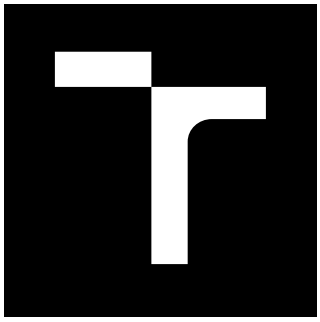
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2020 Dmitrii Scripnic



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

VYUŽITÍ OTEVŘENÝCH NÁSTROJŮ PRO MONITORING PŘÍSTUPOVÉ SÍTĚ
IMPLEMENTATION OF OPEN SOURCE SYSTEM FOR NETWORK MONITORING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Dmitrii Scripnic**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. David Grenar**
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Telekomunikační a informační systémy**

Ústav telekomunikací

Student: Dmitrii Scripnic ***ID:*** 177003
Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Využití otevřených nástrojů pro monitoring přístupové sítě

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je provést teoretický rozbor systémů s otevřenými zdrojovými kódy a realizovat jejich vyhodnocení, a to s ohledem na zaměření sledování okamžitých změn v síti. Na základě teoretického rozboru bude proveden návrh, instalace, konfigurace a ověření vybraného systému, následně pak měření jeho správné funkčnosti. Výstupem práce bude charakterizace jednotlivých monitorovacích systémů a pro vybraný systém vytvoření funkčního skriptu pro kontrolu zpoždění a vyhodnocování vybraných IP služeb.

DOPORUČENÁ LITERATURA:

[1] SPORTACK, Mark A. Směrování v sítích IP: [autorizovaný výukový průvodce: samostudium: kompletní zdroj informací o směrování a protokolech v sítích IP]. Brno: Computer Press, 2004. Cisco systems. ISBN 80 -25-01-7-4.

[2] LAFATA, Pavel a Jiří VODRÁŽKA. Optické přístupové sítě a přípojky FTTx. Praha: České vysoké učení technické v Praze, 2014. ISBN 978-800-1054-635.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 8.6.2020

Vedoucí práce: Ing. David Grenar

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

ABSTRAKT

Bakalářská práce se zabývá monitorováním sítě. V této práci byla provedena teoretická analýza známých monitorovacích systémů a také popsána jejich architektura, vlastnosti a práce s těmito systémy. Následně byla provedena srovnávací charakteristika všech popsaných systémů. Na začátku praktické části byla provedena instalace a konfigurace všech monitorovacích systémů a následně jejich vyhodnocení. Druhá část se zabývá monitorovacímu systému Zabbix, kde byl kladen větší důraz na sledování zpoždění IP služeb.

KLÍČOVÁ SLOVA

Monitorování sítí, SNMP, Zabbix, Nagios, Cacti, NetXMS, open-source, plugin, server, monitorovací systém, CPU, RAM, ping, skript

ABSTRACT

The bachelor thesis deals with network monitoring. In this work, a theoretical analysis of known monitoring systems was performed and their architecture, properties and work with these systems were also described. Subsequently, a comparative characteristic of all described systems was performed. At the beginning of the practical part, the installation and configuration of all monitoring systems and their evaluation was performed. The second part deals with the Zabbix monitoring system, where more emphasis was placed on monitoring IP service delays.

KEYWORDS

Network monitoring, SNMP, Zabbix, Nagios, Cacti, NetXMS,open-source, plugin, server, monitoring system, CPU, RAM, ping, skript

SCRIPNIC, Dmitrii. *Využití otevřených nástrojů pro monitoring přístupové sítě*. Brno, 2019, 55 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. David Grenar

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Využití otevřených nástrojů pro monitoring přístupové sítě“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu semestrální práce panu Ing.Davidu Grenarovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Seznam obrázků

Seznam tabulek

Úvod

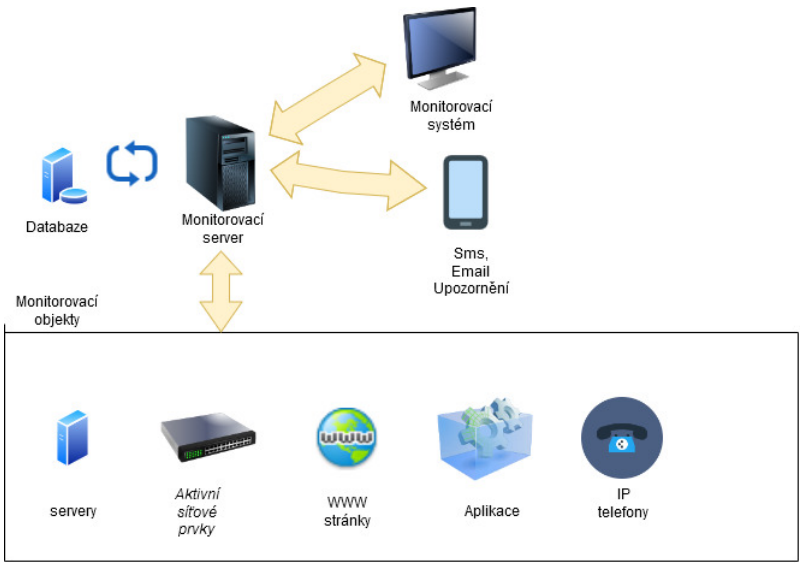
Moderní infrastruktura, ať už ve velkém, nebo malém podniku, vyžaduje neustálé monitorování. Dnes je téměř nemožné sledovat celou síť pro správce sítě bez zvláštních programů. Moderní síť vyžaduje nepřetržité monitorování. Vždy může nastat okamžik, kdy může dojít k poruše systému, což může vést ke ztrátě informací a také dokonce k poškození zařízení. To vše může přinést velké finanční ztráty. Předpokládejme, že máme podnik se 100 zaměstnanci. Společnost má svůj vlastní informační systém, do kterého musí zaměstnanci vstoupit pomocí svého uživatelského jména a hesla. Každý zaměstnanec má počítač, na kterém jsou nainstalovány důležité programy pro práci, také poštovní schránku a důležité informace, které neustále používá. Každý den užívají tiskárny, skenery a další síťová zařízení. Jednoho dne může dojít k selhání sítě a žádný zaměstnanec poté nebude moci smět vstoupit do informačního systému. V tomto okamžiku musí správce systému chybu rychle najít a opravit ji. Každá minuta pozastavení výroby v podniku, může vést k velkým finančním ztrátám. Je možné najít takovou chybu osobně, ale tímto způsobem by to trvalo déle, než kdyby měla firma nainstalovaný monitorovací systém, který by okamžitě automaticky chybu našel. Jedním z hlavních důvodů, proč je nutné monitorování, je okamžitá detekce. To znamená, že v případě síťového problému správce IT chybu co nejdříve najde a opraví. V tom mu pomáhají speciálně vyvinuté monitorovací systémy, které okamžitě reagují na jakékoliv změny v síti. Při výběru vhodného monitorovacího systému je třeba vzít v úvahu několik faktorů jako například:

- Síťová architektura – správce sítě musí vědět, co přesně chce sledovat.
- Funkce tohoto systému – systém musí mít nezbytné funkce pro sledování sítě např. vytvářeni grafu, oznámení o poruše v síti atd.
- Systémové náklady – existují placené i bezplatné systémy s otevřeným zdrojovým kódem. V malých sítích jsou dostačující bezplatné programy. Ve velkých sítích jsou obvykle zapotřebí monitorovací systémy s velkou funkčností. Tyto programy jsou zpravidla placeny.

Následující kapitoly podrobně popisují pojem monitorování sítě, monitorovací protokoly a jaké funkce by měl mít monitorovací systém. Také zde budou popsány open source systémy pro sledování, jako jsou: Zabbix, Nagios, Cacti, NetXMS. Cílem práce je porovnat vybrané monitorovací systémy a provést jejich teoretickou analýzu. Jejich funkce budou porovnány a hlavním účelem tohoto srovnání bude zjištění, jak systémy reagují na okamžité změny v síti. Na základě srovnání bude vybrán nejlepší monitorovací systém podle autora. Poslední praktická část této práce popisuje instalaci a konfiguraci vybraného systému.

1 Základy monitorovacích sítí

Monitorování je jedním z důležitých úkolů IT infrastruktury. Aby se předešlo selhání sítě, musí se neustále monitorovat. Monitorování se provádí u všech komponentů v síti a to počínaje směrovači, přepínači, mosty a konče různými aplikacemi. Proto aby byla zaručena funkčnost sítě, je třeba sledovat všechny její komponenty. V malých sítích, kde se síťová architektura sestavuje z několika komponentů, postačí osobní kontrola nad monitorováním. Zde v poměrně rychlém čase můžeme problém identifikovat a opravit ho sami. Ve velkých sítích, které se obvykle vyskytují ve velkých firmách, konvenční osobní monitorování nestačí, protože správce sítě bude muset trávit spoustu času identifikováním problému. Monitorovací systémy vytvořené vývojáři proto přicházejí k záchraně. Jedním z hlavních úkolů těchto systémů je okamžitá detekce a změny v síti. Jakákoli chyba a porucha v systému je okamžitě detekována a program informuje správce o poruše. Tímto způsobem pomáhá výrazně zkrátit čas správci IT na řešení problémů. Díky monitorovacímu systému má správce sítě všechny informace o stavu IT infrastruktury. V moderní síťové infrastruktuře se žádný podnik neobejde bez těchto systémů. Pomáhají společnosti ušetřit peníze. Tyhle programy však nejen rychle nalézají problémy v síti, ale také je mohou automaticky odstranit, nebo varovat předem o možném selhání konkrétního zařízení. To umožňuje síti pracovat nepřetržitě. V průběhu studia tohoto tématu nastává jedna z nejdůležitějších otázek. Který monitorovací systém zvolit? A který je z nich nejlepší? Následující kapitoly podrobně popisují, jaké monitorovací metody existují, které protokoly jsou odpovědné za monitorování a jaké nejlepší systémy zvolit.



Obr. 1.1: Monitorování síťové infrastruktury

1.1 Metody sledování sítí

Aktivní monitorování

Aktivní monitorování zahrnuje dotazování ohledně funkčnosti zařízení s určitou frekvencí. Obvykle se dotazování provádí formou odesílání paketů, aby se určila dostupnost samotných zařízení a služeb, které kontrolují aktuální stav zařízení, například procenta využití CPU a disku. Monitorovací systém například dotazuje server, aby získal hodnotu aktuálního zatížení procesoru. Server odpoví a pošle nějakou hodnotu zpět, jako třeba 90 %. Jakmile systém přijme hodnotu, porovná se s maximální přípustnou. Pokud má více než 90 %, měl by monitorovací systém automaticky upozornit sám. Můžeme tedy říci, že se jedná o aktivní monitorování. Hlavní nevýhodou aktivního monitorování je to, že výrazně ovlivňuje provoz.

Pasivní monitorování

Pasivní monitorování se používá ke sledování poruch systému a shromažďování různých dat v režimu čtení. Tato data mohou být: zatížení procesoru, spotřeba RAM paměti nebo teplota zařízení. Po shromáždění dat může operátor zobrazit přijaté informace. Formát zobrazení může mít podobu grafu nebo zprávy. Pasivní monitorování na rozdíl od aktivního nepřidává provoz do sítě a nemění provoz, který již v síti existuje. To může hrát důležitou roli pro síť, pokud není schopna zpracovat velké množství informací.

Kombinace aktivního a pasivního monitorování

Kromě čistě aktivního nebo pasivního monitorování jsou i metody využívající kombinace obou přístupů (vhodné například pro měření ztrátovosti paketů), metody zpracovávající data získaná z komponentů sítové infrastruktury (např. pomocí SNMP nebo protokolu Netflow)[1].

1.2 Funkce monitorovacích systémů

Moderní monitorovací systémy by měly sledovat vše, co se děje v síti. Měly by být schopny sledovat servery, kontrolovat soubory, složky, databáze a také vykonávat kontrolu nad procesy a službami v počítačích uživatelů. Sledování sítě vždy začíná objevem. Monitorovací systémy na začátku detekují všechna zařízení v dané síti, která zahrnují servery, směrovače, přepínače, brány firewall, tiskárny a další. Systém poté vytvoří síťovou mapu, kde každému detekovanému zařízení automaticky přiřadí roli. Poté jsou provedeny různé testy jako například příkaz ping, který kontroluje funkčnost spojení mezi dvěma síťovými uzly.

V případě problémů systém pošle oznámení e-mailem, telefonem nebo jiným způsobem. Poté se vytvoří smlouva o úrovni služeb SLA (Service Level Agreement). Podle SLA systém shromažďuje informace o kvalitě poskytování IT služeb. Níže jsou uvedeny nejdůležitější funkce, které by měl mít každý monitorovací systém sítě:

Monitorování sítí a aplikací Moderní nástroje pro monitorování a diagnostiku výkonu sítě umožňují sledovat objekty na všech úrovních síťové architektury. Je důležité mít představu o všech síťových komponentách, aby byl včas nalezen a opraven problém, který se objeví. Protokol, který pomáhá shromažďovat různá data ze síťových zařízení a serverů, se nazývá SNMP. Toto je hlavní protokol pro monitorování sítě. S ním můžeme nejen shromažďovat data, ale také sledovat stav různých hardwarů, napájecích zdrojů, využití paměti atd. Existují také monitorovací nástroje, které pomáhají přijímat a odesílat zprávy protokolu Syslog. Toto je protokol, který je standardem pro zprávy protokolu všech zařízení v síťové infrastruktuře. Tyto zprávy jsou přenášeny na server monitorovacího systému, který ukládá, analyzuje a upozorňuje správce IT o možném narušení běžného provozu systému[2].

Detekce problémů Jednou z hlavních funkcí monitorovacích systémů je detekce problémů. Neexistují žádné dokonalé systémy. V systému se mohou kdykoli vyskytnout poruchy nebo chyby. Pokud je v systému nějaký problém, je pro správce obtížné tuto chybu najít. Monitorovací systémy jsou dobré proto, že problém v síti najdou a upozorní na to správce.

