VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

# DIPLOMOVÁ PRÁCE

Brno, 2018 Bc. Stanislav Vodehnal


# 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


# IDENTIFIKACE DOSTUPNOSTI ZAŘÍZENÍ V TECHNOLOGICKÝCH SÍTÍCH
IDENTIFICATION OF DEVICE AVAILABILITY IN TECHNOLOGICAL NETWORKS

**DIPLOMOVÁ PRÁCE**
MASTER'S THESIS


**AUTOR PRÁCE** — **Bc. Stanislav Vodehnal**
AUTHOR

**VEDOUCÍ PRÁCE** — **doc. Ing. Vladislav Škorpil, CSc.**
SUPERVISOR



**BRNO 2018**

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

# Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

***Student:*** Bc. Stanislav Vodehnal ***ID:*** 158266
***Ročník:*** 2 ***Akademický rok:*** 2017/18

**NÁZEV TÉMATU:**

## Identifikace dostupnosti zařízení v technologických sítích

**POKYNY PRO VYPRACOVÁNÍ:**

Věnujte se problematice technologických sítí v oblasti energetiky, jejich specifikaci a hlavním požadavkům na identifikaci dostupnosti zařízení v tomto typu sítě. Proveďte funkční analýzu relevantních řešení a to jak open source tak komerčně dostupných. Stanovte relevantní kritéria pro vícekriteriální analýzu a na jejím základě vybere nejvhodnější pro implementaci v konkrétním prostředí technologických sítí. Na základě získaných poznatků navrhněte a realizujte nasazení vybraného systému a optimalizujte jej pro konkrétní využití v prostředí energetických technologických sítích. Závěrem vyhodnoťte pilotní nasazení a funkcionalitu nasazeného systému pro identifikaci dostupnosti zařízení v technologických sítích.

**DOPORUČENÁ LITERATURA:**

[1] KOCJAN, Wojciech a Piotr BELTOWSKI. Learning Nagios. Birmingham, United Kingdom: Packt Publishing Limited, 2016. ISBN 9781785885952.

[2] KNAPP, Eric. Industrial network security: securing critical infrastructure networks for Smart Grid, SCADA , and other industrial control systems. Waltham, MA: Syngress, c2011. ISBN 9781597496452.

***Termín zadání:*** 5.2.2018 ***Termín odevzdání:*** 21.5.2018

***Vedoucí práce:*** doc. Ing. Vladislav Škorpil, CSc.
***Konzultant:*** Mgr. Josef Horálek, Ph.D., Univerzita HK

**prof. Ing. Jiří Mišurec, CSc.**
*předseda oborové rady*

## Abstrakt

Tato diplomová práce se zabývá monitoringem síťovích prvků technologických síti a systémů distribuční soustavy. Jsou zde uvedeny důvody, proč a jaké hodnoty chceme monitorovat. Následně jsou vybrány tři monitorovací systémy, u kterých jsou popsány jejich vlastnosti a funkce. Na základě jejich předností je vybrán jeden systém pro nasazení do testovacího prostředí. V praktické části je pak konfigurace vybraného systému a jeho následné nasazení do sítě.

## Klíčová slova

Icinga, ICMP, IEC 60870, IEC 60870-5, IEC 61850, Monitoring, Nagios, Network monitoring, SNMP, Technologická síť, Technologické systémy, Zabbix, Zabbix agent

## Abstract

This diploma thesis deals with the monitoring of network elements of technological networks and distribution systems. There are described reasons why and what kind of values we want to monitor. Three monitoring systems are then selected, described their properties and functions. Based on their merits, one system for deploying the test environment is selected. The practical part is the configuration of the selected system and its subsequent deployment to the network.

## Keywords

Icinga, ICMP, IEC 60870, IEC 60870-5, IEC 61850, Monitoring, Nagios, Network monitoring, SNMP, Technological network, Technological systems, Zabbix, Zabbix agent

VODEHNAL, S. *Identifikace dostupnosti zařízení v technologických sítích.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 79 stran, 1 příloha. Vedoucí diplomové práce doc. Ing. Vladislav Škorpil, CSc..

**Prohlášení**

Prohlašuji, že svou diplomové práci na téma „Identifikace dostupnosti zařízení v technologických sítích“ jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové 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 následku porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonu (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.

V Brně dne ..... …….....

podpis autora

**Poděkování**

Rád bych tímto poděkoval konzultantovi práce panu Mgr. Jesefu Horálkovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne ..... .....

podpis autora

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Technicka 12, CZ-61600 Brno, Czechia

http://www.six.feec.vutbr.cz

Výzkum popsaný v této diplomové práci byl realizovaný v laboratořích podpořených projektem Centrum senzorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operačního programu Výzkum a vývoj pro inovace.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

# Obsah

# Seznam obrázků

# Seznam tabulek

# Úvod

Distribuční soustava se skládá z několika hlavních částí, kterými jsou distribuční síť, rozvodny a v neposlední řadě také elektrárny pro výrobu elektrické energie. Uzlové rozvodny neslouží pouze k distribuci elektrické energie, ale dochází zde také k propojení distribuční a přenosové soustava. Přenosovou soustavu (400 kV a 220 kV) v České republice provozuje ČEPS a.s.. Distribuci elektřiny zajišťují v České republice tři společnosti, kterými jsou ČEZ Distribuce a.s., PREdistribuce a.s., E.ON Distribuce a.s., kde každá z těchto společností působí na jiném území ČR a má svou vlastní distribuční soustavu.

V uzlových rozvodnách, kde se střetává přenosová a distribuční soustava je transformováno napětí z přenosové soustavy 400 kV a 220 kV na 110 kV pro přenos v distribuční soustavě. Distribuční soustava pak slouží k přenosu elektrické energie ke koncovým zákazníkům a je složena z podružných rozvoden/transformoven, které převádí napětí 110 kV vyvedené z uzlové rozvodny na 35 kV, 22 kV, 10 kV, 6 kV a 3 kV. Na koncové trafostanici, nazývanou DTS (Distribuční trafostanice) je pak přímo k zákazníkovi distribuováno napětí 400/230 V. Samozřejmě jsou také zákazníci, kteří jsou připojeni k vyšší napěťové hladině.

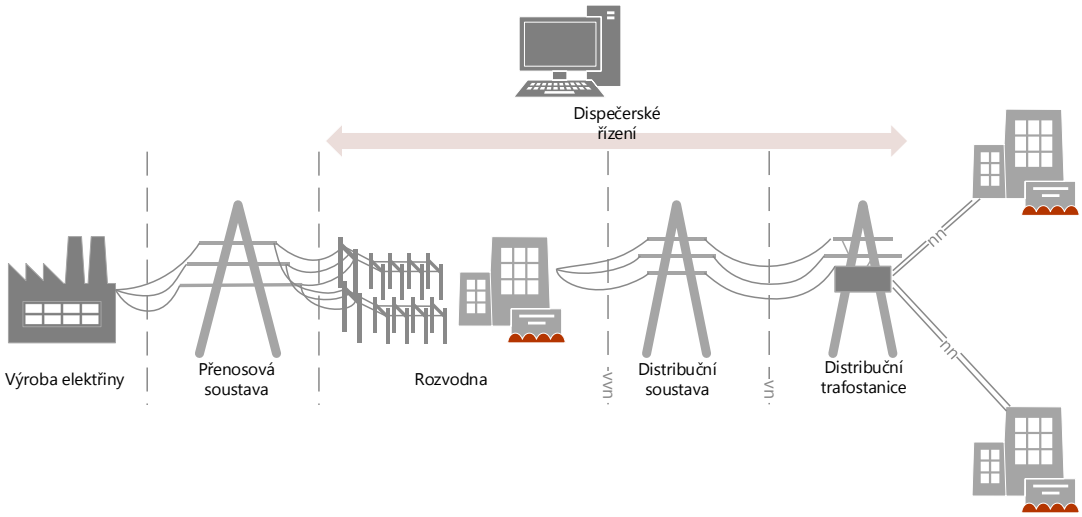
Distribuční síť spolu s rozvodnami je monitorována, ovládána a řízena systémem dispečerského řízení a sběru dat SCADA, případně jiným HMI. Tento systém ovládá na rozvodnách zařízení IED, RTU, sbírá naměřená data, ovládá ochrany a provádí mnoho dalších funkcí. Tato práce se zabývá systémem pro monitorování síťových prvků distribuční soustavy, přesněji řečeno technologickými systémy a sítěmi. Technologické systémy a sítě se skládají ze serverů, switchů, routerů, firewallů a jiných síťových prvků, které patří k distribuční soustavě včetně systému pro řízení distribuční soustavy. Pro lepší pochopení jsou zde popsány standardizované komunikačními protokoly IEC 61850 a IEC 60870 používané na rozvodnách pro přenos komunikace mezi zařízeními. Jsou zde obecně popsána zařízení používaná na rozvodnách, která slouží pro řízení ochran nebo sběr dat. Pro přenos k dispečerskému systému SCADA se využívají ethernetové sítě (případně mobilní sítě) s různými síťovými prvky.

Tyto síťové prvky jsou omezeným způsobem monitorovány systémem SCADA, který není k tomuto účelu určený a proto jsou funkce monitoringu přes systém SCADA výrazně omezené. Úkolem této diplomové práce je vybrat několik systému pro monitorování síťových zařízení, určit jaké služby bude potřeba monitorovat a na základě tohoto výběru zvolit monitorovací systém, který bude nejvíce vyhovovat distribuční soustavě společnosti ČEZ Distribuce a.s..

V závěru diplomové práce bude provedeno vyhodnocení zvoleného monitorovacího systému zaměřené na efektivitu, dosažené výsledky, uživatelskou přívětivost, výhody a nevýhody zvoleného systému.

# 1 Současný stav distribuční soustavy

Celá elektrizační soustava se skládá z výroben elektřiny, přenosové soustavy a distribuční soustavy, ze které je elektřina distribuována k zákazníkovi. Všechny tyto systémy je třeba nějakým způsobem řídit. Řízení distribuční sítě má na starosti velké množství systémů a jedním z nejdůležitějších je systém dispečerského řízení. Tato práce se zabývá pouze problematikou distribuční soustavy, proto jsou zde uvedeny pouze zmínky o přenosové soustavě. Obrázek 1 zobrazuje příklad distribuce elektřiny od výroby až po distribuci k zákazníkovi.




**Obrázek 1: Příklad distribuce elektřiny**

Distribuční soustava se skládá z prvků, jako jsou rozvodny, transformovny, distribuční trafostanice, vedení vvn, vn, nn a mnoho dalšího. Pro správnou funkcionalitu distribuční soustavy musí existovat způsob jejího řízení (resp. manipulace), kde právě k tomuto účelu slouží dispečerský řídicí systém (DŘS). Dispečerský řídicí systém pro manipulace na distribuční soustavě využívá systém SCADA, který shromažďuje naměřená data z rozvoden, hlásí poruchové stavy, ovládá prvky na rozvodně, atd.. Nashromážděná data se pak zobrazují pracovníkům na dispečerském pracovišti, kteří na základě takto získaných informací provádějí manipulace na distribuční soustavě.

Rozvodny fungují v bezobslužném režimu (automatizované systémy), to znamená, že nepotřebují trvalou lokální obsluhu. Každá rozvodna však musí disponovat svým vlastním řídicím systémem, který realizuje vzdálené dispečerské řízení (příkazy od dispečerů). Tento systém slouží také jako ochrana v případně nechtěné nebo nesprávné manipulace na rozvodně, která by mohla způsobit škody v distribuční soustavě. V neposlední řadě je na rozvodně umístěno lokální dispečerské pracoviště určené pouze pro danou rozvodnu, tzv. HMI.

Na rozvodnách a transformovnách jsou instalovány ochrany, které chrání distribuční soustavu, vedení ale také samotné transformátory v transformovnách. Na každý vývod jsou nasazeny ochrany, které sledují aktuální stav provozu. Tyto ochrany pomocí úsečníků zajišťují vypnutí nebo odpojení části distribuční soustavy, které jsou v poruchovém stavu (vypnou pouze místo nebo část sítě, která je v poruchovém stavu). Ochrany jsou kritickou součástí DS a musí splňovat přísné požadavky na spolehlivost, rychlost vypínání, přesnost a citlivost. [1][2][3][4]

### 1.1.1 Hromadné dálkové ovládání

Systém Hromadného dálkového ovládání (HDO) reguluje spotřebu energie a přispívá ke stabilnímu chodu distribuční soustavy. Pomocí HDO lze vzdáleně ovládat spotřebiče u zákazníka a regulovat tak spotřebu energie. Pokud je elektrické energie přebytek nebo naopak nedostatek je vyslán povel HDO pro připnutí nebo odepnutí zátěže. HDO se ve velké míře používá například pro veřejné osvětlení ve městech, spínání ohřívačů vody, spínání elektrokotlů v domácnostech a mnoho dalšího. Veškerá zařízení, která využívají povely HDO mají spotřebu energie nastavenu na nízký tarif. Systém HDO je zmíněný zvlášť, protože se jedná o systém, který není kritický pro chod DS. [3][4]

# 2 Komunikační protokoly pro rozvodny

Distribuční soustava se neskládá pouze ze soustavy pro přenos elektrické energie a dispečerského řídícího systému, ale patří sem také ovládací, dohledové a řídicí systémy stanic (SCADA, HMI, RTU, IED…). Tyto systémy musí komunikovat s prvky na rozvodnách, proto byly mezinárodní technickou komisí IEC standardizovány komunikační protokoly. Protokoly jsou definovány jak pro sériovou komunikaci, tak pro komunikaci pomocí TCP/IP. Obě metody se v současné době používají, přestože je snaha sériovou komunikaci postupně nahrazovat komunikaci pomocí TCP/IP. V této části práce jsou obecně popsány komunikační protokoly IEC 60870, které zahrnují jak sériovou komunikaci, tak komunikaci pomocí TCP/IP a protokol IEC 61850, který je založený pouze na komunikaci pomocí TCP/IP.

## 2.1 Standard IEC 60870

Jedním z uvedených komunikačních protokolů je standard IEC 60870 (ČSN EN 60870), definován Mezinárodní elektrotechnickou komisí, který je určen pro Systémy a zařízení pro dálkové ovládání (systém dispečerského řízení), nebo obecně pro systémy SCADA. Takto označované systémy se používají nejen pro řízení sítě (distribuční soustavy) sloužící pro přenos elektrické energie ale také dalších průmyslových odvětvích (např. automobilový průmysl). Norma IEC 60870 obsahuje šest částí, které definují obecné informace týkající se normy, provozních podmínek, elektrických rozhraní, výkonových požadavků a protokolů přenosu dat. Norma IEC 60870 byla vypracována technickou komisí IEC 57 (pracovní skupina 03). Tabulka 1 níže obsahuje seznam jednotlivých částí normy. [5][6]

**Tabulka 1: Norma IEC 60870 a její části**

| Označení normy | Obsah normy |
|---|---|
| IEC 60870 - 1 (1-5) | Obecná úvaha (Základní principy, specifikace…) |
| IEC 60870 - 2 (1-2) | Provozní podmínky |
| IEC 60870 - 3 | Elektrické vlastnosti rozhraní |
| IEC 60870 - 4 | Výkonové požadavky |
| IEC 60870 - 5 - X | Komunikační / přenosový protokol |
| IEC 60870 - 6 - X | Normalizace funkčních profilů v elektrizačních soustavách |

### 2.1.1 Důležitá zařízení a podpůrné systémy rozvodnové sítě

Než se podíváme na obecný popis protokolu, je nutné se seznámit s důležitými zařízeními a systémy, které jsou součástí distribuční soustavy a jejího řízení. Níže je uveden jejich obecný popis pro jednoduché pochopení k čemu daná zařízeni nebo systémy slouží.

**RTU**

Jednotky RTU (Remote Telemetry Unit), občas nazývané Terminál slouží v energetickém průmyslu pro vzdálené řízení, měření a sběr dat. Velké systémy rozvodnových stanic lze snadno pomocí RTU vzájemně propojovat a tím zvyšovat například zvýšení počtu vstupů a výstupů, kde jedna z těchto jednotek RTU slouží jako hlavní komunikační rozhraní pro systém SCADA. Jednotky RTU mají integrované HMI (Humane Machine Interface), ochranné funkce (jedno RTU může nahradit až několik ochran), možnost výpočtu vyrobeného výkonu (například ve větrné elektrárně), možnost programování nových funkcí RTU a mnoho dalších vlastností. Jednotky RTU fungují především jako datové koncentrátory, které v jednom uzlu rozvodnové sítě shromažďují data a následně je odesílají do nadřazeného systému SCADA. Jednotky RTU bývají zpravidla propojeny s několika zařízeními IED a především podporují komunikační standardy popsané normách IEC 60870 a IEC 61850.[7][8][9][10]

**HMI**

HMI (Humane-Machine Interface) je rozhraní určené pro konfiguraci IED nebo RTU například pomocí webové aplikace (GUI). Prakticky jde o software, který umožnuje jeho operátorovi zadávat různé příkazy, umožnuje zobrazení měřených veličin v určitém období v grafickém znázornění. Jedná se o klasický PC, který je vybavený speciálním softwarem. HMI zpravidla bývají umístěné na rozvodnách. Software, který PC využívá, by měl umožnovat komunikaci pomocí používaných komunikačních protokolů v daném prostředí, například IEC 60870 a IEC 61850 případně jiné používané komunikační protokoly. Dříve bylo HMI známo pod názvem MMI (Man-Machine Interface). [11]

**IED**

Jednotka IED (Intelligent Electronic Device) je zařízení, které je přímo určeno pro chránění, ovládání, monitorování a měření rozvodnových stanic energetických společností. Na rozdíl od RTU jednoty IED přímo slouží jako ochranné systémy, ovládací prvky nebo pro měření. RTU data z těchto jednotek IED koncentruje v jednom bodě a přeposílá je do nadřazeného systému. Zařízení IED podporují řadu komunikačních protokolů, jako jsou DNP3, Modbus, ale především IEC 60870, IEC 61850 při použití horizontální komunikace pomocí GOOSE a normy IEC 60870-5-103, která je přímo určená pro komunikaci s ochranami. [1][2]

**SCADA**

Supervisory Control And Data Acquisition, jejíž zkratka je SCADA, v překladu znamená Systémy pro dohled, řízení a sběr dat (nebo také Systém dispečerského řízení a sběru dat). Systém je určený pro monitorování, ovládání technologických procesů a sběr dat, v energetických sítích distribuční soustavy. Systém je určený pro dispečery, kteří dohlížejí na distribuční soustavu, zajišťují její správný chod a provádějí případnou parametrizaci. SCADA shromažďuje naměřená data přes vzdálený terminál RTU

do centrálního místa, pro následnou analýzu, řízení a zobrazení těchto hodnot. Jedním z významných českých zástupců je systém Reliance. [1][2][6][11][12][13][14]

### 2.1.2 Komunikační protokol IEC 60870-5

Pátá část standardu IEC 60870-5 definuje podrobnějším způsobem požadavky na systémy dálkového ovládání (SCADA) a přenos dat v technologických systémech. Protokol je určen pro zařízení a systémy dálkového ovládání se sériovým a TCP/IP přenosem ať už se jedná o přenos dat mezi jednotlivými zařízeními IED na rozvodně nebo sběr měřených dat pomocí koncentrátoru RTU. Standard je určen především pro použití v energetickém průmyslu, díky systému SCADA však není vyloučeno použití v jiných odvětvích, než je energetický průmysl. Pátá část IEC 60870-5 obsahuje mnoho oddílů, obecně pro komunikaci jsou nejdůležitější oddíly IEC 60870-5-101/102/103/104. Tabulka 2 obsahuje důležité oddíly této části normy. [15][19]

**Tabulka 2: Přehled oddílů části normy IEC 60870-5**

| Označení normy | Obsah normy |
|---|---|
| IEC 60870 – 5 – 1 | Formáty přenosového rámce |
| IEC 60870 – 5 – 2 | Procedury linkového přenosu |
| IEC 60870 – 5 – 3 | Obecná struktura aplikačních dat |
| IEC 60870 – 5 – 4 | Definice a kódování aplikačních informačních prvků |
| IEC 60870 – 5 – 5 | Základní aplikační funkce |
| IEC 60870 – 5 – 6 | Směrnice pro zkoušení shody pro společné normy EN 60870-5 |
| IEC 60870 – 5 – 7 | Bezpečnostní rozšíření pro protokoly IEC 60870-5-101 and IEC 60870-5-104 |
| **IEC 60870 – 5 – 101** | **Společná norma pro základní úkoly dálkového ovládání** |
| IEC 60870 – 5 – 102 | Společná norma pro přenos integrovaných součtových hodnot v elektrizačních soustavách |
| IEC 60870 – 5 – 103 | Přenosové protokoly – Společná norma pro informační rozhraní ochran |
| **IEC 60870 – 5 – 104** | **Přenosové protokoly – Síťový přístup pro IEC 60870-5-101 používající normalizované transportní profily** |

### 2.1.3 Komunikace pomocí sériového rozhraní

Historicky, kdy byla norma dokončena (v roce 1995), pokrývala pouze přenos přes po sériové komunikační lince jako je například RS-232 nebo RS-485 s relativně malou šířkou pásma komunikačního kanálu. Tuto komunikaci definuje oddíl IEC 60870-5-101. V současné době se komunikace přes sériovou linku stále používá především na rozhraní mezi přenosovou soustavou a distribuční soustavou (komunikace mezi ČEPS a ČEZ na uzlových rozvodnách). S rozvojem komunikační technologie byla definována norma IEC 60870-5-104 umožňující komunikaci prostřednictvím IP sítě pomocí sady komunikačních protokolů TCP/IP. V rámci modernizace rozvoden je snaha

o přechod ze sériové komunikace na komunikaci pomocí TCP/IP (IEC 60870-5-104 nebo IEC 61850).[15][16][17][18][19][20]

Struktura protokolu vychází z referenčního modelu ISO/OSI, přičemž je zde definována architektura se zvýšenou výkonností EPA (Enchanced Performance Architecture) pro systémy s omezenou šířkou pásma, která neobsahuje prezentační, relační, transportní a síťovou vrstvu:

- **Fyzická vrstva** umožňuje fyzický přenos dat po požadovaném médiu. Přenáší tedy jednotlivé bity po sériovém kabelu (RS-232 nebo RS-485).
- **Spojová vrstva** obsahuje několik procedur spojového přenosu jako je například zvolení správného formátu rámce datové jednotky ASDU viz kapitola Formáty rámců a požadavky na přenos dat. Spojová vrstva zajišťuje výběr režimu pro přenos a to vyvážený nebo nevyvážený režim. Nevyvážený režim se použije v případě, že spoje z hlavní řídicí stanice sdílejí společný fyzický kanál na podřízené stanice. V ostatních případech se použije vyvážený režim.
- **Aplikační vrstva** obsahuje několik „Aplikačních funkcí“. Definuje vhodné formáty aplikačních datových jednotek ASDU v přenášených rámcích definovaných v IEC 60870-5-1.[15][16][17][20][21]

### 2.1.4 Komunikace pomocí protokolu TCP/IP

Oddíl IEC 60870-5-104 definuje přenos aplikačních datových jednotek ASDU (Application Service Data Unite) z IEC 60870-5-101 a pomocí protokolu TCP/IP pro zařízení dálkového ovládání a systémy SCADA a přidává další v oddíle IEC 60870-5-104. Zjednodušeně řečeno, protokol zajišťuje komunikaci mezi řídicím systémem a zařízeními RTU po IP vrstvě.

Tabulka 3 zobrazuje strukturu protokolu a vychází opět z referenčního modelu ISO/OSI. Na rozdíl od sériové komunikace IEC 60870-5-101 zde již není aplikována architektura EPA, především proto, že pro komunikaci pomocí TCP/IP je třeba použít síťovou a transportní vrstvu, které jsou nezbytné pro komunikaci po IP síti. Přesto je struktura protokolu lehce zjednodušena a nejsou zde vrstvy relační a presentační. Řídící informace Aplikačního protokolu (APCI) zajišťuje mechanizmus zjištění začátku a konce datových jednotek ASDU a také slouží pro účely řízení. ASDU a APCI pak dohromady vytvoří Jednotku dat Aplikačního protokolu (APDU).

Tabulka 3 zobrazuje datové jednotky ASDU a APCI, kde jejich tvorba probíhá na aplikační vrstvě. Následná komunikace mezi systémy je řešená přenosem datagramů/paketů/rámců na jednotlivých vrstvách. [15][16][19][21]

**Tabulka 3: Struktura protokolu**

<table>
<tr><td>Výběr datových jednotek<br>(ASDU)</td><td rowspan="2">Aplikační vrstva</td></tr>
<tr><td>Řídící informace Aplikačního protokolu (APCI)</td></tr>
<tr><td rowspan="4">Přenos pomocí protokolu TCP/IP</td><td>Transportní vrstva</td></tr>
<tr><td>Síťová vrstva</td></tr>
<tr><td>Spojová vrstva</td></tr>
<tr><td>Fyzická vrstva</td></tr>
</table>

### 2.1.5 Formáty rámců a požadavky na přenos dat

Oddíl IEC 60870-5-1 stanovuje normy pro dvě nejnižší vrstvy, fyzickou a spojovou vrstvu ve smyslu referenčního modelu ISO/OSI. Především stanovuje normy pro kódování, formátování a synchronizaci datových formátů s proměnnými a pevnými délkami, které splňují požadavky předepsané pro nenarušitelnost přenášených dat. Oddíl také určuje požadavky na přenos dat v systémech dálkového ovládání, jako jsou:

- Vysoká nenarušitelnost a úplnost dat, která může být narušena nezjištěnými chybami bitů, nezjištěnými chybami rámce, způsobenými chybami synchronizace, nezjištěnou ztrátou informace a podobně.
- Krátká doba přenosu dálkového ovládání s použitím výkonných protokolů pro přenos rámce, především pro zprávy vyvolané událostmi přenášených po přenosových cestách s omezenou šířkou kmitočtového pásma a s nedefinovanými šumovými charakteristikami.
- Podpora bitově orientovaného datového přenosu. [6][15][21]

Oddíl dále definuje tři rozdílné třídy formátů rámce, vhodné pro zvýšené požadavky propustnosti informací a nenarušitelnosti dat v systémech dálkového ovládání. [6][15][21]

- Formát rámce FT1.1 charakterizuje blokový kód s Hammingovou[1] vzdáleností 2, který je vytvořen doplněním spouštěcího bitu, paritního bitu a závěrného bitu do 8 informačních bitů. Formát je používaný pro jednoduché systémy s cyklickou aktualizací dat a požadavky na nízkou třídu nenarušitelnosti dat.
- Formát rámce FT1.2 je sekvence bloků rámce FT1.1 doplněná znakem kontrolního součtu s Hammingovou vzdáleností 4.
- Formát rámce FT2 je charakterizován blokovým kódem s Hammingovou vzdáleností 4, který obsahuje až 15 oktetů uživatelských dat, doplněných jedním

[1] Hammingova vzdálenost (označována *d*) definuje počet bitů, ve kterých se dva bloky bitů liší. Z této vzdálenosti lze také vypočítat kolik chyb lze použitým kódem detekovat ($t_{det} = d_{min} - 1$) a kolik opravit ($t_{opr} = \frac{dmin - 1}{2}$).[22]

kontrolním oktetem. Formáty rámce FT1.2 a FT2 slouží pro řídicí systémy se zvýšenými požadavky na nenarušitelnost dat.

