DIPLOMAMUNKA

Háló Albert Tamás

Debrecen

2010

Debreceni Egyetem

Informatikai Kar

HÁLÓZATI MENEDZSMENT ESZKÖZÖK ALKALMAZÁSA A GYAKORLATBAN

Témavezető:

Dr. Almási Béla

egyetemi docens

Készítette:

Háló Albert Tamás

Programtervező Matematikus

Debrecen

2010

Tartalomjegyzék

1. Bevezetés

1.1. Általános bevezetés

Napjainkban a hálózat megtervezése, beállítása, biztonságos és megfelelő működésének a biztosítása jelentős szerepet kapott. Bármilyen jellegű hiányosság a hálózat nem megfelelő működéséhez, s ebből adódóan egyéb problémákhoz is vezethet. Ezért manapság a vállalatoknál jelentős szerepet fordítanak a hálózat biztonságos felépítésére és felügyelt működtetésére. Ezek beállításában, konfigurálásában, felügyeletében, hibajavításában vannak segítségükre a hálózati menedzsment eszközök.
A vállalati hálózat működése során fontos, hogy bizonyos alkalmazások magasabb prioritást élvezzenek, avagy fix sávszélességet kapjanak a megfelelő működés érdekében, avagy egyes erőforrásokat teljes mértékben megkapjanak bizonyos időre. Ezeket az alkalmazás számára elérhetővé kell tudni tennie a menedzsment eszköznek. A QoS (Quality of Service) jelzéssel ellátott eszközök az ilyen jellegű kéréseket végre tudják hajtani.
A diplomamunkámban bemutatom, mi is a hálózati menedzsment lényege, milyen alapokra épít, hogyan épül fel. Ezután bemutatom röviden, az évek során kialakul protokollokat. Majd végül bemutatok néhány konkrét eszközt, azok működését, beállítási lehetőségeit.

1.2. Hálózati menedzsment

A hálózati menedzsment olyan aktivitásokra, metódusokra, eljárásokra, és eszközökre vonatkozik, melyek hozzátartoznak a működéshez, az adminisztrációhoz, a fenntartáshoz és a hálózati rendszer ellátásához [1].

- A működés alatt a hálózat fenntartását, és a megfelelő működést értjük. Ide tartozik a hálózat monitorozása, ezáltal az esetleges hibák kiszűrése, s kijavítása.
- Az adminisztrációhoz tartozik a hálózati erőforrások nyomon követése, a hálózat teljes felügyelete.
- A fenntartás az esetleges javítások, újítások véghezvitelét jelenti. Esetlegesen egy hibás router cseréjét, egy új switch hozzáadását.
- Az ellátás alatt az erőforrások konfigurálását értjük, adott funkció ellátása érdekében.

Segítségükkel be lehet állítani, hogy az egyes alkalmazások mely erőforrásokat érjék el. Beállíthatjuk, hogy az egyes alkalmazások mekkora sávszélességet kapjanak, ezzel is kevésbé terhelve a hálózatot. Az alkalmazásokhoz prioritásokat rendelhetünk, ezáltal meghatározva, például egyidejű futás esetén melyik alkalmazás mekkora mértékben használhatja a hálózati erőforrásokat. Segítséget nyújt a biztonságos hálózat kialakításához, az esetleges hibák detektálásához, s ezek kijavításához. Ezekkel részletesebben fogok foglalkozni még a diplomamunkámban.

2. Hálózati menedzsment funkciói

A hálózati menedzsment öt funkcionális területre osztható fel. Úgymint: hiba-, konfiguráció-, elszámolás-, teljesítmény-, biztonság menedzsment [13]. Ezek az ISO hálózatmenedzsment modell elemei.

2.1. Hibamenedzsment

A hibamenedzsment célja:

- Az esetleges hibák érzékelése.
- A keletkezett hiba naplózása.
- Szükség esetén a megfelelő személyzet értesítése.
- Optimális esetben a hiba automatikus javítása.

A hibamenedzsment feladatai:

- A hiba körülményeinek meghatározása, a keletkezett hiba izolálása.
- A keletkezett hiba javítása, javítás tesztelése.
- Mind a hiba, mind a javítás dokumentálása.

A hibamenedzsment elvégzésének két különböző módja lehetséges, az aktív illetve a passzív mód.

Passzív hibamenedzsment esetén a hiba bekövetkezésekor a hibás eszközről származó riasztásokat összegyűjti a menedzsment eszköz. Ha az eszköz kellőképp intelligens, akkor a keletkező riasztásokat nem csak naplózza, hanem egy hibaüzenetet generál belőle, melyet a megfelelő eszköznek elküldve automatikusan javítja a hibát. Ellenkező esetben a megfelelő személyzet szükséges a hiba kijavításához.

Aktív hibamenedzsment esetén monitorozzuk az eszközöket, az eszközökre kéréseket küldünk (pl.: PING), melyekre az eszköz működése esetén érkezik válasz, ellenkező esetben nem kapunk választ. Ha egy eszköz nem reagál, akkor a monitorozó eszköz riaszt, s kéri a hiba megfelelő kijavítását.

A hibák dokumentálása fontos, mert jellemezzük vele a hibákat. A dokumentációba belekerülnek a hibák kijavítását szolgáló lépések. Későbbi hiba megjelenése esetén a korábbi hibákból kiindulva egyszerűbb lehet a javítás.

Legfontosabb funkciói:

- Küszöbök meghatározása a lehetséges hibák körülményeihez.
- Rendszerállapot, és a kihasználtság monitorozása.
- Folyamatos vizsgálat a vírusok és a trójai vírusok ellen.
- Általános diagnosztika.
- Távoli hozzáférés a rendszer egyes elemeihez.
- Riasztások, melyek figyelmeztetik az adminisztrátort és a felhasználókat a lehetséges hibaforrásokról és a jelenlegi hibákról.
- Hiba helyének meghatározása, megkeresése.
- Hiba kialakuláshoz vezető körülmények automatikus javítása.
- Automatikus megoldás a kialakult problémára.
- Részletes dokumentáció a rendszer állapotáról, és a futó műveletekről.