Analýza problémů v síti Další důležitou funkcí je analýza problémů v sítí. Nestačí problém detekovat a předcházet mu. Je nutné zajistit, aby se tento problém v budoucnu neobjevil. Proto ve většině monitorovacích systémů existují takzvaní agenti, kteří shromažďují informace ze všech síťových uzlů a odesílají je na server. Tam mohou být data zobrazena ve formě grafu, tabulky nebo jiným způsobem oznámení. Díky tomu můžeme analyzovat stav sítě, porozumět tomu, jak toto nebo dané zařízení funguje, a také předcházet možným problémům v budoucnosti[3].

Hledání hlavní příčiny poruchy nebo problému Je jednou z časově nejnáročnějších funkcí, protože vyžaduje dokonalou konfiguraci všech zařízení a monitorovacích systémů. Pokud například není čas zařízení synchronizován pomocí protokolu NTP (Network Time Protocol), bude skutečný a zaznamenaný čas událostí odlišný. To může nepříznivě ovlivnit probíhající analýzu, což nakonec vede k nesprávnému závěru o hlavní příčině incidentu nebo problému. Pokud se však nainstaluje a nakonfiguruje, jednou automatizované nástroje pro nalezení kořenové příčiny problému ušetří obrovské množství času, které by jinak byly vynaloženy na řešení problémů.[2].

Upozornění Upozornění je pátá nejdůležitější funkce. Jakmile program zjistí problém nebo jakoukoliv jinou změnu v síti, okamžitě o tom informuje správce systému. To pomáhá okamžitě reagovat na jakékoli změny v síti.

2 Protokoly a nástroje pro sledování síťového provozu

Tato část popisuje protokoly, které se používají k řízení síťových zařízení.

2.1 SNMP

SNMP je standardní internetový protokol pro správu zařízení v IP sítích na základě architektur TCP / UDP. Existuje několik verzí SNMP. První verze SNMPv1 (RFC 1067, RFC 1057, RFC 1213) se objevila v roce 1988 a v současné době, i když je považována za zastaralou, je stále velmi populární. Během vývoje první verze se o bezpečnost skoro nikdo nestaral, takže v SNMPv1 nebyla žádná ochrana. Verze číslo 2 se objevila v dubnu 1993 jako SNMPv2 (RFC 1441-RFC 1452) a byla nekompatibilní s první verzí. Hlavními novinkami druhé verze protokolu byla výměna informací mezi řídicími počítači. Navíc se objevil nový příkaz, který přijímal několik proměn najednou (GetBulk). Bezpečnost však zůstávala velkým problémem. V roce 2004 se objevila třetí verze SNMPv3 (RFC 3411-3418), která přináší mnoho změn. Zabezpečení je však výrazně vylepšeno. Nyní každá zpráva obsahuje nastavení zabezpečení, která jsou kódována jako řetězec oktetu. V praxi je v implementacích SNMP často podporováno několik verzí: v1, v2c a v3[4].

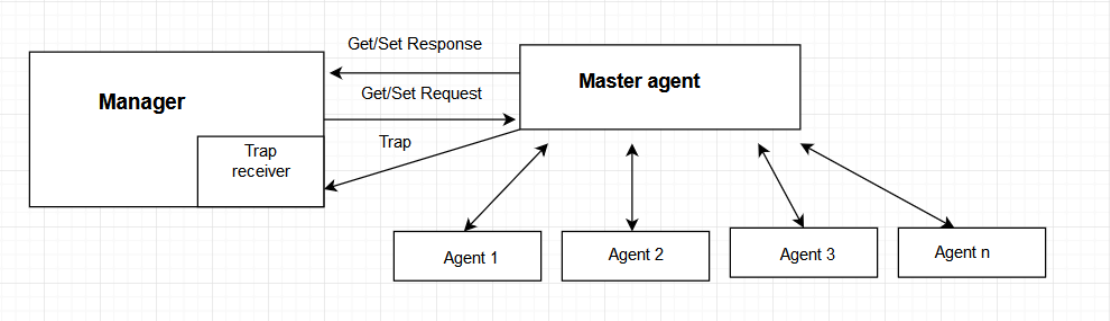
Architektura SNMP

Síť, která používá pro správu SNMP, obsahuje tři hlavní součásti:

- **Manager SNMP** – je to software nainstalovaný v počítači správce. Toto je klient, který žádá agenta, aby získal informace, které potřebuje. Funguje na portu UDP/162.
- **Agent SNMP** – Software běžící na síťovém uzlu, který je monitorován. Můžeme říci, že agent je určitá služba běžící na některém zařízení, které zpracovává požadavky na portu UDP/161. Agent shromažďuje informace o zařízení a předává je manažerovi.
- **SNMP MIB** - je manažerská informační základna. Tato součást systému poskytuje strukturovaná data vyměňovaná mezi agenty a manažery. Ve skutečnosti se jedná o druh databáze ve formě textových souborů. Agenti shromažďují informace o zařízení a zapisují shromážděná data do proměnných hodnot v databázi MIB. Tím je zpřístupněn manažerům.

Agenti a manažeři používají několik hlavních příkazů:

- **Trap** - jednosměrné oznámení od agenta SNMP k manažerovi o jakékoli události.
- **GetReponse** – odezva agenta na manažera, který vrací požadované hodnoty proměnné.
- **GetRequest** – požadavek agenta od manažera, který se používá k získání hodnoty jedné nebo více proměnných.
- **GetNextRequest** – požadavek agenta od manažera, který se používá k získání další hodnoty proměnné v hierarchii.
- **SetRequest** – požadavek agenta na nastavení hodnoty jedné nebo více proměnných.
- **GetBulkRequest** – žádost agentovi o přijetí pole dat (vyladěn GetNextRequest) (Tento příkaz byl představen v SNMPv2.).
- **InformRequest** - jednosměrné oznámení mezi manažery. Může být použit například pro výměnu informací o MIB. Jako odpověď generuje správce podobný balíček, jako potvrzení, že zdrojová data byla přijata[5].



Obr. 2.1: Princip komunikace protokolu SNMP

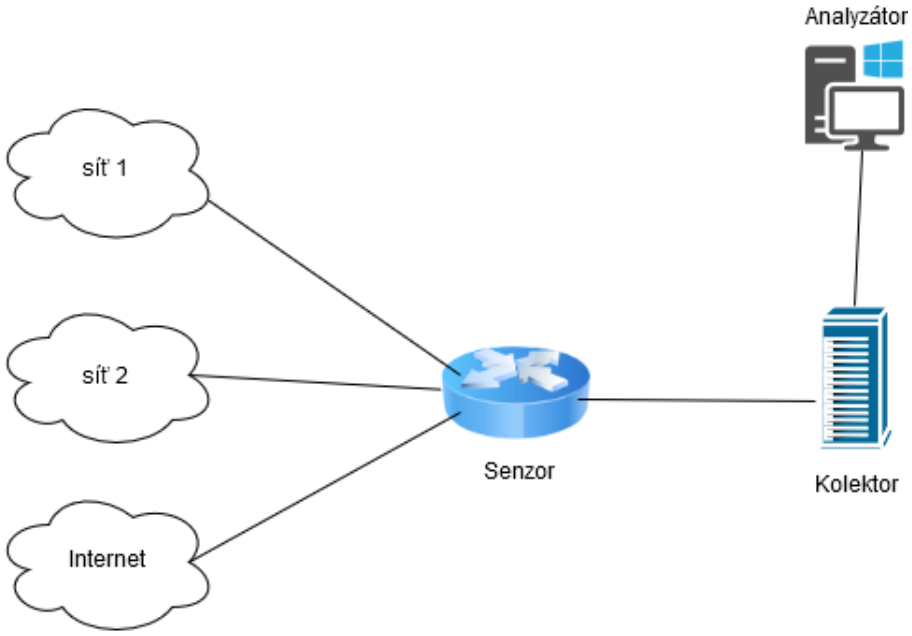
2.2 NetFlow

Netflow je protokol původně vyvinutý společností Cisco pro sledování síťového provozu. K dnešnímu dni tento protokol používá mnoho výrobců ve svých zařízeních. Existuje několik verzí protokolu Netflow a funkční rozdíly jsou mezi všemi verzemi velmi malé. Architektura Netflow se skládá ze tří částí: senzoru, kolektoru a analyzátoru. Senzor shromažďuje provozní data a přenáší všechny přijaté statistiky do kolektoru. Senzorem může být router nebo přepínač. Kolektor stejně jako senzor shromažďuje různé statistiky ze zařízení, přijímá data ze senzoru a ukládá je. Kolektor běží na serveru. Analyzátor, jak je již patrné z názvu, analyzuje a zpracovává data, která jsou v kolektoru. Poskytuje také různé zprávy, grafy atd. Kolektor shromažďuje veškeré informace z tzv. paketových toků[6].

Tok je skupina paketů, které obsahují:

- adresu odesílatele,
- adresu příjemce,
- zdrojový port (pro TCP (Transmission Control Protocol) nebo UDP (User Datagram Protocol)),
- port příjemce (pro TCP nebo UDP),
- typ a kód zprávy (pro ICMP),
- hodnotu pole ToS,
- rozhraní, na kterém se paket objevil.

Můžeme tedy říci, že proud je sada paketů, které se pohybují v jednom směru z bodu A do bodu B. Všechny shromážděné informace jsou zasílány ve formě záznamů, které mohou obsahovat takové parametry jako: číslo verze protokolu, číslo záznamu, příchozí a odchozí síťové rozhraní, čas začátku a konce proudu, počet bajtů a paketů v proudu, zdrojová a cílová adresa, zdrojový a cílový port, číslo protokolu IP. Aby se snížilo zatížení procesoru, lze použít „sampled NetFlow“. V tomto případě senzor neanalyzuje vše, ale každý n-tý paket, kde n může být nastaven správcem sítí, nebo náhodně. Při použití vzorkovaného NetFlow nebudou výsledné hodnoty přesné.



Obr. 2.2: Architektura Netflow

2.3 Diagnostické nástroje

Pro základní analýzu sítí jsou dostačující v operačních systémech integrované nástroje, jako jsou ping, iperf, Nmap atd.Tyto nástroje jsou nezbytné pro správce,

protože pomáhají rychle odhalit vzniklé problémy v sítí. Níže jsou popsané jedny z nejpoužívanějších nástrojů v operačních systémech Windows a Linux. [15]

2.3.1 Ipconfig

Příkaz ipconfig je jedním z nejčastěji používaných síťových nástrojů v systému Windows, protože umožňuje rychle a hlavně pohodlně prohlížet nastavení síťových adaptérů v systému Windows a provádět některé jednoduché, ale důležité úkoly související se správou počítače.V systému Linux je to příkaz ifconfig[2].

2.3.2 Nmap

Název Nmap je zkratka pro Network mapper. Samotný nmap je sada nástrojů pro skenování sítě. Může být použit k ověření bezpečnosti, jednoduše k určení služeb běžících v uzlu, k identifikaci operačního systému a aplikací, k určení typu brány firewall použité na skenovaném uzlu[17].

2.3.3 Iperf

Iperf je nástroj typu klient-server, který umožňuje měřit šířku pásma kanálu. Tento nástroj funguje tak, že pro měření propustnosti mezi dvěma uzly bude nutné spustit iperf na jednom uzlu v režimu server a na druhém uzlu v režimu klient[4].

2.3.4 Ping

Ping je diagnostický nástroj, který testuje spojení mezi dvěma uzly nebo zařízeními v síti. Nástroj ping nabízí dva hlavní účely: zkontrolovat, zda je hostitel přístupný, a změřit, jak dlouho bude odpověď trvat. Příkaz PING je jedním z nejčastěji používaných rozhraní příkazového řádku[4].

2.3.5 Fping

Fping je malý nástroj příkazového řádku pro odesílání „pingů“ do síťových uzlů podobných ping, ale mnohem produktivnější, když se pinguje více hostitelů. Fping je zcela odlišný od ping v tom, že na příkazovém řádku můžeme zadat libovolný počet hostitelů nebo zadat celý rozsah IP adres v rámci celé sítě pro ping.

3 QoS - Kvalita služeb v počítačových sítích

Qos(Quality of service) - je technologie pro rezervaci a řízení datových toků. Aby nedošlo ke snížení kvality síťových služeb, zajišťuje tato technologie pro každou službu dostatek prostředků v přenosovém kanálu. Jinými slovy lze říci, že každá IP služba se snaží doručit data od jednoho uzlu k druhému bez jakýkoliv problémů a o to se stará QoS. Proto tady jsou zavedené tzv. priority, kde provoz s vyšší hodnotou priority má vždy přednost před provozem s nižší hodnotou priority. Tato technologie se využívá především u služeb v reálném čase jako jsou VoIP, videokonference, online hry atd[4][17]. Mezi hlavními parametry QoS patří:

- latence [ms]
- jitter [ms]
- propustnost [kb/s]
- ztrátovost paketu[%]

3.1 Zpoždění

Zpoždění je doba po kterou trvá přenos datových jednotek od zdroje k cíli. Zpoždění může být jednosměrné, nebo obousměrné. Obousměrné zpozdění neboli RTT(Round Trip Time) se liší od jednosměrného zpoždění tím, že zahrnuje nejen čas potřebný ke zpracování a přenesení dat od zdroje k cíli, ale i čas potřebný ke zpracování a přenesení dat od cíle ke zdroji. Latence je jedním z nejdůležitějších parametrů pro služby v reálném čase(VoIP,Videokonference,on-line hry atd. hry)[4][17].

3.2 Jitter

Jitter neboli kolísání zpoždění můžeme chápat jako nerovnoměrnost časových intervalů pro doručování paketu od zdroje k cíli. Datové jednotky jsou vždy odesílány od zdroje k cíli v rovnoměrných intervalech a teoreticky by měly dojít k cíli ve stejných intervalech. Ale realita je trošku jiná. Proto jitter vzniká tehdy, když je síť zatížená nebo, když se odeslaný paket zdržel ve frontě atd. Jitter hraje klíčovou roli pro služby v reálném čase[2].