- Formát rámce FT3 je charakterizován blokovým kódem s Hammingovou vzdáleností 6, který obsahuje až 16 oktetů uživatelských dat, doplněných dvěma kontrolními oktety. Tento formát je vhodný pro systémy se zvláště vysokými požadavky na nenarušitelnost dat.

## 2.2 Standard IEC 61850

Soubor norem IEC 61850 specifikuje jednotné metody komunikace a komunikačního protokolu nezávisle na výrobcích jako jsou ABB, Alstom, Siemens a řada dalších. Standard je určený pro automatizační systémy rozvodných stanic. To znamená, že je především určen pro komunikaci mezi zařízeními na rozvodnách. Zařízení od takto různých výrobců, která jsou spojená komunikační sítí, nesou název Intelligent Electronic Devices (zkratka IED). Standard neobsahuje pouze definici komunikačních protokolů, ale také standardy pro řídicí funkce, programovací jazyk a engineering rozvoden. Tabulka 4 obsahuje jednotlivé části normy, z nichž některé jsou několikadílné. [23]

První čtyři části souboru IEC 61850 specifikují prostředí, požadavky na zařízení, terminologii a další. Části 5-9 se zabývají komunikací a poslední část 10 definuje zkoušky a shody používané v automatizovaných systémech rozvoden. [12][13][24][25][26]

**Tabulka 4: Norma IEC 61850 a její části**

| Označení normy | Obsah normy |
|---|---|
| IEC 61850 - 1 | Přehled a úvod do problematiky |
| IEC 61850 - 2 | Terminologie výrazů |
| IEC 61850 - 3 | Všeobecné požadavky |
| IEC 61850 - 4 | Systémové a projektové řízení |
| IEC 61850 - 5 | Požadavky na komunikaci |
| IEC 61850 - 6 | Konfigurační popisový jazyk pro komunikaci |
| IEC 61850 - 7(1-4) | Základní komunikační struktura |
| IEC 61850 - 8 | Mapování specifických komunikačních funkcí (SCSM) - MMS |
| IEC 61850 - 9(1-2) | Mapování specifických komunikačních funkcí (SCSM) |
| IEC 61850 - 10 | Přizpůsobení testování |

### 2.2.1 Historie vzniku IEC 61850

Historicky můžeme považovat za první komunikační prostředek pro rozvodnové systémy telefonní přístroj. V případě nutnosti provedení manipulace přímo na rozvodně byl kontaktován technik, který tuto práci manuálně obstaral. Existovaly také systémy, které měly přímo vytočením čísla pomocí telefonu možnost manipulovat s prvky na rozvodně.

S postupem času a rozvojem digitální komunikace byl v 60. letech 20. století nainstalován systém DAS (Data Aquisition System). Jedná se o systém pro sběr měřených dat z rozvodných stanic. Tento sběr dat byl však v té době velmi omezený šířkou pásma (propustností) komunikačního kanálu. Z toho důvodu, byl systém historicky navržený a přizpůsobený takovým způsobem, aby mohl fungovat přes omezenou šířku pásma komunikačního kanálu. Toto omezení již neplatí a šířky pásma pro komunikaci je dostatek. Tento pokrok v technologii umožnil rozvoj systému.[1][12][13][23]

Za vypracováním souboru norem IEC 61850 stála Technická komise 57 (TC57), která se skládala z přibližně 60 členů rozdělených do několika pracovních skupin, kteří na normě začali pracovat v roce 1997. První edice normy však vyšla až v roce 2004.

Důvodem zavedení tohoto standardu bylo v minulosti velké množství protokolů a velké množství různých výrobců, kteří byli navzájem nekompatibilní. Podstatný problém byl, že firmy si své protokoly chránily patentem či jiným způsobem, aby zamezily jeho použití v zařízeních jiných výrobců. Záměrem IEC tedy bylo nalezení společného řešení, které by splňovalo požadavky na automatizované řízení, chránění a měření rozvodných zařízení. Vznikl tedy standard IEC 61850 (pro ČR ČSN EN 61850), který sjednocuje komunikační protokol, rozhraní a datové modely. Standard tak výrazně zjednodušuje výměnu zařízení, kdy při poruše může být poškozené nebo vadné zařízení nahrazeno kompatibilním kusem od jiného výrobce. [1][12][13][23]

Komunikační protokol standardu IEC 61850 je založený na Ethernetu, standard ISO/IEC 8802-3 (IEEE 802.3) a komunikační vrstvě TCP/IP. Je rozdělen do dvou komunikačních úrovní (horizontální a vertikální komunikace). Vertikální komunikace je určená pro komunikací mezi systémy (např. SCADA - IED) a horizontální komunikace je určená pro přenos kritických dat mezi zařízeními IED. Komunikace je typu klient-server, ale právě díky horizontální komunikaci je možný rychlý přenos kritických dat v reálném čase mezi jednotlivými IED (IED při takové komunikaci pracuje jako klient nebo server). Komunikace mezi uzly používá několik typů zpráv (celkem 7), které jsou rozděleny dle důležitosti a maximální možné doby pro přenos.[12][13][23]

### 2.2.2 Komunikační protokol

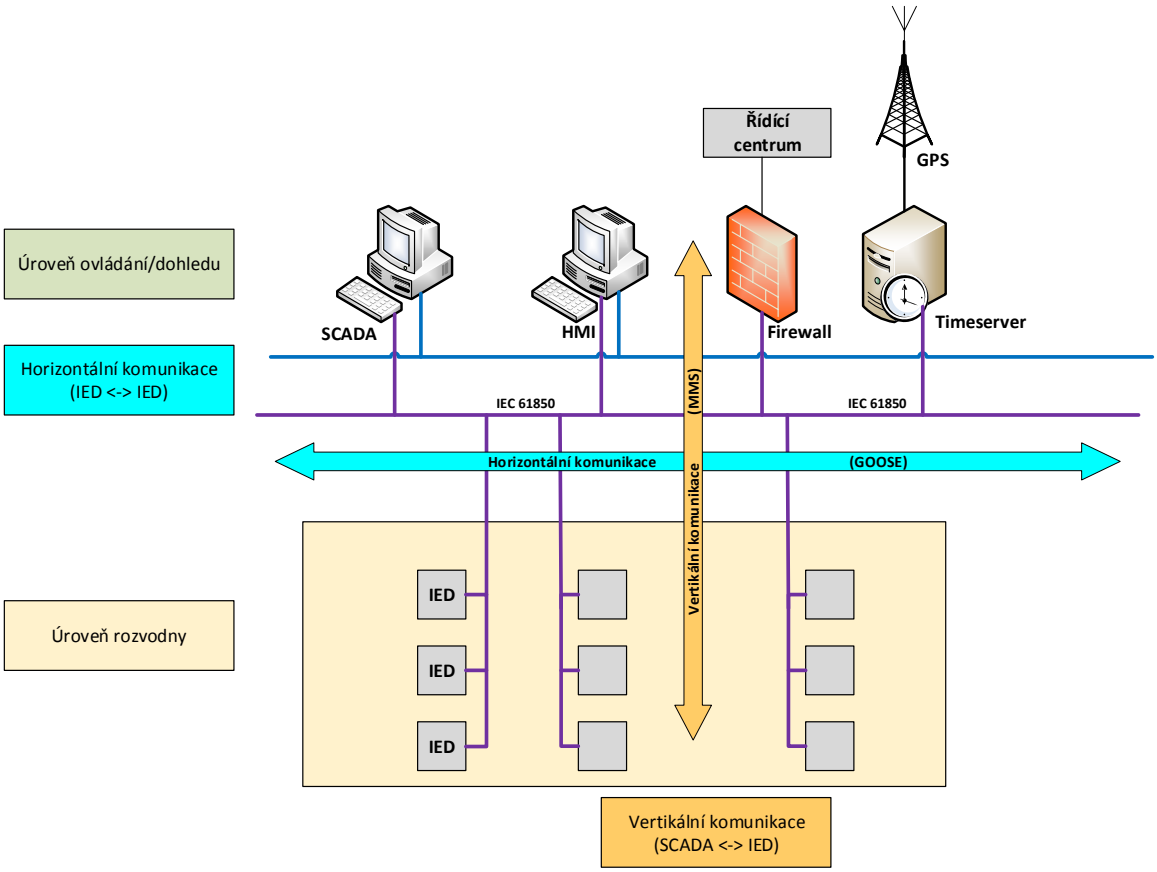
Abstraktní datový a objektový model IEC 61850 definuje standardizovanou metodu, která dovoluje všem zařízením IED prezentovat data s použitím jednotné struktury. Abstraktní v tomto případě znamená, že zde jsou vyzdviženy společné vlastnosti různých systémů. Model ACSI (Abstract Communication Service Interface) definuje soubor služeb a způsob odpovědí na tyto služby, které dovolují zařízením IED, aby se z pohledu chování sítě všechna IED chovala stejným způsobem, přestože jsou od různých výrobců. ACSI definuje mapování hned na několik služeb, mezi které se řadí MMS, GOOSE nebo SMV. Komunikační protokol je rozdělen do dvou komunikačních modelů, kterými jsou vertikální a horizontální komunikace. Každý model je specifický pro daný typ komunikace a nelze je mezi sebou zaměňovat. [12][23]

Obrázek 2 zobrazuje příklad komunikace obou modelů. Obrázek 3 pak zobrazuje zprávy obou modelů. Jsou zde zobrazeny zprávy, které se přenáší po linkové nebo síťové vrstvě. Z toho lze odvodit také důležitost jednotlivých zpráv.[2]

**Vertikální komunikace**

Vertikální komunikace je určená pro předávání nekritických dat a služeb mezi úrovněmi řídicího systému rozvodny SAS (Substation Automation System) a vertikálně

přenášená data mají oproti horizontálnímu přenosu nižší prioritu. Jedná se například o předávání zpráv mezi RTU, který v tomto případě vystupuje jako server a systémem SCADA, který vystupuje jako klient získávající data. V zásadě se jedná o ovládací a monitorovací funkce. Norma IEC 61850 definuje MMS (Machine Messaging Standard), který je určený pro komunikaci mezi monitorovacími a řídicími systémy (vertikální komunikace). MMS je spojově orientovaný a pracuje jak s komunikační vrstvou TCP/IP tak s modelem ISO/OSI viz Obrázek 3. Metoda využívá komunikační metody typu klient-server. [12][13][25][26][27]




**Obrázek 2: Využití horizontální a vertikální komunikace**

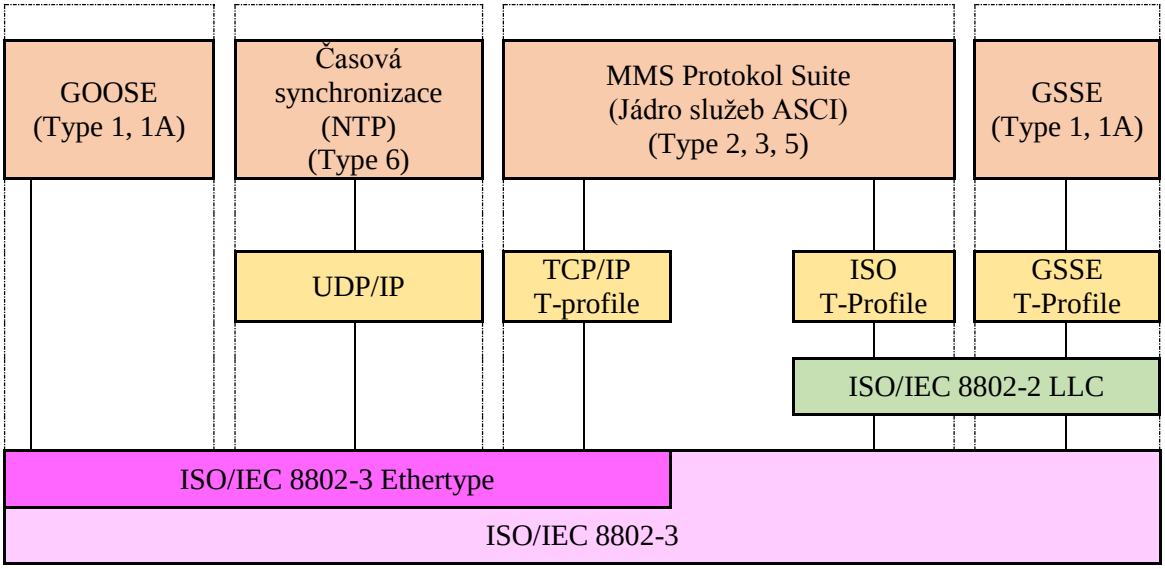
### Horizontální komunikace

Horizontální komunikace je pro komunikaci mezi IED – IED a přenášejí se zde kriticky důležité informace, automatizační/řídicí informace, kde tato data mají vysokou prioritu. Tyto informace se vysílají pomocí všesměrového vysílání, aby bylo zajištěno, že takto důležité informace dostane každé zařízení IED. Jedná se o krátké informace, které je potřeba doručit s velmi nízkým zpožděním do několika milisekund (do 4 ms), což je významné z hlediska rychlé výměny dat pro funkce chránění. Komunikace je typu peer-to-peer (rovný s rovným) nebo metoda point-to-point. Rozdíl oproti vertikální komunikaci je především takový, že komunikace nevyužívá TCP/IP, ale komunikuje pouze po linkové vrstvě, viz Obrázek 3.[12][13]

Zprávy, které jsou přenášeny při horizontální komunikaci se nazývají generické události ve stanici (GSE - Generic Substation Event). Tento model umožňuje rychlé a spolehlivé předávání vstupních a výstupních dat v celém systému. Model GSE poskytuje účinný způsob dovolující současné předání téže informace o GSE na více fyzických zařízení použitím služeb hromadného vysílání. [12][13]

Model GSE definuje dvě třídy řízení a struktury zpráv (GOOSE a GSSE):

- Generická objektově orientovaná událost ve stanici (**GOOSE** - Generic Object Oriented Substation Events) zajišťuje výměnu rozsáhlých obvykle obecných dat organizovaných pomocí datového souboru. GOOSE zprávy obsahují informace, které umožňují příjemci rozeznat, že byl změněn stav a zjistit čas poslední změny stavu. Nově aktivované zařízení musí po zapnutí vyslat aktuální hodnotu datového objektu (stav) nebo hodnoty jako počáteční GOOSE zprávu. Tato data obdrží pouze zařízení, která se zaregistrují, aby obdržovala tento typ zpráv. [12][13][23][25][28]
- Generická stavová událost ve stanici (**GSSE** - Generic Substation State Events) umožňuje předávat informace o změnách stavů (obsahuje pouze stavové informace). [12][13][23][25][28]




**Obrázek 3: Přehled rozdělení jednotlivých datových modelů**

### 2.2.3 Konfigurační jazyk standardu IEC61850

Konfigurační jazyk pro rozvodnové stanice se nazývá Substation Configuration Language (SCL - rozvodnový konfigurační jazyk) a je založený na jazyku XML (eXtensible Markup Language). Definice syntaxe jazyku SCL je uvedena jako W3C XML schéma. Obrázek 4 ukazuje příklad syntaxe jazyka SCL ve formátu XML. [13]

```
<xs:complexContent>
    <xs:extension base="tEquipmentContainer">
        <xs:sequence>
            <xs:element name="VoltageLevel" type="tVoltageLevel" maxOccurs="unbounded">
                <xs:unique name="uniqueBayInVoltageLevel">
                    <xs:selector xpath="./scl:Bay"/>
                    <xs:field xpath="@name"/>
                </xs:unique>
                <xs:unique name="uniquePowerTransformerInVoltageLevel">
                    <xs:selector xpath="./scl:PowerTransformer"/>
                    <xs:field xpath="@name"/>
                </xs:unique>
                <xs:unique name="uniqueGeneralEquipmentInVoltageLevel">
                    <xs:selector xpath="./scl:GeneralEquipment"/>
                    <xs:field xpath="@name"/>
                </xs:unique>
                <xs:unique name="uniqueChildNameInVoltageLevel">
                    <xs:selector xpath="./*"/>
                    <xs:field xpath="@name"/>
                </xs:unique>
            </xs:element>
            <xs:element name="Function" type="tFunction" minOccurs="0" maxOccurs="unbounded">
                <xs:unique name="uniqueSubFunctionInFunction">
                    <xs:selector xpath="./scl:SubFunction"/>
                    <xs:field xpath="@name"/>
                </xs:unique>
                <xs:unique name="uniqueGeneralEquipmentInFunction">
                    <xs:selector xpath="./scl:GeneralEquipment"/>
                    <xs:field xpath="@name"/>
                </xs:unique>
            </xs:element>
        </xs:sequence>
    </xs:extension>
</xs:complexContent>
```

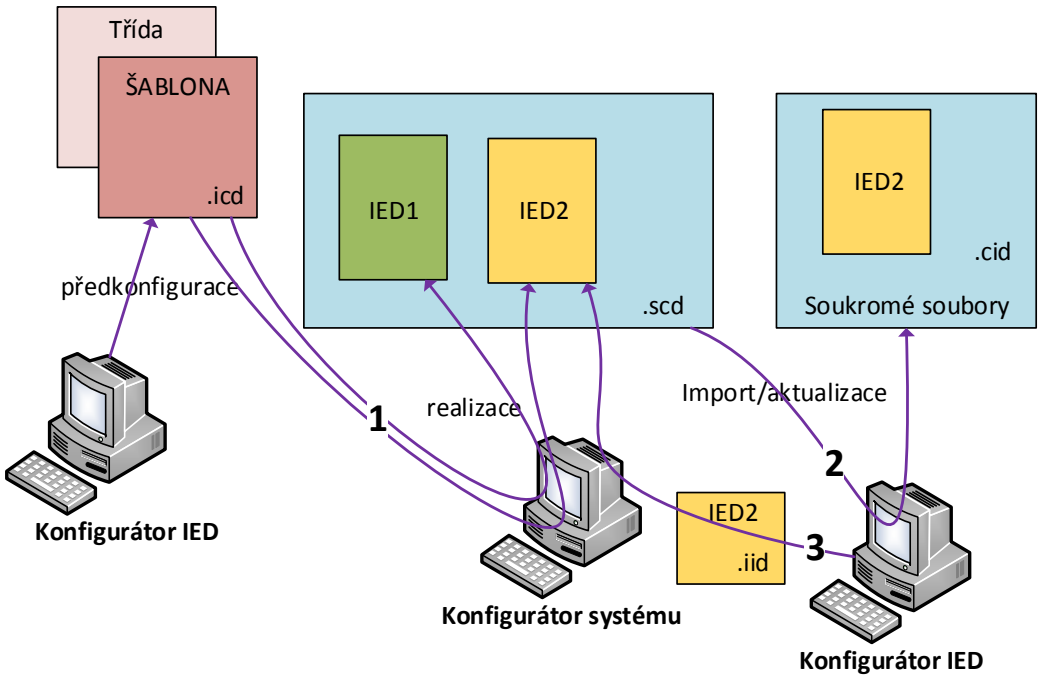
**Obrázek 4: Příklad syntaxe XML pro jazyk SCL[29]**

Pro výměnu konfiguračních dat mezi různými IED pomocí konfigurátorů jsou definovány čtyři základní typy SCL souborů. Zvolený konfigurační jazyk určuje hierarchii konfiguračních souborů, které umožňují popis více úrovní systému v jednoznačných a standardizovaných souborech XML. Jazyk je založen na starších značkovacích jazycích. Obrázek 5 zobrazuje obecnou návaznost jednotlivých souborů. Jednotlivé typy SCL souborů jsou vypsány níže[30]:

- Jedním ze souborů je soubor s příponou **.ICD**, jehož součástí jsou základní funkce IED. Soubor popisuje schopnosti daného IED, to znamená, že definuje jeho funkce a technické možnosti. Tento soubor musí být dodán každým výrobcem daného IED, bez toho souboru není možná konfigurace systému. Soubor lze použít jako šablonu pro tvorbu souboru **.CID**. Název IED musí být **ŠABLONA.**
- Výměna dat od konfigurátora IED na konfigurátora systému pro IED, které je speciálně určeno pro specifický projekt. V případě tohoto souboru má IED název přímo specifický pro projekt, pro který je dané IED určeno. Přípona souboru musí být **.IID** pro popis konkrétního IED.
- Soubor s příponou .SSD popisuje jednopólové schéma rozvodny a funkční požadavky logických uzlů. Jedná se o výměnu dat od prostředku pro specifikaci systému na konfigurátor systému. Soubor obsahuje popis stanice, případně

šablony jiných typů dat a v neposlední řadě může obsahovat definici logických uzlů. Rozšíření souboru musí být s příponou **.SSD** pro popis specifikace systému.

- Výměna dat od konfigurátoru systému na konfigurátory IED. Tento soubor obsahuje kompletní popis konfigurace rozvodny, tzn. že zahrnuje všechna IED, která obsahují konfigurovaný tok dat a potřebné šablony typu dat. Rozšíření souboru musí být s příponou **.SCD** popisující konfiguraci stanice. Soubor se skládá ze souborů **CID**.
- Výměna dat od konfigurátoru IED na IED. Popisuje část týkající se komunikace pro konkrétní IED v projektu, tzn. IP adresu zařízení, ICD soubor a technický klíč. Jedná se o soubor .SCD, který musí znát příslušné IED. Soubor musí mít příponu **.CID**. [26]
- Výměna dat mezi konfigurátory systému u různých projektů. Tento soubor popisuje rozhraní jednoho projektu použitá jiným projektem a opětný import dodatečně navržených propojovacích spojení mezi projekty. Jedná se podmnožinu .SCD souboru. Rozšíření souboru musí být **.SED** pro Popis Výměny v Systému.[12][13][30]




**Obrázek 5: Obecné schéma předávání informací jazykem SCL**

**Konfigurátor IED** je softwarový prostředek přímo specifický pro výrobce, případně je přímo specifický pro dané IED. Konfigurátor IED musí být schopen exportovat nebo nahrávat soubory definované v IEC 61850-6. Pomocí konfigurátoru IED pak lze získat nastavení IED, vygenerovat konfigurační soubor a následně nahrávat konfiguraci IED do jiného IED. Kompatibilitu lze zajistit dvěma způsoby. Prvním způsobem je obsah souboru s příponou .ICD, který popisuje možnosti daného IED. Druhým způsobem

je možný import systémového SCL souboru pro nastavení IED, který lze použít jako systémový soubor s příponou .SCD. [12][13][30]

**Konfigurátor systému** je na rozdíl od konfigurátoru IED určený k tomu, aby byl schopný, nezávisle na zařízení/výrobci IED, exportovat nebo nahrávat konfigurační soubory do a z IED, které jsou definované v části IED 61850-6. V normě je definováno, že konfigurující technik je povinen použít konfigurátor systému pro doplnění systémových informací různých zařízení IED. Konfigurátor systému musí umět vygenerovat konfigurační soubor, ze kterého je zřejmá konfigurace daného IED. [12][13][30]

## 2.3 Shrnutí rozdílů mezi IEC 61850 a IEC 60870-5-104

Pokud srovnáme obě normy z hlediska komunikace, je zřejmé, že oba standardy jsou vesměs stejné, přestože byly vyvíjeny nezávisle na sobě. Oba definují standard pro přenos dat pomocí protokolu TCP/IP. Pokud zajdeme do hloubky, tak služby a protokoly v reálném čase, informační modely a především programovací jazyk SCL nejsou ve standardu IEC 60870 definovány. Komunikační protokol z normy IEC 61850 by měl také do budoucna nahradit komunikaci pomocí IEC 60870-5-101 a IEC 60870-5-104. Z pohledu komunikace jsou protokoly vzájemně hodně podobné a lze obecně říci, že komunikační protokol normy IEC 61850 je modernizací IEC 60870-5. Zde ale podobnost končí a pokud chceme normy porovnávat i z jiného pohledu jednoduše nemáme co porovnávat, protože možnosti, které nabízí norma IEC 61850, norma IEC 60870 a jiné nenabízejí. Porovnání lze nalézt také v tabulce níže. [5][7][12][13][31]

**Tabulka 5: Porovnání norem pro použití v distribuční soustavě**

| Norma | Komunikace pomocí sériové linky | Komunikace pomocí TCP/IP | Časová synchroniz-ace | Konfigurační jazyk | Horizontální/ Vertikální komunikace |
|---|---|---|---|---|---|
| IEC 61850 | NE | ANO | ANO | ANO | ANO |
| IEC 60870-5-101 | ANO | NE | ANO | NE | NE |
| IEC 60870-5-104 | NE | ANO | ANO | NE | NE |

# 3 Technologické systémy a síť

Obecná definice pro technologické systémy a sítě není obecně uznávaná, proto autor vychází z vlastních zkušeností a obecně tyto systémy a sítě popisuje. Technologické systémy jsou takové systémy, které jsou provozovány a používány pro řízení, ovládání a monitoring distribuční soustavy. Patří sem systémy SCADA, HDO, HMI a jiné řídící systémy včetně prvků na rozvodnách (IED, RTU…).

Běžnou síť si můžeme představit jako síť internetového providera, odkud mohou uživatelé do internetu a nejsou téměř žádným způsobem omezeni. Technologická síť se stejně jako každá jiná síť skládá ze síťových prvků, pracovních stanic, serverů a dalších zařízení. Systémy a komunikační protokoly zmiňované v přechozích kapitolách se používají v rámci technologických systémů. O technologických sítích lze obecně tvrdit, že poskytují komunikaci a prostředky pro běh technologických systémů. Jedná se o uzavřenou a chráněnou síť oddělenou pomocí DMZ od vnějších sítí.

Ochrana technologických systémů a sítí je v první řadě zabezpečena fyzicky, to znamená omezený přístup do serveroven pouze pro vybrané správce (přístup přes kartu nebo přes pin), sledování kamerovým systémem a samotná fyzická ochrana technologických prostor. Fyzická ochrana by měla být řešena centrálním systémem STO (Systém Technické Ochrany), který všechny výše uvedené funkce slučuje.