2.2. Konfigurációmenedzsment

A konfigurációmenedzsment feladata, hogy a hálózati-, és a rendszerkonfigurációkról szolgáltasson információkat. A rendszer aktuális állapotáról tartalmaz információkat.
A konfigurációmenedzsment célja:

- Összegyűjtse, és tárolja a hálózati eszközök konfigurációját.
- Egyszerűbbé tegye az eszközök konfigurálását.
- Nyomon kövesse a konfigurációban bekövetkezett változásokat.
- Kialakítsa az útvonalat a „non-switched” hálózatokon keresztül.

Segítségével a különböző verziójú hardver és szoftver elemek hálózati működése nyomon követhető és kezelhető. A konfigurációmenedzsment alrendszerek ezen információkat egy rendezett adatbázisban tárolják. Hiba esetén kinyerhetőek belőle a nyomravezető adatok.

2.3. Elszámolás menedzsment

Célja a hálózatot leíró paramétereknek a mérése, ezek segítségével a hálózat működéséről statisztikákat készíthetünk. Mérhető a felhasználók hálózathasználata, s megfelelő eszközökkel szükség esetén szabályozható, ezzel csökkentve a hibák számát, a hiba kialakulásának esélyét, valamint hogy a felhasználók egyenlő mértékben kihasználhassák a hálózatot.

A rendszer elsősorban meghatározza az egyes hálózati erőforrások kihasználtságát. Az eredmények által lehetőség van arra, hogy a túlterhelt erőforrásokhoz beállítsunk egy küszöböt, az optimális működés érdekében. A küszöbérték meghatározása nehézséget okoz, mivel figyelni kell arra, hogy olyan küszöbértéket válasszunk, mellyel a felhasználó erőforrás elérése továbbra is problémamentes.

Mérhető paraméterek:

- Lemez kihasználtság.
- Kapcsolatok kihasználtsága.
- A CPU terheltség.

2.4. Teljesítménymenedzsment

A teljesítménymenedzsment célja a hálózat teljesítményének mérése különböző paraméterek segítségével, valamint a hálózat teljesítményének szinten tartása. Az adatokat a továbbítás céljából kis egységekre bontjuk (pl.: keretek, csomagok). A továbbítás során számos tényező játszik szerepet:

- Késleltetés: Időbe telhet, amíg egy csomag kézbesítve lesz a célszámítógépre. Megfelelő protokoll esetén a kapó fél visszajelez, hogy megkapott minden adatot a küldő félnek. Ez az idő mérhető. A jel terjedési sebessége nagyjából 5 mikro-szekundum kilométerenként. Így a két csomópont közötti „távolság” aktív tényező.
- Csomagvesztés: Néha a csomag elveszik a hálózatban. Ez hibát okozhat, avagy túlterhelheti a közvetítő réteget.

- Újraküldés: Mikor egy csomag elveszik, akkor újra kell küldeni. Ezáltal többszörösen is csúszik az adatátvitel.
- Átbocsátóképesség: a hálózati forgalmat mérik vele, általában a mértékegysége Kb/s, de manapság egyre gyakrabban elterjedt a Mb/s, s kezd megjelenni a Gb/s is.
- Ütközés.

Egyéb teljesítményt meghatározó tényezők

- Erőforrás használat: az alkalmazások szinte minden alkalommal használják a központi erőforrásokat.
- Válaszidő: a felhasználó egy kérésére mind hardver, mind szoftver miatti késések jelennek meg. A teljesítmény szempontjából fontos válaszidő ezen késések kombinációjából áll elő.

Ezek, s még más tényezők befolyásolják a hálózat teljesítményét. Ezáltal bizonyos esetekben (pl.: hálózat túlterhelése, sérülés) a hálózat nagyon lassú, vagy akár működésképtelen, ezáltal hatással van az alkalmazásokra.

A teljesítménymenedzsment magában foglalja a hálózat mérését, modellezését, tervezését, optimalizálását, hogy a hálózat sebessége, megbízhatósága, teljesítménye az adatforgalom számára megfelelő legyen, figyelembe véve azt is, hogy az egyes alkalmazások más-más erőforrás igénnyel jelennek meg a hálózaton.

A teljesítménymenedzsment figyeli, hogy az egyes meghatározó tényezők milyen értékek között mozognak, s ha egy bizonyos küszöböt átlépnek, akkor jelez a menedzsment rendszer felé.

Hálózat tervezése, avagy bővítése esetén lehetőség van arra, hogy szimulációk segítségével meghatározzuk, hogy milyen hatással lesz a teljesítményre, esetleges hibákat fog-e okozni. Ezáltal felhívja a rendszer üzemeltetőinek a figyelmét a lehetséges problémákra.

2.5. Biztonságmenedzsment

A biztonságmenedzsment feladata, hogy vezérelje a hálózati erőforrásokhoz való hozzáférést a lokális elveknek megfelelően, hogy a hálózat működését ne lehessen szabotálni se

véletlenül, se szándékosan, valamint a bizalmas információk megfelelő jogosultság nélkül ne kerülhessen ki. A biztonságmenedzsment rendszer monitorozza a felhasználók által igénybe vett erőforrásokat, valamint megtagadja a hozzáférési jogot a nem megfelelő belépési kísérlet esetén.

A biztonságmenedzsment rendszerek a hálózati erőforrásokat szétválasztják engedélyezett, és nem engedélyezett részekre. Így lehet olyan felhasználó, aki semmilyen hálózati erőforráshoz nem férhet hozzá, például, aki külsősként dolgozik egy cégnél. Lehet olyan felhasználó, aki a cégen belül dolgozik, de még sincs jogosultsága ahhoz, hogy az összes hálózati erőforrást igénybe vegye.

Az ilyen rendszerek külön kezelik a bizalmas adatokhoz való hozzáférést. Meghatározzák, hogy ezekhez az adatokhoz milyen jogosultsággal bíró felhasználók férhetnek hozzá. A rendszer monitorozza a belépési pontokat, valamint feljegyzi a jogosulatlan belépési kísérleteket.