3.3 Propustnost

Propustnost je maximální rychlost přenosu datových jednotek pro zadanou cestu. Propustnost se obvykle měří v bitech za sekundu (bps), ale v reálných sítích jsou nejběžnějšími jednotkami kbps nebo Mbps. Obecně lze říci, že čím vyšší je propustnost,

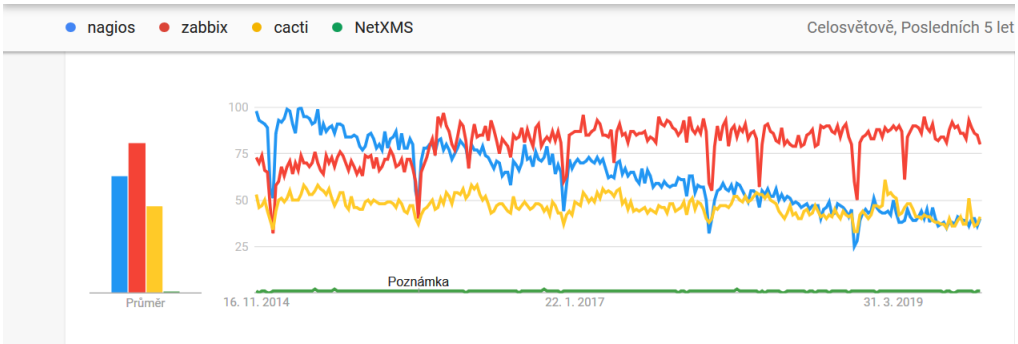
tím je lepší výsledná kvalita služeb.[4][17]

3.4 Ztrátovost paketů

Při odesílání paketů ze zdroje do cíle mohou být pakety ztraceny a nedosáhnout cíle. To může být způsobeno tím, že je síť zatížena, nebo jsou problémy se síťovým hardwarem atd. Ztráta paketů může ovlivnit jakoukoli aplikaci, ale ve většině případů ovlivňuje hlavně ty, které závisí na přenosu dat v reálném čase[2].

4 Přehled systémů

Monitorovací systémy kontrolují kromě vyhledávání chyb dostupnost služeb, zobrazují data o využití sítě, vytvářejí seznamy uzlů v síti, přistupují ke sdíleným složkám, zobrazují data při zatížení procesoru, RAM paměť, pevného disku, komunikačního kanálu atd. Jednotlivé programy v této kategorii mohou nejen shromažďovat statistiky všech procesů probíhajících v síti, ale také sledovat jejich kvalitu, analyzovat chyby a na základě shromážděných informací sestavit zprávu se závěry. Systémy monitorování můžou výrazně zjednodušit život správce systému, zejména pokud je pod jeho dohledem mnoho síťových zařízení. K dnešnímu dni byly vyvinuty desítky softwarových produktů, které umožňují sledovat síť. Všechny lze podmíněně rozdělit na placené a bezplatné. Ideální program však neexistuje. Každý produkt má své výhody a nevýhody. Správci proto zpravidla monitorují síť pomocí několika programů. Uvažujme některé z nich. V průběhu studia tohoto tématu byly vybrány nejznámější monitorovací systémy, jako jsou: Zabbix, Nagios, Cacti. Byl také vybrán systém NetXMS, který je méně využíván, aby bylo možné ho porovnat s jeho známějšími produkty. Statistiky požadavků uživatelů na Google za posledních 5 let jsou uvedeny níže.Je vidět, že v poslední době je nejoblíbenějším monitorovacím systémem Zabbix, zatímco NetXMS téměř nikdo nepoužívá.



Obr. 4.1: Statistika vyhledávání monitorovacích systémů v Google

4.1 Monitorovací systém Zabbix

Zabbix je bezplatný monitorovací systém pro sledování stavu různých počítačových síťových služeb, serverů a síťových zařízení. Zabbix je podnikové řešení s otevřeným zdrojovým kódem, které může provádět komplexní monitorování infrastruktury (servery, síťová zařízení a virtuální stroje). Také může vizualizovat přijaté informace v grafech, sledovat zatížení a výkon zařízení pomocí vlastních agentů, podporovaných téměř všemi operačními systémy.

4.1.1 Architektura

- **Monitorovací server** (jádro) - pravidelně dotazuje, přijímá data, zpracovává, analyzuje je a také spouští skripty pro zasílání výstrah. Může vzdáleně kontrolovat síťové služby. Jedná se o úložiště, ve kterém jsou uložena veškerá konfigurační, statistická a provozní data. Nelze jej umístit na server, který řídí operační systém rodiny Windows nebo OpenBSD.
- **Proxy** - shromažďuje údaje o výkonu a dostupnosti ze serveru Zabbix. Všechna shromážděná data se ukládají do vyrovnávací paměti místně a přenášejí se na server Zabbix, který vlastní proxy server. Zabbix proxy je ideálním řešením pro centralizované vzdálené monitorovací umístění poboček, sítí a místních správců. Může být také používán k distribuci zátěže jednoho serveru Zabbix. V tomto případě proxy shromažďuje pouze data, takže server načte menší zatížení procesoru a řídí jednotku dovnitř nebo ven.
- **Agent** je speciální démon, který běží na monitorovacích objektech a poskytuje data serveru. Řídí místní zdroje a aplikace, jako jsou pevné disky, paměť, statistika procesorů atd. V síťových systémech, tj. systémy by měly fungovat se spuštěným agentem Zabbix, monitorování však lze provádět nejen pomocí agentů, ale také prostřednictvím SNMP verze 1, 2, 3 spuštěním externích skriptů, které vydávají data, a několika typů předdefinovaných vestavěných kontrol, jako je ping, požadavek na http, ssh, ftp a další protokoly, stejně jako zpožděná doba odezvy pro tyto služby. Agenti Zabbix jsou velmi efektivní díky použití vestavěných systémových volání pro shromažďování statistik.
- **Webové rozhraní** - prostředky vizuální reprezentace Zabbix implementované pomocí PHP. Pro spuštění je nutné mít webový server[7].

4.1.2 Vlastnosti

- Monitorovací scénáře;
- Automatické vyhledávání;
- Centralizované monitorování protokolu;
- Webové rozhraní pro správu a konfiguraci;
- Podávání zpráv;
- Sledování SLA;
- Podpora vysoce výkonných agentů (zabbix-agent) pro téměř všechny platformy;

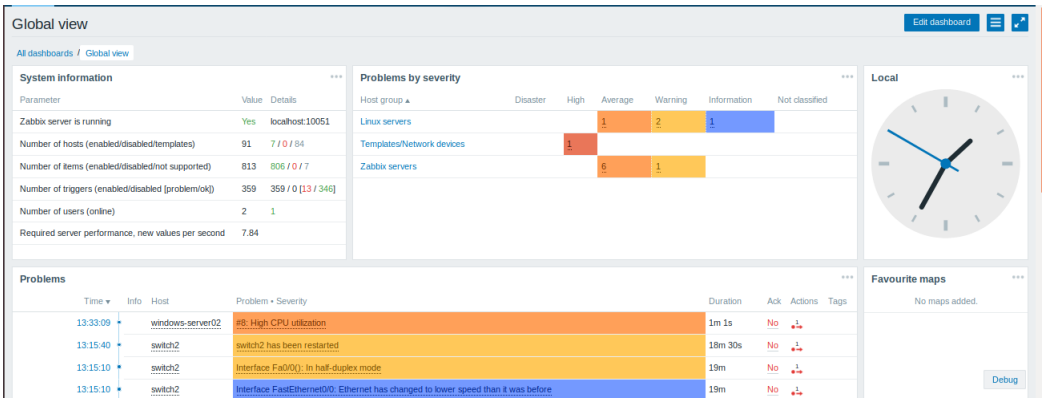
- Komplexní reakce na události;
- Podpora SNMP v1,2,3;
- Podpora sledování aplikací JMX;
- Rozšíření prostřednictvím provádění externích skriptů;
- Flexibilní systém šablon a skupin;
- Schopnost vytvářet síťové mapy[7].

4.1.3 Podporované platformy

Podporované platformy (server a agent): AIX, FreeBSD, HP-UX, Linux, Mac OS, OpenBSD, SCO OpenServer, Solaris, Tru64 / OSF. Kromě toho jsou implementováni agenti pro operační systémy Novell Netware a Windows[7].

4.1.4 Práce se systémem

Program můžeme nainstalovat z balíčků pro jednotlivé operační systémy nebo kompilací ze zdrojových kódů, které jsou na oficiálních webových stránkách produktu Zabbix. Instalace Zabbix trvá asi 30 minut. Po instalaci můžeme přidat vlastní síťové uzly, které chceme monitorovat nebo nakonfigurovat hotovou šablonu pro automatickou detekci potřebných uzlů. Po přidání a konfiguraci uzlů Zabbix začne sledovat. V případě jakýchkoli změn o tom informuje správce. Na obrázku níže je vidět, jak systém reaguje na změny v síti. Při velkém zatížení procesoru Zabbix okamžitě upozorní na tento problém. Také proběhlo restartování přepínače, takže systém nás o této události také okamžitě informoval.



Obr. 4.2: Okamžité změny v síti v Zabbix

4.2 Monitorovací systém Cacti

Cacti je open source monitorovací systém, který poskytuje pohodlné rozhraní pro RRDTool (zkratka pro round-robin databázový nástroj). S ním můžeme ovládat velké množství různých parametrů, jako jsou zaváděcí systémy a sítě s výstupem všech druhů grafů[8]. Cacti bude bez problémů pracovat na sítích všech velikostí, malých i velkých, se složitou topologií s rozvětveným řetězcem. Pro sběr dat lze použít libovolné externí příkazy nebo skripty s libovolnými parametry, které je potřeba shromáždit. V Cacti je implementována podpora SNMP. Rozhraní je napsáno v PHP jazyku, všechny shromážděné informace jsou uloženy v databázi MySQL. Hlavní součástí tohoto systému jsou grafy. Všechny kontrolované parametry a nastavení jsou vázány na graf. Cacti je licencován pod GNU GPL[9].

4.2.1 Architektura

Infrastruktura Cacti se skládá ze svazku Apache-PHP-MySql, kde první je server, na kterém běží systém. Druhý je platforma, na které je napsán Cacti, a na třetí je databáze, kde jsou uložena všechna nastavení tohoto programu. Podstata Cacti práce je následující: pomocí přidělených úkolů periodicky začíná dotazování agentů sběru dat ze zařízení, na kterých se provádí monitorování. Je to v podobě, že jsou shromažďovány v databázi RRD - tzn. v kruhových vyrovnávacích pamětích, jako jsou soubory. Dále lze nahromaděná data z RRD zobrazit prostřednictvím webového rozhraní Cacti, kde je lze zobrazit ve formě grafů v různých hodinových intervalech jako v minutách, hodinách, dnech, měsících atd[10].

4.2.2 Vlastnosti

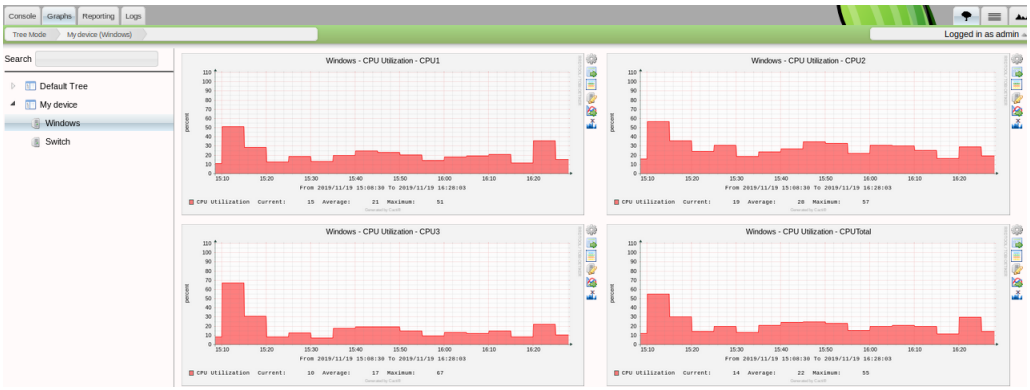
- Neomezený počet grafů,
- podpora automatického vyplňování grafů,
- manipulace s grafickými daty,
- flexibilní zdroje dat,
- skripty sběru uživatelských dat,
- podpora SNMP,
- grafické šablony,
- šablony zdrojů dat,
- šablony zařízení,
- správa a zabezpečení uživatelů a uživatelů,
- vzdálený sběr dat,
- objev sítě.

4.2.3 Podporované platformy

Cacti pracuje na všech Unix systémech, jako jsou BSD, Solaris, Linux atd, dále také i na Microsoft Windows.

4.2.4 Práce se systémem

Instalace trvá přibližně 30 minut. Po instalaci se můžeme přihlásit do systému. Zde si můžeme vytvořit své vlastní hosty, které chceme sledovat. Musíme také vybrat šablonu pro hosty, které jsou potřebné pro shromažďování různých informací ze zařízení. Cacti má několik standardních šablon: Cisco Router, Generic SNMP Device, Local Linux Machine, Net-SNMP Device a Windows Device. Pokud žádná z těchto šablon neodpovídá zařízení, které chceme sledovat, můžeme si vytvořit vlastní, nebo si stáhnout z oficiálních stránek produktu hotové šablony. Můžeme také vytvářet nové grafy pro zařízení. Grafy mají také funkci sledování v reálném čase. To nám umožní rychle reagovat na jakékoli změny v síti.



Obr. 4.3: Grafy v Cacti

4.3 Monitorovací systém Nagios

Nagios (původně Netsaint) je freewarový program pro monitorování síťových systémů. Pomocí tohoto programu je k dispozici komplexní monitorování IT infrastruktury, identifikace problémů bezprostředně po výskytu, schopnost sdílet získané informace, sledování bezpečnosti systému a v důsledku toho zkrácení času potřebného pro komerční ztráty. Instalace a konfigurace konfiguračního souboru Nagios trvá dost dlouho (přibližně hodinu pro zkušeného správce Systému Unix). Všechna nastavení monitorování jsou také prováděna v konfiguračních souborech a na zařízeních se musí nakonfigurovat SNMP, protože tento systém funguje podle tohoto protokolu.

Jednoduchá architektura rozšiřujících modulů (plaginu) umožňuje použití libovolného programovacího jazyka (Shell, C ++, Perl, Python, PHP a další) pro snadný vývoj vlastních způsobů kontroly služeb, je vydáván pod GPL licencí[11].