Další ochrana technologických systémů a sítí je na síťové úrovni, kde přístupy po TCP/IP musí být omezovány pomocí firewallů a to nejen pro vstup do technologické sítě, ale pravidla musí být aplikována také v rámci vnitřní sítě. V tomto případě se jedná o DMZ (demilitarizovaná zóna). Samozřejmě jsou zde také různé úrovně oprávnění pro uživatelské účty.

Ve většině případů technologických sítí je potřeba vzdálená správa technologických systémů. Způsob vzdálené správy se řeší pomocí vzdáleného přístupu VPN. Aby se zajistila co největší bezpečnost, je potřeba, aby vytvořený VPN tunel měl přístup pouze do technologické sítě a vnější komunikace na vzdáleném klientovi byla zakázána. Pro řešení problémů nebo incidentů je vhodné veškerý provoz přes VPN odesílat pomocí logů na log server nebo lépe nahrávat celé spojení. Komunikace VPN spojení musí být šifrovaná.

Všechny systémy (SCADA, HDO, HMI) a zařízení (IED, RTU, HMI) zmiňované v přechozích kapitolách využívají technologickou síťovou infrastrukturu a musejí být chráněny od vnějšího internetu pomocí DMZ. Síť by měla být monitorovaná, především pro snadnější řešení problémů v síti a také proto, aby se předešlo různým problémům s nedostupností služeb. Tato práce se tedy věnuje výhradně technologickým systémům a technologické síti.

# 4 Obecné požadavky na monitorovací systém

Od monitorovacího systému se požaduje především kontrola stavu jednotlivých prvků v síti a případná detekce problémů v síti, na které by měl monitorovací systém reagovat adekvátním způsobem, kterým může být například výpis problému na monitorovací plochu, zaslání notifikace administrátorovi systému pomocí SMS nebo emailem. Od monitorovacích systémů se neočekává, že budou do problému zasahovat, případně problém jakýmkoliv způsobem analyzovat. K tomu slouží ve většině případů jiné systémy, které by měly být spolu s monitorovacím systémem nasazeny. Tato funkce však není u monitorovacích systémů vyloučena. Monitorovací systémy mají především upozornit na daný problém, aby ho mohl administrátor systému, pracovník dohledového centra nebo zodpovědná osoba řešit.

Obecně o monitorovacích systémech existuje spousta článků, zabývajících se jejich problematikou. Autoři v článku [32] popisují obecně typy a topologii jednotlivých monitorovacích systémů. V článku popisují, jakým způsobem by měl být monitorovací systém testován. Monitorovací systémy lze obecně rozdělit na základní, rozšířené a proaktivní monitorovací systémy. V článku [32] je autory popsán monitorovací systém typicky pracující s protokolem ICMP pro základní zajištění dostupnosti daného prvku v síti. To je však pro komplexnější monitoring sítě nedostačující. Proto jsou v článku uvedeny rozšířené monitorovací systémy, které umějí pracovat s protokoly jako jsou SNMP, CDP, SSH atd.. Systém, který umí pracovat s těmito protokoly, je již pro monitoring sítě dostačující a lze s ním monitorovat všechny hodnoty, které z daného prvku v síti lze vyčíst. V praxi se však monitorují pouze relevantní hodnoty, které jsou pro dané prostředí užitečné. Autoři [32] zde uvádějí také proaktivní monitorovací systémy, které jsou schopny zasahovat do konfigurací prvků v síti. Proaktivním monitorovacím systémem je například systém NETBRAIN.

Architekturu monitorovacích systémů lze podle článku [32] rozdělit na dvě základní architektury, centralizované a decentralizované monitorovací systémy. Centralizovaný systém je takový systém, který je nainstalovaný pouze na jednom místě a veškeré jeho systémy jsou spuštěny pouze na jednom serveru. Centralizovaný monitorovací systém se obecně hodí na monitoring menších sítí. Jeho výhodou je jednoduchost a rychlost implementace při instalaci. Je také vhodný pro otestování funkcí decentralizovaného systému.

Pro rozsáhlejší sítě je vhodný decentralizovaný monitorovací systém, který je možné rozdělit do několika lokalit a není nutné se spoléhat na jeden systém v celé síti. Příkladem může být vetší společnost, která má rozdělené své působení do několika lokalit. Každá z těchto lokalit je monitorovaná lokálním systémem a naměřená data jsou pak zasílána do centrálního bodu monitorovacího systému. V případě výpadku jedné lokality jsou data z ostatních lokalit stále zasílána do centrálního bodu. Naopak v případě výpadku

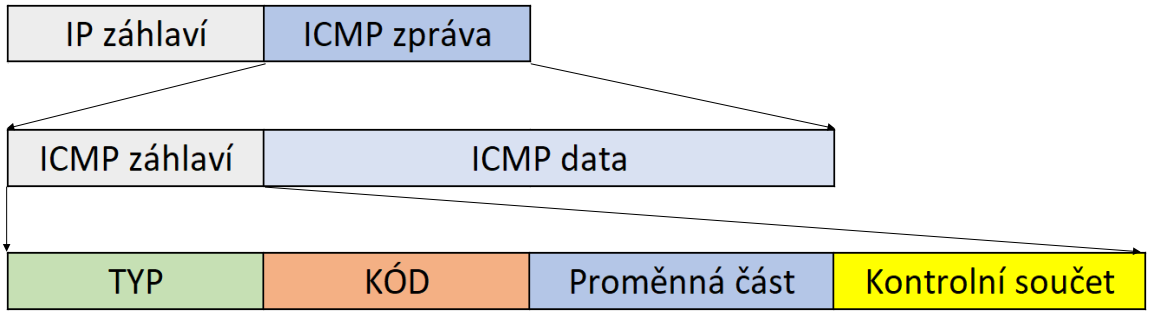
centrálního bodu jsou data v jednotlivých lokalitách stále sbírána a po zpřístupnění centrálního bodu zaslána pro zpracování. Většinu známých monitorovacích systémů je možné nainstalovat jak centralizovaně, tak decentralizovaně. Samozřejmě volba dané architektury záleží především na daném prostředí a na tom, co vše je třeba monitorovat. [32]

## 4.1 Protokoly používané v monitorovacích systémech

Základní protokoly pro monitorování jsou ICMP, SNMP, SSH a mnoho dalších. ICMP protokol je základní protokol architektury TCP/IP (Internet Control Message Protocol) a má hned několik využití. V IP síti je tento protokol využívaný například pro oznamování nedostupnosti prvku v síti a jiné odesílání služebních informací. V případě použití v některém z monitorovacích systémů s ním lze monitorovat dostupnost zařízení, odezvu na dané zařízení a kolísání zpoždění.

### 4.1.1 Struktura protokolu ICMP

Protokol ICMP generuje žádosti a odpovídá na ně. Tyto zprávy jsou vkládány za záhlaví IP datagramu. Protokol ICMP odesílá zprávy typu Echo Request, Echo Reply, Destination Unrechable, Redirect a podobně. O jaký typ zprávy se jedná, je definováno v poli TYP v hlavičce ICMP zprávy. Obrázek 6 níže zobrazuje formát protokolu ICMP. [33][34][35][36]




**Obrázek 6: Formát ICMP zprávy**

- **TYP (1B)** – pole typ definuje typ zprávy ICMP. Například pokud v tomto poli bude typ 8, bude se jednat o zprávu ICMP Echo Request.
- **KÓD (1B)** – kód ve struktuře protokolu blíže specifikuje typ zprávy ICMP. Například zpráva Destination Unrechable má více podtypů. Kódem zprávy se blíže vyspecikuje podtyp této zprávy, který může být například Destination port Unrechable, Destination host unrechable a podobně.
- **Proměnná část (4B)** – definuje další atributy zprávy ICMP, které jsou závislé na poli TYP a KÓD.

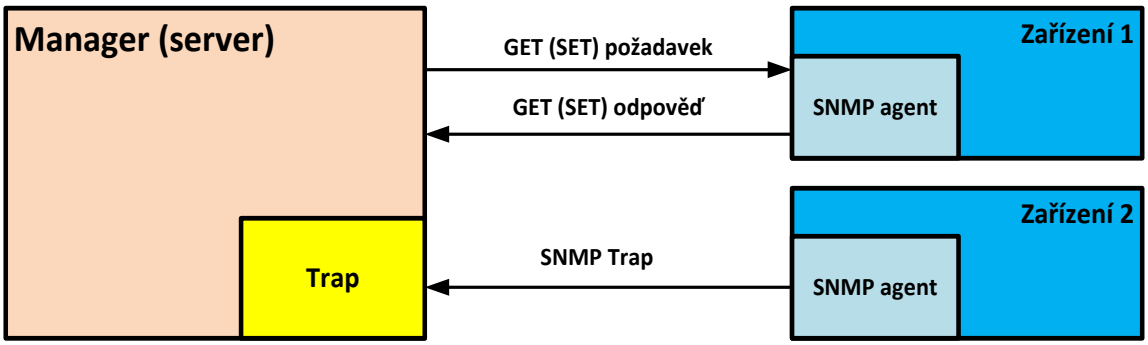
- **Kontrolní součet (2B)** – kontrolní součet ICMP zprávy vypočítaný ze záhlaví a dat zprávy ICMP. Kontrolní součet zajišťuje ochranu před poškozením zprávy. [33][34][35][36]

### 4.1.2 Simple Network Management Protocol

SNMP (Simple Network Management Protocol) protokol slouží pro sběr dat z prvků v síti. Z hlediska monitoringu sítě je to základní a jeden z nejužitečnějších protokolů pro sběr informací. Protokol je standardizovaný a podporuje jej široká škála zařízení jako jsou aktivní síťové prvky (switche, routery, FW), různé OS a jiná síťová zařízení. Pro správné vyčítání informací je třeba nastavit komunitu (angl. community). Tato komunita může být jak read-only (RO) tak read-write (RW). Je tedy možné SNMP používat v proaktivních monitorovacích systémech pro zásah do konfigurace. V problematice SNMP figurují dvě entity, kde jednou z nich je manager, což je server, který získává data ze zařízení. Druhou entitou je agent, který odesílá informace ze zařízení. [37][39][40]

Informace ze zařízení lze získávat dvěma způsoby. Prvním způsobem je vyžádání dat **GET request**/požadavek, se správnou komunitou ze strany manageru (serveru), kde zařízení, na kterém je spuštěn SNMP agent na základě přijatých OID (vysvětleno bude později v této kapitole) odešle serveru informace **GET response**/odpověď. Druhým způsobem získávání informací je zaslání informací od síťového prvku pomocí SNMP agenta k serveru pomocí **SNMP Trap**. Od SNMPv2 odpovídá server na zprávu SNMP Trap zprávou **Inform**. Výhodou druhé možnosti je, že při vzniku problému na straně monitorovaného prvku vzniká velmi nízké zpoždění mezi vznikem a zjištěním problému. V případě vyžádání této informace od serveru by zde byla prodleva, protože server kontroluje síťový prvek v pravidelných intervalech dle nastavení. Obrázek 7 uvádí příklad obou komunikací. [37][39][40]

Požadavky **SET request** upravují hodnoty jedné nebo více OID v seznamu. Tento požadavek je vysílán od managera k agentovi, který provede požadovanou úpravu a na tuto zprávou odpovídá zprávou **SET response**. Pomocí požadavků SET lze měnit konfiguraci zařízení.




**Obrázek 7: Příklad komunikace mezi SNMP agentem a managerem**

Dalšími typy zpráv, kterými může server požádat agenta o informace je požadavek **GETNEXT**, který vyčte pouze jednu následující hodnotu od zadaného OID (viz výpis z konzole níže GETNEXT). Tento dotaz vypisuje hodnoty lexiograficky navazující na předchozí hodnoty OID. **GETBULK**, dostupný od SNMPv2, optimalizuje dotaz GETNEXT tím způsobem, že aplikuje požadavek GETNEXT vícekrát v jednom požadavku. Lze tedy pomocí něho vyčíst zbytek stromu od zadaného OID (viz výpis z konzole níže GETBULK). Pro lepší pochopení přidávám výpisy z konzole pro oba požadavky. [37][39][40]

**GETNEXT:**

```
snmpgetnext -c ZBXGETRO -v 2c 10.252.129.5 ifDescr.83886080
IF-MIB::ifDescr.151060481 = STRING: Vlan1
```

**GETBULK:**

```
snmpbulkget -c SNMPCOMM -v 2c 10.252.129.5 ifDescr.83886080
IF-MIB::ifDescr.151060481 = STRING: Vlan1
IF-MIB::ifDescr.151060492 = STRING: Vlan12
IF-MIB::ifDescr.151060493 = STRING: Vlan13
IF-MIB::ifDescr.151060494 = STRING: Vlan14
IF-MIB::ifDescr.151060495 = STRING: Vlan15
IF-MIB::ifDescr.151060496 = STRING: Vlan16
IF-MIB::ifDescr.151060497 = STRING: Vlan17
IF-MIB::ifDescr.151060498 = STRING: Vlan18
IF-MIB::ifDescr.151060499 = STRING: Vlan19
IF-MIB::ifDescr.151060500 = STRING: Vlan20
```

Protokol SNMP byl postupem času vyvinut do několika verzí, kterými jsou verze SNMPv1, SNMPv2c, u kterých je nutné použít SNMP komunitu pro ověření komunikace pro vyčtení žádaných informací. Verze protokolu SNMPv3, kde komunikace pomocí této verze je šifrovaná, využívá navíc oproti předchozím verzím ověřování uživatelským jménem a heslem. SNMP využívá pro přenos informací protokol UDP. Standardní

komunikační protokol pro komunikaci ze serveru na SNMP agenta je port 161 a komunikace pro zasílání informací pomocí SNMP agenta na server (SNMP Trap) využívá port 162. Obrázek 8 zobrazuje strukturu protokolu. [37][39][40]


**Obrázek 8: Struktura protokolu SNMP**

V celkové struktuře protokolu jsou pro nás důležité především pole verze, komunita a PDU typ.

- **Verze** – definuje použitou verzi SNMP protokolu popsané výše.
- **Komunita** – v tomto poli je uvedena použitá komunita, kterou musí mít stejnou manager i dané zařízení (resp. SNMP agent).
- **PDU typ** – toto pole určuje požadavek protokolu SNMP. Jedná se tedy o Get/Set request, Get Next response, Get Bulk Request, Trap a Inform. [37][39][40]

**Management Information Base**

Každé zařízení, které podporuje SNMP, má ve své paměti uloženou databázi MIB (Management Information Base). Tato databáze se označuje jako MIB, MIB tabulka nebo MIB strom. Databáze obsahuje seznam všech objektů OID, ze kterých je schopna vyčítat data. Jedná se o užitečné informace jako stav paměti, stav procesoru, vytížení procesoru atd. Záleží na konkrétním zařízení. Ve většině monitorovaných zařízení lze MIB databázi vypsat například příkazem „snmpwalk“ v linuxu nebo ji lze stáhnout z oficiálních stránek výrobce. Obrázek 9 uvádí příklad souboru MIB, konkrétně HOST-RESOURCES-MIB. Soubory MIB lze do systému běžně naimportovat, tyto soubory však musí být ve správném formátu a začátek souboru musí vždy obsahovat název MIB + DEFINITIONS ::= BEGIN, následuje obsah souboru MIB a konec souboru je ukončen END:

```
HOST-RESOURCES-MIB DEFINITIONS ::= BEGIN
        .
        .
        .
END
```

V praxi se však zřídka setkáme se soubory, které by nebyly v uvedeném formátu. Toto je vhodné vědět pro případné kopírování obsahu souboru do systému nebo pro případný troubleshooting v případě nefunkčnosti daných MIB. [41][42]

```
-- The Host Resources System Group

hrSystemUptime OBJECT-TYPE
    SYNTAX     TimeTicks
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The amount of time since this host was last
        initialized.  Note that this is different from
        sysUpTime in the SNMPv2-MIB [RFC1907] because
        sysUpTime is the uptime of the network management
        portion of the system."
    ::= { hrSystem 1 }

hrSystemDate OBJECT-TYPE
    SYNTAX     DateAndTime
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The host's notion of the local date and time of day."
    ::= { hrSystem 2 }

hrSystemInitialLoadDevice OBJECT-TYPE
    SYNTAX     Integer32 (1..2147483647)
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The index of the hrDeviceEntry for the device from
        which this host is configured to load its initial
        operating system configuration (i.e., which operating
        system code and/or boot parameters).

        Note that writing to this object just changes the
        configuration that will be used the next time the
        operating system is loaded and does not actually cause
        the reload to occur."
    ::= { hrSystem 3 }

hrSystemInitialLoadParameters OBJECT-TYPE
    SYNTAX     InternationalDisplayString (SIZE (0..128))
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "This object contains the parameters (e.g. a pathname
        and parameter) supplied to the load device when
        requesting the initial operating system configuration
        from that device.
```

**Obrázek 9: Příklad obsahu souboru MIB HOST-RESOURCES-MIB[41]**

Každé zařízení, které podporuje SNMP, disponuje alespoň jednou databází MIB, kterou lze vyčíst. Pokud však server, ze kterého pomocí příkazu „snmpwalk“ chceme tyto informace vyčíst, dané MIB neobsahuje, příkazem informace automaticky nevyčteme. Jedním způsobem, jak informace ze zařízení vyčíst, je získání daného číselného OID a informace tak vyčíst. Druhou, mnohem efektivnější možností je import potřebných MIB souborů přímo na server, odkud informace vyčítáme a jejich následné použití.

### OID

Získaná data pomocí SNMP jsou sama o sobě nic neříkající, proto existuje tzv. Management Information Management (MIB) popisující strukturu získaných dat. Soubor MIB popisuje hodnoty, které jsou definovany jako OID (Object Identifier). OID může být například ve tvaru 1.3.6.1.4.1.9.9.117.1.1.2, kde je samozřejmě tato hodnota nic neříkající a pomocí příkazu „snmpwalk“ v OS Linux vyčteme pouze hodnotu. V případě,

kdy máme v systému zavedeny dané soubory MIB, tak OID uvádíme ve tvaru HOST-RESOURCES-MIB::hrSystemInitialLoadDevice. Význam tohoto OID najdeme v tabulce MIB, kde také zjistíme, jakých hodnot může toto OID nabývat a o jaký typ informace se jedná (INT, STRING…). OID (Object IDdentifier) označuje jeden konkrétní objekt v databázi MIB, kterým je například teplota zařízení, vytížení procesoru a mnoho dalšího.

### 4.1.3 SSH

SSH (Secure Shell) je protokol transportní vrstvy, který disponuje silným šifrováním, kryptografií autentizace a ochranou integrity při přenosu. Tento protokol se primárně používá pro šifrovaný přístup pro vzdálenou konfiguraci zařízení a nahrazuje nezabezpečený protokol telnet. U monitorovacích systémů, lze tento protokol využít pro získání potřebných informací obdobně jako u SNMP. Z hlediska bezpečnosti je však nutné, aby účet, který se pro tento účel požívá, měl omezená práva. Tento problém lze řešit například založením lokálního uživatele v systému s právy read-only. Pokud organizace používá doménové účty, lze využít například servisní účet, který bude zařazen do skupiny s omezenými právy. Z bezpečnostního hlediska není vhodné SSH používat pro monitoring, protože zde existuje potencionální riziko zneužití tohoto účtu v případě získání přístupových údajů útočníkem.[38]

### 4.1.4 Analýza provozní historie zařízení

Na monitoring se lze dívat více způsoby. Výše je zmíněný monitoring zařízení z hlediska jejich HW, dostupnosti a provozních informací jako datový tok na portech atd.. Určitým způsobem lze monitoring provádět na úrovni analýzy provozu v síti, přístupů a změn v konfiguraci zařízení a analýzy událostí v systémech. Tyto informace jsou zasílány například pomocí logů na server, který tato data analyzuje a na základě nastavených pravidel informuje správce o neobvyklých událostech v síti. Systémů, které takovou analýzu umožňují je velké množství, kde mezi jejich zástupce patří například řešení SIEM (Security Information and Event Management).

# 5 Výběr vhodného monitorovacího systému

Technologické systémy (SCADA, HMI, řídící systémy) zpravidla používají komunikaci pomocí TCP/IP. Tyto systémy jsou od rozvodny vzdáleny několik desítek nebo stovek kilometrů (kromě lokálních dispečinků HMI) a datový přenos je realizován pomocí TCP/IP. V rámci možností, je snaha při upgradu rozvoden o nahrazení komunikace sériové linky komunikací TCP/IP, která je následně použitá pro komunikaci mezi systémy technologické sítě. Z praxe však není tento upgrade vždy realizovatelný, proto se v některých případech volí možnost převodníků sériové komunikace na ethernet.

Samotná IED, RTU a jiná zařízení na rozvodnách monitoruje systém SCADA nebo HMI a do jisté míry jsou těmito systémy monitorovány některé síťové prvky. Úkolem této diplomové práce je navrhnout a zprovoznit dohledový systém, který bude separátně od systému SCADA monitorovat síťové prvky v technologické síti, jako jsou především routery, switche a jiné síťové prvky. Jedná se o sledování parametrů, kterými jsou propustnost na jednotlivých portech switche, monitoring a ohlašování změny datového toku pod/nad určenou úroveň, sledování jednotlivých procesů spuštěných na serveru a mnoho dalšího.

Pro monitoring síťových zařízení jsou vybrány tři vhodné systémy, které budou v této části práce popsány a na závěr porovnány jejich vlastnosti, přednosti a možnosti. Na základě porovnání bude vybrán nejvhodnější systém pro implementaci do testovacího prostředí. Porovnávat se zde budou monitorovací systémy Zabbix, Icinga a Nagios.

Rozbor distribuční sítě v předchozích kapitolách jasně vypovídá o tom, kterými systémy je distribuční síť řízena a monitorována. Není zde však žádný komplexní monitoring sítě a síťovích prvků. Tato práce by měla především navrhnout způsob, jak tento nedostatek vyplnit.

## 5.1 Které veličiny je vhodné monitorovat

Z vlastní praxe autora je určeno několik hlavních veličin, které je vhodné monitorovat. Samozřejmě musíme vzít v úvahu, že pro každé síťové zařízení bude ve výsledku trochu jiný seznam monitorovaných veličin. Seznam vhodných veličin je popsán níže:

- Dostupnost zařízení, ať už se jedná o fyzické zařízení nebo virtuální server.
- Dostupnost jednotlivých služeb. To znamená především, zda je daná služba spuštěna či nikoliv. Může se jednat o web server, různé agenty, databáze a podobně. Vhodné je také monitorovat, zda služby nebyly restartovány.
- Monitoring datové propustnosti na jednotlivých rozhraních switche/routeru, jejich stavy a změny těchto stavů, sledování drop-rate na rozhraní, rychlost daného rozhraní.

- Z hlediska provozu je pak vhodné monitorovat uptime systému, teplotu zařízení a okolního vzduchu, stav hlavního a záložního napájení, pokud jím zařízení disponuje nebo stavy chlazení zařízení.
- Pro monitoring výkonu síťových prvků je pak vhodné monitorovat velikost volného a využitého místa na disku, vytížení procesorů nebo jednotlivých jader, velikost a využití pamětí RAM.
- Vyčítání informativních hodnot jako jsou umístění zařízení, název, verze OS nebo počet přihlášených uživatelů.
- Z bezpečnostního hlediska pak sledovat změny souborů, například /etc/passwd.

Na základě těchto dat pak spouštět různé triggery a alarmy s příslušnou severitou a upozornit tak dohledového pracovníka o problému. Může se jednat například o pokles datového toku na určitém portu pod/nad nastavenou úroveň, nebo reboot zařízení, restart služby nebo změna stavu rozhraní.

## 5.2 Zabbix

Zabbix jako jeden z mála monitorovacích systémů poskytuje své rozhraní v českém jazyce. Tato výhoda však není podmínkou pro naší implementaci. Při správném nastavení tzn. dostatečný výkon nebo použitá databáze, je schopný monitorovat až 100 000 zařízení. Zabbix je Open-Source monitorovací systém a většina omezení připadá na samotného administrátora systému, jak s daným systémem bude pracovat a co vše na něm bude schopný využít. Support je v první řadě řešen pomocí oficiálního fóra, případně pomocí Wiki, kterou mají vývojáři pěkně zpracovanou. Ve Wiki, je poměrně dopodrobna popsán postup instalace/konfigurace Zabbix serveru. Placený Support je rozdělen do několika úrovní Bronze, Silver, Gold, Platinum a Enterprise.

Architekturu Zabbixu lze rozdělit do několika úrovní. Jedná se o samotný Zabbix server, databázi, proxy server a GUI (Front-End). Každá z těchto úrovní může být nainstalována separátně nebo mohou být na jednom serveru. Může být tedy vhodné oddělit například samotnou databázi a server především pro případ havárie jednoho ze serverů. Při volbě jakéhokoliv způsobu instalace, je vždy vhodné provádět zálohy veškerých důležitých částí systému, jimiž jsou především server a databáze. Níže jsou popsány funkce a vlastnosti jednotlivých úrovní: [43][44][45][54]

- **Zabbix server** – jádro monitorovacího systému. Vyhodnocuje data, která přijal na základě kritérií nastavených v Zabbix GUI. Server lze nainstalovat na operační systémy Linux.
- **Zabbix database** – databáze založená na MySQL, InnoDB nebo PostgreSQL, přičemž na oficiálních stránkách distributora, jsou uvedena doporučení na velikost paměti a počtu jader pro optimální chod DB při určitém počtu monitorovaných zařízení. Pokud nefunguje databáze, služba Zabbix serveru

je stále spuštěna, ale server není schopný zpracovávat data, protože je nemá kam ukládat.

- **Zabbix proxy –** proxy je určená především proto, aby odlehčila samotnému serveru při zpracování dat. Proxy tedy může zpracovávat data například z jedné lokality a může tak ulehčit HW zátěži serveru. Proxy používá pro ukládání zpracovaných dat svou vlastní databázi. Server následně na zpracovaná data aplikuje pravidla nastavená v GUI. [43][44][45][54]
- **Zabbix GUI –** nebo také Front-End je grafické rozhraní, kde jsou zobrazeny veškeré události, které se aktuálně dějí (monitoring sítě). Dále je zde prováděna veškerá konfigurace pravidel pro monitoring, vytváření šablon pro monitoring a přidávání hostů.
- **Zabbix API –** API slouží pro doprogramování jistých funkcionalit, které by administrátorovi chyběly. Příkladem může být propojení databáze s jinými systémy.