Két fő funkciója van a jogosultságkezelés, és a titkosítás.

3. Hálózat menedzsment protokollok

A hálózatok folyamatos bővülése és kezelhetősége miatt a hálózatok menedzselése egyre nehezebb feladatnak bizonyult. Emiatt szükség volt az irányelvek meghatározására.

3.1. Protokollok

TMN (Telecommunications Management Network)

Az ITU-T által definiált protokoll a kommunikációs hálózatok rendszerének kezeléséhez. Olyan keretrendszerrel rendelkezik, mely megvalósítja az összekapcsolódást és a kommunikációt különböző műveleti rendszerek és telekommunikációs hálózatok között. Használható ISDN, ATM, GSM hálózatok kezelésére.

CMIP (Common Management Information Protocol)

Hálózat menedzsment protokoll. A CMIS (Common Management Infomation Service) szolgáltatásainak implementációját foglalja magában, lehetővé téve a kommunikációt két hálózatmenedzsment alkalmazás között. A CMIP az OSI menedzsment protokollból fejlődött ki az ITU-T X.700-as sorozat ajánlásai alapján.

CMOT (CMIP Over TCP/IP)

CMIP-et használó hálózati menedzsment, az IP alapú hálózatok kezelésére.

SNMP (Simple Network Management Protocol)

Egyszerű hálózat menedzsment protokoll. A TCP/IP család része, melyet az IETF definiált. Leginkább a hálózati menedzsment rendszerekben alkalmazzák, hogy monitorozzák a hálózatra kapcsolt eszközök állapotát. Részletesebben a 4.1.-ben.

WBEM (Web-Based Enterprise Management)

A WBEM a nagykiterjedésű hálózatok felügyeleti információinak elérését és megosztását szolgáló szabványos, de nem szabadalmaztatott megoldások kifejlesztését célzó kezdeményezés. Felügyeleti infrastruktúrára vonatkozó szabványokat fogalmaz meg, valamint

lehetővé teszi többféle hardver- és szoftver-felügyeleti rendszerből származó információk összegzését.

WSDM (Web Services Distributed Management)

A WSDM egy webes szolgáltatás, mely mérvadó más szolgáltatások kezeléséhez és monitorozásához. A célja egy olyan jól definiált hálózati protokoll, mely más, WSDM-kompatibilis szolgáltatásokat ellenőriz.

3.2. Információs modellek

MIB (Management Information Base)

Egy virtuális adatbázis, mely tartalmazza a hálózaton vezérelt összes eszköz leírását. Az adatbázis egyes objektumai eltérnek egymástól aszerint, hogy egy adott eszközön futó alkalmazás állapotát írja le, avagy szöveges formában jellemzi az adott eszközt név-leírás formában. Ezen információk pontosan definiáltak, így a különböző hálózat menedzsment rendszerek is elérik, és fel tudják használni. Részletesebben a 4.2-ben.

CIM (Common Information Model)

A CIM egy nyílt szabvány, mely a WBEM részét alkotja. Meghatározza, hogy az informatikai környezetben kezelt elemek, általános objektumhalmazokként és a közöttük levő kapcsolatokkal legyenek reprezentálva. Ezáltal lehetőség van ezen elemekre vonatkozóan, egy konzisztens menedzsmentre, gyártótól függetlenül.

A CIM szabvány magában foglalja a CIM alap specifikációt, és a CIM sémát.

- A specifikáció definiálja CIM szerkezetét, mely magában foglalja a sémát leíró nyelvet, és a CIM leképezését szolgáló metódusokat a többi információs modell számára.
- A séma leírja az egyéni objektumhalmazokat és a közöttük levő kapcsolatokat, melyek az információs környezetben kezelt elemeket jellemzi.

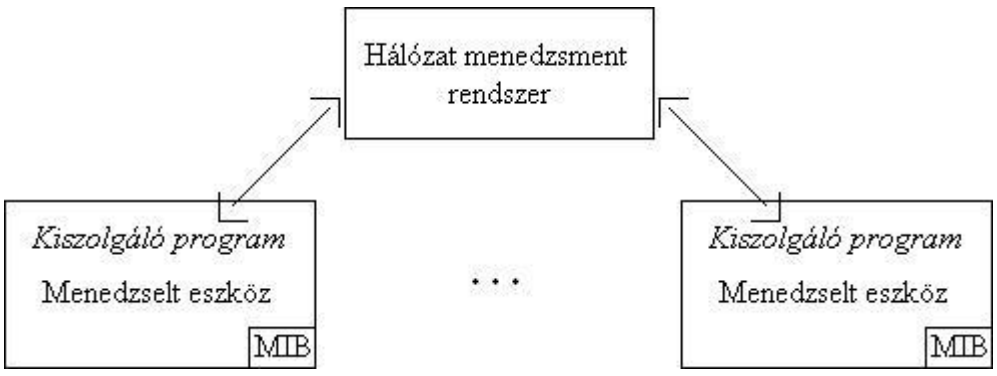
4. Az SNMP és a MIB

4.1. SNMP (Simple Network Management Protocol)

A 80-as évek közepén fejlesztették ki, a TCP/IP protokollhierarchia része. Megoldást jelentett az egyre növekvő hálózatok menedzselésére. Ahogy a neve is tartalmazza, egyszerű a működése, lényegében egy kérés-válasz protokoll. Eredetileg csak átmeneti megoldásnak szánták, ezért nem kerültek bele komplexebb megoldások. A leváltására két protokollt fejlesztettek ki, a SMNPv2 ami az elődjétől elég sokat örökölt, valamint az összetettebb CMIP-et. Mégis az egyszerű, könnyű használatósága miatt a népszerűsége megmaradt.

A kommunikációt az UDP felett valósítja meg, általában a 161-es és a 162-es portokon. Habár TCP-n keresztül is megvalósítható a kommunikáció, nem igazán elterjedt. Az UDP kevésbé biztonságos, de gyorsabb.
Három fő alkotóelemből áll:

- menedzselt eszköz,
- kiszolgáló program,
- hálózati menedzsment rendszer.