4.3.1 Architektura

Nagios se skládá ze dvou hlavních složek. První a nejdůležitější součástí systému Nagios je server, který pracuje téměř na všech unixových systémech. Hlavním úkolem serveru je zpracovat přijatá data od agentů nebo jiných externích programů, upozorní nás také na selhání systému. Druhou komponentou v architektuře Nagios je agent. Agenti sledují výkon serverů a systémových zařízení a přenášejí přijaté informace na server Nagios. Nagios má modulární architekturu s rozšiřitelností. To znamená, že Nagios používá pluginy (Nagios pluginy) a rozšíření (Nagios addony), což může výrazně zvýšit funkčnost tohoto systému. Pluginy se používají k získání volného místa na disku, teploty zařízení, doby odezvy vzdáleného hostitele atd. Je také možné vytvořit si vlastní pluginy. Pojem „rozšíření“ (addon) byl zaveden, aby nebyl zaměňován s pluginy, protože jsou rozšíření navržena tak, aby vytvářela nové funkce, nebo integrovala s jinými produkty[11].

4.3.2 Vlastnosti

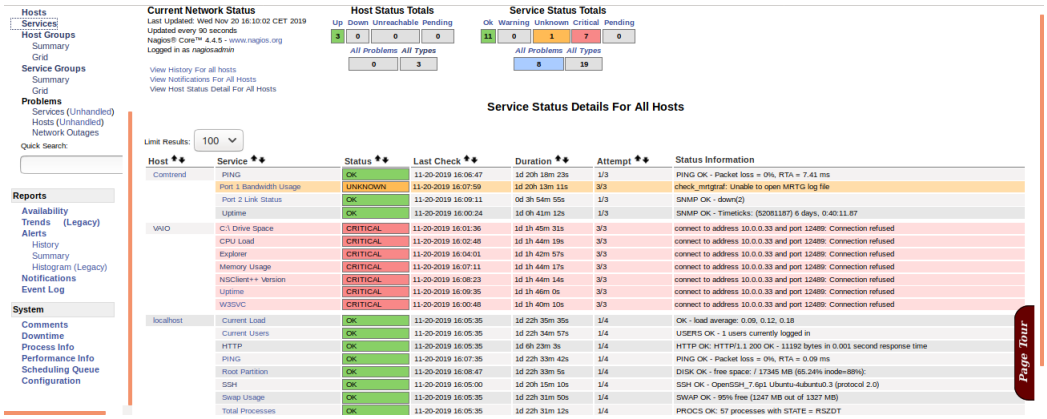
- Monitorování síťových služeb,
- sledování stavu hostitelů (načítání procesoru, použití disku, systémové protokoly) ve většině síťových operačních systémů,
- podpora vzdáleného monitorování pomocí šifrovaných tunelů SSH nebo SSL,
- schopnost vytvářet síťové mapy,
- jednoduchá architektura rozšiřujících modulů (plug-inů) umožňuje použití libovolného programovacího jazyka (Shell, C ++, Perl,27Python, PHP a další),
- schopnost určit hierarchii hostitelů pomocí „nadřazených“ hostitelů umožňuje detekovat a rozlišovat mezi hostiteli, kteří jsou mimo řád a kteří nejsou k dispozici,
- zasílání oznámení v případě problémů se službou nebo hostitelem (pomocí pošty, pageru, SMS nebo jakékoli jiné metody určené uživatelem prostřednictvím systému),
- možnost organizovat společnou práci několika monitorovacích systémů s cílem zvýšit spolehlivost a vytvořit distribuovaný monitorovací systém,
- zahrnuje obslužný program nagiostats, který zobrazuje souhrn všech sledovaných hostitelů[12].

4.3.3 Podporované platformy

Původně navržený pro systémy Linux, nyní funguje stejně dobře jako v systémech Sun Solaris, FreeBSD, AIX a HP-UX.

4.3.4 Práce se systémem

Nagios je považován za výkonný monitorovací nástroj a je určen pro monitorování ve velkých sítích. Instalace a konfigurace trvá o něco déle než Cacti nebo Zabbix. Kromě samotného serveru musíme nainstalovat i pluginy Nagios. Nagios má jasné webové rozhraní, kde můžeme vidět nejen počet monitorovaných hostitelů, ale také síťovou mapu, kterou Nagios vytváří automaticky. Na webovém serveru Nagios nemá funkci „přidat hostitele“ (jako v Cacti nebo Zabbix). Chceme-li na server přidat hostitele, musíme napsat skript. Nagios však již má připravené šablony pro takové skripty, kde stačí přidat pouze jméno hostitele, kterého potřebujeme, jeho IP adresu, a přidat jej do skupiny, kam se budou přidávat stejné typy hostů pro lepší přehled v případě větších sítí, kde je spousta těchto hostitelů. Budeme také muset psát procesy, které bude Nagios sledovat, například: zatížení procesoru, dostupnost síťového portu atd. Po přidání hostitele, skupiny a procesů je třeba tento skript uložit do konfiguračních souborů Nagios. Tento hostitel by se měl objevit ve webovém rozhraní. Nagios reaguje velmi rychle na změny v síti. V případě problému o tom okamžitě informuje správce. Na obrázku níže je vidět, jak se v případě odpojení od hostitele ve sloupci stav objeví „Kritický“ stejně jako informace o problému.



Obr. 4.4: Reagování Nagios na okamžité změny v síti

4.4 Monitorovací systém NETXMS

NetXMS je software s otevřeným zdrojovým kódem pro monitorování počítačových sítí. Používá se ke sledování celé IT infrastruktury. Je vydán pod licencí GNU v2. Systém má modulární architekturu a umožňuje snadné rozšíření funkčnosti a také možnost instalace na jakékoli platformě[14].

4.4.1 Architektura

NetXMS má třístupňovou architekturu: informace shromažďují monitorovací agenti (agenti NetXMS nebo agenti SNMP) a dodávají se monitorovacímu serveru ke zpracování a uložení. Správce sítě má přístup ke shromážděným datům pomocí konzole pro správu napříč platformami, webového rozhraní nebo konzole pro správu systému Android[13].

4.4.2 Vlastnosti

- Automatické zjišťování sítě
- sběr dat buď prostřednictvím SNMP, nebo prostřednictvím nativního agenta NetXMS,
- konfigurace sběru dat na základě šablony pro zjednodušenou správu velkých sítí,
- Sada šablon akcí (například ukončení nebo restartování konkrétního procesu pro jakýkoli počítač založený na Windows nebo Linuxu, restartování pro jakýkoli typ zařízení atd.),
- podpora pro SNMP verze 1, 2c, 3,
- má webové rozhraní,
- shromažďuje data o výkonu serveru,
- data jsou uložená pro následnou analýzu,
- automaticky detekuje nové hostitele a síťová zařízení,
- upozorňování správce na problémy prostřednictvím e-mailu nebo SMS[13].

4.4.3 Podporované platformy

Jsou podporovány následujícími platformami: Windows, Linux, Solaris, AIX, HP-UX, FreeBSD

4.4.4 Práce se systémem

Instalace se téměř neliší od instalace monitorovacích systémů uvedených výše. Na oficiálních webových stránkách produktu je dokumentace, která popisuje instalaci

programu. Můžeme nainstalovat webový server i administrační konzole. Také jako v jiných monitorovacích systémech musíme na začátek vytvořit uzel, který musíme sledovat. Nejprve musíme v „Infrastructure Services“ vytvořit kontejner s libovolným názvem (například switch). Poté můžeme vytvořit uzel a zadat zobrazovaný název a jeho IP adresu. Automatické zjišťování uzlů můžeme také povolit na kartě „Configuration-Network Discovery“. Tato funkce je velmi užitečná, jelikož je v síti mnoho síťových uzlů a jejich přidání ručně by vyžadovalo spoustu času. Poté můžeme pro tento uzel vytvořit šablonu. Chceme-li například shromáždit informace o době odezvy uzlu (ping), musíme přidat nový parametr. V poli „Origin“ zvolíme interní, a vybereme „Ping Time“. Můžeme také změnit dobu dotazování uzlu (ve výchozím nastavení je to 60 sekund, tj. pokud router přestane reagovat okamžitě poté, co byl dotazován monitorovacím systémem, bude to trvat téměř minutu, než systém zjistí, že s ním něco není v pořádku) a poté zazní alarm.

5 Srovnávací charakteristika monitorovacích systémů

V průběhu studia monitorování sítě byly vybrány nejznámější a nejpoužívanější monitorovací systémy, jako jsou Nagios, Zabbix, Cacti, NetXMS. Níže je uvedena tabulka pro srovnávání charakteristik monitorovacích systémů. Podle tabulky mají skoro všechny systémy stejné funkce. Porovnáním funkcí monitorovacích systémů, jejich výkonu a složitosti konfigurace můžeme dojít k závěru, že nejvhodnějším systémem by mohl být každý ze čtyř produktů. Každý z výše uvedených produktů má své výhody i nevýhody. Cacti a NetXMS se snadno nastavují. Mají však menší funkčnost než Nagios a Zabbix. Cacti nemá vestavěný varovný systém a k tomu budeme muset nainstalovat další pluginy. Nagios má skvělou funkčnost, ale všechna nastavení se provádějí v konfiguračních souborech v terminálu (neexistuje způsob, jak přidat zařízení z webového rozhraní a vytvářet samostatné procesy) a je také třeba restartovat službu Nagios, aby se nastavení aktualizovala. Nevýhodou NetXMS je malá komunita. Pro vyřešení problému bude nalezení řešení trvat dlouho.Lze poznamenat, že Zabbix je obtížné konfigurovat. Ale vzhledem k tomu, že Zabbix má docela velkou komunitu a spoustu dokumentace, je nastavení velmi rychlé. Z výše uvedeného můžeme shrnout, že Cacti a NetXMS jsou vhodné pro začátečníky, kteří teprve začínají monitorovat síť, zatímco Nagios a Zabbix jsou vhodné spíše pro zkušenější uživatele. Všechny výše uvedené produkty mají zároveň téměř stejnou funkčnost, takže si můžeme vybrat jakýkoli monitorovací systém. Autor však dává přednost Zabbix, protože tento konkrétní monitorovací systém má na internetu mnoho informací, což nám umožní lépe se s tímto produktem seznámit. Tento systém také může rychle reagovat a upozornit nás na jakékoli změny v síti. Proto je Zabbix vhodný pro další studium tohoto tématu. V průběhu studia monitorovacích systémů a na základě výše popsaných funkcí vybraných produktů byla sestavena tabulka, kde byly vybrány hlavní parametry, které měl moderní monitorovací systém mít. Mezi ně patří: podpora agentů, použití pluginů, vytváření síťových map atd.

Tab. 5.1: Srovnávací charakteristika systémů

Název	**Grafy**	**Zprávy**	**SLA**	**Syslog**	**SNMP**	Pluginy	Mapy	Alarmy	Agent	Licence
Zabbix	Ano	Ano	Ano	Ano	Ano	Ano	Ano	Ano	Ano	GNU/ GPL
NetXMS	Ano	Ano	Ano	Ano	Ano	Ano	Ano	Ano	Ano	GNU/ GPL
Cacti	Ano	Ano	Přes plugin	Ano	Ano	Přes plugin	Ano	Ano	Ano	GNU/ GPL
Nagios	Ano	Přes plugin	Přes plugin	Přes plugin	Ano	Ano	Ano	Ano	Ano	GNU/ GPL

Dalším důležitým kritériem je spotřeba RAM paměti, zatížení procesoru každým systémem. Server má parametry: Intel Core 2 Duo, HDD 100 Gb, 4 GB RAM, OS Ubuntu 18.04.3 LTS. Všechny testovací systémy monitorování nasazené na server v kontejnerech. Najednou jsou spuštěny kontejnery spojené pouze s jedním monitorovacím systémem. V tab. 5.2 ukazuje hodnoty získané během testování sledovaných monitorovacích systémů.

Tab. 5.2: Spotřeba systémových prostředků

System	**CPU** [%]	**RAM** [%]	**Počet přidaných uzlů**[-]	**Počet přidaných metrik**[-]	**Interval sběru dat**[min]
Zabbix	15	31	100	500	5
Nagios	17	30	100	500	5
Cacti	11	26	100	500	5
NetXMS	13	24	100	500	5

6 Praktická část

6.1 Instalace systémů

Tato část nebude popisovat úplnou instalaci monitorovacích systémů, pouze hlavní části. Budou nainstalovány monitorovací systémy Cacti, Nagios, Zabbix, NetXMS. Podrobný popis instalace vybraných monitorovacích systémů je uveden níže v literatuře[20][21][22][23].

Pro instalaci všech monitorovacích systémů byl vybrán operační systém Ubuntu 18.04.3 LTS, který je nainstalován ve virtualizovaném prostředí VitrualBox VM. Během procesu instalace systém požádá o zadání nezbytných údajů pro úspěšnou instalaci systému:

- vybrat zemi a region;
- zvolit klávesnici a rozložení;
- pojmenovat počítač;
- dát uživatelské jméno;
- nastavit heslo;
- vybrat časové pásmo;
- vybrat disk a způsob označení disku;
- nastavení proxy serveru;
- výběr aktualizace;
- výběr role serveru;
- výběr zavaděče systému.