Možností, kterými lze monitorovat síťová zařízení je mnoho. Základní vlastnosti monitorovacího systému Zabbix jsou následující:

- Sběr dat lze pak provádět pomocí SNMPv1, SNMPv2, SNMPv3, SSH, Telnet, Zabbix agent a dalších možností.
- Real-time monitoring zobrazovaný ve Front-Endu (Dashboard).
- Notifikace emailem, SMS nebo pomocí Jabber.
- Velká přizpůsobivost systému na základě konfigurace administrátora. Schopnost zpracovávat více než 50 000 aktivních kontrol za vteřinu (podmínka vysokého výkonu serveru).
- Vytváření map a závislostí jednotlivých zařízení.
- Grafy naměřených dat (upload/download) u jednotlivých zařízení.
- Další vlastnosti a funkce lze nalézt v [43][44][45][54].

## 5.3 Nagios

Nagios je monitorovací systém, který je stejně jako předchozí systém určený pro monitoring sítě. Stejně jako v případě Zabbix serveru, schopnost monitorování velkého počtu zařízení v sítí je spíše omezená dostupným HW výkonem než samotným systémem. Nagios je stejně jako předchozí systém Open-Source s tím, že je rozdělen do dvou rozdílných distribucí Nagios Core a Nagios XI. Nagios Core je volně šiřitelný pod licencí GPL, ale má omezené funkce oproti Nagios XI, který je zdarma pouze na 60 dní (pro max. 7 monitorovaných zařízení je neomezená licence zdarma). Nejlevnější varianta začíná na $1995. Stejně jako v případě Zabbix serveru, jeho support je řešen oficiálním fórem Nagiosu a předplaceným Supportem přímo od Nagiosu. Oficiální distributor vytvořil svou vlastní Wiki, kde jsou veškeré důležité návody. [46][47][48][49][50][54]

Architekturu monitorovacího systému Nagios lze rozdělit stejně jako v případě Zabbix serveru do několika úrovní. Jedná se o samotný Nagios (Core, XI), pluginy vyvíjeny jak samotnými vývojáři Nagiosu tak komunitou a databáze. Popis funkcí a vlastností jednotlivých úrovní:

- **Nagios (Core, XI)** – stejně jako u Zabbixu nejdůležitější část monitorovacího systému, bez kterého nemůže fungovat. Udržují se na něm nastavené konfigurace nebo zpracovává přijatá data.
- **Pluginy** – slouží k doplnění funkcí Nagiosu, které v základu nemá. Může se jednat o možnost vytvoření grafů, vytváření mapových podkladů a grafické znázornění síťových propojení, přes úpravu a customizaci dashboardu, až po monitoring různých síťových prvků.
- **Nagios databáze** – databáze, které může Nagios používat, jsou MySQL/MariaDB a PostgreSQL. Obdobně jako v předchozím případě jsou na oficiálním serveru parametry na HW požadavky serveru, především se jedná o počet jader a paměť RAM serveru.

Základní vlastnosti monitorovacího systému Nagios XI jsou popsány níže:

- Sběr dat lze provádět pomocí SNMPv1, SNMPv2, SNMPv3, SSH, Telnet a dalších možností.
- Dashboard pro zobrazení real-time monitoringu.
- Notifikace emailem nebo SMS, případně jiným způsobem pomocí Pluginu.
- Možnosti automatického vytvoření grafické infrastruktury.
- Zobrazování grafů naměřených hodnot (upload/download).
- Další vlastnosti lze nalézt v [46][47][48][49][50][54].

## 5.4 Icinga2

Icinga2 je obdobně jako předchozí monitorovací systémy Open-Source pod licencí GNU GPLv2, který monitoruje téměř jakékoliv síťové zařízení, kontroluje jejich dostupnost, případně informuje uživatele o nedostupnosti daného zařízení nebo generuje naměřená data ve formě grafů (upload/download) a podobně. Oproti předchozím dvěma monitorovacím systémům je Icinga2 navržena tak, aby si ji mohl uživatel přizpůsobit přesně svým požadavkům. Prostředí Icinga2 je programovatelné a mělo by být schopné se přizpůsobit jakémukoliv požadavku na funkčnost, jako může být například propojení s jiným systémem pro správu uživatelů v sítí a jiné. Samotná Icinga2 je napsaná v jazyce C++ a obsahuje moduly, které jsou programovatelné a vyžadují znalost programovacího jazyka PHP, Perl, Python nebo Ruby. Support je řešen podobně jako u předchozích systémů a to buď přes oficiální fórum nebo přes placenou profesionální podporu přímo od Icinga2. [51][52][53]

Obdobně jako předchozí systémy, Icinga2 se dělí na několik hlavních částí, jimiž jsou:

- **Icinga2 serve**r – stejně jako u předchozích dvou systémů se u Icinga2 jedná o tu nejdůležitější část systému, bez které by systém nefungoval. Toto jádro systému Icinga2 lze nainstalovat na několik různých distribucí Linuxových systémů. Na systém Windows tento server nainstalovat nelze.
- **Icinga Web 2** – jedná se o webovou část systému, tzv. Front-end. V této části systému jsou zobrazovány veškeré naměřené hodnoty, dostupnost systémů a podobně.
- **Databáze** – velmi důležitá část systému. Databáze může běžet na MySQL nebo na PostgreSQL. Obdobně jako v předchozích systémech s rostoucím množstvím monitorovaných zařízení, stoupá náročnost na HW požadavky.
- **API** – využívá se především pro vytváření nových aplikací, které pracují s Icinga2. API je nutné použít i v případě integrace do grafů. Dále je to velmi mocný nástroj k propojení Icinga2 s jinými systémy. Jako příklad může být propojení s ticketovacím systém a v případě problému automaticky vytvořit ticket.
- **Moduly** – Icinga2 v základní instalaci neobsahuje vše a například tvorba grafů nebo vytváření map je potřeba do systému doplnit pomocí modulů. Seznam některých modulů lze dohledat na stránce https://www.icinga.com/products/icinga-web-2-modules/. [51][52][53]

Základní vlastnosti monitorovacího systému Icinga2 jsou následující:

- Sběr dat je možný pomocí SNMPv1, SNMPv2, SNMPv3, SSH, Telnet, vlastního agenta atp.
- Real-time monitoring, jehož stav je zobrazován na Dashboardu.
- Oznamování nežádoucích stavů pomocí SMS, emailem nebo vlastními řešeními.
- Report pomocí modulu, například do grafického znázornění.
- Další vlastnosti lze nalézt v [51][52][53].

## 5.5 Vyhodnocení monitorovacích systémů

Všechny výše zmíněné systémy splňují požadavky z kapitoly 5.1. Níže je přiložena tabulka se srovnáním jednotlivých vlastností a možností systémů, kterými Zabbix, Nagios XI a Icinga2 disponují.

**Tabulka 6: Srovnání monitorovacích systémů[54]**

| | Open source | Upozornění | Uživatelské rozhraní | Podpora | Tvorba grafů, map a dalších dodatečných funkcí | Cena |
|---|---|---|---|---|---|---|
| **Zabbix** | Ano, licence GNU GPLv2 | Email, SMS nebo vlastní | Ano | Komunita, fórum, Zabbix wiki, HelpDesk, telefon | Ano | Zdarma |
| **Nagios XI** | Ano, licence GPL | Email, SMS nebo vlastní | Ano, ale v základní verzi Nagios Core omezené možnosti | Komunita, Placená podpora, Fórum, Wiki | Ano ale v základní verzi Nagios Core omezené možnosti, případně instalací plugynů | Zdarma pouze pro 7 zařízení nebo 60 dní |
| **Icinga** | Ano, licence GNU GPLv2 | Email, SMS nebo vlastní | Ano | Komunita, placená podpora, fórum | Ano - omezenější možnosti v základu, lze řešit instalací modulů | Zdarma |

Každý ze systémů disponuje potřebnými vlastnostmi, z nichž některé z nich mohou být přidány různými způsoby. Jedním z těchto způsobů může být aplikace grafického znázornění Icinga2 vs Zabbix. Zatímco Zabbix má již tyto funkce integrované, v systému Icinga2 je potřeba je doinstalovat ve formě modulu/addonu. Nagios XI disponuje stejnými vlastnostmi jako Zabbix a Icinga2, problém však nastává v případě monitoringu pro více jak 7 zařízení, kde následuje placená verze začínající na $1,995. [54]

Pro následnou implementaci do testovacího prostředí je tedy zvolen monitorovací systém Zabbix, který z tabulky výše vychází s nejlepším hodnocením. Pro instalaci tohoto systému je však třeba dobrá znalost systému Linux z důvodu, že Zabbix Server nelze nainstalovat na Windows Server. Složitost systému Zabbix oproti předchozím dvěma je irelevantní, přestože má nejsložitější rozhraní pro samotnou konfiguraci. I přes tyto překážky je systém zcela ideální pro implementaci do testovacího, případně produkčního prostředí technologických systémů, především proto, že není nutná žádná dodatečná instalace plugynů, či modulů. Výhodou je samozřejmě také to, že systém je volně šiřitelný a není nijak omezený z hlediska počtu monitorovaných zařízení.[54]

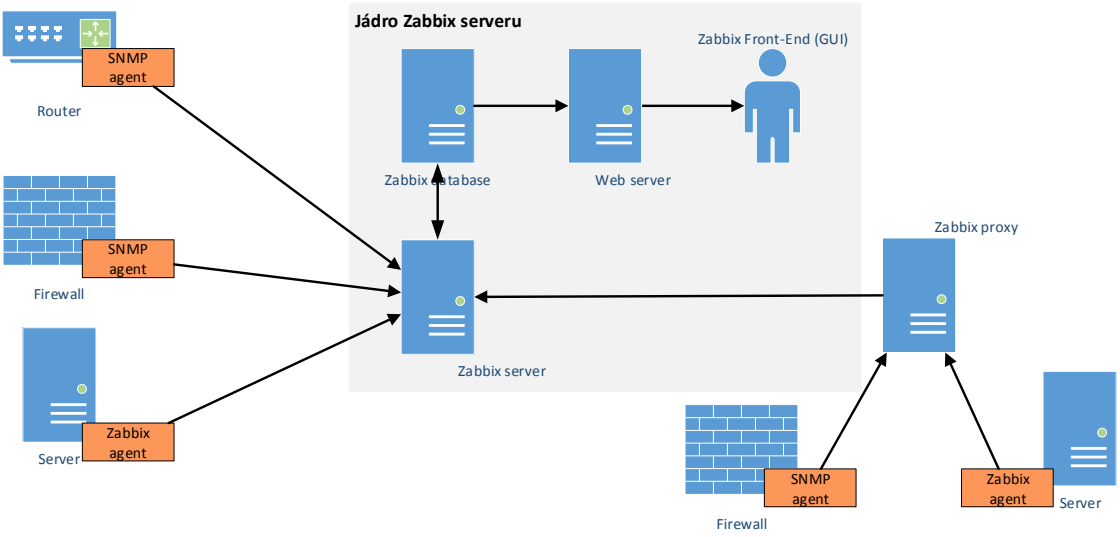
# 6 Implementace monitorovacího systému Zabbix

Monitorovací systém Zabbix je vedený jako Open-Source pod licencí GNU GPLv2. Open-Source znamená, že jeho zdrojový kód je veřejně dostupný a může být upravován jinými vývojáři. Instalace Zabbix serveru v této práci bude probíhat na OS Linux Centos 7 a pro testovací účely je zvolena SQL databázi Maria DB. V této práci bude probíhat monitorování pomocí Zabbix agenta, který je vhodný pro monitoring jak OS Linux, tak Windows. Práce se dále věnuje monitorování síťových zařízení, jako jsou servery, switche, routery, datová úložiště a jiná zařízení, která podporují SNMP.

V další části praktické ukázky monitorovacího systému bude uvedeno vytváření templatů pro nastavení monitorovaných hodnot jak u Zabbix agenta, tak pomocí SNMP. Vytváření triggerů a grafů pro zobrazování naměřených provozních dat je také nedílnou součástí této práce, která bude předvedena spolu s vytvářením templatů, které s triggery a grafy úzce souvisí.

Jak již bylo zmíněno, tato práce se bude věnovat především monitorování pomocí SNMP a Zabbix agenta. Monitoring pomocí SSH je záměrně vyřazený, vzhledem tomu, že všechny potřebné informace lze získat pomocí výše zmíněných. Protokol je určen především pro vzdálené přihlašování na zařízení a lze jej použít v proaktivních monitorovacích systémech pro úpravu konfigurace. Monitoring pomocí SSH lze zadat dvěma způsoby. Prvním z nich je zadání *username* a *password* a druhým je autentizace pomocí *username* a *veřejného klíče*. Dále je nutná znalost jednotlivých příkazů pro vyčtení potřebných informací. Více o monitoringu pomocí SSH lze nalézt na *https://www.zabbix.com/documentation/3.4/manual/config/items/itemtypes/ssh_checks*.

Přestože Zabbix umožnuje decentralizovanou instalaci systému, v této práci je pro ulehčení a snadnější konfiguraci zvolen centralizovaný způsob instalace. Z centralizovaného systému lze jednoduše vytvořit decentralizovaný pomocí Zabbix Proxy serverů. Lze také samostatně nainstalovat databázi a zmigrovat data do nové databáze a GUI Zabbixu. Obrázek 10 zobrazuje obecnou architekturu monitorovacího systému Zabbix.




**Obrázek 10: Obecná architektura monitorovacího systému Zabbix**

Veškerá konfigurace kromě úprav samotného serveru provádí v GUI Zabbix serveru. Systém je nakonfigurovaný tak, aby ověřoval uživatele vůči doménovému kontroléru pomocí LDAP.

## 6.1 Instalace Zabbix server, DB a GUI

Jako předpoklad instalace Zabbix serveru autor spoléhá na to, že čtenář disponuje nějakými zkušenostmi s OS Linux a nebudou zde uváděny všechny podrobnosti instalace jistých daemonů či doplňků jako jsou Apache, MariaDB, konfigurace firewallu… Instalace Zabbix serveru je prováděna na OS Linux Centos 7. Při samotné kompilaci s největší pravděpodobností narazíte na problém s chybějícími knihovnami, balíčky atd., proto autor předem doporučuje nainstalovat následující balíčky a knihovny, v případě instalace Zabbix serveru s databází na MariaDB:

```
yum install gcc libxml2-devel net-snmp-devel libevent-devel curl
libcurl-devel php php-gd
```

1. Ve virtuálním prostředí nainstalujeme Linux Centos
    - Po instalaci provedeme upgrade serveru
2. Z oficiálních stránek Zabbixu stáhneme balíček. Konkrétně v našem případě se jedná o balíček s verzí Zabbixu Zabbix 3.4 který obsahuje Server, Proxy, Agent a GUI.
    - Po stažení balíček rozbalíme do námi zvolené složky

```
[root@localhost]# tar -xvzf zabbix-3.4.2.tar.gz
```

3. Pro správnou funkci Zabbix procesů je nutné vytvořit speciální účet. Nutnost tohoto účtu je zde uvedena proto, protože pokud je daemon spuštěný z *root* účtu, je tento proces přesunut pod správu uživatele „zabbix“. Samozřejmě je možné

spouštět Zabbix procesy přímo pod uživatelem *root*, ale tato možnost není doporučovaná, především proto, že takto spuštěnými procesy vystavíte server bezpečnostnímu riziku v případě „zadních vrátek“ (uživatel *root* má neomezený přístup) v daném procesu nebo jiném způsobu narušení bezpečnosti serveru.

```
[root@localhost]# groupadd zabbix
[root@localhost]# useradd -g zabbix zabbix
```

4. Dalším krokem je přidání databáze pro Zabbix server s názvem *zabbix*. Máme výběr mezi mnoha různými databázemi a pro tuto instalaci byla zvolena databáze MySQL neboli MariaDB.
    - Jako první vytvoříme databázi *zabbix* s kódováním utf-8 a všechna práva pro tuto databázi dáme uživateli *zabbix*.

```
MariaDB [(none)]> create database zabbix character set
utf8 collate utf8_bin;
MariaDB [(none)]> grant all privileges on zabbix.* to
zabbix@localhost identified by '<password>';
```

    - Dále je potřeba do vytvořené databáze *zabbix* naimportovat několik defaultních schémat pro počáteční funkčnost databáze. Umístění souborů je ve složce, kterou jsme stáhli z oficiálních stránek distributora.

```
[root@localhost]# mysql -uzabbix -p zabbix < schema.sql
[root@localhost]# mysql -uzabbix -p zabbix < images.sql
[root@localhost]# mysql -uzabbix -p zabbix < data.sql
```

5. Nejdůležitějším krokem samotné instalace Zabbix serveru je kompilace databáze. Pro zobrazení veškerých možností kompilace je možné zobrazit pomocí příkazu.

```
./configure—help
```

Samotné minimální doporučení/návrh přímo od tvůrců Zabbixu je kompilace.

```
./configure --enable-server --enable-agent --with-mysql
-- nable-ipv6 --with-net-snmp --with-libcurl --with-libxml2
```

Naše instalace OS Centos byla čistá, proto bylo nutné doinstalovat služby/knihovny, které bránili kompilaci:

```
yum install gcc mariadb-devel libxml2-devel net-snmp-devel
libevent-devel curl curl-devel php-bcmath php-mbstring php-
xmlwriter php-xmlreader php-mysql
```

Poté už následují příkazy *make* a *make install*. Instalace se defaultně provádí do složky */user/local* a to lze změnit parametrem *–prefix*.

6. Důležitým souborem Zabbix serveru je soubor *zabbix_server.conf*. V souboru je uložena veškerá konfigurace, jako je například uživatel a heslo k DB, maximální možný počet spuštěných procesů pro jednotlivé procesy monitoringu a další. Umístění souboru je */etc/zabbix/zabbix-server.conf*.

- Před spuštěním služby zabbix-server je nutné nakonfigurovat *zabbix_server.conf* a zadat do něj výše zmíněné údaje pro DB.
- Po instalaci Zabbix serveru je vhodné nainstalovat také Zabbix agenta, který monitoruje samotný server. Z hlediska bezpečnosti se však doporučuje, aby proces, který spouští daemona zabbix_agent, byl spuštěný pod jiným uživatelem než server, který je spuštěný pod uživatelem zabbix. Zabbix agent pak může být spuštěný například pod uživatelem *zabbix-agent*. Instalaci agenta lze spustit pomocí příkazu:

  ```
  yum install zabbix-agent
  ```

- Spustíme procesy zabbix-server a zabbix-agent a nastavíme, aby se pouštěli po startu serveru

  ```
  systemctl start zabbix-agent
  systemctl enable zabbix-agent
  systemctl start zabbix-server
  systemctl enable zabbix-server
  ```

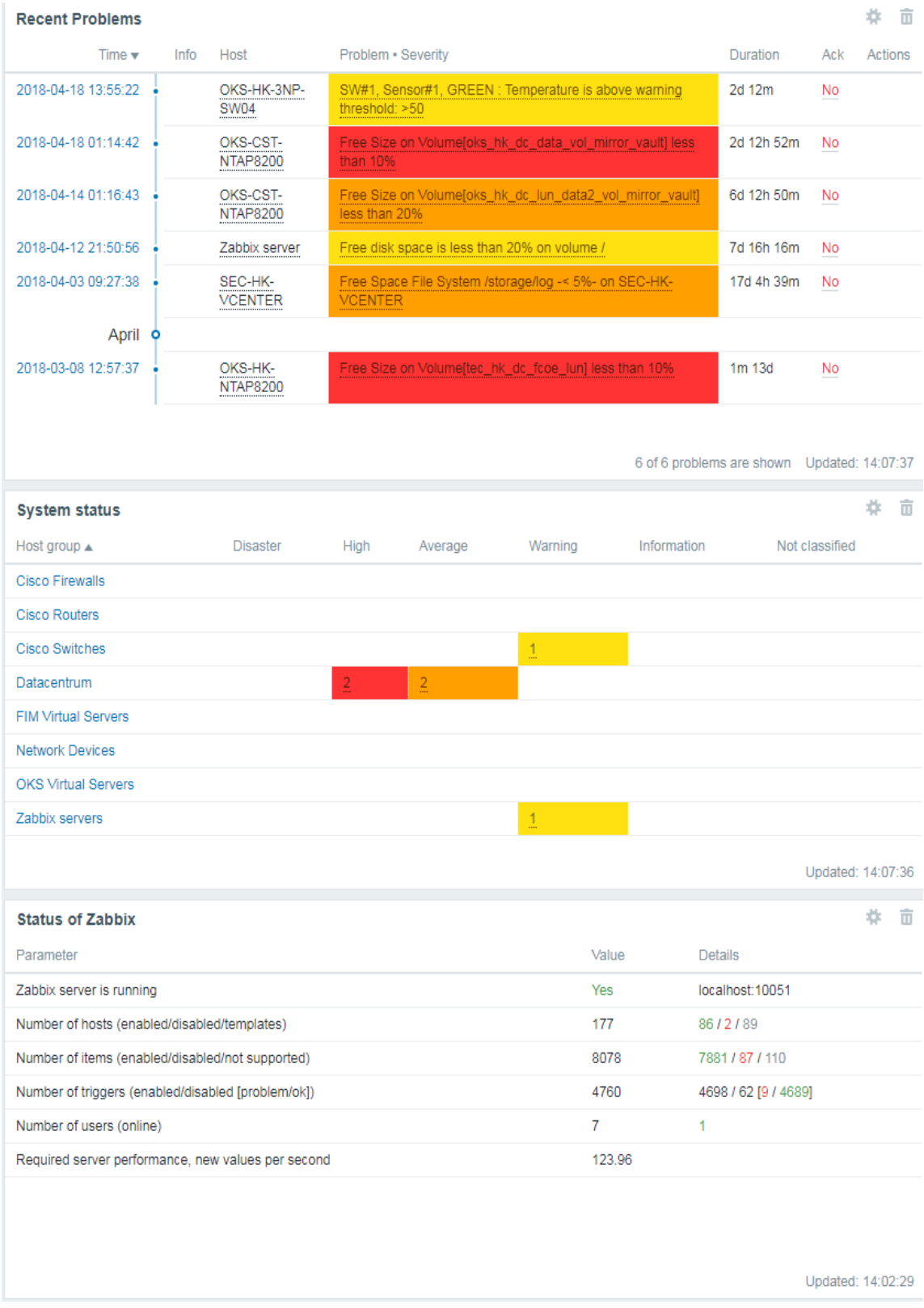
7. Nyní máme nainstalovaný server, prozatím bez GUI. Frontend neboli web rozhraní (GUI) se „instaluje“ jednoduchým způsobem, a to pouze zkopírováním celé složky *frontends/php* ze staženého souboru zabbix-3.4.2. do složky */var/www/html*. Dále je nutné udělit zkopírovaným souborům/složce správná práva (přidělit práva daemonu pro httpd) a také správně nastavit soubor *httpd.conf*. Následně ve webovém prohlížeči otevřeme stránku *http://<adresaServeru>/zabbix* případně jinou adresu dle nastavení *httpd.conf*.
   - První nastavení úvodní stránky je intuitivní a není zde třeba zacházet do detailů.

Po úspěšném nastavení první stránky se nám zobrazí přihlašovací stránka pro webové rozhraní, kde defaultní uživatel je **Admin** a heslo **zabbix**. Po přihlášení se dostaneme na hlavní stránku dashboardu. Obrázek 11 a Obrázek 12 zobrazuje úvodní stránku (Dashboard). Pro další práci se systémem je vhodné nastavit NTP server pro synchronizaci času. OS Linux Centos 7 používá jako defaultní NTP klient balíček Chrony. Jeho konfigurační soubor je umístěn v adresáři */etc/chronyd.conf*, kde stačí pouze upravit IP adresu pro NTP server. V případě privátních adres je nutné doplnit danou IP nebo celý rozsah adres, které budou povoleny pro synchronizaci času. Bez této výjimky nebude klient synchronizovat čas, protože nebude server považovat za důvěryhodný.

Pro následnou orientaci v textu, který bude následovat, jsou zde vysvětleny některé výrazy, které se budou v textu vyskytovat:

- **Item** – jedná se monitorovaný prvek na daném zařízení nebo systému. Za item můžeme považovat například teplotu zařízení.

- **Trigger** – pomocí triggeru se zobrazují dané problémy nebo informace v dashboardu Zabbix serveru. Jejich spuštění je závislé na překročení mezní hodnoty monitorovaného itemu nebo změny hodnoty itemu. Příkladem může být trigger, který je nastaven na item pro měření teploty zařízení. Trigger je nastavený tak, že pokud hodnota v itemu stoupne nad 50 °C, spustí se trigger a zobrazí varování dashboardu.
- **Host** – jedná se zařízení nebo systém, který má přiřazenou IP adresu, název a je přidaný do nějaké skupiny zařízení. Jde tedy přímo o zařízení nebo systém, který chceme monitorovat.
- **Template** – šablona, ve které je nastavený veškerý monitoring pro daný typ zařízení. Obsahuje itemy, nastavení triggerů, jejich závislosti na jiné triggery, případně vytváření grafů, dicovery, automatické vytváření hostů a podobně.
- **Severita** – určuje, jak kritický je problém v nastaveném triggeru. Severitu lze nastavit od úrovně Not classified až po Disaster dle obrázku 11 v tabulce System status. Pro lepší přehled jsou jednotlivé úrovně barevně rozlišeny.
- **Latest data** – zobrazuje poslední získanou/naměřenou hodnotu daného itemu a jejich historii.