A menedzselt eszközben (router, switch, híd, nyomtató, számítógép, …) implementálva van a SNMP interfész, ami lehetővé teszi, hogy az adott eszközre jellemző adatokhoz hozzá lehessen férni. A menedzselt eszköz, a rendszer által küldött kérésre válaszolva, a megfelelő adatokat elküldi. Csak olyan eszközök menedzselésére van lehetőség, melyek képesek a magukat leíró állapotváltozókat elküldeni.

A kiszolgáló program egy hálózat menedzsment szoftver, amely a menedzselt eszközön található. Saját tudással rendelkezik, a menedzsment szempontjából lényeges, az eszközre vonatkozó információkból, melyeket átalakít az SNMP formátumára, és vissza. Alapfeladata a menedzsment rendszer által üzenetei alapján:

- a MIB változók értékeinek vizsgálata, és elküldése,
- az információk módosítása.

A hálózati menedzsment rendszeren futnak az eszközöket vezérlő és monitorozó alkalmazások. SNMP utasítások segítségével lekérhetjük, valamint kezelhetjük a kiszolgáló programokat. Napjainkban számos ilyen rendszer van jelen. Erről részletesebben az 5. fejezetben.

Az SNMP csomag felépítése:

- SNMP-csomag fejléc:
  - Verziószám
  - Community string: az authentikációt foglalja magában.
- PDU

Az SNMP üzenet formája az egyes SNMP verziókban megegyezik, csak a PDU rész különbözik.
(További leírások: [3][4][5])

4.1.1. SNMPv1

A GetRequest, a GetNextRequest, a GetResponse és a SetRequest PDU felépítése [4]:

PDU type (4 byte)	Request ID (4 byte)	Error status (4 byte)	Error index (4 byte)	Object 1, value 1	Object 2, value 2	…

- PDU type: meghatározza a PDU típusát.
  - 0 – GetRequest
  - 1 – GetNextRequest
  - 2 – GetResponse
  - 3 – SetRequest

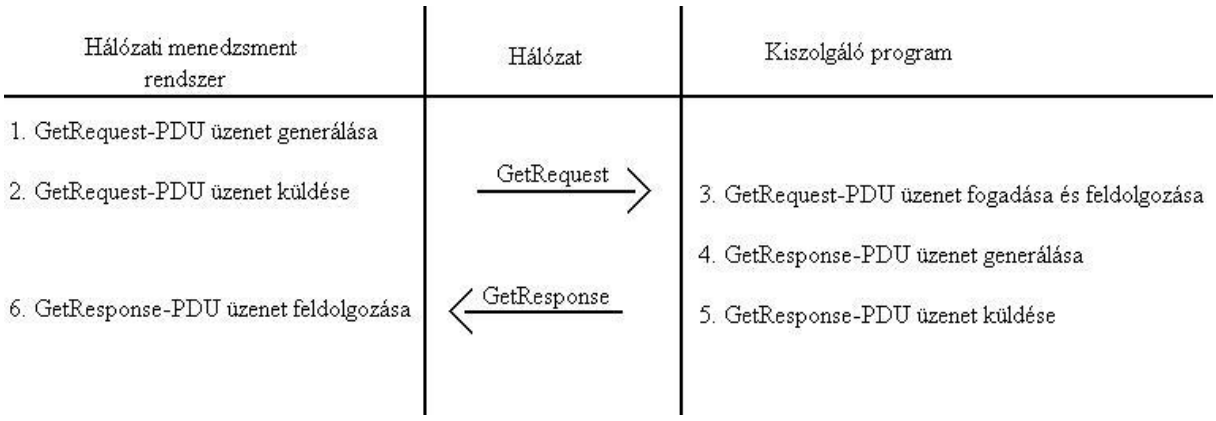
- Request ID: segítségével egy kérés megkülönböztethető a többi, még feldolgozásra váró kéréstől. A kiszolgáló program a válaszában, a kérés azonosítóját küldi vissza.
- Error status, error index: csak válasz esetén állítódhat neki érték, minden más esetben 0. Esetleges hiba esetén állítódnak be az értékei, mely segítségével leírja a hibát.
- Object *x*, value *x*: adott változó lekérését, illetve beállítását szolgálja.

A Trap PDU felépítése:

PDU type (4 byte)	Enterprise	Agent address (4 byte)	Generic trap type (4 byte)	Specific trap type (4 byte)	Timestamp (4 byte)	Object 1, value 1	…

- PDU type: meghatározza a PDU típusát (4 – Trap).
- Enterprise: meghatározza, hogy mely menedzselő modulban generálódott az üzenetet.
- Agent address: a menedzselt eszköz IP címe.
- Generic trap type: megmondja, hogy milyen esemény váltotta ki az üzenetet.
- Specific trap type: az esetben van értéke, ha az üzenetet valamely egyedi modul váltotta ki.
- Timestamp: a sysUpTime objektum értékét tartalmazza, időbélyeggel látja el.

GetRequest-PDU: A menedzsment rendszer generálja, és elküldi a kiszolgáló programnak. A kiszolgáló program a kért változók kikeresi a MIB-ből. Ha a keresés és az változó értékének kiolvasása során nem történt semmilyen hiba, akkor egy GetResponse-PDU-ban visszaküldi a kérés azonosítóját, és a kért változó értékét. Hiba esetén a GetResponse-PDU „error” mezői is értéket kapnak, mely a hiba leírását szolgálja.



GetNextRequest-PDU: A menedzsment rendszer generálja, és elküldi a kiszolgáló programnak. A kiszolgáló program kikeresi azt a változót, amely a kért változó azonosítóját lexikális sorrendben követi. Ha a keresés és az változó értékének kiolvasása során nem történt semmilyen hiba, akkor egy GetResponse-PDU-ban visszaküldi a kérés azonosítóját, és a kért változó értékét. Hiba esetén a GetResponse-PDU „error" mezői is értéket kapnak, mely a hiba leírását szolgálja.

GetResponse-PDU: A kiszolgáló program generálja válaszul a kapott kérésekre.