Po instalaci operačního systému můžeme začít instalovat monitorovací systémy. Instalace všech monitorovacích systémů je přibližně stejná. Budeme muset nainstalovat webový server Apache, protože webové rozhraní je napsáno v PHP pro všechny monitorovací systémy (s výjimkou NetXMS, který je napsán v Javě), takže k jeho spuštění budeme potřebovat webový server s podporou PHP. A budeme také potřebovat databázi. Byla vybrána Mysql databáze, protože je nejsnadnější s ní pracovat. K instalaci produktů používáme příkaz:

```
apt-get install
```

Podrobnosti o instalaci webového serveru a databáze můžeme nalézt na internetu a na oficiálních webových stránkách monitorovacích systémů v sekci instalace programu. Také je nutné nastavit správné datum a čas. To lze provést rychle pomocí příkazu:

```
dpkg-rekonfigure tzdata
```

Budeme potřebovat správné datum a čas pro správné zobrazení statistik v monitorovacích systémech a pro jejich bezchybný provoz. Poté můžeme začít instalovat naše monitorovací systémy. Protože instalace všech monitorovacích systémů je přibližně stejná, budou zde popsány hlavní instalační kroky bez jakýchkoli podrobností. Podrobné informace o instalaci a popis každého jednotlivého instalačního kroku můžeme najít na oficiálních stránkách monitorovacích systémů se všemi potřebnými dokumenty. Pro instalaci vybraných produktů použijeme postupně příkazy:

```
wget a tar - zxvf
```

První příkaz stáhne vybraný soubor. Chceme-li to provést, po příkazu wget musíme přidat odkaz s úplnou cestou k tomuto souboru. Například:

```
wget https://assets.nagios.com/downloads/nagioscore/
releases/nagios-4.4.5.tar.gz
```

Druhý příkaz daný soubor rozbalí:

```
tar - zxvf
```

V dalším kroku náš soubor nakonfigurujeme pomocí příkazu:

```
./configure
```

Jak je tento příkaz správně použit v jednotlivých instalacích monitorovacích systémů, je popsán na oficiálních stránkách těchto produktů. Příkaz:

```
 make install
```

nainstaluje soubory, jako jsou server nebo agent monitorovacího systému, po konfiguraci souboru. Po stažení a instalaci produktu můžeme vytvořit databázi. Pomocí příkazu:

```
mysql -u root -p <heslo>
```

přejdeme do databáze Mysql a vytvoříme novou databázi a uživatele:

```
CREATE DATABASE <název databáze>;
CREATE USER 'jméno' @ '%' IDENTIFIED BY 'heslo␣databáze';
GRANT ALL PRIVILEGES ON <database>. * TO 'jméno' @ '%';
```

poslední příkaz znamená, že vytvořenému uživateli je uděleno oprávnění pro vytvořenou databázi. V dalším kroku musíme upravit konfigurační soubor serveru monitorovacího systému v sekci „Database“ a změnit hodnoty na nové, které byly vytvořeny ve výše uvedeném kroku. Poté zbude spustit všechny služby, jako jsou server, agent a webový server Apache. Také upravíme soubor webového rozhraní monitorovacího systému a v části „date.timezone“ musíme uvést správné časové pásmo.

Chceme-li získat přístup k webovému rozhraní nainstalovaného monitorovacího systému, musíme do prohlížeče napsat:

```
http://ip-adresa serveru/název systému.
```

Při prvním otevření webového rozhraní je nutné jej nakonfigurovat. Jak nakonfigurovat webové rozhraní, je na oficiálních stránkách systémů.

6.2 Přidání zařízení a šablon

Po instalaci všech monitorovacích systémů můžeme začít monitorovat síť. Chceme-li zkontrolovat, jak systém reaguje na změny v síti, budeme muset přidat zařízení a také šablony. Ke každému monitorovacímu systému bude přidáno několik zařízení (např. router a počítač s operačním systémem Windows).
V Nagiosu neexistuje funkce pro přidávání zařízení ve webovém rozhraní. Chceme-li přidat zařízení, musíme otevřít konfigurační soubor, který již existuje v konzole, a přidat hotovou šablonu pro přidání hostitele. Otevřeme soubor pomocí příkazu:

```
vi /usr/local/nagios/etc/objects/windows.cfg
```

a do souboru přidáme svá zařízení. Po přidání nového zařízení musíme také přidat procesy, které chceme sledovat, jako je využití procesoru, ping, volné místo na disku C atd. Poté v konfiguračním souboru Nagios, který je umístěn v

```
/usr/local/nagios/etc/nagios.cfg
```

pak odkomentujeme řádek

```
cfg_file = / usr/local/nagios/etc/objects/windows.cfg
```

a soubor uložíme. Pak je třeba zkontrolovat chyby u všech změn, které jsme vytvořili. Chceme-li to provést, použijeme příkaz:

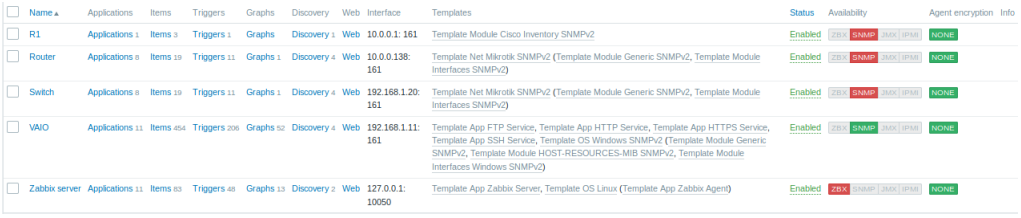
```
/usr/local/nagios/bin/nagios -v/usr/local/nagios/etc/nagios.cfg
```

Pokud neexistují žádné chyby, restartujeme služby Nagios. Nyní můžeme přejít na webové rozhraní Nagios a otevřít kartu „Hosts“. Tam by se měl objevit náš přidaný hostitel. Na kartě „Services“ můžeme vidět, které služby byly přidány pro tohoto hostitele. Nyní přidejme hostitele do Cacti. Zde je přidání mnohem jednodušší než v Nagiosu. V hlavní nabídce vybereme možnost „Create new device“. Do pole „Host name“ přidáme IP adresu našeho zařízení.

Do pole „Device Templates“ přidejme potřebnou šablonu. V našem případě vyberme šablonu „Templates Windows“. V části „SNMP Version“ zvolme verzi protokolu. Musíme také napsat identifikátor do pole „SNMP Community“, které nám umožní přístup ke statistikám zařízení. Zbývající pole lze ponechat beze změny.

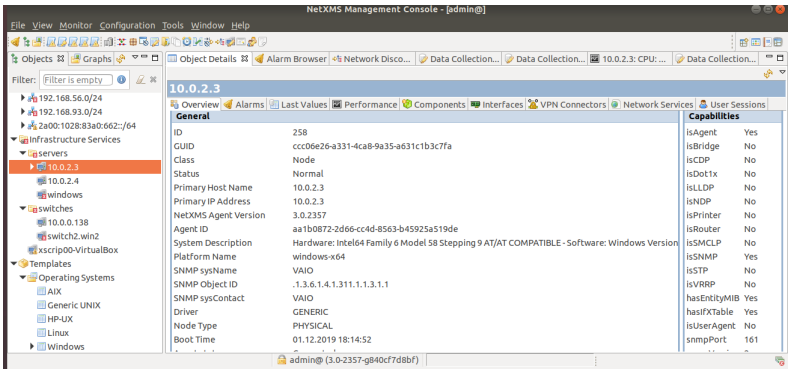
Obr. 6.1: Příklad přidání hostitele v Cacti

V Zabbixu je proces přidání hostitele přibližně stejný jako v Cacti. V hlavní nabídce vyberme „Configuration — Hosts — Create Host“. Do pole „Host name“ napíšeme název našeho zařízení. V poli „Groups“ napíšeme název skupiny, ve které bude tento hostitel umístěn. Pak si můžeme vybrat, pomocí čeho bude Zabbix shromažďovat informace o zařízení, v závislosti na tom, který agent je na zařízení nainstalován. V našem případě zvolíme monitorování pomocí SNMP a napíšeme IP adresu zařízení, na kterém je agent nainstalován. Musíme také přidat šablonu pro naše zařízení. Vybereme připravenou šablonu „Template OS Windows“ a klikneme na tlačítko „Add“. Nyní, když jsme přidali zařízení, klikneme znovu na tlačítko „Add“. Zařízení bylo úspěšně přidáno. Pokud bylo vše provedeno správně a agent byl na zařízení nainstalován, měl by SNMP v sekci „Hosts“ v poli přidaného zařízení svítit zeleně. To znamená, že agent začal svou práci na shromažďování informací ze zařízení.



Obr. 6.2: Seznam hostitelů v Zabbix

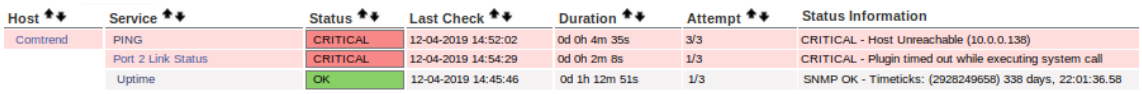
Nejjednodušší použitelný monitorovací systém byl NetXMS. Programové rozhraní je pro uživatele velmi pohodlné a srozumitelné. Existují dva způsoby, jak přidat zařízení: automatické a manuální. V automatickém režimu stačí nastavit pouze rozsah IP adres a systém sám detekuje všechna dostupná zařízení v tomto rozsahu. Přidáme zařízení manuálně. Musíme kliknout pravým tlačítkem a vybrat možnost „Create – Node“. V okně, které se otevře, napíšeme název zařízení a jeho IP adresu a klikněme na OK. Nyní musíme do tohoto zařízení přidat šablony. Zde můžeme buď vytvořit vlastní šablonu, nebo použít stávající. Přejdeme na kartu „Templates“ a vyberme šablonu v závislosti na tom, které zařízení bylo přidáno. V našem případě, když přidáváme počítač s OS Windows, vybereme šablonu s názvem Windows a pravým tlačítkem myši otevřeme „Data Collection Configuration“. Poté vybereme šablony, které potřebujeme, a zkopírujeme je do našeho zařízení. Tím je přidávání zařízení dokončeno.



Obr. 6.3: Webové rozhraní NetXMS

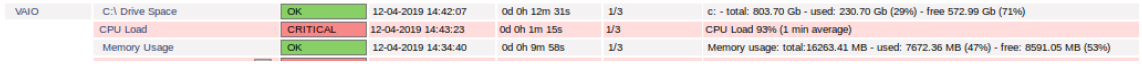
6.3 Testování systému

Jakmile jsou zařízení přidána do monitorovacího systému, můžeme začít sledovat jejich správnou funkci. Hlavním cílem této práce bude zjistit, jak systém reaguje na okamžité změny v síti, a pak určit, který systém nejrychleji detekuje okamžité změny v síti. Aby bylo zkontrolováno, jak rychle Nagios reaguje na změny v síti, bylo přidáno několik zařízení, jako jsou router a počítač, s operačním systémem Windows. Do zařízení byly přidány šablony jako je ping, Uptime (doba provozu), načítání CPU, využití RAM paměti atd. Poté byl tento monitorovací systém testován zatížením procesoru Windows zařízení a odpojením routeru od sítě. Přibližná doba pro detekci těchto chyb v Nagiosu byla 2 minuty a 30 sekund.



Host	Service	Status	Last Check	Duration	Attempt	Status Information
Comtrend	PING	CRITICAL	12-04-2019 14:52:02	0d 0h 4m 35s	3/3	CRITICAL - Host Unreachable (10.0.0.138)
	Port 2 Link Status	CRITICAL	12-04-2019 14:54:29	0d 0h 2m 8s	1/3	CRITICAL - Plugin timed out while executing system call
	Uptime	OK	12-04-2019 14:45:46	0d 1h 12m 51s	1/3	SNMP OK - Timeticks: (2928249658) 338 days, 22:01:36.58

(a) Odpojeni routeru od síti

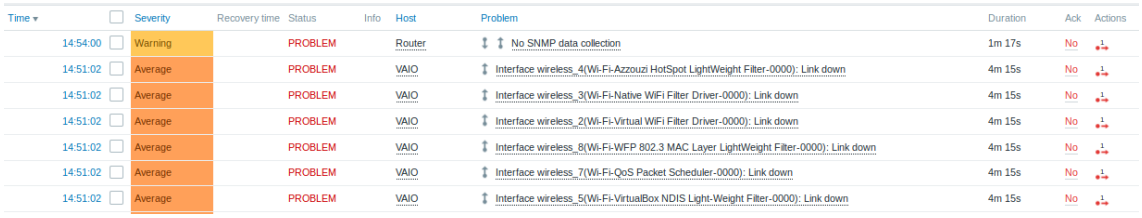


VAIO	C:\ Drive Space	OK	12-04-2019 14:42:07	0d 0h 12m 31s	1/3	c: - total: 803.70 Gb - used: 230.70 Gb (29%) - free 572.99 Gb (71%)
	CPU Load	CRITICAL	12-04-2019 14:43:23	0d 0h 1m 15s	1/3	CPU Load 93% (1 min average)
	Memory Usage	OK	12-04-2019 14:34:40	0d 0h 9m 58s	1/3	Memory usage: total:16263.41 MB - used: 7672.36 MB (47%) - free: 8591.05 MB (53%)

(b) Zatížení procesoru

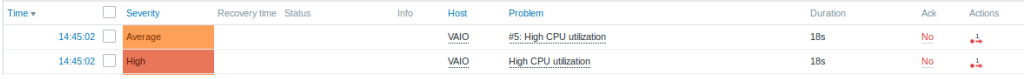
Obr. 6.4: Okamžité změny v Nagios

Do monitorovacího systému Zabbix bylo přidáno také několik zařízení jako v Nagios. Pro testování bylo také nutné přidat potřebné šablony (viz předchozí kapitola). Přibližná doba detekce poruchy zařízení byla 2 minuty a 10 sekund.



Time	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack	Actions
14:54:00	Warning		PROBLEM		Router	No SNMP data collection	1m 17s	No	
14:51:02	Average		PROBLEM		VAIO	Interface wireless_4(Wi-Fi-Azzouzi HotSpot LightWeight Filter-0000): Link down	4m 15s	No	
14:51:02	Average		PROBLEM		VAIO	Interface wireless_3(Wi-Fi-Native WiFi Filter Driver-0000): Link down	4m 15s	No	
14:51:02	Average		PROBLEM		VAIO	Interface wireless_2(Wi-Fi-Virtual WiFi Filter Driver-0000): Link down	4m 15s	No	
14:51:02	Average		PROBLEM		VAIO	Interface wireless_8(Wi-Fi-WFP 802.3 MAC Layer LightWeight Filter-0000): Link down	4m 15s	No	
14:51:02	Average		PROBLEM		VAIO	Interface wireless_7(Wi-Fi-QoS Packet Scheduler-0000): Link down	4m 15s	No	
14:51:02	Average		PROBLEM		VAIO	Interface wireless_5(Wi-Fi-VirtualBox NDIS Light-Weight Filter-0000): Link down	4m 15s	No	

(a) Odpojení routeru v Zabbix

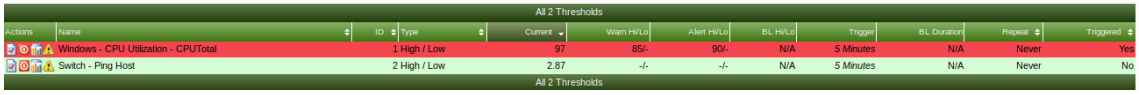


Time	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack	Actions
14:45:02	Average				VAIO	#5: High CPU utilization	18s	No	
14:45:02	High				VAIO	High CPU utilization	18s	No	

(b) Zatížení CPU v Zabbix

Obr. 6.5: Okamžité změny v Zabbix

Poté byla zařízení přidána do systému Cacti. A také byl nainstalován plugin Thold, který je potřebný pro upozornění na změny v síti. Po instalaci pluginu musíme přidat šablony zařízení. Přejděme na kartu „Management – Thresholds“ a klikněme na znaménko plus v pravém horním rohu. Poté přidáme naše zařízení a potřebnou šablonu a klikneme na tlačítko „Create“. Poté můžeme vyzkoušet naše zařízení. Budeme testovat stejně jako v předchozích systémech (zatížení procesoru a odpojení routeru od sítě). Když byl procesor zatížen na více než 90 %, systém nás o tom informoval. Přibližná doba detekce problému byla 2 minuty a 40 sekund.