**Recent Problems**

| Time ▾ | Info | Host | Problem • Severity | Duration | Ack | Actions |
|---|---|---|---|---|---|---|
| 2018-04-18 13:55:22 | | OKS-HK-3NP-SW04 | SW#1, Sensor#1, GREEN : Temperature is above warning threshold: >50 | 2d 12m | No | |
| 2018-04-18 01:14:42 | | OKS-CST-NTAP8200 | Free Size on Volume[oks_hk_dc_data_vol_mirror_vault] less than 10% | 2d 12h 52m | No | |
| 2018-04-14 01:16:43 | | OKS-CST-NTAP8200 | Free Size on Volume[oks_hk_dc_lun_data2_vol_mirror_vault] less than 20% | 6d 12h 50m | No | |
| 2018-04-12 21:50:56 | | Zabbix server | Free disk space is less than 20% on volume / | 7d 16h 16m | No | |
| 2018-04-03 09:27:38 | | SEC-HK-VCENTER | Free Space File System /storage/log -< 5%- on SEC-HK-VCENTER | 17d 4h 39m | No | |
| April | | | | | | |
| 2018-03-08 12:57:37 | | OKS-HK-NTAP8200 | Free Size on Volume[tec_hk_dc_fcoe_lun] less than 10% | 1m 13d | No | |

6 of 6 problems are shown Updated: 14:07:37

**System status**

| Host group ▲ | Disaster | High | Average | Warning | Information | Not classified |
|---|---|---|---|---|---|---|
| Cisco Firewalls | | | | | | |
| Cisco Routers | | | | | | |
| Cisco Switches | | | | 1 | | |
| Datacentrum | | 2 | 2 | | | |
| FIM Virtual Servers | | | | | | |
| Network Devices | | | | | | |
| OKS Virtual Servers | | | | | | |
| Zabbix servers | | | | 1 | | |

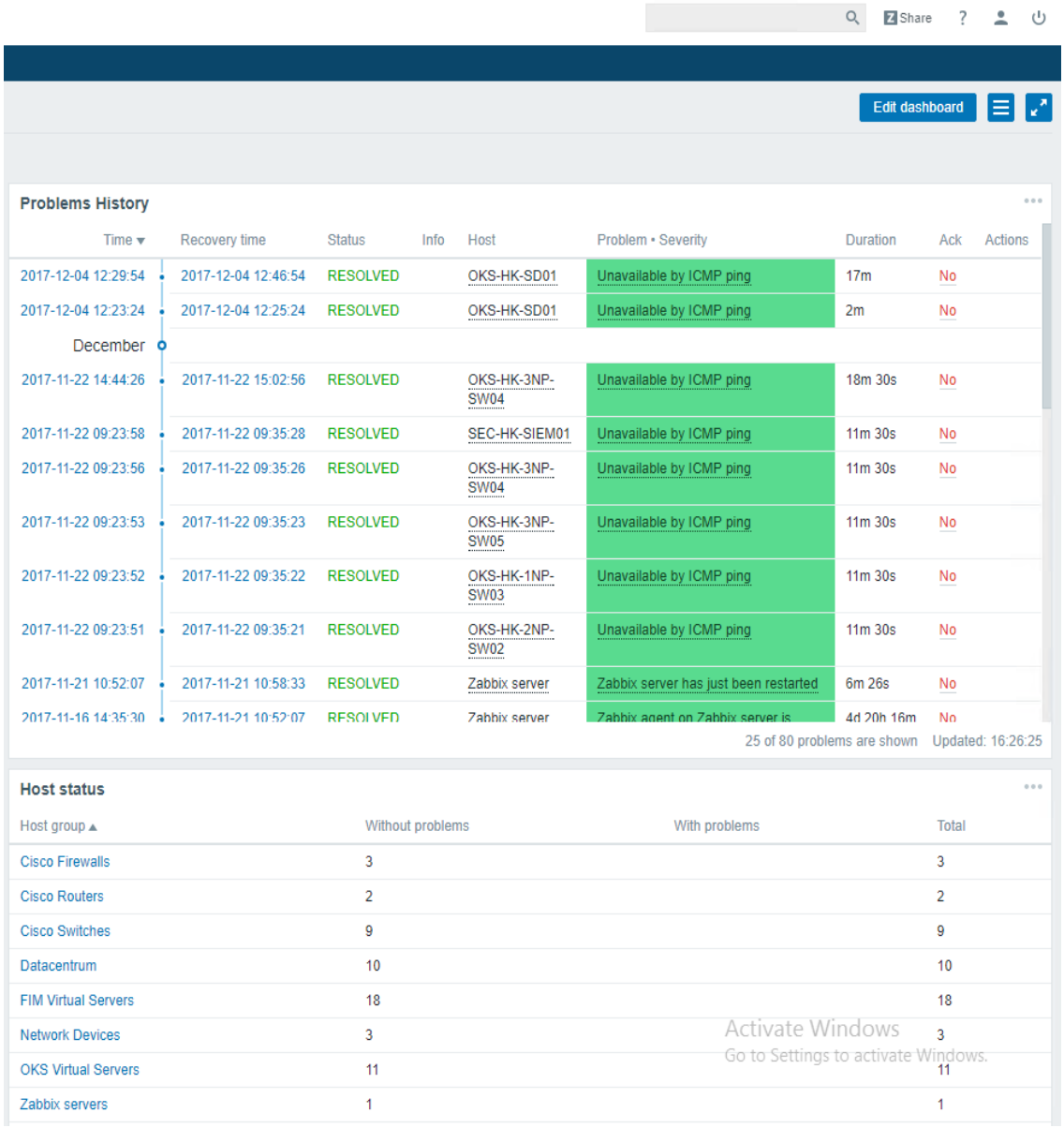
Updated: 14:07:36

**Status of Zabbix**

| Parameter | Value | Details |
|---|---|---|
| Zabbix server is running | Yes | localhost:10051 |
| Number of hosts (enabled/disabled/templates) | 177 | 86 / 2 / 89 |
| Number of items (enabled/disabled/not supported) | 8078 | 7881 / 87 / 110 |
| Number of triggers (enabled/disabled [problem/ok]) | 4760 | 4698 / 62 [9 / 4689] |
| Number of users (online) | 7 | 1 |
| Required server performance, new values per second | 123.96 | |

Updated: 14:02:29

**Obrázek 11: Dashboard Zabbix serveru (1. část)**




**Obrázek 12: Dashboard Zabbix serveru (2. část)**

Na obrázcích výše je vidět základní dashboard monitorovacího systému. Obrázek 11 zobrazuje tabulku Recent Problems, která zobrazuje problémy, jenž je potřeba vyřešit nebo je o nich potřeba alespoň vědět. Ideální stav je takový, kdy v této tabulce nejsou žádné problémy. Důležité je vědět, že v tabulce Recent Problems nejsou zobrazeny pouze problémy, ale také informace typu restart zařízení, výměna zařízení, změna stavu portů a podobně. Tabulka zobrazuje tedy spuštěné triggery, které se na základě nastavené podmínky spustí. Tabulka System status zobrazuje počet spuštěných triggerů v jednotlivých sloupcích dle severity jednotlivých triggerů. Problémy jsou barevně vyznačeny s tím, že barva je volitelná. Pokud si monitorované prvky dobře nastavíme, je možné podle tabulky Systém status zjistit nejkritičtější problém v síti. Poslední tabulka Status of Zabbix zobrazuje stav Zabbix serveru, celkový počet přidaných hostů, nastavených triggerů a itemů a počet přihlášených uživatelů.

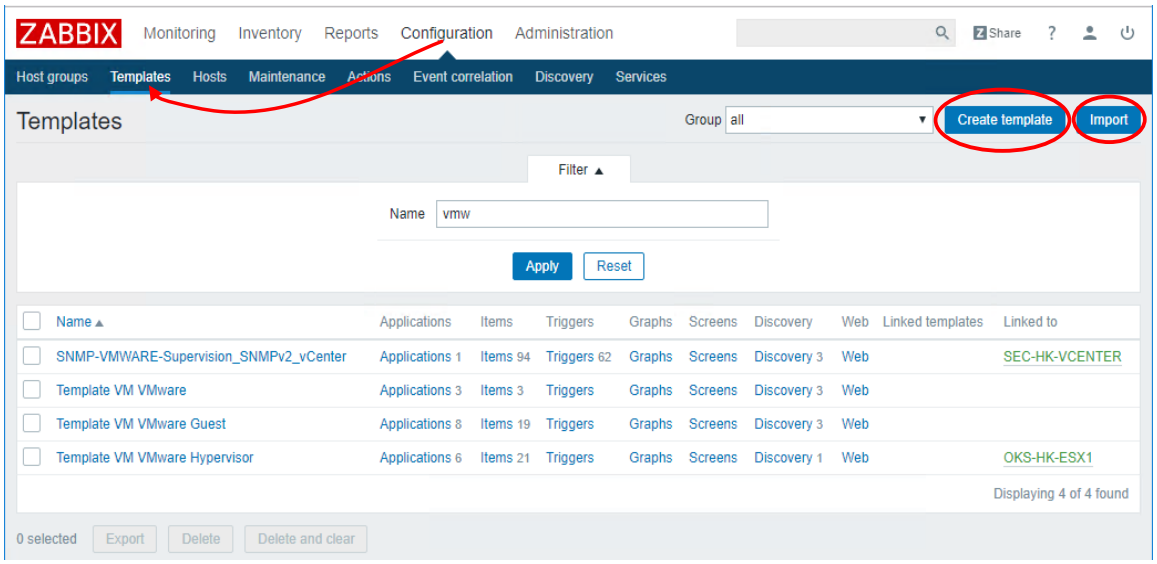
Obrázek 12 zobrazuje historii problémů (spuštěných i vyřešených) a také status jednotlivých hostů. Tabulka Host status zobrazuje celkový počet přidaných hostů v jednotlivých skupinách a také zobrazuje, u kolika prvků v dané skupině jsou spuštěné triggery s tím, že barva, která se zobrazuje v tomto poli je vždy barva problém s nejvyšší severitou.

## 6.2 Monitoring pomocí SNMP

Template je šablona, ve které je nakonfigurované vše potřebné pro monitorování. Nastavené hodnoty (itemy, triggery, grafy…) lze také přidávat jednotlivě na každého nově vytvořeného hosta zvlášť. Tento způsob je však považován za vysoce neefektivní způsob konfigurace, který zabere velkou spoustu času. Templaty mají tu výhodu, že pokud jsou již svázány s vytvořenými hosty, stačí přidat, upravit, zakázat, povolit nebo vymazat nějaký item a tato změna se automaticky projeví na všech hostech, kteří jsou s tímto templatem svázány. Další výhodou templatu je především možnost konfigurace discovery, která umožňuje automatické vyhledávání například na základě OID a následné vytvoření itemů. Jako příklad uvedeme 48 portový switch, na který by bez discovery bylo nutné vytvořit 48 (například pro vyčítání názvu portů) různých itemů zvlášť. Pomocí discovery si ulehčíme práci a všech 48 portů přidáme díky jednoduché konfiguraci a možnostem systému Zabbix.

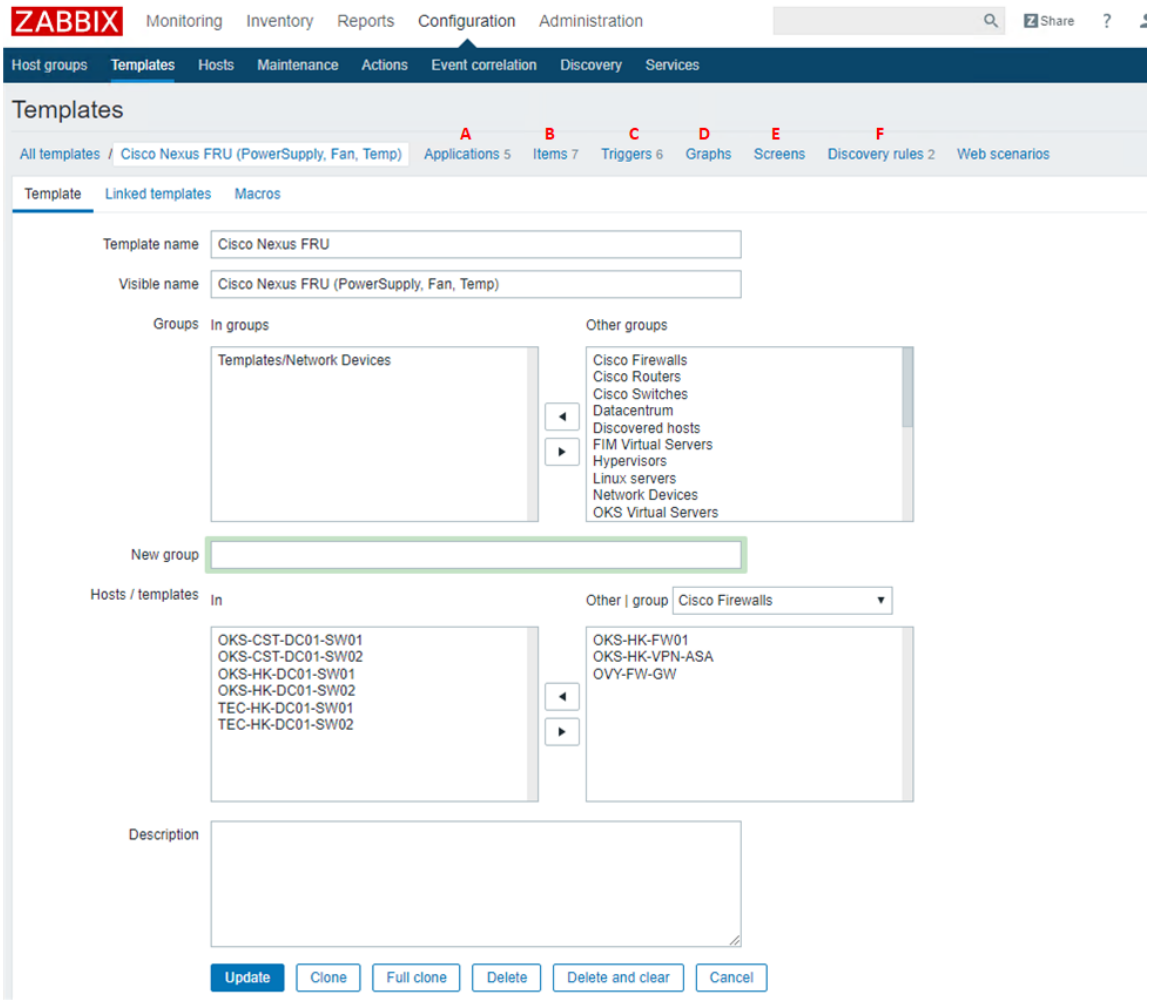
### 6.2.1 Vytvoření templatu

Template, je obecně velice užitečný, a to především z hlediska ulehčení práce administrátorovi systému. Vytvoření nového templatu je možné dvěma způsoby. Jedním způsobem je import z XML souboru a druhým způsobem je manuální vytvoření nového nebo úprava již založeného templatu (Obrázek 13 červeně zakroužkováno). Import je vhodný v případě, pokud máme nový server a všechny templaty předpřipravené z jiné instalace serveru. Velké množství templatů pro import (vytvořené komunitou) lze také nalézt na internetu. Vytvoření nového templatu nebo import z XML souboru se provádí v **Configuration → Templates** (Obrázek 13 označeno červenou šipkou).




**Obrázek 13: Postup pro vytvoření a import templatu**

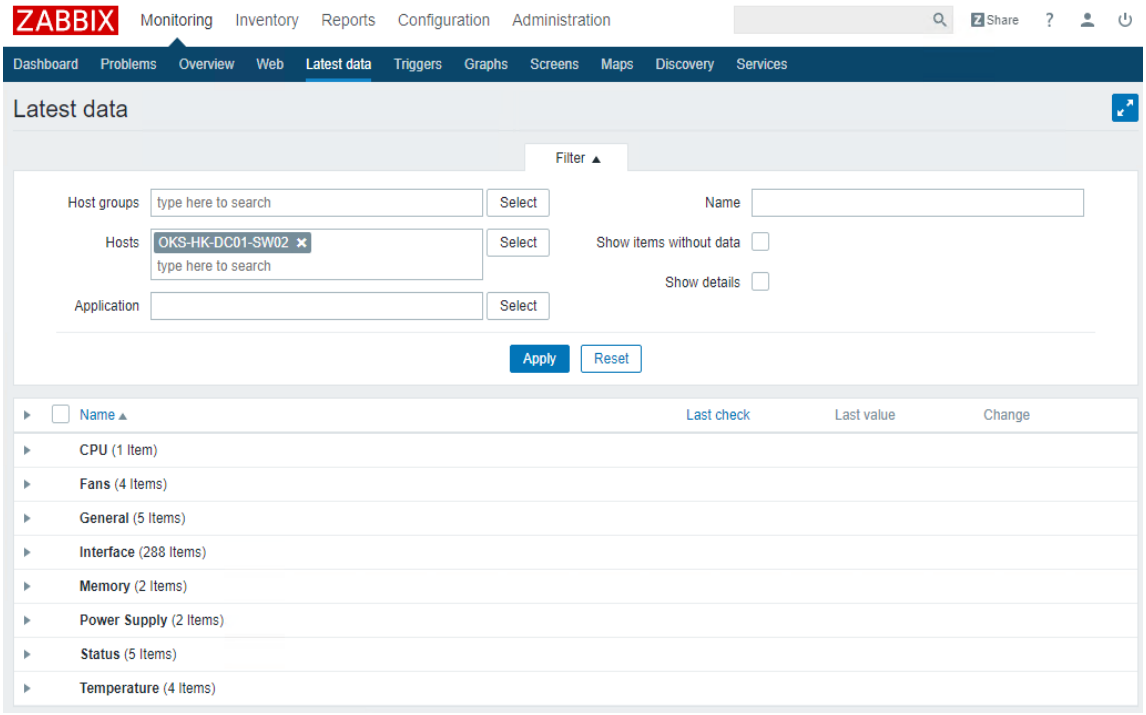
Pro vytvoření nového templatu tedy zvolíme položku **Create template**. Jako příklad je uvedený template pro zařízení Cisco Nexus. Konkrétně se bude jednat o vytvoření templatu pro monitoring napájení, chlazení, výkon procesoru a meření teploty zařízení. Obrázek 14 zobrazuje úvodní stránku pro vytvoření nového templatu.




**Obrázek 14 : Úvodní stránka pro vytvoření nového template**

Obrázek 14 obsahuje mnoho možností, které lze nastavovat. Na úvodní stránce se nastavují pouze údaje jako je název templatu (Template name) a zobrazovaný název (Visible name). V tomto případě je nastavený název, který se bude zobrazovat jiný, než název templatu. Template lze také přidat do určité skupiny (Groups). Obrázek 14 zobrazuje template Cisco Nexus FRU přidaný do skupiny Templates/Network Devices. Jak je vidět v položce Hosts/Templates jsou již přidaná síťová zařízení Cisco Nexus, která jsou pomocí toho template monitorována. Template může být také součástí jiného templatu. Obrázek 14 je označen písmeny A-F a ukazuje jednotlivé sekce, které je možné nastavovat. Jednotlivé sekce jsou vysvětleny následovně:

A. **Application** – definuje jednotlivé aplikace, které monitorujeme a ke kterým se přiřazují jednotlivé itemy. Není považována za povinnou, je však vhodné ji nastavit pro lepší orientaci v měřených datech. V tomto případě jsou zde zadány aplikace Power Supply, CPU, Fans, Temperature. Tyto hodnoty jsou pak přehledně seřazeny do jednotlivých aplikací například v Latest Data, kde jsou zobrazována naměřená data jednotlivých itemů viz Obrázek 15.




**Obrázek 15 : Latest data a jetnolivé aplikace**

B. **Items** – zde se provádí nastavení monitorování jednotlivých Itemů. Je zde vyspecifikováno, jaké OID chceme ze zařízení vyčítat, o jakou hodnotu se jedná, pomocí jaké funkce ji chceme monitorovat (SNMP, SSH…). Podrobněji bude probráno v kapitole Vytvoření Itemu.

C. **Triggers** – zde se nastavují jednotlivé triggery pro spuštění alarmů na dashboard monitorovacího systému Zabbix. Podrobněji bude probráno v kapitole Vytvoření triggeru.

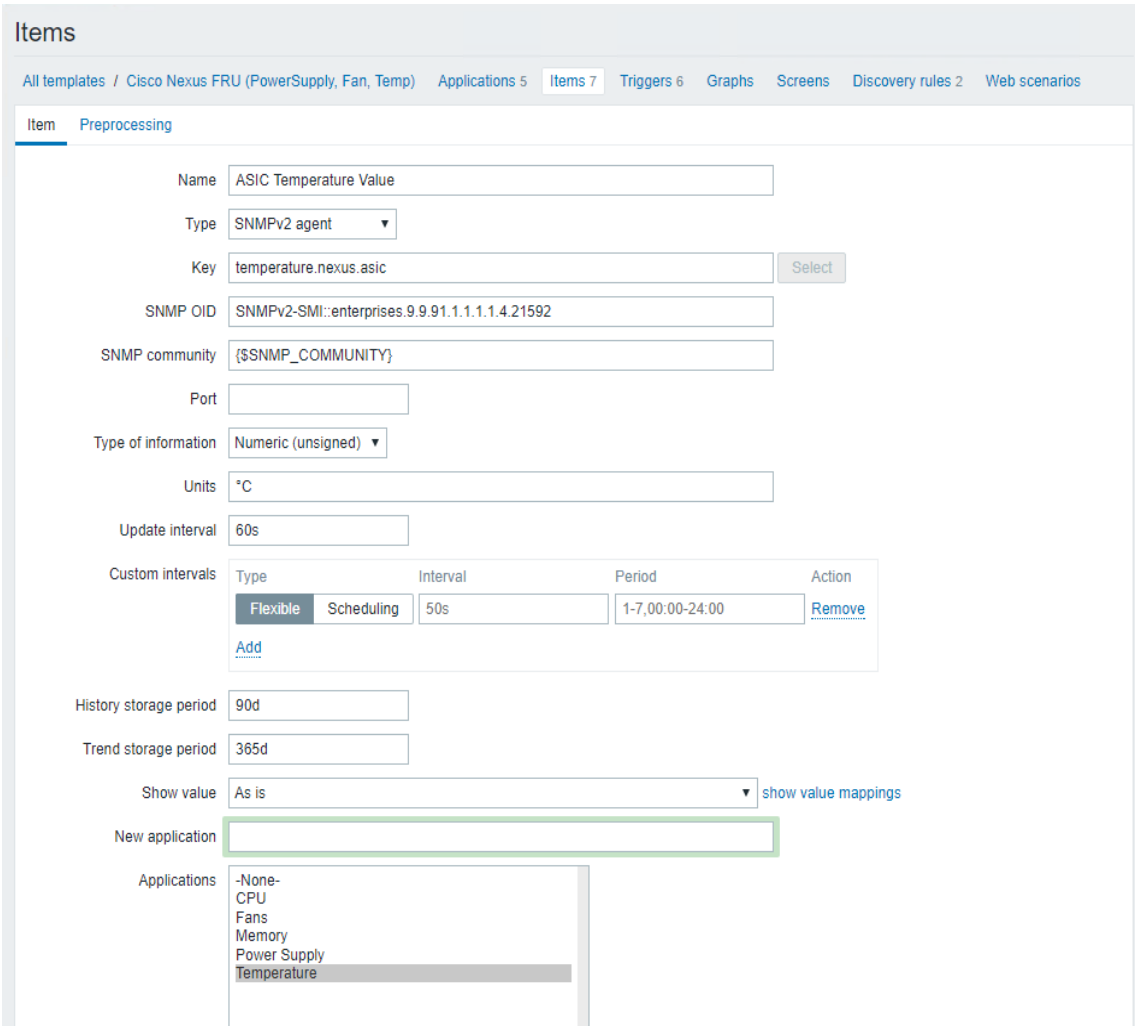
D. **Graphs** – zde se vytvářejí grafy z naměřených hodnot. Jako příklad můžeme uvést naměřená data z jednotlivých rozhraní SW, kde se zvlášť měří datový tok pro Input a Output na jednotlivých rozhraních zařízení. Tyto hodnoty je pak vhodné vložit do grafu, kde se stávají přehlednější (Input a Output stejného rozhraní v jednom grafu). Více v kapitole Vytvoření grafu.

E. **Screens** – screens úzce souvisí s vytvořením grafu. Abychom nemuseli přeskakovat mezi jednotlivými grafy rozhraní, je možné vytvořit screen, kde budou všechna data zobrazena najednou na jedné stránce. Do položky screens lze přidat prakticky vše, co administrátor systému uzná za vhodné pro lepší přehled v naměřených datech.

F. **Discovery rules** – jedná se o sekci pro automatické vytváření itemů, triggerů a grafů pomocí specifické konfigurace. Více o discovery rules bude probráno v kapitole Vytvoření discovery.

### 6.2.2 Vyvoření itemu

Pro vytvoření nového itemu, přejdeme do položky Items a pro vytvoření nového itemu zvolíme „Create item“. Následně se zobrazí stránka viz Obrázek 16.




**Obrázek 16: Vytoření nového itemu**

Obrázek 16 zobrazuje všechna pole, která je možné nastavit. Postupně projdeme všechny možnosti a vysvětlíme si je:

- **Name** – v položce Name nastavujeme pouze název Itemu. ASIC Temperature Value specifikuje, že se jedná o vnitřní teplotu zařízení vyjádřenou číslem.
- **Type** – zde lze nastavit, pomocí jakého protokolu, aplikace nebo agenta chceme monitorovat. Každá z možností má jiné možnosti nastavení. Například pro Zabbix agenta nebudeme potřebovat položku SNMP OID ani SNMP komunitu. Zde je nastaveno monitorování pomocí SNMPv2, u kterého je nutné vyspecifikovat správné OID, typ informace, formát vyčítané hodnoty a komunita.
- **Key** – unikátní název, který musí být unikátní pro celý systém. Na základě toho klíče se následně vytvářejí triggery nebo vypočítávají hodnoty, například pro procentuální využití místa na disku a podobně.

- **SNMP OID** – zde je hodnota se specifickým OID *SNMPv2-SMI::enterprises.9.9.91.1.1.1.1.4.21592* pro zařízení SW Nexus. V tomto případě se jedná konkrétně o hodnotu teploty naměřené uvnitř zařízení. SNMP OID jsou běžně dostupná na oficiálních stránkách výrobce. Číslo na konci OID, konkrétně .21592 je index pro teplotu uvnitř zařízení. V případě chyby při zadávání indexu nebo jeho úplné absence buď vyčteme jinou hodnotu, celý strom nebo nevyčteme nic.
- **SNMP community** – jedná se komunitu pro komunikaci se zařízeními. V tomto případě je nastavené jako globální makro *{$SNMP_COMMUNITY}*, které se nastavuje v **Administration → General → Macros**. Toto makro je pak platné pro všechny hosty nebo templaty, které mají v SNMP komunitě nastavené toto makro. SNMP komunita, která je použita v rámci této práce je ZBXGETRO.
- **Port** – nastavení portu pro komunikaci. Pokud je na zařízení použit standardní port 161 pro komunikaci pomocí SNMP není třeba nastavovat. Číslo portu lze také nastavit pomocí makra jako v případě komunity.
- **Type of information** – jedná se o specifikaci, o jaký typ informace ze zařízení se jedná (INT, STRING). V případě tohoto itemu se jedná o INT, tedy číslo (vyčítáme teplotu). Tuto hodnotu si lze také ověřit pomocí příkazu snmpwalk v linuxu.
- **Units** – jak již napovídá anglický název, jedná se pouze o jednotku, jakou budeme zobrazovat. V tomto případě jsou to °C.
- **Update interval** – zde nastavujeme, jak často se bude daná hodnota vyčítat. V tomto případě budeme vyčítat hodnotu každých 60 sekund.
- **History storage period** – hodnota udává, jak dlouho se budou ukládat data do historie. Do historie se ukládají nová data každých 60 sekund dle pole Update interval. Všechna data starší 90 dnů jsou z historie buď mazána nebo ukládána pomocí Trend storage period.
- **Trend storage period** – hodnota udává, jak dlouho budou data ukládána v historii v agregovaném stavu po době History storage period, než se smažou. Agregace je v tomto případě nastavena na 1 hodinu.
- **Show value** – tato položka se využívá v případech, kdy vyčítáme hodnoty INT, které mohou nabývat více hodnot. Například čidlo teploty se může dostat do několika stavů (1-5), kde tyto hodnoty jsou samy o sobě nevypovídající. Pomocí položky Show value namapujeme jednotlivé hodnoty na jednotlivé stavy. Například hodnota 1 se bude v případě stavu čidla zobrazovat jako UP a hodnota 5 jako CriticalStatus. Hodnoty je nutné namapovat v **Administration → General → Value mappings** a zde každou z hodnot popsat, podle toho, jaký má význam. Toto není povinná položka, slouží pouze pro orientaci v Latest data jednotlivých Hostů.