SetRequest-PDU: A menedzsment rendszer generálja, és küldi a kiszolgáló programnak. A kiszolgáló program kikeresi azt a MIB változót, melyre a kérés hivatkozik, és beállítja a kapott értékre. Ha a keresés és az változó értékének beállítása során nem történt semmilyen hiba, akkor egy GetResponse-PDU-ban visszaküldi a kérés azonosítóját, és a kért változó értékét. Hiba esetén a GetResponse-PDU „error" mezői is értéket kapnak, mely a hiba leírását szolgálja.

Trap-PDU: Olyan speciális PDU, mely a kiszolgáló program oldaláról automatikusan generálódik, és elküldi a menedzsment rendszernek, ha egy meghatározott állapotot érzékel. Az ilyen állapotokat a következők lehetnek:

- cold_start: az entitás re-inicializálta magát, emiatt a kiszolgáló program konfiguráció módosulhat.
- warm_start: az entitás re-inicializálta magát, de a kiszolgáló program konfiguráció nem módosult.
- link_down: a kiszolgáló program egyik kommunikációs kapcsolatában hiba történt.
- link_up: a kiszolgáló program egyik kommunikációs kapcsolata helyre állt.
- auth_fail: sikertelen az azonosítás, jogosulatlan hozzáférés.
- egp: jelzi, hogy az entitás (pl.: router) az egyik kapcsolatát elvesztette
- enterprise: jelzi, hogy néhány gyártó specifikus esemény történt.

Előnyei:

- Interneten használták, tesztelték.
- Egyszerű a használata, könnyű implementálni az eszközökre.

– Kicsi az erőforrás igénye.

Hátrányai:

– Biztonság hiánya, nincs titkosítás.
– Hatékonysági problémák vannak a kommunikációs protokollban.
– Nagytömegű adatlehívás nem megoldható.

4.1.2 SNMPv2

A 2-es verzió átvette elődjének a jó tulajdonságait. Belekerült a GetBulk utasítás, mely segítségével megoldható a nagytömegű adatlehívás, valamint megvalósították a menedzser-menedzser kommunikációt. Megpróbálták megvalósítani a biztonságot, de a komplexitása miatt nem lett elterjedt.

Az SMTPv2c verzió annyiban tér el a 2-es verziótól, hogy átvette az 1-es verzió a „Community string”-es biztonsági megoldást.

A 2-es verzió inkompatibilis az 1-es verzió két kulcsfontosságú területén is. Az üzenetformátumokban és a protokoll műveletekben. Az üzenetei eltérnek mind a fejlécben, mind a PDU szerkezetében az 1-es verzió üzeneteitől.

A Get, a GetNext, a Response, Inform, Set és a Trap PDU felépítése:

PDU type (4 byte)	Request ID (4 byte)	Error status (4 byte)	Error index (4 byte)	Object 1, value 1	Object 2, value 2	…

A GetBulk PDU felépítése:

PDU type	Request ID	Non repeaters	Max repetitions	Object 1, value 1	Object 2, value 2	…

– Non repeaters: Meghatározza azon változók számát, amit csak egyszer kell lekérdezni a kapott kérés kezdetétől számítva. Általában a skalár értékű változók értékét kérjük le egyszer.
– Max repititions: Meghatározza, hogy a maradék változókat hányszor kell lekérdezni. Ez jellemzően a dinamikusan változó értékek lekérdezésére szolgál.

Hátránya, hogy továbbra sem tartalmaz titkosítást.

4.1.3 SNMPv3

SNMP 3-as verziója teljesen új biztonsági szinttel rendelkezik – ami mind az 1-es, mind a 2-es verzióból hiányzik – különválasztva az azonosítást és a jogosultságkezelést [4].

Az USM az alapértelmezett biztonsági modul a 3-as verzióban. Bővebb leírás olvasható az USM-ről a [6]-ben.

A VACM az alapértelmezett hozzáférés-kezelés modul a 3-as verzióban. Meghatározza, hogy melyik felhasználónak van jogosultsága, hogy információt kapjon a MIB-ből. Lehetőséget ad arra, hogy különböző hozzáférési szinteket állítsunk be, a MIB fa különböző részeihez. Bővebb leírás olvasható a VACM-ról a [7]-ban.

A PDU szerkezete megegyezik a 2-es verzióban használt szerkezettel.
A 3-as verzió fontos biztonsági eszközei:

- Message integrity (üzenet megszorítás): az átvitel során az adatokat ne lehessen módosítani.
- Azonosítás: biztosítja, hogy az üzenet érvényes forrásból származik.
- Titkosítás: jogosulatlan forrás ne tudjon információkat szerezni.

Az SNMPv3 a jelenlegi hivatalos SNMP szabvány.

4.2. MIB (Management Information Base)

Szoros kapcsolatban van az SNMP-vel, mivel egy kérés során a kiszolgáló program ebből nyeri az információt. A MIB a Menedzselt-eszközök Információs Bázisa. Ez egy fa szerkezetű virtuális adatbázis, mely az egyes hálózati eszközöket írja le.

Az MIB-ben lévő objektumok az ASN.1 (Abstract Syntax Notation One) segítségével vannak definiálva.

Minden objektumnak van neve, ami egyedileg azonosítja. Igazából minden objektumnak két neve van, egy szöveges neve (Object Descriptor), valamint egy numerikus neve (Object Identifier), mely meghatározza az objektum helyét a MIB hierarchiájában. A szöveges név a felhasználó-barát kezelhetőség miatt van jelen.

Az objektumok szintaxisa meghatározza a típusát, és a struktúráját. A típusa határozza meg, hogy milyen információt tartalmazhat az objektum. Két alapkategóriája létezik:

- Szokásos adattípusok, melyek az információ egyszerű részét képezik, mint például az INTEGER, az OCTET STRING, …
- Táblázatos adatok, mely összetett adatok kollekciója. Például egész értékekből álló táblázat készíthető értékhalmazok megjelenítésére

Az objektumok elérhetősége szerint lehet:

- read-create: írható, olvasható, létrehozható,
- read-write: írható, olvasható,
- read-only: csak olvasható,
- accessible-for-notify: csak SNMP értesítésekre használható,
- not-accessible: speciális célokra használható elérhetőség.