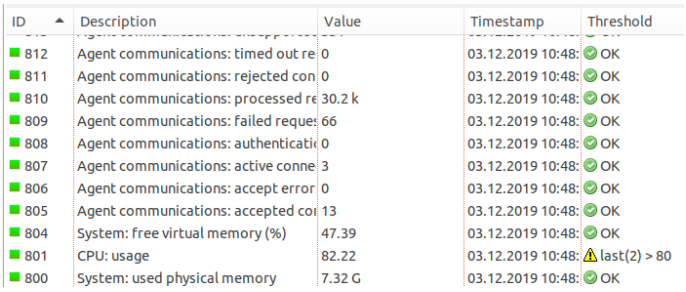


(a) Zatížení procesoru v Cacti

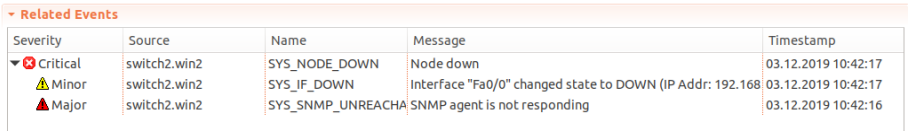
(b) Odpojení routeru v Cacti

Obr. 6.6: Okamžité změny v síti v Cacti

Byl testován poslední monitorovací systém NetXMS. Po přidání zařízení a šablon (viz předchozí kapitola) můžeme začít testovat. Směrovač byl opět odpojen od sítě a procesor byl také zatížen. Pak do 3 minut nás systém varoval o selhání těchto zařízení.



(a) Zatížení procesoru NetXMS



(b) Odpojení routeru v NetXMS

Obr. 6.7: Okamžité změny v NetXMS

6.4 Výsledky měření

Po instalaci a konfiguraci všech produktů byla přidána zařízení se šablonami a všechny monitorovací systémy byly testovány. Byly nainstalovány nejnovější verze všech systémů. Měření bylo provedeno 20krát za sebou. To bylo provedeno za účelem ověření, jak tyto systémy budou reagovat v případě opakovaných problémů a pak zjistit průměrný čas detekce problému. Na všech monitorovacích systémech byl také nastaven čas sběru informací ze zařízení na 5 minut. Protože po prostudování vybraných produktů bylo zjištěno, že nejlepší volbou je 5 minut, protože systém nevyžaduje velké prostředky a také stačí pro naše účely, pokud nás systém informuje až 5 minut po výskytu problému. Z dvaceti měření bylo zjištěno, že Zabbix byl nejrychlejší monitorovací systém, zatímco NetXMS byl pomalejší než ostatní systémy. Níže můžeme vidět výsledky měření.

Tab. 6.1: Výsledky měření

Název	Čas sběru dat[min]	Počet měření[-]	Využití RAM[%]	CPU[%]	Rychlost detekce problému[min]
Zabbix	5 min	20	25	12	2,1
Nagios	5 min	20	25	14	2,3
Cacti	5 min	20	22	10	2,4
NetXMS	5 min	20	23	11	3

7 Práce s monitorovacím systémem Zabbix

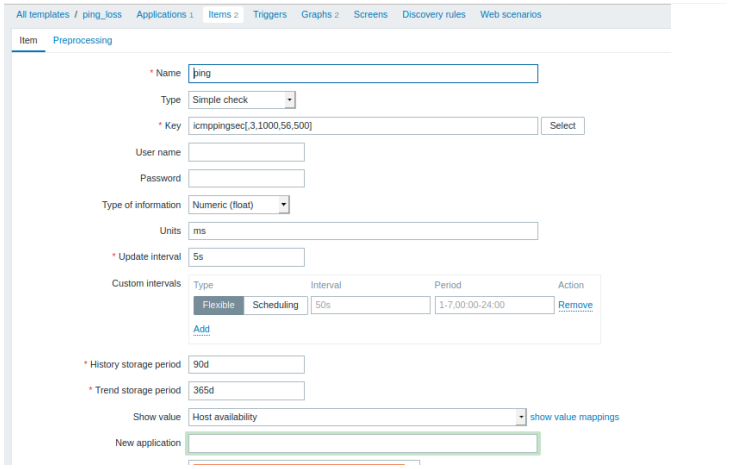
Jak již bylo popsáno výše, Zabbix má velké množství funkcí. Tento monitorovací systém má nejen už dokončené šablony, ale také má možnost si vytvářet i své vlastní. Především je tato funkce užitečná, když potřebujeme zkontrolovat například nějakou službu, která není zadaná v hotové šabloně. Tato kapitola popisuje proces vytváření nové šablony pro kontrolu zpoždění (ping) a proces vytváření šablony pro ztrátu paketů. Pro zpracování ICMP ping Zabbix používá externí nástroj *fping*, který není součástí distribuce Zabbix. Bude ho tedy nutné dodatečně nainstalovat[24].

7.1 Vytvoření šablony

Chceme-li vytvořit novou šablonu, přejdeme na kartu *Configuration — Templates*. V pravém horním rohu vyberme funkci *Create Template*. V zobrazeném okně musíme vyplnit další dvě pole. Budeme muset napsat název šablony a vybrat si skupiny, pro které bude tato šablona použita. V našem případě pojmenujeme šablonu *ping loss*. Dále také přidáme potřebné skupiny, kde jsou umístěni hostitelé, které chceme sledovat. Následně přidáme 4 skupiny, jako jsou: Discovered hosts, Linux servers, routers, Zabbix servers. Tento proces dokončíme uložením naší nové šablony kliknutím na tlačítko *Add*[27].

7.2 Vytvořeni datové položky pro zpoždění

Vytvořili jsme si šablonu, ale zatím je prázdná. To znamená, že v této šabloně nejsou žádná data a neexistuje žádná služba, kterou chceme sledovat. Proto musíme k tomu přidat i službu. V našem případě ji vytvoříme pro sledování zpoždění (ping). Na kartě *Templates* vybereme naši novou šablonu. V okně, které se objeví nahoře, zvolme funkci *Items — Create item*.



Obr. 7.1: New item

Otevře se nové okno pro vytvoření nové datové položky. Do pole *Name* zadáme název našeho datového prvku. V poli *Type* vybereme název *Simple check*. Jednoduché kontroly se používají, pokud nechceme používat agenty jako např. Zabbix agent, SNMP atd. Proto server Zabbix, který je zodpovědný za zpracování jednoduchých kontrol, přebírá sám veškerou práci. Dále v poli *Key* si vybereme službu, kterou budeme potřebovat, jelikož chceme sledovat dobu zpoždění na zařízení, tak si vybereme funkci:

```
icmppingsec [target,packets,interval,size,timeout,mode]
```

Tato funkce vrací čas odezvy ping pomocí protokolu ICMP. Funkce má šest parametrů:

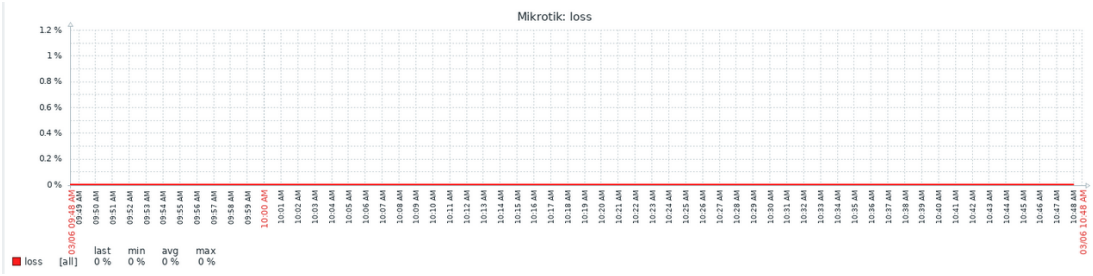
- target - IP hostitele nebo jeho DNS jméno, které chceme sledovat;
- pakety - počet paketů;
- interval - doba mezi přenosem úspěšných paketů v milisekundách;
- size - velikost paketu v bajtech;
- timeout - timeout v milisekundách;
- režim - jeden z min, max, avg (výchozí).

Ve výchozím nastavení byly nastaveny následující hodnoty:

- počet paketů - 3;
- interval - 1000 ms;
- velikost paketu: 56 bytů;
- zpoždění - 500 ms.

V dalším poli *Type of information* vybereme *Numeric (float)*, jelikož chceme získávat desetinná čísla. Protože chceme získat hodnoty v milisekundách, zapíšeme *ms* do pole *Units*. V poli *Update interval* nastavíme čas aktualizace na 5 sekund. Zbývající pole lze ponechat beze změny, pro naše účely nám budou tyto nastavené parametry

stačit. Ve výchozím nastavení si Zabbix uchová historii 90 dnů. Nyní můžeme přidat pomocí tlačítka *Add* datový prvek, který jsme vytvořili. Předpokládejme, že chceme nejen zkontrolovat ping, ale také i ztrátu paketů. Chceme-li toto provést, vytvoříme další datový prvek. Princip vytvoření datového prvku pro ztrátu paketů je stejný jako pro ping[28].



Obr. 7.2: Ztráta paketů

7.3 Vytvoření datové položky pro ztrátu

Znovu vytvoříme a pojmenujeme další nový datový prvek. Pojmenujeme ho například *loss*. V poli *Type* vybereme možnost *Simple check*. Protože chceme zkontrolovat ztrátu paketů, vybereme v poli *Key*, že chceme sledovat dobu zpoždění zařízení, vybereme z dostupných služeb

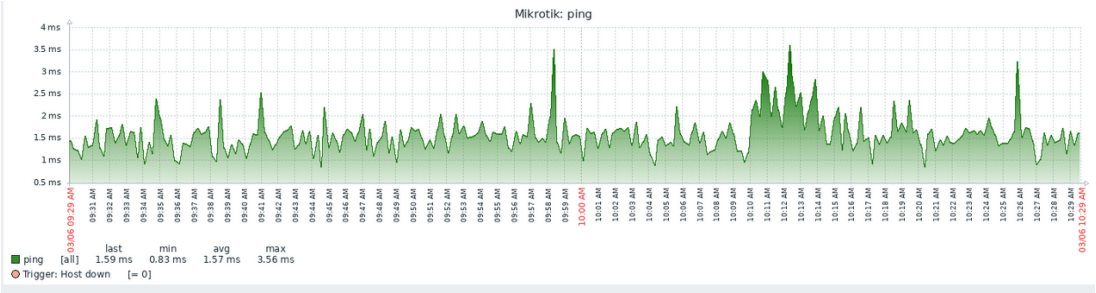
```
icmppingloss [target,packets,interval,size,timeout]
```

V poli *Type of information* vybereme *Numeric(unsigned)*. To znamená, že chceme získat hodnoty bez znaménka. Do pole *Units* napíšeme, co se bude měřit. Ztráta paketu se měří v procentech, takže napíšeme znak %. V poli *Update interval* znovu nastavíme hodnotu na 5 sekund. Zbývající pole ponecháme beze změny a klikneme na tlačítko *Add* .

7.4 Vytvoření grafu pro zpozdění a ztrátu

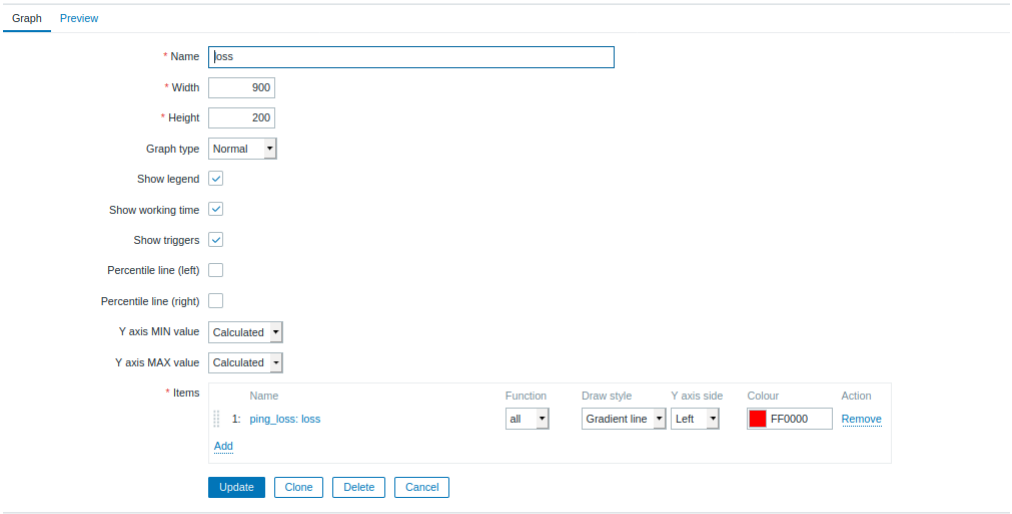
Nyní jsme vytvořili dvě datové položky. Jedna sleduje zpoždění, druhá sleduje ztrátu paketů. Abychom však usnadnili analýzu získaných informací, musíme také přidat grafy.

Znovu vybereme šablonu *ping_loss*, kterou jsme vytvořili dříve, v horní části obrazovky vybereme funkci *Graphs—Create Graph*.



