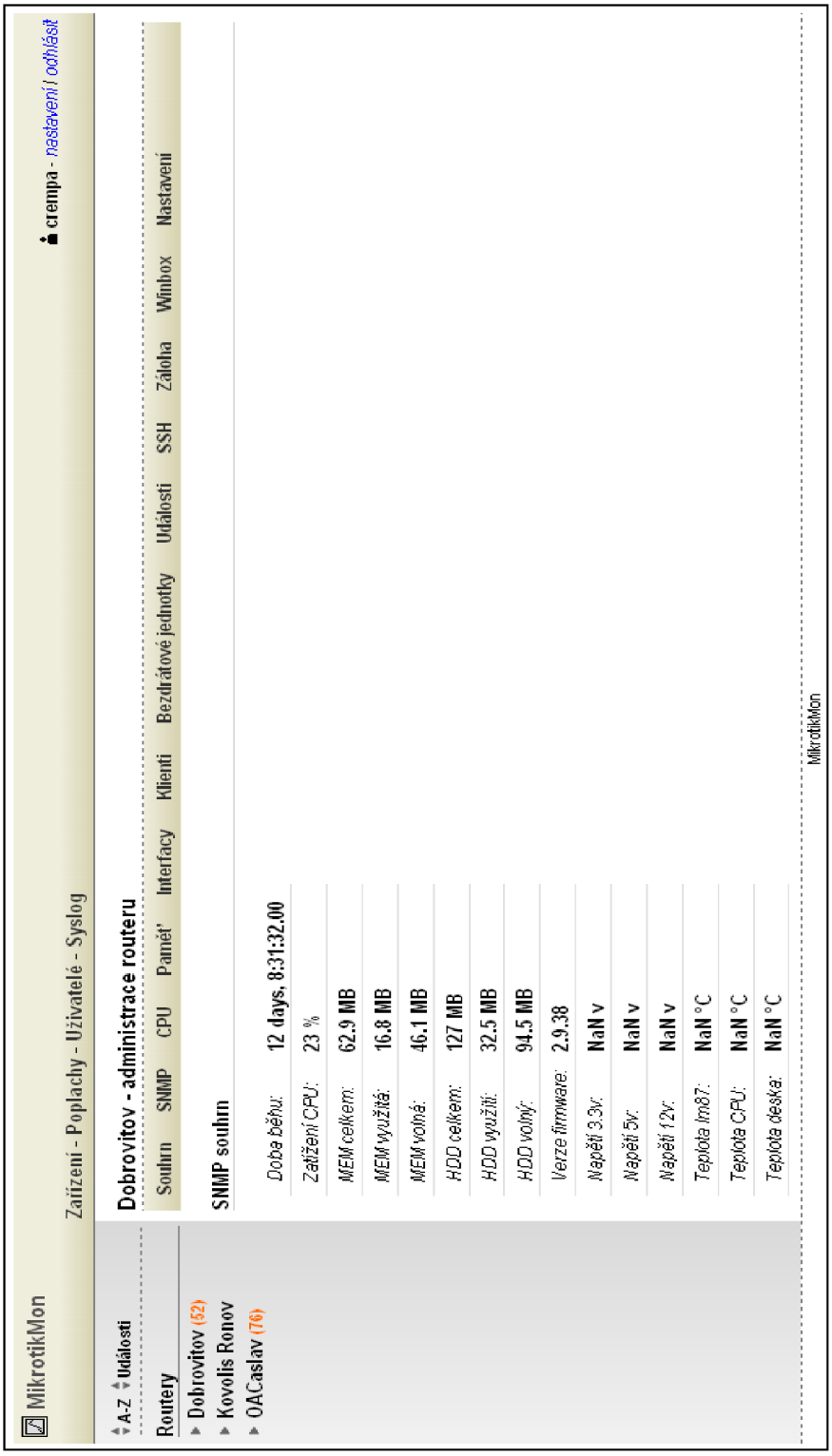
Příloha 13. Digram tříd



Příloha 14. Datový diagram



Příloha 15. Ukázka uživatelského rozhraní – SNMP souhrn



Příloha 16. Ukázka uživatelského rozhraní – zatížení routeru