A státusza lehet kötelező, opcionális, vagy nem használt.
Valamint tartalmaz még egy szöveges leírást.

Példa az objektumdefiniálására [15]:

```
SysUpTime OBJECT-TYPE
        SYNTAX TimeTicks
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                ” The time (in hundredths of a second) since the network management portion
                  of the system was last re-initialized.”
        ::= { system 3 }
```

A példa által definiált objektum a MIB objektum név hierarchiájában a függelék 1. képének megfelelő helyre kerül.

4.3. ASN.1 (Abstract Syntax Notification One)

A számítógépes hálózatoknál az ASN.1 alapvető és rugalmas jelölésrendszert ad az adatstruktúrák megjelenítéséhez, kódolásához, dekódolásához és módosításához. Magában foglal – az objektumok szerkezetének leírásához – számos formális szabályt, mely független a gépfüggő kódolási technikáktól, valamint jól definiált, és ellentmondás mentes. Az ASN.1-ben az adatok típusokba vannak sorolva. Az ASN-1 főbb típusai a függelék Táblázat 1.-ban szerepelnek.

Lehetőség van saját típus megadására is. A saját típus definiálása esetén a típusnevünk nagybetűvel kell, hogy kezdődjön. Hivatkozást '::=' szimbólum segítségével tudunk létrehozni. Összetett típusok definiálása esetén, a típusok elemei egyedi azonosítóval rendelkeznek, melyek kisbetűvel kezdődnek.
Például: *Person ::= SEQUENCE {name PrintableString, age INTEGER}*

Az ASN-1 többféle kódolási szabállyal is rendelkezik, mely segítségével létre lehet hozni az átviteli szintaxist, ami nem más, mint egy meghatározott oktett sztring halmaz, mely az absztrakt szintaxis egy értéke. A fogadó számítógép a kapott halmazt egyértelműen tudja dekódolni, amennyiben megkapta a kódolási formára vonatkozó információt. Ilyen kódolási szabályrendszerek a következők:

- BER (Basic Encoding Rules)
- CER (Canonical Encoding Rules)
- DER (Distinguished Encoding Rules)
- XER (XML Encoding Rules)
- PER (Packed Encoding Rules)
- GSER (Generic String Encoding Rules)

(További leírások: [16][17][18])

4.3.1. BER (Basic Encoding Rules)

A kódolt adat 4 komponensből állhat, azonosító, hossz , tartalom, valamint „tartalom-lezáró” elem. A „tartalom-lezáró” elemre csak akkor van szükség, ha nem meghatározott hosszúságú alakkal dolgozunk.

azonosító	hossz	tartalom	(„tartalom-lezáró”)

Azonosító elem

Ez a komponens tartalmazza érték típusának az ASN.1 megfelelőjét a kódolt formáját, mely egy osztályból és egy számból áll.

- A 8. és a 7. bit határozza meg az osztályt:
  - univerzális (00_2)
  - alkalmazás (01_2)
  - környezetfüggő (10_2)
  - privát (11_2)
- A 6. bit 0_2, amennyiben az érték típusa egyszerű, valamint 1_2, amennyiben az érték típusa összetett.
- Az 5-1. bit értéke a típus száma, amennyiben ez a szám 0..30 tartományba esik. Egyéb esetben az 5-1. bit értéke 11111_2 lesz.

Amennyiben a típus száma 31-nél nagyobb, vagy egyenlő, abban az esetben az azonosító további oktettekből fog állni, mely alakja:

- 8. bitje 1_2, egészen a legutolsó oktettig, melynek a 8. bitje 0_2.
- 7-1. bitje együttesen összefűzve, a típus számának a bináris értékével egyenlő.
- Az első ezen oktettek közül 7-1. bitje nem lehet 0000000_2.

Például:

Típus	Száma	Azonosító elem
INTEGER	2	00000010_2
SajatPrivat	300	11111111_2 10000001_2 11001000_2

Hossz elem

Két alakja lehet a hossz elemnek. A határozott, és a határozatlan alak. A küldő

- a határozott alakot használja, ha primitív az érték típusa.
- tetszőleges alakot használhatja, ha az érték típusa összetett és mindegyik érték azonnal elérhető.
- a határozatlan alakot használja, ha az érték típusa összetett, s nem mindegyik érhető el azonnal.

Határozott alak esetén a hossz elem egy vagy több oktettből áll, mely meghatározza, hogy a tartalom hány oktettből fog állni. Megadására két lehetőség van. A rövid és a hosszú alak.

A rövid alak csak akkor alkalmazható, hogy ha a tartalom kevesebb, mint 128 oktettből áll. Ebben az esetben a hossz elem egyetlen oktettből áll, melynek a 8. bitje 0_2, a 7-1. bitje pedig a tartalom oktettjeinek száma bináris alakban.

A hosszú alak esetén a hossz elem két vagy több oktettből áll. Az első oktett meghatározza, hogy utána hány oktett szerepel még. Ezen oktett:

- 8. bitje 1_2.
- 7-1. bitje a további oktettek száma bináris alakban.
- Nem lehet 11111111_2.

Az ezt következő oktettek összefűzve megadják bináris alakban, hogy a tartalom elem hány oktettből áll.

Például:

Tartalom elem hossza	Hossz elem	
	Rövid alak	Hosszú alak
25	00011001_2	10000001_2 00011001_2
300	*Nincs*	10000010_2 00000001_2 00101100_2

Határozatlan alak esetén a hossz elem egyetlen oktettből áll, mely értéke 10000000_2. Ebben az esetben nincs meghatározva, hogy a tartalom hány oktettből fog állni, ilyenkor van szükség a tartalom után a „tartalom-lezáró” elemre.

Tartalom elem

A tartalom elem nulla, egy vagy több oktettből áll, az értéktől függően.

„Tartalom-lezáró” elem

Akkor van jelen, hogy ha a hossz elem határozatlan alakú, azaz nem derül ki, hogy hány oktettből áll a tartalom. Ez az elem, két részelemből áll. Az azonosító elemből, mely 00000000_2, valamint a hossz elemből, mely 00000000_2. Tartalom elem nincs.