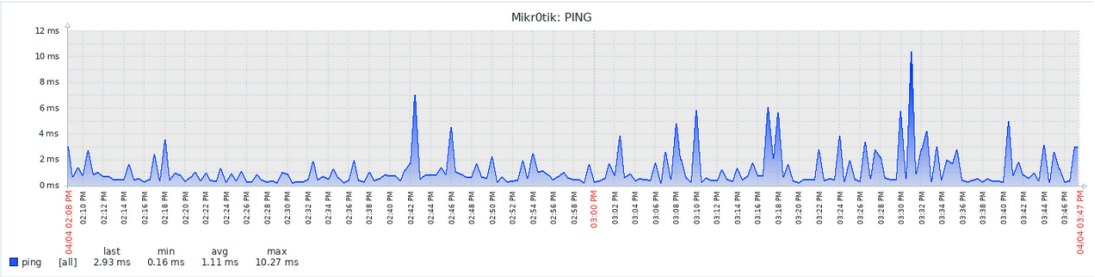
Obr. 7.3: Vytváření grafu ping

Nejprve vytvoříme graf pro zpoždění. Do pole *Name* napíšeme název našeho nového grafu (např. *ping*). V poli *Graph type* vybereme *Normal*. V poli *items* klikněme na tlačítko *Add* a vybereme datový prvek *ping*, který jsme vytvořili. Poté vybereme všechny funkce, které chceme v grafu vidět. V nastavení můžeme také zvolit styl nebo barvu grafu. Zbytek můžeme nechat beze změny a stisknout tlačítko *Add*. Podobně také přidáme graf pro ztrátu paketů. Pouze v poli *items* již vybereme datový prvek *loss* a přidáme náš graf[25].

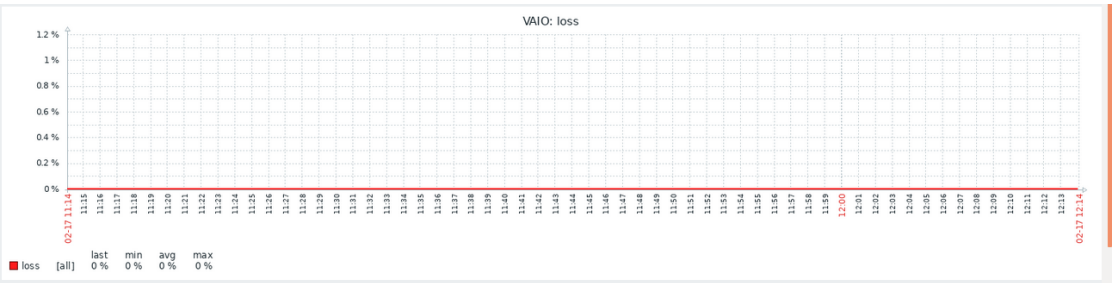


Obr. 7.4: Vytváření grafu loss

Chceme-li zobrazit naše vytvořené grafy, přejděme na kartu *Monitoring—Graphs*. Vybereme hostitele, kterého potřebujeme, a vybereme grafy, které jsme vytvořili. Na obrázku níže můžeme vidět graf zpoždění a ztráty paketů.



Obr. 7.5: Graf zpoždění

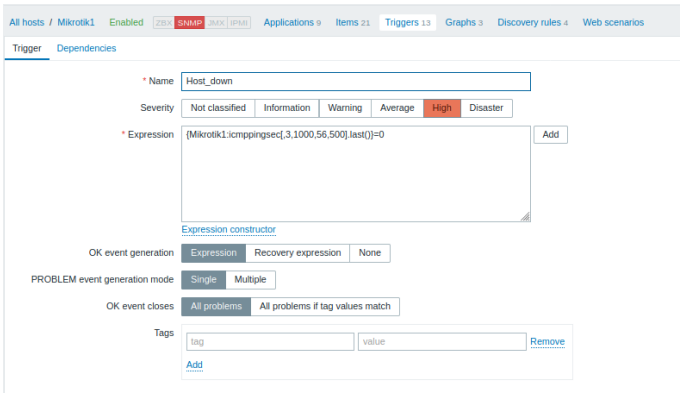


Obr. 7.6: Graf ztráty paketů

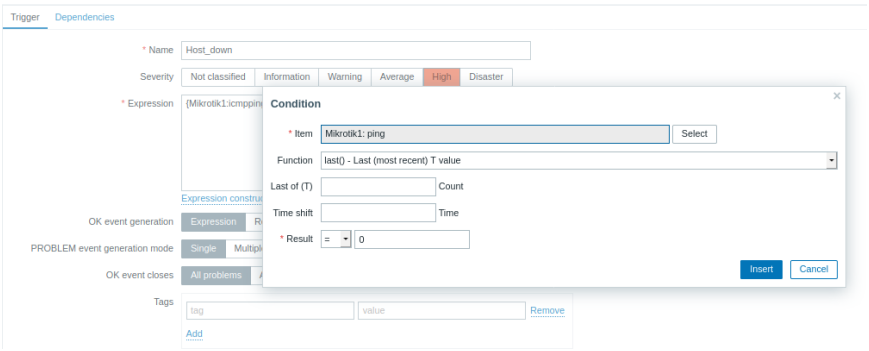
7.5 Vytvoření spouštěče

Byly vytvořeny datové prvky, které shromažďují data ze zařízení. Také byly vytvářeny grafy, díky nimž je možné shromažďovaná data zobrazit v grafické podobě. Nyní musíme vytvořit spouštěč, který nás upozorní na možný problém (například pokud hostitel není k dispozici). Spouštěče jsou užitečné v tom, že vyhodnotí přijatá data, a pokud jsou správně nakonfigurovány, automaticky detekují problém. Jinými slovy, spouštěče jsou logické výrazy, které fungují na principu True-False. Pokud bude výraz nepravdivý, pak to znamená, že zařízení je v normálním a funkčním stavu. Pokud bude naopak výraz pravdivý, znamená to, že se se zařízením něco pokazilo a že se zde vyskytuje nějaká chyba. Když jsou v síti stovky až tisíce hostitelů, znamená to, že spouštěče hrají důležitou roli při správném fungování sítě. Spouštěč lze vytvořit dvěma způsoby. Pokud jej vytvoříme přímo v šabloně, použije se na všechny hostitele, kteří tuto šablonu používají. Můžeme si ale také vytvořit individuální spouštěč pro samostatného hostitele zvlášť, kterého chceme sledovat. Tento samostatný spouštěč vytvoříme tak, že přejdeme do záložky *Configuration—Hosts*

a vybereme hostitele, pro kterého chceme spouštěč udělat. V nastavení hostitele zvolíme sekci *Triggers-New trigger*.



Obr. 7.7: Vytvoření spouštěče



Obr. 7.8: Trigger expression

Do pole *Name* zadáme název spouštěče. V poli závažnosti nastavíme spouštěcí úroveň závažnosti. V poli *Expression* klikneme na tlačítko *Add*. V zobrazeném okně vybereme datovou položku, kterou bude spouštěč kontrolovat. V našem případě vybereme náš datový prvek ping. V dalším poli názvem *Function* musíme určit, co bude spouštěč následně kontrolovat, jelikož chceme zkontrolovat nepřístupnost hostitele, zvolíme si tuto funkci:

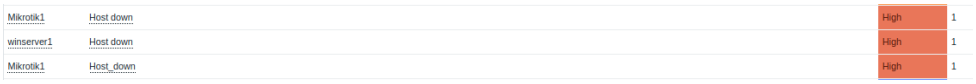
```
last () -Last (last) T value
```

. V poli Výsledek vybereme znaménko rovnosti a stiskněme tlačítko Insert. Výsledkem je následující výraz:

```
{Mikrotik: icmppingsec [, 3,1000,56,500] .last ()} = 0
```

- Mikrotik - název hostitele;
- icmppingsec [, 3,1000,56,500] - datový prvek, který bude spouštěč kontrolovat;
- last () - zkontroluje poslední hodnotu.

Spouštěč je následně připraven a kliknutím na tlačítko *Add* jej přidáme do své vytvořené předchozí šablony[26]. Na obrázku níže lze vidět příklad fungování spouštěčů, když host není k dispozici a neodpovídá, spouštěč nám to automaticky oznámí sám.



Obr. 7.9: Host down

7.6 Vytvoření skriptu pro kontrolu zpozdění

Dalším způsobem, jak zkontrolovat latenci Ping, je vytvoření vlastního skriptu. Zabbix může pracovat s externími skripty, které nejsou součástí distribuční sady monitorovacího systému. Stačí zadat cestu k vytvořeným skriptům v konfiguračním souboru Zabbix. Poté Zabbix uvidí všechny vytvořené skripty v sekci, která byla zaregistrována v konfiguračním souboru, a díky tomu může uživatel tyto skripty používat přímo v samotném monitorovacím systému Zabbix.

Pro kontrolu zpoždění byl napsán jednoduchý bash skript.

Výpis 7.1: Skript pro kontrolu zpožděni

```
#!/bin/bash
pinghost=$1

ping_results=$(ping $pinghost -c 60 | awk -F',|/'
'/rtt/{print␣x␣"␣"␣$5}'| sed 's/[^0-9␣.]*//g');

if [ -z '$ping_results' ];
then
ping_results="0000"
echo $ping_results
else
echo $ping_results
fi
```

Vstupním parametrem je IP adresa zařízení. Daný skript nám vrátí průměrnou hodnotu zpoždění v milisekundách. Pokud odpověď neobdržíme, tak skript nám vrátí výsledek 0000. Vytvořený skript uložíme do souboru *Externalscripts* a v konfiguračním souboru Zabbix zapíšeme cestu k tomuto skriptu.

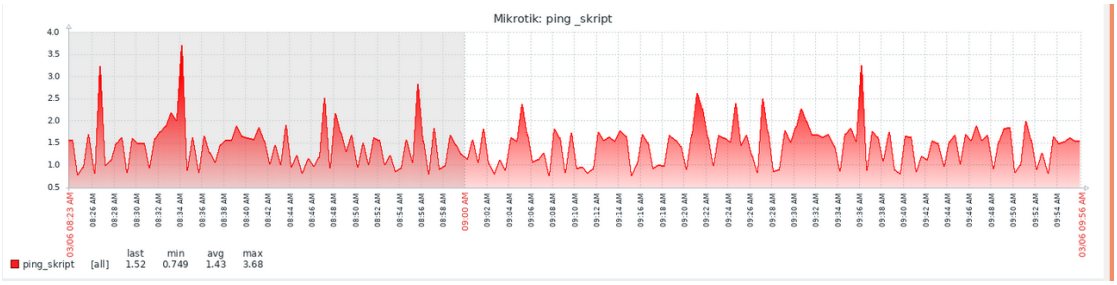
```
### Option: ExternalScripts
#       Full path to location of external scripts.
#       Default depends on compilation options.
#       To see the default path run command "zabbix_server --help".
#
# Mandatory: no
# Default:
#ExternalScripts=${datadir}/zabbix/externalscripts
ExternalScripts=/usr/local/share/zabbix/externalscripts

### Option: FpingLocation
#       Location of fping.
#       Make sure that fping binary has root ownership and SUID flag set.
#
# Mandatory: no
# Default:
 FpingLocation=/usr/local/sbin/fping
```

Obr. 7.10: Konfigurační soubor Zabbix

Také musíme pro vytvořený skript nastavit práva. Uděláme to pomocí dvou příkazů:

```
chown root:zabbix /usr/local/share/zabbix/externalscripts/
ping1.sh
chmod 550  /usr/local/share/zabbix/externalscripts/ping1.sh
```

Pak restartujeme Zabbix server. Nyní musíme vytvořit šablonu, datový prvek a graf. Proces vytváření šablony a grafů byl popsán výše. Proto okamžitě přistoupíme k vytvoření datového prvku. Pojmenujeme znovu naši novou datovou položku (např. Ping_script). V poli *Type* vyberme *External check*. Do pole *Key* napíšeme název našeho skriptu a do něj přidáme parametr *HOST.CONN*, který bude nahrazen názvem zařízení nebo jeho IP adresou. Do pole *Units* znovu zapíšeme *ms*, protože hodnoty dostáváme v milisekundách. Přidáme naši novou datovou položku do šablony. Dále vytvoříme graf stejným způsobem, jak je popsáno výše. Hotovou šablonu přidáme do zařízení, které chceme sledovat[29].



Obr. 7.11: Vytváření grafu ping

7.7 Měření zpoždění

K vyhodnocení výkonu sítě podle doporučení IETF RFC 2544[30] nebo ITU-T Y.1564[31] se nejčastěji používají čtyři hlavní parametry:

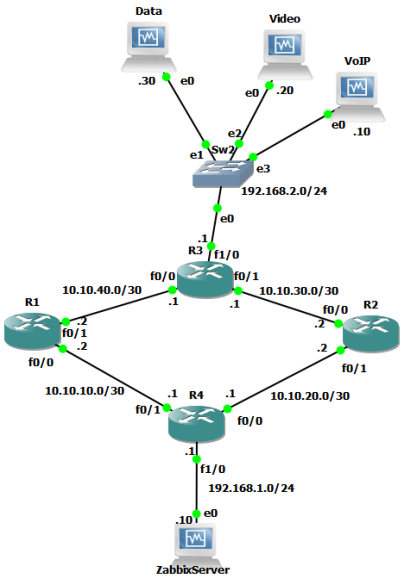
- Propustnost;
- Zpoždění;
- Jitter;
- Ztráta paketu.

Tyto parametry ovlivňují kvalitu poskytovaných služeb, jako jsou VoIP, videokonference, online hry atd. Bez podrobností chci poznamenat, že čím vyšší je propustnost, tím lepší je kvalita služeb. Čím menší je zpoždění, ztráta jitteru nebo paketu, tím lepší je kvalita služeb. Pro zlepšení kvality služeb byly zavedeny mechanismy QoS jako: DiffServ, Int-Serv, MPLS. Podrobný popis těchto mechanismů je nad rámec tohoto textu.

Tab. 7.1: Parametry QoS

Služba	Zpoždění[ms]	Jitter[ms]	ztrátovost[%]
Voip	150-400	<20	<5
Video	4-5 s	<50	<5
Data	nezáleží	nezáleží	nezáleží

V této části praktické práce byla vytvořena topologie sítí v simulačním prostředí GNS3(vzhledem k malému počtu fyzických síťových prvků bylo rozhodnuto, si vytvořit topologii v dánem simulačním prostředí). V dáne topologii byly použité routery s OS Cisco IOS a také jeden směrovač. Jako koncové zařízení byly použity virtuální počítače. Následně byla provedena základní konfigurace všech síťových prvků, jako přidělení IP adresy, směrovacího protokolu(byl zvolen protokol OSPF) atd. Na obrázku můžeme vidět navrženou topologii.