- **Applications** – zde volíme, pod jakou aplikaci bude item spadat. V tomto případě se jedná o aplikaci Temperature.

Obrázek 16 zobrazuje vedle položky item také položku Preprocessing. Ta se využívá pro úpravu získaných hodnot. Například při měření datového toku na zařízeních Cisco získáváme data v bytech a vzhledem k tomu, že se hodnota běžně udává v bitech, je nutné ji vynásobit osmi.

Po zadání všech hodnot uložíme item. Vzhledem k tomu, že tento item je založený v templatu, po uložení začne v pravidelných intervalech monitorovat hodnotu teploty na všech zařízeních, která mají tento template přidaný.

### 6.2.3 Vytvoření triggeru

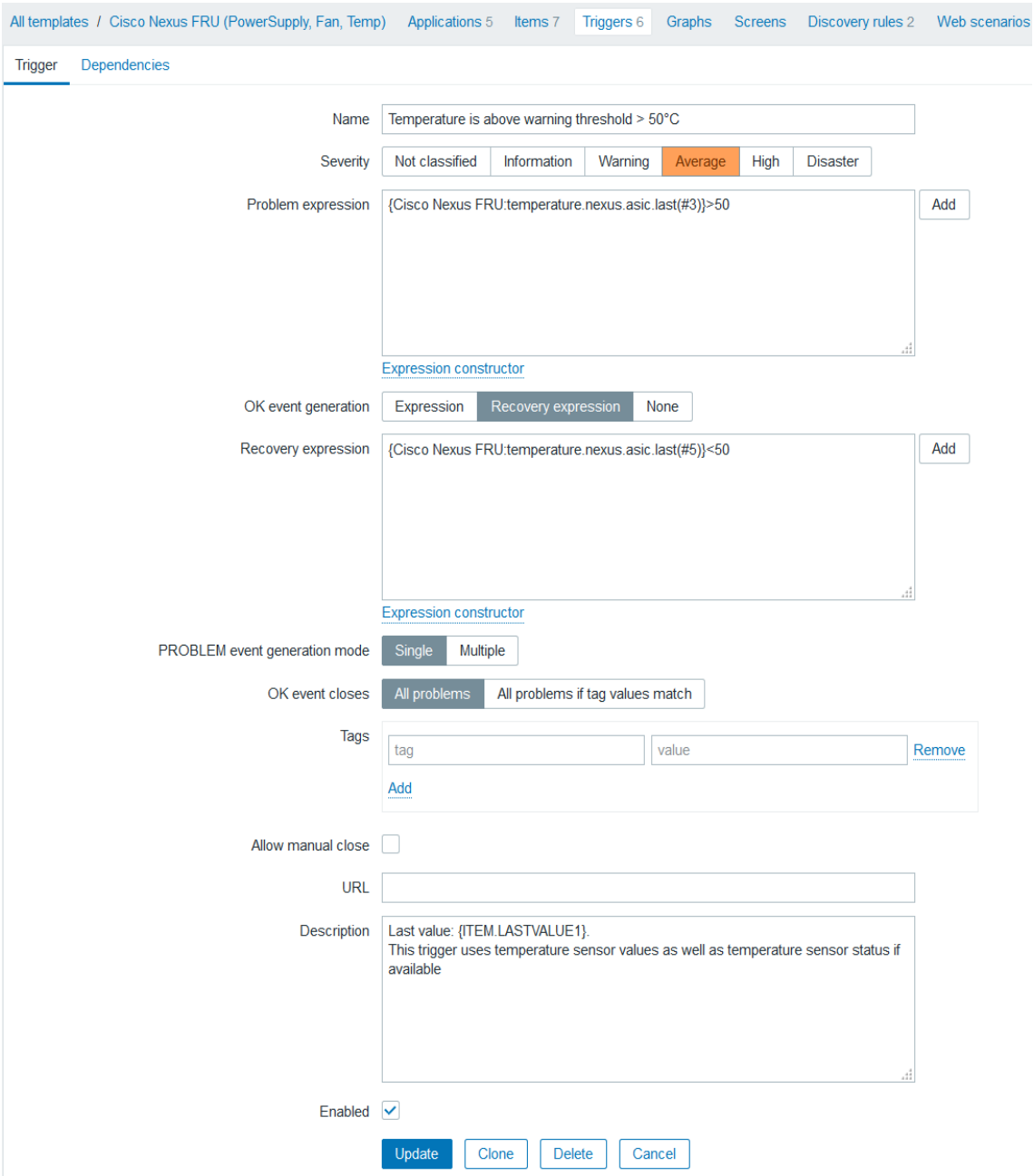
Vytvoření nového triggeru úzce souvisí s vytvořenými itemy. Důležitá hodnota u itemu je právě pole Key, zadávané při vytváření itemu. Na základě tohoto Key se konfigurují triggery. Trigger by se měl vytvářet pouze v případě, je-li nutné změnu hodnoty nebo přesáhnutí daného tresholdu oznamovat na dashboard Zabbix serveru. Na jeden item může být aplikováno libovolné množství triggerů. Obrázek 17 zobrazuje hodnoty, které je vhodné nakonfigurovat, pro správné spouštění triggeru.

- **Name** – jedná se o název triggeru, pod kterým se bude zobrazovat na Dashboardu.
- **Severity** – zde se nastavuje barevně severita daného triggeru. V tomto případě je nastavena jako Average, protože při 50°C je switch stále ve stavu, ve kterém může běžně pracovat, ale je vhodné zkontrolovat, z jakého důvodu se switch přehřívá.
- **Problem extension** – v tomto poli nastavujeme, za jakých podmínek se trigger spustí a zobrazí se na dashboardu. Výraz vypadá následovně ***{****Cisco Nexus FRU:temperature.nexus.asic.last(#3)}>50* a je vysvětlený níže.
    - Cisco Nexus FRU – název templatu.
    - temperature.nexus.asic – unikátní Key v předchozím nastavení daného itemu.
    - last(#3)>50 – zde je nastaveno, že trigger se spustí pouze v případě, jsou-li předchozí naměřené hodnoty vyšší než 50°C. V případě, že budou získané hodnoty v řadě za sebou 50 – 51 – 50 trigger se nespustí. Naopak v případě hodnot 52 – 51 – 51 se trigger spustí. V tomto případě se vždy uvažují pouze poslední tři hodnoty v historii Itemu (Latest data).
- **Recovery extension** – v tomto poli nastavujeme, za jakých podmínkách se trigger považuje za vyřešený a z Dashboardu zmizí. Výraz pro deaktivaci problému je *{Cisco Nexus FRU:temperature.nexus.asic.last(#5)}<50*. Nastavení je podobné jako v případě spuštění triggeru, pouze se liší v tom, že pokud hodnota klesne pod 50°C a zároveň se bude jednat o posledních 5 hodnot, které klesnou pod 50°C, tak se trigger deaktivuje a zmizí z Dashboardu. Toto pole není povinné a nastavuje se

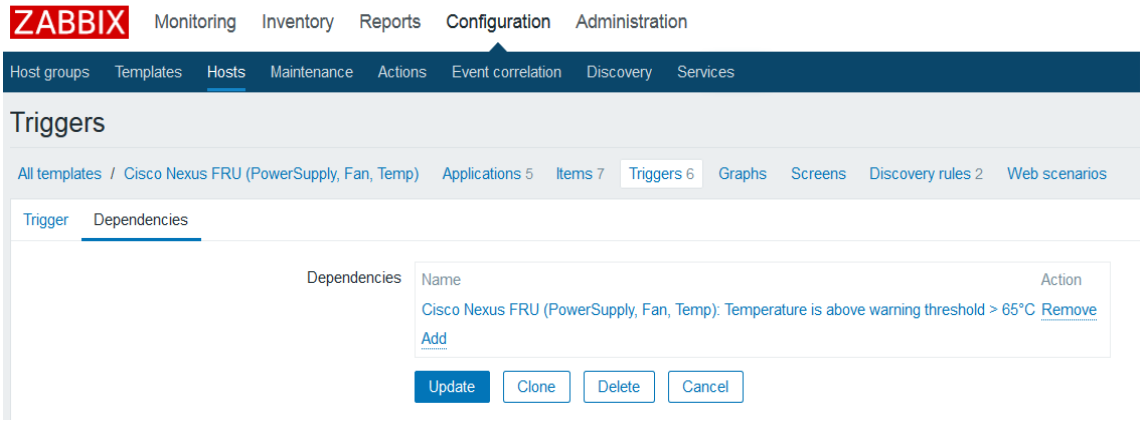
pouze v případě, vybraných hodnot. Bez nastavení tohoto pole se trigger deaktivuje v případě, jakmile teplota klesne pod 50 °C.

- **Description** – zde se udává popis triggeru. Záměrně je vše popsáno v angličtině pro lepší přehled a univerzálnost systému. Je zde také uvedena hodnota Last value: *{ITEM.LASVALUE1}*. Tato hodnota zobrazuje poslední naměřenou hodnotu po najetí kurzoru myši na trigger v dashboardu, kde uvidíme aktuální teplotu zařízení. Položka není povinná, ale je vhodné ji nastavit pro lepší přehled.

Ostatní položky jsou volitelné a není třeba je nastavovat.




**Obrázek 17: Tvorba nového triggeru**

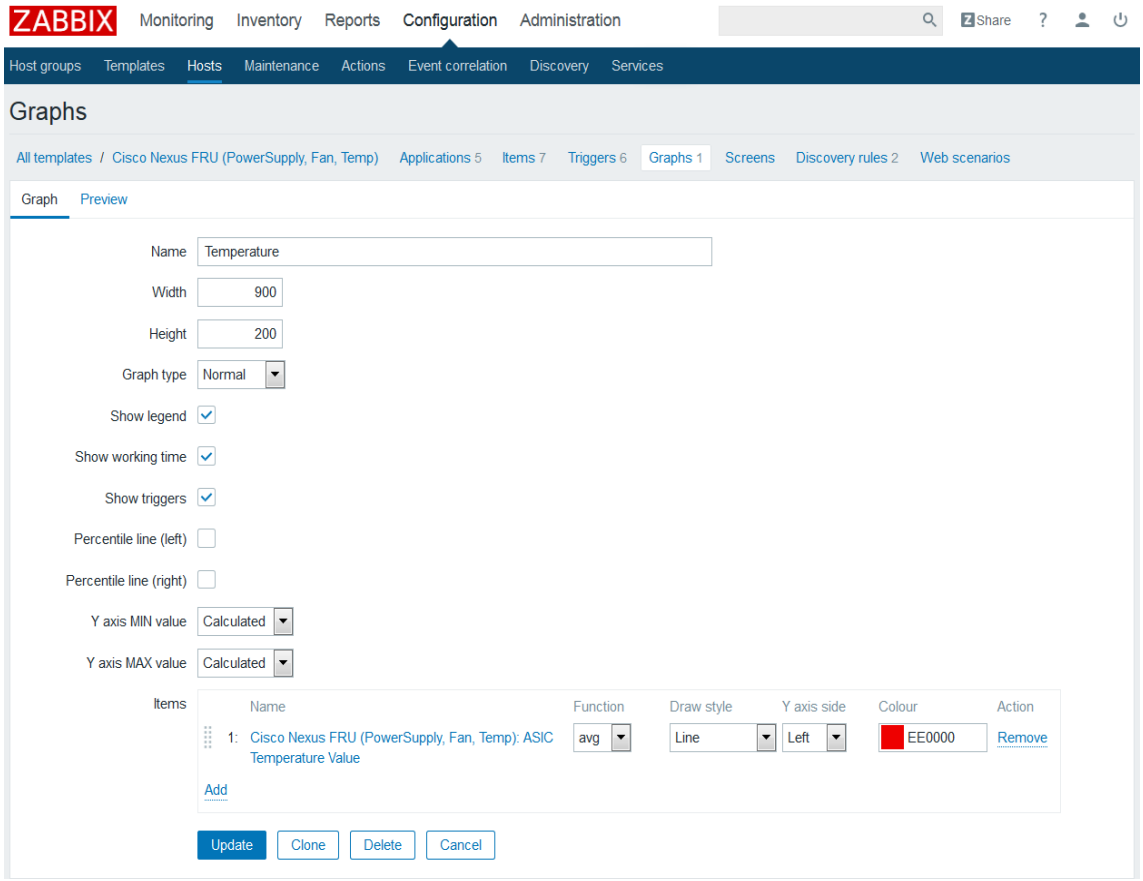



**Obrázek 18: Vytvoření zavislostí daného triggeru**

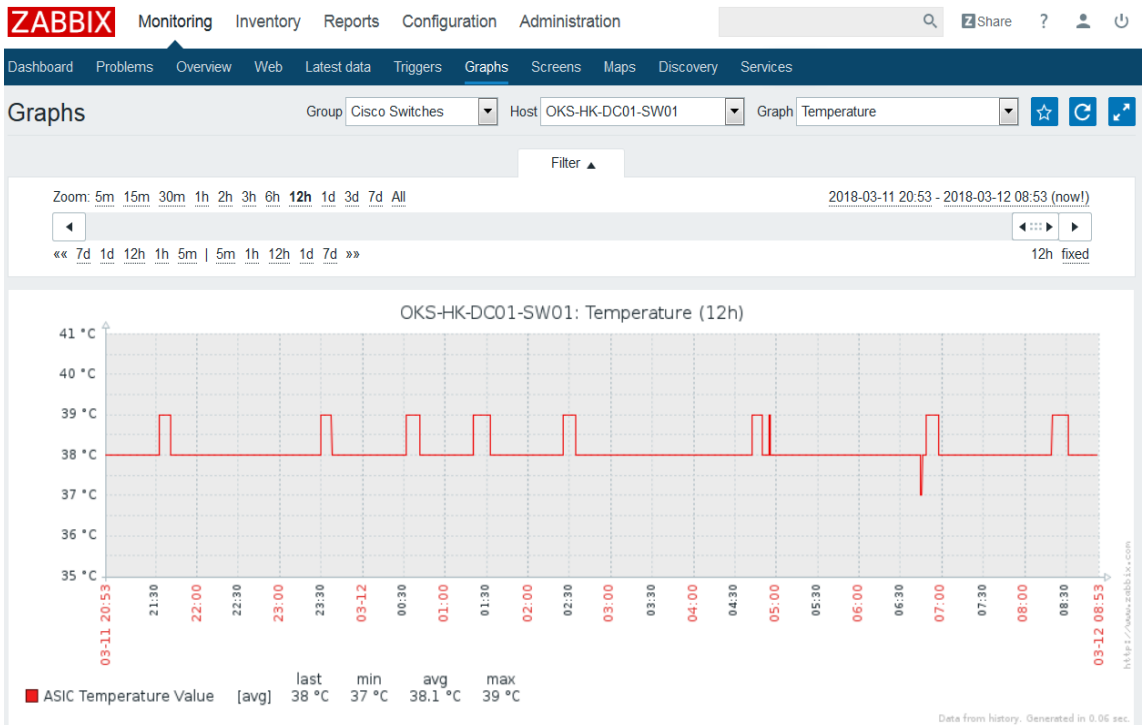
Při nastavování triggeru je také vhodné nastavit závislost (Dependencies) na jiném triggeru. Obrázek 18 ukazuje, že tento trigger se automaticky deaktivuje, pokud se spustí trigger s vyšší severitou a samozřejmě vyšší hodnotou teploty. Dependencies je určené k tomu, aby se nezobrazovalo více triggerů, než je potřeba. V tomto případě, pokud se spustí trigger na teplotu zařízení, které má vyšší teplotu než 65°C, tak se předchozí trigger deaktivuje a zobrazí se pouze jeden trigger a to ten, který je nadřazený tomuto triggeru. Tento způsob nastavování řeší větší přehlednost při řešení problému z Dashboardu Zabbix serveru.

### 6.2.4 Vytvoření grafu

Tvorba grafů je velice jednoduchá a intuitivní. V daném template stačí pouze kliknout na položku Graphs, zvolit Create graph, zadat název grafu, v tomto případě se jedná o graf naměřené teploty s názvem Temperature a v položce Items vybrat item Cisco Nexus FRU (PowerSupply, Fan, Temp): ASIC Temperature Value. Potvrdíme vytvoření a následně se nám již bude u každého zařízení, které má template Cisco Nexus FRU, zobrazovat graficky teplota zařízení. Do grafu lze vkládat libovolný počet itemů. Pro představu je přiložený Obrázek 19, kde je zobrazená konfigurace grafu a Obrázek 20, kde je graficky zobrazená naměřená hodnota.




**Obrázek 19: Konfigurace grafu**




**Obrázek 20: Graficky vyzobrazená hodnota naměřené teploty zařízení**

### 6.2.5 Vytvoření discovery

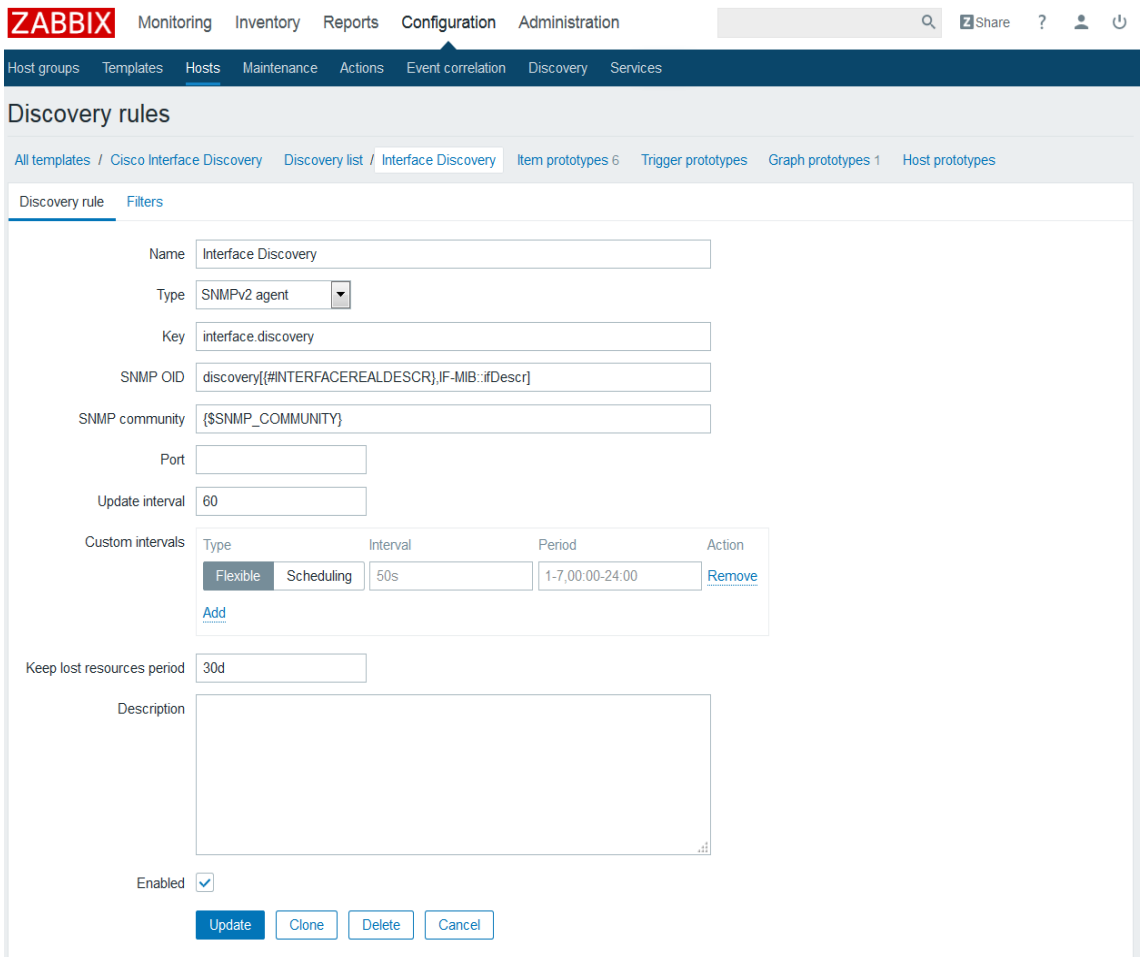
Pro jednoduchou tvorbu více itemů najednou slouží v templatu položka Discovery rules, do které se dostaneme pomocí **Configuration → Templates → Template, na kterém chceme nastavovat Discovery rule → Discovery rules.** Na tomto příkladu bude ukázáno, jak nastavit Discovery rule pro automatické vyhledávání a následné vytvoření všech rozhraní zařízení Cisco jako nový item. Obrázek 21 zobrazuje nastavené pravidlo pro automatické vyhledávání rozhraní na zařízení Cisco. Toto pravidlo samo o sobě ještě nic nevytváří, ale pouze vyhledává rozhraní na zařízení a ukládá si do paměti jejich indexy. Samotná tvorba pak probíhá v položce **Configuration → Templates → Template, na kterém chceme nastavovat Discovery rule → Discovery rules → Item prototypes.** Popis, jak postupovat při zakládání itemů pomocí Discovery rules, bude popsán následovně. Triggery a grafy se vytvářejí stejně jako v případě klasického itemu, proto zde nebudou uvedeny.

Jako první bude popsáno základní pravidlo Discovery rule, bez kterého se vytváření Item prototypes neobejde. Postup je podobný jako v případě tvorby samostatného itemu s tím rozdílem, že je zde přidaná funkce „discovery“, která slouží právě pro vyhledávání jednotlivých rozhraní na zařízení. Hlavní rozdíl je tedy v SNMP OID, kde zadaná syntaxe pro automatické vyhledání rozhraní musí být ve tvaru *discovery[{#INTERFACEREALDESCR},IF-MIB::ifDescr]*. Toto OID *IF-MIB::ifDescr* vypisuje hodnoty ve formátu STRING a obsahuje hodnoty jako GigabitEthernet 1/0/1 nebo Ethernet 0/1 atd.. U tohoto OID *IF-MIB::ifDescr* je důležité vědět, že se jedná o celý strom. Vyhledá všechny hodnoty, které jsou v tomto stromě a uloží si všechny jejich nalezené indexy do paměti.

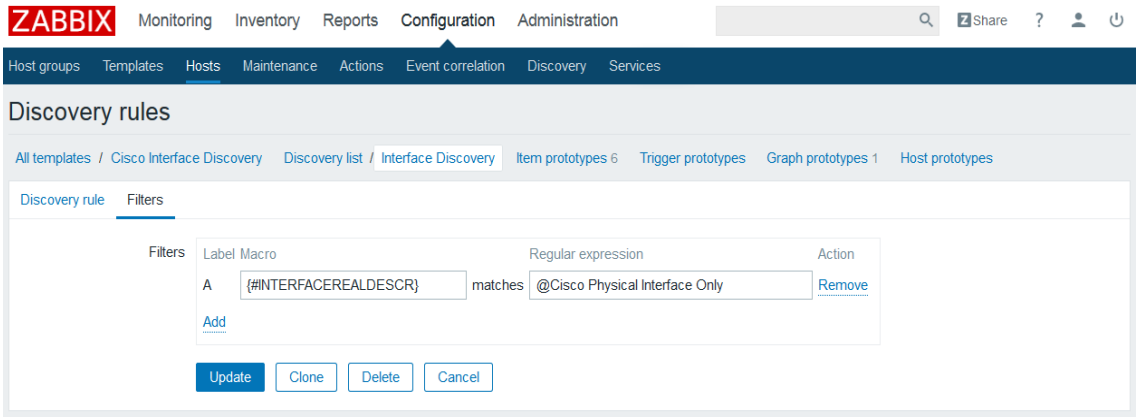
Popis formátu discovery zadaného jako *discovery[{#INTERFACEREALDESCR},IF-MIB::ifDescr]* je následovný:

- **discovery** – jedná se funkci Zabbix serveru, která má za účel vyhledávat a ukládat do paměti všechny indexy, které najde pod daným OID *IF-MIB::ifDescr*.
- **{#INTERFACEREALDESCR}** – jedná se o makro, do kterého se budou ukládat hodnoty typu STRING ze zadaného OID, například GigabitEthernet 1/0/1 nebo Ethernet 0/1 atd..

- **IF-MIB::ifDescr** – je samotné OID, které vypíše všechny rozhraní na zařízení (celý strom tohoto OID), které jsou pod tímto OID. Toto OID vyhledá vše včetně rozhraní VLAN, loopback i klasických GigabitEthernet atd.. Vzhledem k tomu, že není ve většině případů žádoucí vytvářet itemy na jednotlivé VLAN nebo rozhraní loopback, lze tyto hodnoty vyfiltrovat pomocí Filters. Do Filters lze vkládat Regular expression (regulární výrazy) pro vyhledávání pouze určitých názvů rozhraní a vyfiltrovat tak rozhraní VLAN nebo loopback z vyhledávání. Obrázek 22 zobrazuje přesně takový filtr, který je možné k tomuto vyhledávání přidat. Obsahuje makro *{#INTERFACEREALDESCR}*, které je obsaženo v samotném SNMP OID a dále obsahuje odkaz na Regular expression *@Cisco Physical Interface Only*, které je nakonfigurované v **Administration → General → Regular Expressions.**

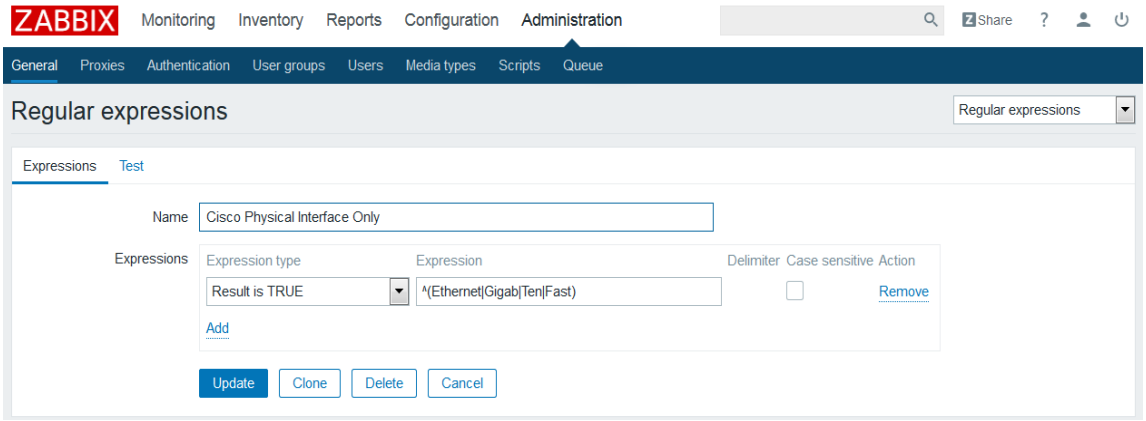



**Obrázek 21: Nastavení pravidel pro discovery**




**Obrázek 22: Filtr pro Discovery rule**

Obrázek 23 zobrazuje samotnou syntaxi regulárního výrazu, který byl pro tento template použit. Konfigurace obsahuje výraz *^(Ethernet|Gigab|Ten|Fast)*, kde jsou vyspecifikovány pouze fyzická rozhraní daných zařízení, která se na zařízeních Cisco vyskytují. Nastavené makro *{#INTERFACEREALDESCR}* vyhledává pouze tato rozhraní a nic jiného.