Például:

Azonosító	Hossz	Tartalom	Azonosító	Hossz	Tartalom
30_{16}	80_{16}	…	00_{16}	00_{16}	*Nincs*

Néhány univerzális típus, s a hozzá tartozó érték kódolása:

BOOLEAN

Primitív típus. A tartalom egy oktettből áll, mely értéke

- hamis esetén: 00000000_2,
- igaz esetén: tetszőleges nem nulla érték.

Például:

BOOLEAN	Hossz	Hamis
00000001_2	00000001_2	00000000_2

INTEGER

Primitív típus. A tartalom egy vagy több oktettből áll. Ha több oktettből áll, akkor az első oktett minden bitje, és a második oktett 8. bitje, nem lehet mind 1-es, avagy mind 0. Ez biztosítja, hogy az érték a lehető legkevesebb oktetten legyen lekódolva.

Például:

INTEGER	Hossz	200
00000010_2	00000010_2	00000000_2 11001000_2

NULL

Primitív típus. Nincs tartalom.

NULL	Hossz	Tartalom
00000101_2	00000000_2	Nincs

SEQUENCE

Összetett típus. A tartalom, a sequence minden egyes elemét tartalmazza kódolva, olyan sorrendben, ahogy az definiálva volt.

Például:

Típus: SEQUENCE {nev IA5String, nem ENUMERATED {ferfi (0) no (1)}}

Érték: {nev Zsolt, nem 0}

Azon	Hossz	Tartalom					
		Azon.	Hossz	Tartalom	Azon.	Hossz	Tartalom
30_{16}	$0A_{16}$	36_{16}	05_{16}	$5A736F6C74_{16}$	$0A_{16}$	01_{16}	00_{16}

OBJECT IDENTIFIER

Primitív típus. A tartalom oktettek a rögzített sorrendű részazonosítók kódolt állapota. Minden oktett 8. bitje 1_2, ha a részazonosító kódolásához további oktettek vannak igénybe véve, egyébként 0_2. Minden részazonosítót egy vagy több oktett határoz meg (kivéve az első kettőt). Figyelembe véve, hogy a gyökér csúcsból 3 érték van engedélyezve, s ezek közül is a 0. és az 1. értéknek legfeljebb 39 gyermeke van, ezért az első két részazonosítót összevonjuk, mégpedig az (X * 40) + Y képlettel, ahol X az objektumazonosító első részazonosítója, az Y a második részazonosítója.

Például: sysUpTime, melynek OID-ja {1 3 6 1 2 1 1 3 }

OBJECT IDENTIFIER	Hossz	Tartalom
06_{16}	07_{16}	$2B060102010103_{16}$

(Továbbá [18])

5. Hálózat menedzsment eszközök

A fejezetben három hálózat menedzsment eszközt mutatok be. Az általam kiválasztott alkalmazások mindegyike ingyenesen elérhető, használható. Az általam bemutatott eszközöket lokális környezetben tesztelem. A Windows alatt futtatható alkalmazást a saját konfigurációm alatt, míg a Linux-os környezetet igénylő alkalmazásokhoz pedig a Oracle VM VirualBox 3.2.10 alkalmazást vettem igénybe, melyre az Ubuntu 10.04-t telepítettem. Mivel az eszközök az SNMP segítségével kommunikálnak, így nagyobb kiterjedésű hálózati rendszerben is ugyanaz a működésük, mint lokális környezetben. Néhány program azonban igényel kliens oldali alkalmazást is.

Az egyes eszközök során leírom, a telepítés menetét, hogy milyen beállításokat igényel, majd bemutatom a lehetőségit.

Mivel ingyenes alkalmazásokról van szó, leginkább csak a hálózat monitorozására alkalmas szoftverekről esik szó. A komolyabb hálózat menedzsment eszközök jelentősebb összegekbe kerülnek, melyeket inkább csak a nagyvállalatok tudnak megfizetni.

Az általam választott programok a SpiceWorks, Munin, Nagios. A SpiceWorks Windows operációs rendszeren futó alkalmazás, míg a Munin és a Nagios alkalmazások Linuxos környezetre lettek tervezve.

5.1. SpiceWorks

Általánosan:

A SpiceWorks egy ingyenesen alkalmazható rendszer felügyeleti alkalmazás. A szoftver felismeri a hálózatra kapcsolt számítógépeket, hálózati eszközöket, valamint az eszközökön futó alkalmazásokat is. A hálózaton található eszközöket monitorozza, mint például az eszköz állapotát, a lemezek kapacitását, rajtuk futó alkalmazásokat, stb. Lehetőséget ad egyénileg definiált események monitorozására is. A hálózati eszközök monitorozása valamely böngésző segítségével követhetőek nyomon.

Első lépések:

A szoftver telepítése nem igényel nagy energia befektetést. Az alkalmazás telepítője letölthető a szoftver hivatalos oldaláról. Telepítés után, az első indításkor felméri a hálózatot, a hálózatra kapcsolt eszközöket, és azok kapcsolatát. Valamint az eszközökön engedélyezni kell a megfelelő portot.

Használat:

A navigáció az összes oldalon megtalálható baloldali menü segítségével történik.

Dashboard (vezérlő)

Áttekintést szolgáltat a hálózatról. Felhasználó által személyre szabható, hogy milyen, a SpiceWorks alkalmazás által nyújtott információkat szeretne látni. Az „Add Content" segítségével lehet új elemet felvenni. Ha az egyes elemekre ráállunk, akkor az elem címsorában megjelennek az elemen végrehajtható műveletek, melyek általában a szerkesztés, és a törlés.

Inventory (eszköztár)

A hálózaton található aktív és inaktív eszközök felsorolását tartalmazza kategóriákba csoportosítva. Új kategória vihető fel, avagy létező törölhető / szerkeszthető. Szerkesztés során megadhatjuk feltételek segítségével, hogy az adott kategóriába, mely hálózati eszközök kerüljenek bele. A SpiceWorks beépítve tartalmazza, hogy milyen feltételek közül válogathatunk. Ezeket paraméterezhetjük, s érhetjük el vele a kívánt lekérdezést. Tartalmaz egy úgy nevezett „Environment Summary", amely a teljes hálózatot jellemzi.