Obr. 7.12: Topologie v simulačním prostředí GNS3

V rámci bakalářské práce byl vytvořen skript pro měření zpoždění. Ve skriptu byl použit nástroj Ping, který používá protokol ICMP pro odesílaní paketů. Výsledkem skriptu bude průměrné obousměrné zpoždění tzv. RTT(round-trip time) zpoždění za 1 minutu. Test bude trvat 15 minut pro každou službu.Čas měření byl zvolen na základě přečtení odborné literatury a po prostudování doporučení jako jsou IETF RFC 2544 [30] nebo ITU-T Y.1564[31].Podrobný popis těchto doporučení lze nalézt na internetu. Měření budou prováděna pro zatíženou, nezatíženou síť a také pro jednotlivé služby (VoIP, Data, Videostream). Výsledkem měření bude porovnání jednotlivých služeb.Zatížení sítě bude simulováno pomocí programu VLC, kde bude spuštěn videostream, a také stažením velkého souboru ze serveru a uskutečněním telefonního hovoru pomocí aplikace X-lite. Všechny služby proto budou spuštěné najednou v jeden okamžik. Prvním krokem bylo změřit zpoždění, když nebyla spuštěna žádná služba. U nezatížené sítě bylo průměrné zpoždění 1,56 ms. Pomocí aplikace X-lite byl uskutečněn telefonní hovor. Aplikace používá kodek G.711, který vytváří datový tok 64 kbit / s. Při použití této služby bylo průměrné zpoždění 3,58 ms. Pro tuto službu zpoždění by nemělo překročit hodnotu 150 ms, jinak by došlo k výraznému zhoršení kvality hlasu. V tomto případě bylo zpoždění této služby velmi nízké, a proto byla kvalita přenášeného hlasu na velmi dobré úrovni. Pro testování zpoždění videa byl použit program VLC. Na straně serveru byl generován datový tok pomocí kodeku Xvid standardu MPEG-4. Rychlost přenosu videa byla přibližně 6400 kbit/s. Průměrné zpoždění při použití videostreamu bylo 5,45 ms. Poslední testovanou službou byla Data. Byl stažen velký soubor. Průměrné zpoždění bylo

10,23 ms. Pro simulaci zatížené sítě byly spuštěny všechny služby současně. Tímto měřením se zpoždění během testovaného času silně zvýšilo.
To by mohlo vést k velkému jitteru, což je pro službu VoIP nežádoucí. Průměrné zpoždění se výrazně zvýšilo a činilo 105,3 ms. Níže je uvedena tabulka s výsledky měření.

Tab. 7.2: Výsledky měření zpoždění

Služby	Zpoždění[ms]
VoIP	3,58
Video	5,45
Data	10,23
Nezatížená	1,56
Zatížená	105,3

7.8 Vyhodnocení měření

Během testování bylo zjištěno, že zpoždění ovlivňuje většinu služeb v reálném čase (VoIP, videostream, online hry atd.). U Data zpoždění neovlivnilo kvalitu služby ani rychlost stahování. Důležitějším parametrem pro tuto službu je šířka pásma. Ze všech testovaných služeb mělo VoIP minimální zpoždění, protože tato služba vytváří malý datový tok a klade nízké nároky na šířku pásma. Proto můžeme říci, že čím menší je velikost přenášených dat, tím menší je zpoždění. Naopak u služby Data bylo zpozdění nejvyšší. Při testování zatížené sítě bylo průměrné zpoždění kolem 100 ms, což je již vysoká hodnota, ale stále je zahrnuta v přijatelných hodnotách VoIP.

Závěr

Na začátku této práce bylo popsáno, co je monitorování, jaké způsoby monitorování existují a jaké funkce by měl mít každý monitorovací systém. Následující kapitola popsala protokoly, které se používají ke sledováni zařízení. Díky ním mohou monitorovací systémy přijímat data z různých zařízení. Poté byly vybrány a popsány 4 monitorovací systémy, jejichž hlavním cílem bylo vzájemně je porovnat a zjistit, který monitorovací systém rychleji detekuje problémy. V úvodu praktické části byl nainstalován operační systém Ubuntu, kam byly nainstalovány monitorovací systémy. Instalace OS Ubuntu a všech monitorovacích systémů nebyla plně popsána, pouze některé její části, protože se autor domnívá, že podrobná instalace již byla na oficiálních webových stránkách produktů popsána se všemi potřebnými dokumenty a čtenář tam najde mnohem více odpovědí na všechny své otázky. Prvním cílem této práce, jak již bylo popsáno výše, bylo identifikovat nejlepší monitorovací systém z hlediska rychlého odhalení problémů v síti. Proto bylo do všech monitorovacích systémů přidáno několik zařízení, jako jsou počítač se systémem Windows OS a router a byly také přidány šablony. Bylo testováno, jak rychle systém upozorní na velké zatížení procesoru a jak rychle bude informovat, pokud je router odpojen od sítě. Experiment ukázal, že všechny monitorovací systémy nás informovaly až 3 minuty po výskytu problému. Je třeba poznamenat, že doba oznámení systému závisí na tom, jak často chce správce získávat informace o zařízení. Mohli bychom nakonfigurovat systémy pro rychlejší sběr dat ze zařízení, ale pak by systémy spotřebovaly více systémových prostředků. Proto byl na všech monitorovacích systémech nastaven čas sběru informací 5 minut. Závěrem lze říci, že každý popsaný produkt v této práci má své výhody a nevýhody. Proto pro výběr vhodného monitorovacího systému nebude stačit znát jen teoretickou část. Musíme nainstalovat systém a otestovat jej. Teprve po testování bude možné identifikovat jejich výhody a nevýhody a pochopit, zda tento monitorovací systém splňuje naše požadavky. Ve druhé části praktické práce byla popsána metoda pro kontrolu latence pomocí nástroje fping, což je jedna z hlavních metod pro kontrolu latence, jež Zabbix nabízí. Také byl vytvořen skript pro kontrolu zpoždění a byl zaveden do monitorovacího systému Zabbix. V poslední části bylo provedeno základní měření latence.Testování bylo prováděno pro jednotlivé služby jako VoIP, Video, Data a také pro zatíženou a nezatíženou síť. Cílem měření bylo otestovat vytvořený skript a také vyhodnotit vybrané IP služby. Z měření bylo zřejmé,že pro VoIP službu zpoždění bylo nejmenší. Je to způsobeno tím, že pro komunikaci mezi dvěma účastníky služba využívá malé velikosti datových jednotek. Proto je přenos dat podstatně rychlejší než u ostatních služeb a tím pádem i zpozdění bude nižší. Také bylo testováno zpoždění pro zatíženou síť, kde během měření se zpoždění výrazně zvýšilo. A i přestože průměrné zpoždění pro zatíženou

síť nebylo kritické pro jednotlivé služby je doporučeno využívat QoS mechanizmy, kde by nejvyšší prioritu by měla mít služba VoIP jako služba, která má vyšší nároky na síť. Podrobný popis QoS mechanizmu je nad rámec dané práce.

Literatura

[1] UBIK, Dr. Ing. Sven. Monitorování vysokorychlostních počítačových sítí. *Sdelovaci Technika* [online]. 2006, , 7 [cit. 2019-10-12]. ISSN 0036-9942. Dostupné z: https://www.istlobster.org/publications/articles/sdel_tech.pdf

[2] KABELOVÁ, Alena a Libor DOSTÁLEK. *Velký průvodce protokoly TCP/IP a systémem DNS*. 5., aktualiz. vyd. Brno: Computer Press, 2008. ISBN 978-802-5122-365

[3] Pokročilá analýza provozu datových sítí (4. díl). *IT SYSTEMS* [online]. 2015, **2015**(6), 40-42 [cit. 2019-10-12]. Dostupné z: : https://www.systemonline.cz/it-security/sledovani-vykonu-site-a-aplikaci.html

[4] TCP/IP v kostce. *TCP/IP v kostce*. 2., upr. a rozš. vyd. České Budějovice: Kopp, 2009, s. 476-481. ISBN 978-80-7232-388-3.

[5] PUŽMANOVÁ, Rita. *Moderní komunikační sítě od A do Z: [technologie pro datovou, hlasovou i multimediální komunikaci]*. 2., aktualiz. vyd. Brno: Computer Press. ISBN 80-251-1278-0.

[6] *Bringing IT together* [online]. Zoho, 2019 [cit. 2019-10-20]. Dostupné z: https://www.manageengine.com/products/netflow/what-is-netflow.html

[7] *Enterprise-Class Monitoring Solution for everyone* [online]. Zabbix, 2009 [cit. 2019-10-22]. Dostupné z: https://www.zabbix.com/documentation

[8] *Center for Applied Internet Data Analysis* [online]. Caida, 2019 [cit. 2019-10-22]. Dostupné z: http://www.caida.org/tools/utilities/rrdtool/

[9] *The complete rrdtool-based graphing solution* [online]. Cacti, 2019 [cit. 2019-11-02]. Dostupné z: https://www.cacti.net/index.php

[10] *Documentation and howtos* [online]. The Cacti Group, 2019 [cit. 2019-11-06]. Dostupné z: https://docs.cacti.net/start

[11] *The Industry Standard in IT Infrastructure Monitoring* [online]. Nagios, 2019 [cit. 2019-11-06]. Dostupné z: : https://www.nagios.com/

[12] *Nagios Tutorial for Beginners: What is, Installation, Architecture* [online]. [cit. 2019-11-13]. Dostupné z: https://www.guru99.com/nagios-tutorial.html

[13] SERENKO, Ivan. *Analýza monitorovacího systému NetXMS*. Mladý vědec, 2017, s. 199-205.

[14] PERSCHKE, Susan. *Review: 4 open-source network management tools improve usability, performance* [online]. 2019, , 1-2 [cit. 2019-11-19]. Dostupné z: https://www.networkworld.com/article/3331852/review-4-open-source-network-management-tools-improve-usability-performance.html

[15] LITTLEJOHN SHINDER, Debra. *Počitačové sítě*. Pearson Education, 2001. ISBN 80-86497-55-0.

[16] BOUŠKA, Petr. *Začínáme s monitoringem sítě* [online]. Praha: Petr Bouška, 2009 [cit. 2019-11-25]. Dostupné z: https://www.samuraj-cz.com/clanek/zaciname-s-monitoringem-site/

[17] HUNT, Craig. *Konfigurace a správa sití TCP/IP*. Praha: Computer Press, 1997. ISBN 80-722-6024-3.

[18] KOSTRHOUN, Aleš. *Stavíme si malou síť*. Praha: Computer Press, 2001. Všechny cesty k informacím. ISBN 80-722-6510-5.

[19] *Ubuntu: praktická příručka uživatele Linuxu*. Brno: Computer Press, 2008. ISBN 978-802-5119-006

[20] *Zabbix Documentation* [online]. Zabbix SIA, 2019 [cit. 2019-12-03]. Dostupné z: https://www.zabbix.com/documentation/4.4/en/manual/installation

[21] *The Cacti Manual* [online]. [cit. 2019-12-03]. Dostupné z: https://www.cacti.net/downloads/docs/html/installation.html

[22] *Nagios Support Knowledgebase* [online]. 2015 [cit. 2019-12-03]. Dostupné z: https://support.nagios.com/kb/article/ nagios-core-instal-nagios-core-from-source-96.html#Ubuntu

[23] *Installation* [online]. Raden Solutions, SIA, 2019 [cit. 2019-12-03]. Dostupné z: https://www.netxms.org/documentation/adminguide/installation.html#instal-on-debian-or-ubuntu

[24] Simple checks. *Zabbix Documentation 4.4* [online]. Zabbix LLC., c2001-2020 [cit. 2020-03-27]. Dostupné z: https://www.zabbix.com/documentation/current/manual/config/items/itemtypes/simple_checks

[25] Custom graphs. *Zabbix Documentation 4.4* [online]. Zabbix LLC., c2001-2020 [cit. 2020-03-27]. Dostupné z: https://www.zabbix.com/documentation/current/manual/config/visualisation/graphs/custom

[26] Triggers. *Zabbix Documentation 4.4* [online]. c2001-2020 [cit. 2020-03-28]. Dostupné z: https://www.zabbix.com/documentation/current/manual/config/triggers

[27] Configuring a template. *Zabbix Documentation 4.4* [online]. Zabbix LLC., c2001-2020 [cit. 2020-03-28]. Dostupné z: https://www.zabbix.com/documentation/current/manual/config/templates/template

[28] Creating an item. *Zabbix Documentation 4.4* [online]. Zabbix LLC., c2001-2020 [cit. 2020-03-30]. Dostupné z: https://www.zabbix.com/documentation/current/manual/config/items/item

[29] External checks. *Zabbix Documentation 4.4* [online]. Zabbix LLC., c2001-2020 [cit. 2020-03-30]. Dostupné z: https://www.zabbix.com/documentation/current/manual/config/items/itemtypes/external

[30] IETF. *Benchmarking Methodology for Network Interconnect Devices* [online]. Network Working Group, 1999 [cit. 2020-05-03]. Dostupné z: https://tools.ietf.org/html/rfc2544

[31] ITU-T Recommendation Y.1564. *Ethernet service activation test methodology* [online]. ITU, 2011 [cit. 2020-05-03]. Dostupné z: https://www.itu.int/rec/T-REC-Y.1564-201103-S/en

Seznam symbolů, veličin a zkratek

MIB	Management Information Base
RFC	Request for Comments
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ICMP	Internet Control Message Protocol
ToS	Type of service
IP	Internet Protocol
RAM	Random-Access-Memory
PHP	Hypertext Preprocessor
SLA	Service Level Agreement
Ping	Packet InterNet Groper
SSH	Secure Shell
FTP	File Transfer Protocol
JMX	Java Management Extensions
RRD	Round-robin Database
GNU	General Public License
IT	Information technology
JMX	Java Management Extensions
RRD	Round-robin Database
GNU	General Public License
IT	Information technology
Syslog	System log
CPU	Central processing unit
HDD	Hard Disk Drive
GB	Gigabajt