**Obrázek 23: Příklad konfigurace regulárního výrazu**

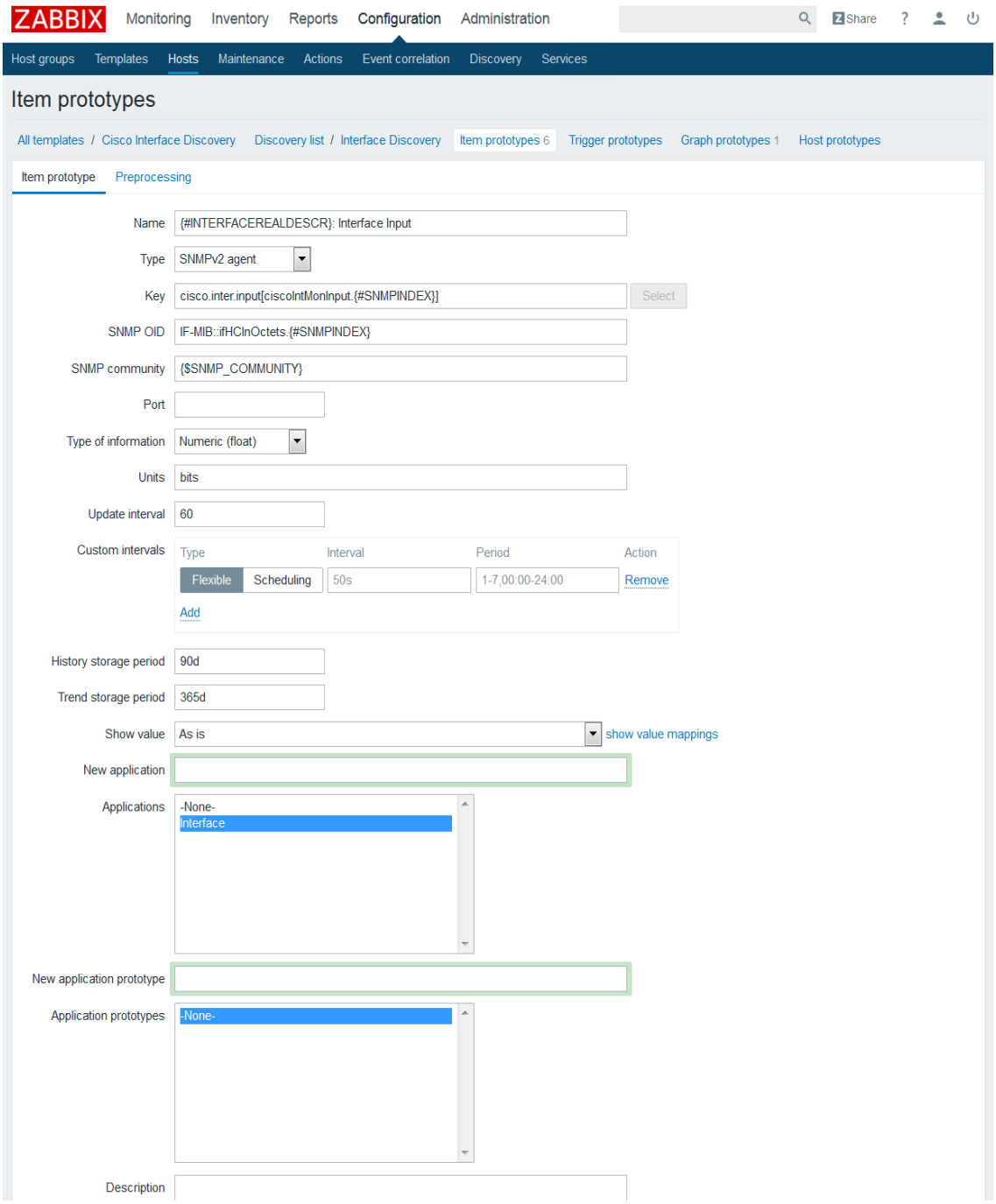
### Založení Item prototypes

Po úspěšném nastavení Discovery rule následuje založení nového itemu ze získaného Discovery rule. Item prototype se nastavuje v **Configuration → Templates → Template na kterém chceme nastavovat Discovery rule → Discovery rules (vybereme dané pravidlo, v našem případě Interface Discovery)→ Item prototypes.** Obrázek 24 zobrazuje příklad vytvoření item prototype, který probíhá téměř shodně s klasickým itemem.

Podstatný rozdíl je v dynamické proměnné {#SNMPINDEX}, která určuje již konkrétní rozhraní. Každé vyhledané rozhraní má unikátní dynamickou proměnnou pro dané zařízení. SNMP OID *IF-MIB::ifHCInOctets* vypisuje datový tok na vstupu rozhraní a bez specifikace *.{#SNMPINDEX}* by byly uloženy všechny hodnoty do jednoho itemu z předchozího pravidla Discovery rule. Bez makra *.{#SNMPINDEX}*

však server nedovolí vytvořit nový item a vypíše chybu. Ve výsledku dynamická proměnná specifikuje každé rozhraní zvlášť indexem uloženým v paměti discovery a výsledná hodnota pro jedno rozhraní vypadá následovně, *IF-MIB::ifHCInOctets.436326400 = Counter64: 296955129330*, kde *.436326400* je index *(.{#SNMPINDEX})* jednoho z rozhraní.

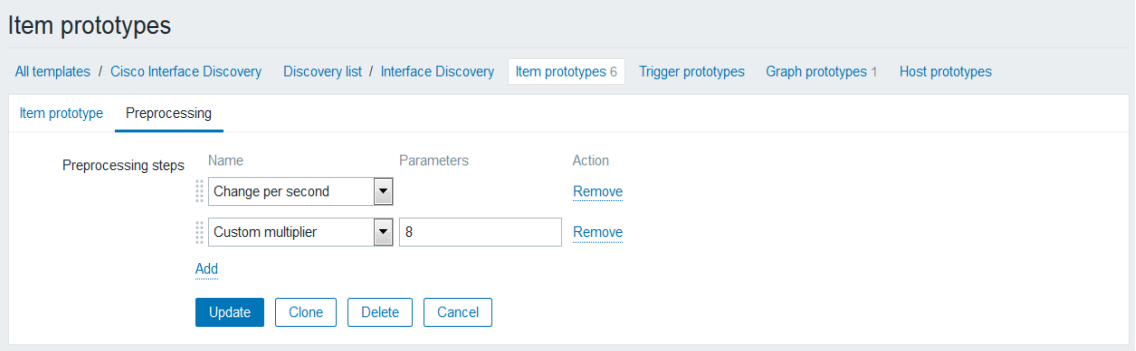
Důležité je přidat makro *.{#SNMPINDEX}* také do položky Key, aby se zaručila její unikátnost.




**Obrázek 24: Příklad založení nového Item prototype**

Abychom věděli, o jaké rozhraní se jedná je v názvu Item prototype přidáno makro *{#INTERFACEREALDESCR}*, který vypisuje konkrétní STRING pro dané rozhraní, například GigabitEthernet 1/0/1.

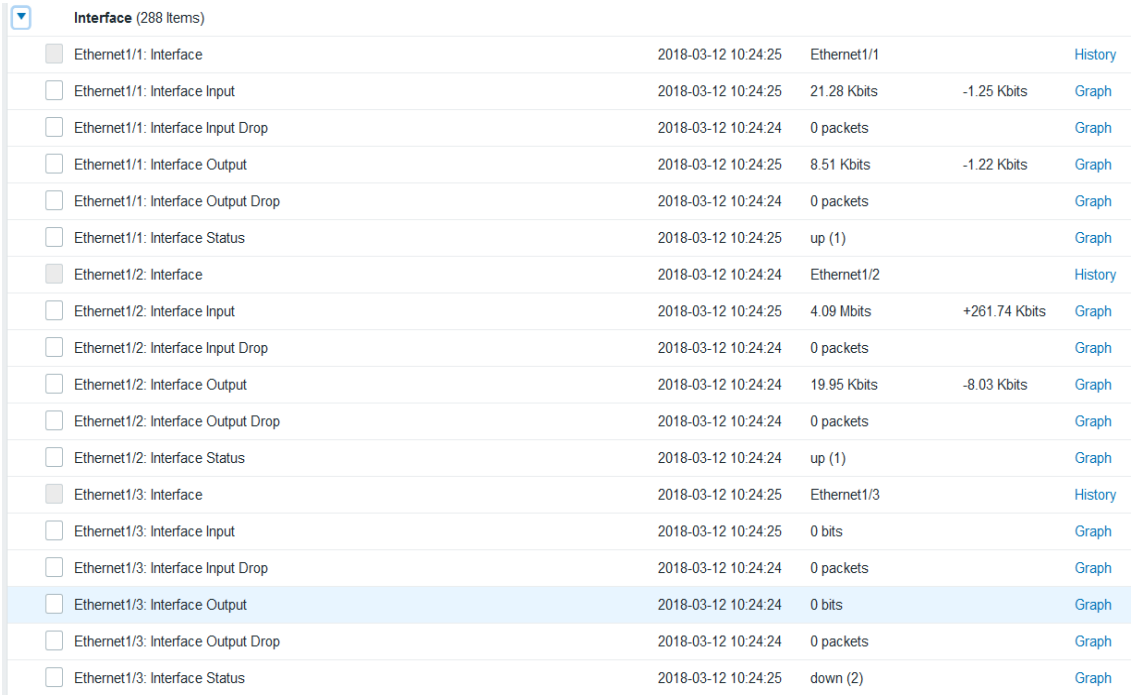
V položce Preprocessing je následně nutné nastavit násobení osmi, protože hodnota, kterou získáváme je v bytech. Důležité je vědět, že Cisco v tomto OID zobrazuje celkový provoz na rozhraní, nikoliv aktuální (viz výsledek vyčtení pomocí SNMP *Counter64: 296955129330*), proto je nutné v Preprocessing nastavit, aby se získávaly pouze změny za sekundu (Change per second). Obrázek 25 ukazuje příklad nastavení záložky Preprocessing.




**Obrázek 25: Příklad nastavení Preprocessing pro nové Rozhraní**

Discovery rule tedy postupně a dle regulárního výrazu projde všechna rozhraní na zařízení a pomocí Item prototype se jeden po druhém vytvoří. To by při zadávání jednotlivých itemů pro každé rozhraní zabralo velkou spoustu času. Pro každé rozhraní bychom museli vyspecifikovat dané OID, které by bylo ve tvaru *IF-MIB::ifHCInOctets.436326400* a podobně. Pomocí discovery je vše zajištěno automaticky pomocí jednoduché konfigurace.

Obrázek 26 zobrazuje výsledek Discovery rule, kde lze vidět jednotlivé získané hodnoty v Latest Data. Obrázek 26 zobrazuje více hodnot, které jsou již v systému nastaveny. Konkrétně se jedná o Input, Input Drops, Output, Output Drops, Port Status a samotný název rozhraní. Vše se nastavuje stejně jako v předchozím případě pro Interface Input, kde stěžejní je právě Discovery rule, na základě něhož se vytvářejí jednotlivé Item prototypes. Důležité je však najít správné OID pro každý item, které jsou však běžně dostupné na oficiálních stánkách distributora.



| Interface (288 Items) | | | | |
|---|---|---|---|---|
| Ethernet1/1: Interface | 2018-03-12 10:24:25 | Ethernet1/1 | | History |
| Ethernet1/1: Interface Input | 2018-03-12 10:24:25 | 21.28 Kbits | -1.25 Kbits | Graph |
| Ethernet1/1: Interface Input Drop | 2018-03-12 10:24:24 | 0 packets | | Graph |
| Ethernet1/1: Interface Output | 2018-03-12 10:24:25 | 8.51 Kbits | -1.22 Kbits | Graph |
| Ethernet1/1: Interface Output Drop | 2018-03-12 10:24:24 | 0 packets | | Graph |
| Ethernet1/1: Interface Status | 2018-03-12 10:24:25 | up (1) | | Graph |
| Ethernet1/2: Interface | 2018-03-12 10:24:24 | Ethernet1/2 | | History |
| Ethernet1/2: Interface Input | 2018-03-12 10:24:25 | 4.09 Mbits | +261.74 Kbits | Graph |
| Ethernet1/2: Interface Input Drop | 2018-03-12 10:24:24 | 0 packets | | Graph |
| Ethernet1/2: Interface Output | 2018-03-12 10:24:24 | 19.95 Kbits | -8.03 Kbits | Graph |
| Ethernet1/2: Interface Output Drop | 2018-03-12 10:24:24 | 0 packets | | Graph |
| Ethernet1/2: Interface Status | 2018-03-12 10:24:24 | up (1) | | Graph |
| Ethernet1/3: Interface | 2018-03-12 10:24:25 | Ethernet1/3 | | History |
| Ethernet1/3: Interface Input | 2018-03-12 10:24:25 | 0 bits | | Graph |
| Ethernet1/3: Interface Input Drop | 2018-03-12 10:24:24 | 0 packets | | Graph |
| Ethernet1/3: Interface Output | 2018-03-12 10:24:24 | 0 bits | | Graph |
| Ethernet1/3: Interface Output Drop | 2018-03-12 10:24:24 | 0 packets | | Graph |
| Ethernet1/3: Interface Status | 2018-03-12 10:24:25 | down (2) | | Graph |

**Obrázek 26: Latest data zařízení Cisco pro správném nastavení Discovery rule a Item prototype**

## 6.3 Monitoring pomocí Zabbix agenta

Zabbix agent je doplněk monitorovacího systému Zabbix. Je určen především pro monitoring OS jako Linux, Windows, NetBSD, Mac OS X a další. Aktuální seznam podporovnaných OS je dostupný na adrese *https://www.zabbix.com/documentation/3.4/manual/concepts/agent*. Velké množství itemů pro monitoring pomocí Zabbix agenta je již předdefinováno v systému ve formě klíče (key), kde jejich správný formát je dostupný na oficiálních stránkách Zabbix na adrese *https://www.zabbix.com/documentation/3.4/manual/config/items/itemtypes/zabbix_agent*. V této ukázce bude uvedena konfigurace Zabbix agenta na systému Linux Centos 7, konkrétně na serveru Dokuwiki. Instalace probíhá pomocí balíčku, který by měl být běžně dostupný v základních repositářích, případně lze chybějící repositáře doplnit. Pokud je použit Zabbix agent na systému, na kterém je spuštěn Zabbix server nebo proxy, je vhodné z hlediska bezpečnosti spouštět agenta pod jiným uživatelem, než jsou spuštěny procesy serveru nebo proxy. Tato poznámka je zde uvedena především proto, že při instalaci agenta z balíčku, je defaultní uživatel, pod kterým je proces spuštěn, uživatel *zabbix*, který je stejný také pro Zabbix server a proxy.

### 6.3.1 Instalace Zabbix agenta

Vzhledem k tomu, že instalujeme na OS Linux Centos z balíčku, je instalace jednoduchá a stačí zadat následující.

```
yum install zabbix-agent
```

Následně je třeba zadat FW pravidla pro komunikaci serveru s agentem. Agent defaultně komunikuje na portu 10050 protokolu TCP. To lze provést pomocí příkazu níže, záleží však na OS a také na používaném FW. Příkazy se proto mohou lišit.

```
firewall-cmd --zone=internal --add-port=10050/tcp --permanent
firewall-cmd --reload
```

V tuto chvíli máme nainstalovaného agenta, kterého je potřeba nakonfigurovat. Konfigurace probíhá v souboru */etc/zabbix/zabbix/zabbix_agentd.conf*. Zde je potřeba upravit parametr Server a ServerActive a zadat zde IP adresu Zabbix serveru pro komunikaci. Dále je potřeba upravit parametr Hostname a zadat název, který budeme používat při vytváření hosta v Zabbix serveru. Bez této úpravy nebude agent schopný komunikovat se serverem a v logu bude vypsána chyba. Konfigurace je následující a parametry lze zadat do předem připravených míst v konfiguračním souboru, nebo je lze zadat na konec souboru. Záleží na zvyklostech administrátora systému.

```
Server = 10.252.128.126
ServerActive=10.252.128.126
Hostname=OKS-HK-DOKUWIKI
```

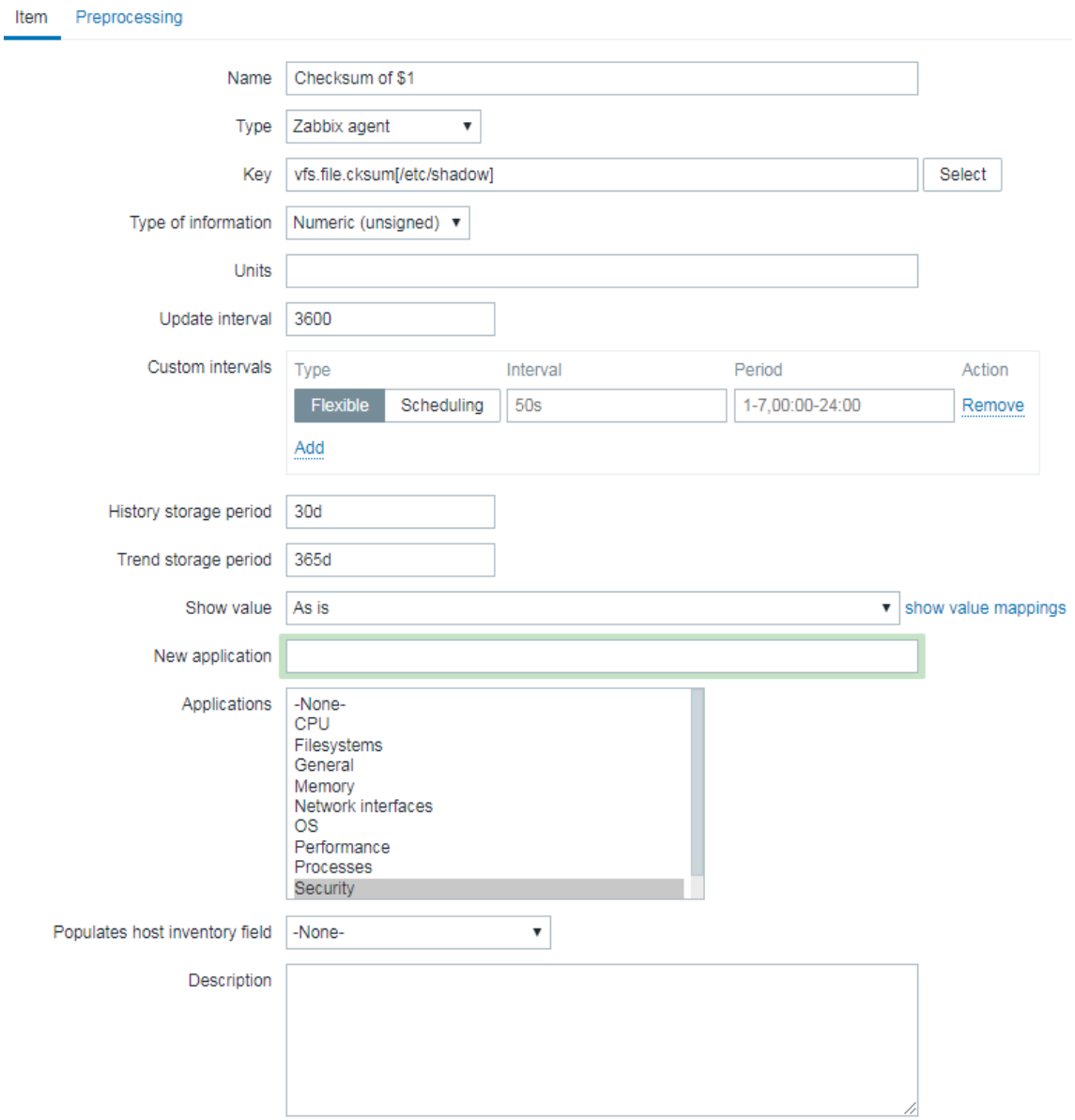
Základní parametry máme tímto nastaveny a zbývající není třeba upravovat. Tímto máme Zabbix agenta připraveného pro komunikaci se serverem. Dalším krokem je vytvoření nového templatu pro agenta a přidání nového hosta na Zabbix server pro monitoring.

### 6.3.2 Vytvoření nového itemu pro Zabbix agenta

Při vytváření nového templatu pro Zabbix agenta postupujeme téměř stejně jako v případě templatu pro SNMP pouze s tím rozdílem, že vybereme v poli Type → Zabbix Agent a použijeme předdefinované klíče pro monitoring systému, jejichž dostupnost byla zmiňována na začátku kapitoly 6.3. Samozřejmě je možné si nadefinovat vlastní hodnoty, které chceme vyčítat, pokud nejsou součástí klíčů předdefinovaných od tvůrců systému. Postup pro vytvoření triggeru, grafů, discovery je stejný jako v kapitole 6.2, proto je zde uvedeno pouze vytvoření nového itemu. Obrázek 27 takový item zobrazuje.

Postupně si popíšeme hodnoty, které je třeba nastavit. V tomto případě se jedná o kontrolu souboru */etc/shadow*, kam se ukládá hash hesel k uživatelským účtům.

- **Name** – stejně jako u itemu pro SNMP se zde jedná o název itemu. Hodnota *$1* je makro, které vypisuje hodnotu v klíči, konkrétně hodnotu napsanou v závorkách */etc/shadow*.
- **Type** – v tomto případě nastaven na Zabbix agent.
- **Key** – klíč, který je zároveň funkcí, kterou chceme pomocí toho itemu dělat. V tomto případě se každou hodinu ukládá checksum souboru */etc/shadow*.




**Obrázek 27: Příklad itemu pro zabbix agenta**

- **Type of information** – typ informace je v tomto případě číselný. Tuto informaci si zjistíme z manuálu, ale logicky je checksum souboru vždy číselný.
- **Units, Update interval, History storage period, Trend storage period, Show value, Applications** – tyto hodnoty představují stejné využití jako v případě jakéhokoliv jiného itemu. Tyto hodnoty byly popisovány v kapitole 6.2.2.

Pokud tento item uložíme, tak u každého hosta, na kterém bude tento template aplikován, bude vyčítána hodnota checksum souboru */etc/shadow*. Trigger je v tomto případě nastavený tak, že pokud některá z předchozích 10 hodnot bude jiná než poslední vyčtená hodnota, spustí se trigger s upozorněním, že došlo ke změně hesla na daném hostovi.

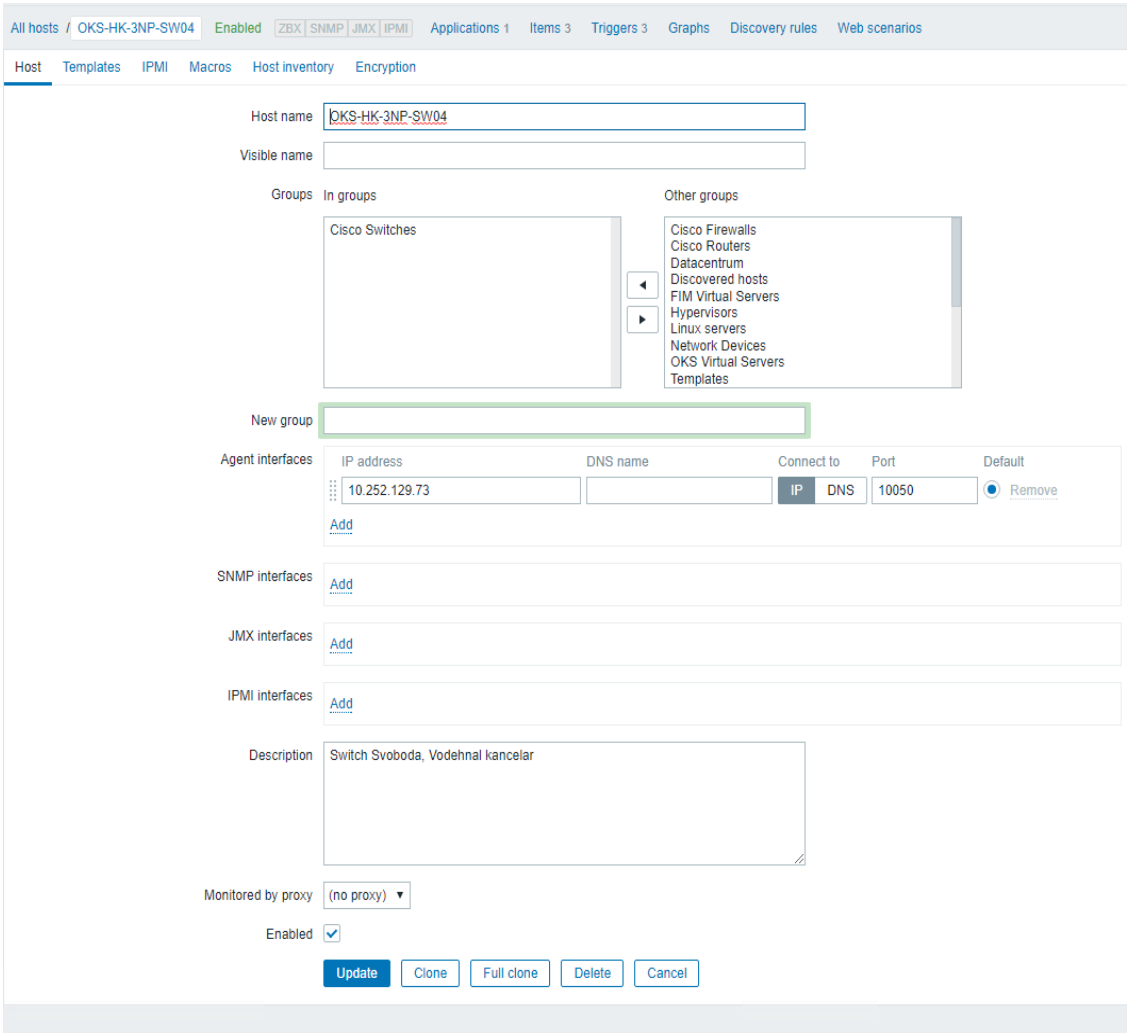
## 6.4 Založení nového monitoringu (přidání nového hosta)

Pro ulehčení výběru jednotlivých možností monitoringu je vhodné si nastavit templaty, které byly popisovány v předchozích kapitolách. Z již nastavených templatů pak vybíráme ty, které jsou vhodné pro daného hosta/zařízení.

Pro lepší orientaci v systému, je vhodné rozdělit hosty do jednotlivých skupin. Tyto skupiny se definují v **Configuration -> Hosts**. Zde jsou vytvořené skupiny Cisco Firewalls, Cisco Switches, Cisco Routers, Datacentrum a podobně. Obrázek 11 tyto skupiny zobrazuje v tabulce System status.

Samotný monitoring nového hosta se nastavuje v **Configuration -> Hosts.** Obrázek 28 zobrazuje ukázku vytvoření monitoringu pro nového hosta. Při vytváření nového hosta můžeme použít kopii již vytvořeného hosta, který je již přiřazen do správné skupiny a má přiřazené správné templaty. V tomto případě stačí pouze změnit název a IP adresu a nový host je připravený pro monitoring. V případě tvorby nového hosta je nutné přidat jej do správné skupiny a přiřadit mu správné templaty. Obrázek 28 zobrazuje příklad vytvoření nového hosta, kde jednotlivé položky jsou vysvětleny následovně:

- **Host name** – musí být unikátní název hosta. Toto pole je důležité především v případě, kdy monitorujeme systém pomocí Zabbix agenta. V tomto případě musí být Host name shodné s Host Name nastaveném v *zabbix_agent.conf* souboru.
- **Visible name** – libovolné jméno. Pokud pole zůstane prázdné, bude se v dashboardu zobrazovat název z Host Name.
- **Groups** – přidáme hosta do správné skupiny. Pole New group vytvoří novou skupinu.
- **Agent interfaces** – zadává se vždy. Pole IP address obsahuje adresu hosta, pole DNS name není povinné, pokud nezaškrtneme položku Connect to -> DNS. Toto rozhraní se používá jak pro klasický simple check jako je kontrola pingem tak pro monitoring pomocí Zabbix agenta. Port 10050 je určen pro komunikaci se Zabbix agentem.
- **SNMP interfaces** – podobné jako u Agent interfaces, s tím rozdílem, že toto pole je použito pro SNMP spojení. Obdobně slouží JMX a IPMI rozhraní.
- **Description** – slouží pro popis hosta.
- **Monitored by proxy** – nastavuje je pouze tehdy, pokud chceme daného hosta monitorovat pomocí proxy. Pokud ne, necháme default.




**Obrázek 28: Vytvoření monitoringu pro nového hosta**

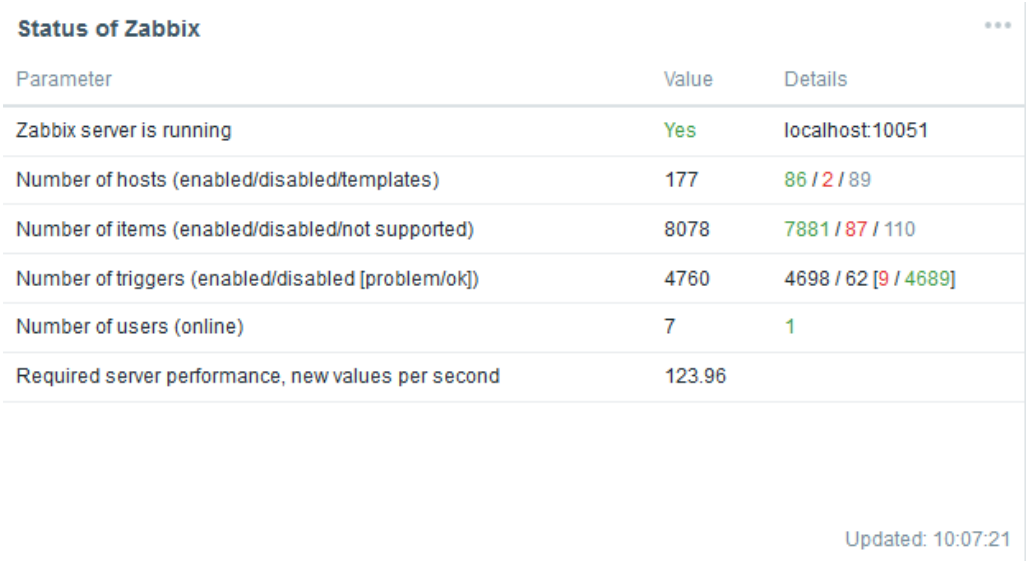
## 6.5 Výsledky monitorovacího systému Zabbix

V současné chvíli je v systému nastaven monitoring na 87 hostech (zařízeních). Na těchto hostech je monitorován datový tok na jednotlivých rozhraních, stavy těchto rozhraní a jejich packet drop. Dále je monitorována teplota, stav čidla teploty, stavy jednotlivých napájecích zdrojů, uptime zařízení a verze OS. Co se týče výkonu jednotlivých zařízení tak především na hypervisorech pro virtuální stroje je monitorováno vytížení procesorů, pamětí RAM a také velikost volného místa na discích. Disky je třeba monitorovat také na samotných datových centrech, kde je spuštěný tzv. thinprovizioning, proto je možné, že na hypervisorech bude jiná hodnota než na samotném datovém úložišti v datovém centru. Z hypervisorů je zároveň vyčítán stav jednotlivých virtuálních stanic tzn. zda jsou ve stavu vypnuto nebo zapnuto, zda je na nich spuštěn agent pro vCentrum a kolik mají přiřazené paměti a procesorů. Detailnější popis najdete v kapitole Detailní popis zařízení a jejich monitoring.

Na relevantní itemy, jako jsou stavy rozhraní, velikost úložiště, uptime zařízení nebo teplotu zařízení jsou nastavené triggery, které se spustí v případě přesáhnutí nastavené

úrovně nebo při změně stavu a spustí alarm na Dashboardu Zabbix serveru. U některých zařízení je nastaveno automatické vytváření grafů a screenů pro lepší přehled v naměřených datech. Jedná se především o taková zařízení, která mají více rozhraní, teplotních čidel nebo procesorových jednotek. V testovacím prostředí je monitorováno 86 zařízení, z těchto zařízení je vyčítáno bezmála 7881 itemů a nastaveno celkem 4798 triggerů viz Obrázek 29.

Celkový výsledek a stav monitorovacího systému Zabbix je pomocí obrázku s popisky přidán do přílohy diplomové práce. Na obrázcích jsou jednotlivá pole zobrazena na hlavní stránce monitorovacího systému, konkrétně Dashboardu. Dále je v příloze zobrazena část přidaných hostů, část vyčítaných itemů pomocí SNMP a Zabbix agenta včetně triggerů, aplikovaných na tyto itemy a také příklad grafického zobrazení pomocí screenu.




**Obrázek 29: Současný stav monitoringu**

### 6.5.1 Detailní popis zařízení a jejich monitoring

Monitoring je prováděn na zařízeních, SW a OS, které jsou popsány níže, kde u všech jmenovaných se monitoruje jejich dostupnost.

Na zařízeních Cisco Routery, Switche a Firewally jsou monitorovány jednotlivá rozhraní a na těchto rozhraních jsou monitorovány názvy rozhraní, datový tok, packet drop a stavy jednotlivých rozhraní. Dále jsou na těchto zařízeních monitorovány napájecí zdroje a chlazení, kde u těchto hodnot je monitorována teplota, jejich stav. V neposlední řadě se monitoruje také vytížení procesoru a využití paměti. Jako doplněk jsou ze zařízení vyčítány informační hodnoty typu název zařízení, verze OS, umístění zařízení, případně kontakt na správce zařízení.

Dalším typem, nad kterým probíhá monitoring jsou zařízení Cisco Unified Computed System (Cisco UCS), které slouží jako HW pro hypervisory VMware ESXi,

na kterém jsou spuštěny virtuální stanice. Na zařízeních Cisco UCS je monitorováno celé šasí, na kterém jsou monitorovány teploty jednotlivých ventilátorů chlazení a všech napájecích zdrojů. U těchto hodnot je také monitorován jejich stav. Dále jsou monitorované jednotlivé „žiletky“, u kterých se vyčítají pouze informační hodnoty jako je počet jader, threadů a teploty procesorů.

Nad hypervisorem VMware ESXi je nastaven monitoring jednotlivých jader a jejich vytížení. Tyto hodnoty jsou pak sumarizovány a je zobrazena průměrná hodnota vytížení celého procesoru. Sleduje se také velikost paměti, její využití a volné místo. Dále jsou monitorována jednotlivá připojená úložiště z diskového pole, jejich využití a velikost volného místa. V neposlední řadě jsou nad VMware ESXi monitorovány jednotlivé virtuální stanice, stav těchto stanic (zda jsou zapnuté nebo vypnuté) a zda je na dané virtuální stanici nainstalován VMware agent. Jako dodatečné informace jsou vyčítány hodnoty typu verze VMware ESXi, uptime daného hypervisoru a jeho název.

Nad diskovým polem NetApp jsou pak monitorovány všechny vložené disky a jejich stavy. Monitorují se jednotlivé datové oddíly, jejich volná kapacita, využití a to jak v agregovaném stavu, tak v reálném místě na disku. Dále je monitorováno, zda jsou disková pole v clusteru a zda umožňují režim HA (High Availability). Stejně jako u přechozích prvků jsou zde monitorována jednotlivá napájení, chlazení, jejich teplota a stav.

# 7 Závěr a vyhodnocení

V úvodu práce je zmíněn základní popis přenosové a distribuční soustavy. Jsou zde uvedeny komunikační protokoly, které se používají pro komunikaci nejen na rozvodnách mezi jednotlivými zařízeními, jako jsou IED nebo RTU ale také mezi rozvodnami a řídicími systémy nebo dispečinkem, který používá pro monitoring distribuční soustavy systém SCADA.

V další části práce je obecně popsána problematika technologických sítí a proč je nutné tyto sítě monitorovat. V kapitole Obecné požadavky na monitorovací systémy je popsána problematika monitoringu síťových zařízení a také jsou zde popsány obecně známé prostředky pro monitoring síťové infrastruktury. Výběr monitorovacího systému pro nasazení do testovacího prostředí v ČEZ Distribuce byl zvolen s ohledem na přehlednost systému a jeho následné využití pro ostrý provoz.

Závěr práce již obsahuje konkrétní informace o systému Zabbix, který je nasazený v testovacím prostředí. Autor popisuje instalaci monitorovacího systému a jeho následnou konfiguraci s podrobnými postupy. Jednotlivé výsledky jsou v práci prezentovány především pomocí obrázků. Vybraný monitorovací systém již sleduje všechny relevantní hodnoty jako jsou teploty zařízení nebo prostor v serverovně, dostupnost virtuálních strojů, propustnost na jednotlivých rozhraní zařízení, stavy napájení a chlazení. Na základě těchto hodnot jsou vytvořeny triggery, které se spouští buď při překročení nastavené hodnoty, kterou může být překročení teploty zařízení přes 50 °C nebo na základě změně předchozí hodnoty, kterou může být například změna stavu portu.

Veškeré spuštění těchto triggerů je zobrazováno na Dashboard monitorovacího systému, který byl měl být přinejmenším kontrolován dohledovým centrem. Systém má řízený přístup pomocí LDAP a jeho účty jsou tedy řízeny pomocí doménového kontroléru, vyjma lokálního administrátorského účtu se silným, minimálně 15 znakovým heslem. V současné chvíli je systém připraven na nasazení do produkčního prostředí a jeho používání pro dohled sítě.

V příloze jsou přiloženy obrázky s podrobnějšími výsledky dosažených při konfiguraci monitorovacího systému. Pro lepší přehled jsou obrázky opatřeny popisky.

# Literatura


[1] *IED pro chránění a ovládání vývodu REF630: Popis a technická data výrobku* [online]. In: . s. 82 [cit. 2017-12-02]. Dostupné z: https://library.e.abb.com/public/608ccc229ef4dab0c1257d2a0024d07b/REF630_pg_757075_CZd.pdf

[2] UCHYTIL, T. *Testování komunikace ochran podle IEC 61850.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 48 s. Vedoucí diplomové práce doc. ing. Petr Toman, Ph.d..

[3] Přenosová a distribuční soustava 3. část - dispečink, systémy chránění, komunikace a HDO. *E.ON Distibuce* [online]. [cit. 2017-12-02]. Dostupné z: https://www.eon-distribuce.cz/o-nas/novinky/media/prenosova-a-distribucni-soustava-3-cast-dispecersky-ridici-system-systemy-chraneni-komunikace-a-hdo

[4] HDO. *Wikipedia* [online]. [cit. 2017-12-02]. Dostupné z: https://cs.wikipedia.org/wiki/HDO

[5] IEC 60870. *Wikipedia* [online]. [cit. 2017-12-02]. Dostupné z: https://en.wikipedia.org/wiki/IEC_60870

[6] CLARKE, Gordon, Deon REYNDERS a Edwin WRIGHT. *Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems*. 200 Wheeler Road, Burlington, MA 01803: Elsevier, 2004 [cit. 2017-12-02]. ISBN 07506 7995.

[7] BUREŠ, Stanislav. Modernizace a rozšíření řídicího systému rozvodny Chotějovice. In: *AllForPower* [online]. [cit. 2017-12-02]. Dostupné z: http://www.allforpower.cz/UserFiles/files/2011/Siemens_chotejovice.pdf

[8] Industrial PC & solutions Profil. In: *ELVAC* [online]. [cit. 2017-12-02]. Dostupné z: https://www.elvac.eu/ipc/download/KATALOG-IPC-Solutions-CZ.pdf

[9] Remote terminal unit. *Wikipedia* [online]. [cit. 2017-12-02]. Dostupné z: https://en.wikipedia.org/wiki/Remote_terminal_unit

[10] *IED pro chránění a ovládání vývodu REF620: Popis a technická data výrobku* [online]. In: . s. 100 [cit. 2017-12-02]. Dostupné z: https://library.e.abb.com/public/6e0ccfff4a4141dabcff49e64e3bd096/REF620_pg_758210_CSb.pdf

[11] BALDA, Pavel. *Vendulka* [online]. 2007-04-24 [cit. 2011-04-12]. SCADA a HMI systémy. Dostupné z WWW: http://vendulka.zcu.cz/Download/Free/IRS1/IRS1-.08_SCADA_HMI.pdf

[12] STODŮLKA, I. Model elektrické stanice s komunikačním protokolem IEC 61850. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 97s.

[13] MACKIEWICZ, R.E. Overview of IEC 61850 and Benefits. In: *2006 IEEE PES Power Systems Conference and Exposition* [online]. IEEE, 2006, s. 623-630 [cit. 2017-11-11]. DOI: 10.1109/PSCE.2006.296392. ISBN 1-4244-0177-1. Dostupné z: http://ieeexplore.ieee.org/document/4075831/

[14] Promotic: SCADA visualization software. In: *Promotic* [online]. Ostrava: MICROSYS [cit. 2018-04-27]. Dostupné z: https://www.promotic.eu/cz/pmdoc/WhatIsPromotic/WhatIsScada.htm

[15] *ČSN EN 60870-5-1 Systémy a zařízení pro dálkové ovládání Část 5: Přenosové protokoly: Oddíl 1: Formáty přenosového rámce*. Praha: Český normalizační institut, 1997.

[16] *ČSN EN 60870-5-101 ed. 2 (334650) Systémy a zařízení pro dálkové ovládání - Část 5-101: Přenosové protokoly - Společná norma pro základní úkoly dálkového ovládání*. Praha: Český normalizační institut, 2005.

[17] *CSN EN 60870-5-102 Systémy a zařízení pro dálkové ovládání – Část 5: Přenosové protokoly – Oddíl 102: Společná norma pro přenos integrovaných součtových hodnot v elektrizačních soustavách*. Praha: Český normalizační institut, 2000.

[18] *CSN EN 60870-5-103 Systémy a zařízení pro dálkové ovládání – Část 5-103: Přenosové protokoly – Společná norma pro informační rozhraní ochran.* Praha: Český normalizační institut, 2000.

[19] *ČSN EN 60870-5-104 ed. 2 (334650) Systémy a zařízení pro dálkové ovládání - Část 5-104: Přenosové protokoly - Síťový přístup pro IEC 60870-5-101 používající normalizované transportní profily*. Praha: Český normalizační institut, 2007.

[20] *ČSN EN 60870-5-2 Systémy a zařízení pro dálkové ovládání - Část 5: Přenosové protokoly Oddíl 2: Procedury linkového přenosu*. Praha: Český normalizační institut, 1997.

[21] *ČSN EN 60870-5-3 Systémy a zařízení pro dálkové ovládání. Část 5: Přenosové protokoly. Oddíl 3: Obecná struktura aplikačních dat.* Praha: Český normalizační institut, 1996.

[22] BURDA, Karel. *Bezpečnost informačních systémů* [online]. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013 [cit. 2018-04-23]. ISBN 978-80-214-4890-2.

[23] IEC 61850: soubor norem pro komunikaci v energetice s velkým potenciálem výhod. *AUTOMA: časopis pro automatizační techniku* [online]. [cit. 2017-12-02]. Dostupné z: http://automa.cz/cz/casopis-clanky/iec-61850-soubor-norem-pro-komunikaci-v-energetice-s-velkym-potencialem-vyhod-2010_03_40771_5154/

[24] *IED pro chránění a ovládání vývodu REF630: Popis a technická data výrobku* [online]. In: . s. 82 [cit. 2017-12-02]. Dostupné z: https://library.e.abb.com/public/608ccc229ef4dab0c1257d2a0024d07b/REF630_pg_757075_CZd.pdf

[25] *ČSN EN 61850-4 ed. 2 Komunikační sítě a systémy pro automatizaci v energetických společnostech – Část 4: Systémové a projektové řízení*. Praha: Český normalizační institut, 2012.

[26] KIRRMANN, Hubert. Introduction to the IEC 61850 electrical utility communication. *Mafiadoc* [online]. 2012, , 59 [cit. 2017-12-09]. Dostupné z: https://mafiadoc.com/download/introduction-to-the-iec-61850-electrical-utility-communication-epfl_59daf3ad1723dde9ef145b3e.html

[27] KAHÁNEK, Michal. *Ethernetové přepínače pro integraci inteligentních elektrických rozvodných stanic s duální podporou MMS a SNMP* [online]. In: . ELVAC [cit. 2017-12-02]. Dostupné z: https://www.elvac.eu/Portals/0/Docs/Clanky/Energetika/2014_03_ethernetove_prepinace_pro_integraci_inteligentnich_rozvodnych_stanic.pdf?ver=2015-11-16-143600-287

[28] ČSN EN 61850-7-2 ed. 2 Komunikační sítě a systémy pro automatizaci v energetických společnostech – Část 7-2: Základní informační a komunikační *struktura – Abstraktní rozhraní pro komunikační služby (ACSI)*. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2011.

[29] APOSTOLOV, Alexander. IEC 61850 Substation Configuration Language and Its Impact on the Engineering of Distribution Substation Systems Notes. In: Slideplayer [online]. [cit. 2018-05-11]. Dostupné z: http://slideplayer.com/slide/4537031/15/images/16/SCL+Examples.jpg

[30] *ČSN EN 61850-6 ed. 2 Komunikační sítě a systémy pro automatizaci v energetických společnostech – Část 6: Konfigurační popisový jazyk pro komunikaci v elektrických stanicích týkající se IED.* Praha: Český normalizační institut, 2010.

[31] SCHWARZ, Karlheinz. Cost for IEC 61850 versus DNP3 or IEC 60870-5-104. *News on IEC 61850 and related Standards* [online]. 2010 [cit. 2018-04-22]. Dostupné z: http://blog.iec61850.com/2010/11/cost-for-iec-61850-versus-dnp3-or-iec.html

[32] MARIK, Ondrej; ZITTA, Stanislav. Comparative analysis of monitoring system for data networks. In: *Multimedia Computing and Systems (ICMCS), 2014 International Conference on*. IEEE, 2014. p. 563-568.

[33] BOUŠKA, Petr. SNMP - Simple Network Management Protocol. *Www.samuraj-cz.com* [online]. 2006 [cit. 2018-04-15]. Dostupné z: https://www.samuraj-cz.com/clanek/snmp-simple-network-management-protocol/

[34] Internet Control Message Protocol. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-04-15]. Dostupné z: https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol

[35] Protokol ICMP. In: *UIS Mendelu* [online]. [cit. 2018-04-15]. Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=597

[36] DOSTÁL, Radim. Sokety a C/C++: ICMP protokol. In: *Root* [online]. 2003 [cit. 2018-04-15]. Dostupné z: https://www.root.cz/clanky/sokety-a-c-icmp-protokol/

[37] BOUŠKA, Petr. SNMP - Simple Network Management Protocol. In: *Samuraj-CZ* [online]. 2006 [cit. 2018-04-15]. Dostupné z: https://www.samuraj-cz.com/clanek/snmp-simple-network-management-protocol/.

[38] Secure Shell. In: *Wikipedia: the free encyclopedia*[online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-05-14]. Dostupné z: https://cs.wikipedia.org/wiki/Secure_Shell

[39] Simple Network Management Protocol. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-04-15]. Dostupné z: https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol

[40] HOLMAN, Jan. SNMP a monitoring sítě. *Fakulta Informatiky: Masarikova universita* [online]. [cit. 2018-04-19]. Dostupné z: https://www.fi.muni.cz/~kas/pv090/referaty/2015-podzim/snmp.html

[41] GRILLO, Pete; WALDBUSSER, Steven. Host resources MIB. 1993.

[42] Management information base. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-04-15]. Dostupné z: https://en.wikipedia.org/wiki/Management_information_base

[43] Zabbix Documentation 3.4: 3 Installation from sources. *Zabbix.com* [online]. [cit. 2017-12-02]. Dostupné z: https://www.zabbix.com/documentation/3.4/manual/installation/install

[44] Zabbix – Enterprise-Class Monitoring Solution for everyone. In: *Https://www.zabbix.com/product* [online]. [cit. 2017-11-11]. Dostupné z: https://www.zabbix.com/files/Brochures/General_Brochure_3.2.pdf

[45] S. V. Mescheryakov, D. A. Shchemelinin, "Analytical overview of Zabbix International Conference 2013", *SPbSPU Journal. Computer Science. Telecommunication and Control Systems*, 2014, no. 1(188), 91–98

[46] Nagios Support Knowledgebase. *Support.nagios.com* [online]. [cit. 2017-11-12]. Dostupné z: https://support.nagios.com/kb/article/nagios-core-installing-nagios-core-from-source.html#CentOS

[47] LPS:Nagios. *Http://support.zcu.cz* [online]. [cit. 2017-11-12]. Dostupné z: http://support.zcu.cz/index.php/LPS:Nagios

[48] Nagios Features & Capabilities. *Www.nagios.org* [online]. [cit. 2017-11-12]. Dostupné z: https://www.nagios.org/about/features/

[49] Nagios XI: Enterprise Server and Network Monitoring Software. *Www.nagios.org* [online]. [cit. 2017-11-12]. Dostupné z: https://www.nagios.com/products/nagios-xi/

[50] Nagios. *Wikipedia* [online]. [cit. 2017-11-12]. Dostupné z: https://sk.wikipedia.org/wiki/Nagios

[51] Icinga 2 API. *Icinga 2* [online]. [cit. 2017-12-03]. Dostupné z: https://www.icinga.com/docs/icinga2/latest/doc/12-icinga2-api/#introduction

[52] Getting started. *Icinga 2* [online]. [cit. 2017-12-03]. Dostupné z: https://www.icinga.com/docs/icinga2/latest/doc/02-getting-started/#configuring-db-ido-mysql

[53] Configuring Icinga 2: First Steps. *Icinga 2* [online]. [cit. 2017-12-03]. Dostupné z: https://www.icinga.com/docs/icinga2/latest/doc/04-configuring-icinga-2/

[54] HERNANTES, Josune; GALLARDO, Gorka; SERRANO, Nicolas. IT infrastructure-monitoring tools. *IEEE Software*, 2015, 32.4: 88-93.

# Seznam symbolů, veličin a zkratek

| | |
|---|---|
| ANSI | American National Standards Institute |
| APCI | Application Protocol Control Information |
| APDU | Application Protocol Data Unit |
| API | Application Programming Interface |
| ASDU | Application Service Data Unit |
| CDP | Cisco Discovery Protocol |
| ČEPS | Česká Energetická Přenosová Soustava |
| ČEZ | České Energetické Závody |
| ČSN | Česká Státní Norma |
| DAS | Data Aquisition System |
| DB | Database |
| DŘS | Dispečerský Řídící systém |
| DTS | Distribuční trafostanice |
| FRU | Field Replaceable Unit |
| FW | Firewall/Firmware |
| GOOSE | Generic Object Oriented Substation Events |
| GSSE | Generic Substation State Events |
| GUI | Graphic User Interface |
| HA | High Availability |
| HDO | Hromadné dálkové ovládání |
| HMI | Humane Machine Interface |
| HW | Hardware |
| ICMP | Internet Control Management Protocol |
| IEC | International Electrotechnical Commission |
| IED | Intelligent Electronic Device |
| IEEE | Institute of Electrical and Electronics Engineers |

| | |
|---|---|
| INT | Integer |
| IP | Internet Protocol |
| ISO/OSI | International Standards Organization / Open Systen Interconnection |
| MIB | Management Information Base |
| MMS | Manufacturing Message Specification |
| NTP | Network Time Protocol |
| OID | Object Identifier |
| OS | Operační Systém |
| PRE | Pražská Energetika |
| RAM | Random Access Memory |
| RO | Read Only |
| RTU | Remote Terminal Unit |
| RW | Read Write |
| SAS | Substation Automation System |
| SCADA | Supervisory Control And Data Acquisition |
| SCL | Substation Configuration Language |
| SIEM | Security Information and Event Management |
| SMV | Sampled Measured Values |
| SNMP | Network Management Protocol |
| SSH | Secured Shell |
| STO | Systém Technické Ochrany |
| SW | Switch/Software |
| TC | Technical Committee |
| TCP | Transmission Control Protocol |
| UCS | Unified |
| UDP | User Datagram Protocol |
| VPN | Virtual Private Network |
| XML | eXtensible Markup Language |

# Seznam příloh