Valamely kategóriát kiválasztva, a megfelelő feltételeknek eleget tevő eszközök – kivéve a Szoftver kategóriát, melyben az eszközökre telepített szoftverek – jelennek meg, valamint a kategóriába tartozó eszközökről egy összesítő jellemzés. Kiválasztott kategóriába manuális módon is vehetünk fel új elemet, avagy törölhetünk egy létezőt.

Ezen belül a konkrét eszköz kiválasztása esetén, csak az adott eszközre jellemző adatok jelennek meg. Az összegzés mezőt fülek további alegységekre bontják, melyek jellemzik a kiválasztott elemet. Ezáltal megtekinthetjük az eszközről származó eseményeket időrendi

sorrendben, az eszközről származó hibákat, és veszélyforrásokat, az általános információit, a konfigurációját, az eszközre telepített programokat. Lehetőség van az egyes eszközökhöz jegyzeteket készíteni, valamint hozzájuk tartozó dokumentációt feltölteni. A „Complete Profile” segítségével az eszközről, minden információ megtekinthető.

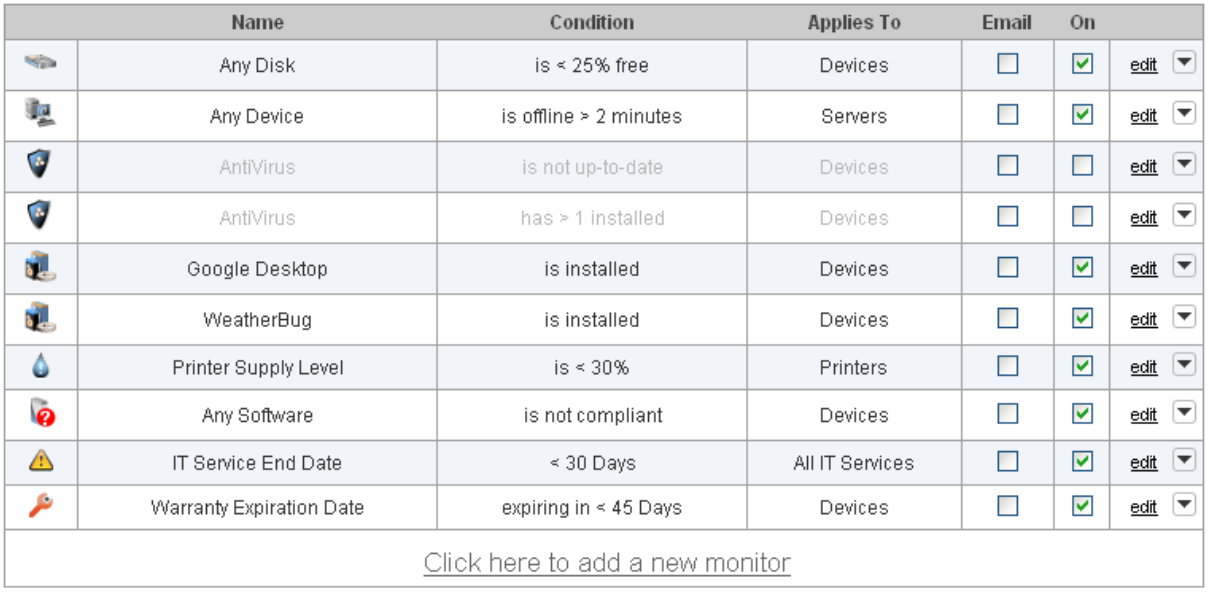
Hiba esetén segítséget nyújt a hibajavításhoz. Lehetőséget biztosít távoli asztal kapcsolat kialakítására, pingelésre, az eszköz és a szerver közötti útvonal meghatározására, valamint az eszközön futó alkalmazások elemzésére, esetleges befejeztetésére.

Monitoring (monitorozás)

Ezen az oldalon állítható be, hogy mely eseményeket szeretnénk monitorozni, valamint, hogy mely események hatására jelezzen a rendszer. Meg kell adni, hogy a monitorozott esemény során milyen feltétel esetén küldjön riasztást a felhasználónak. Az egyes események figyelését be, illetve ki lehet kapcsolni, ezáltal lehetőséget adva arra, hogy bizonyos eseményeket csak időközönként legyenek figyelve. Az alkalmazás lehetőséget ad beállított levelező rendszer esetén, hogy riasztás esetén a megfelelő e-mail címekre is elküldje a értesítőt, annak minden információjával együtt, ezáltal esetleges hiba, vagy veszélyforrás esetén a megfelelő személyzet akkor is értesül, ha nincs lehetősége épp a rendszert figyelni. Be kell állítani, hogy az egyes monitorozott események mely eszközökre, avagy eszközcsoportokra vonatkozzanak. Az 1. ábrán az alkalmazás által előre beállított események láthatóak, de természetesen van lehetőség ezek törlésére, szerkesztésére, valamint új események felvitelére, mely során megadjuk, hogy milyen típusú eseményt szeretnénk figyelni, milyen feltétel bekövetkezése esetén riasszon, mely eszközökre vonatkozzon a felügyelet.

Előre rögzített monitorozható események vannak, melyekből választhatunk:

- Anti-vírus program létezése, állapota,
- eszközök létezése, aktivitása,
- az eszközökben lévő merevlemezek kapacitása, állapota
- szoftverek, szolgáltatások, gyors javítások létezése, állapota
- hálózati azonosítók létezése,
- nyomtatók állapota,
- eszközök érvényessége, hitelessége,
- hibajegyek állapota.



	Name	Condition	Applies To	Email	On	
	Any Disk	is < 25% free	Devices	☐	☑	edit
	Any Device	is offline > 2 minutes	Servers	☐	☑	edit
	AntiVirus	is not up-to-date	Devices	☐	☐	edit
	AntiVirus	has > 1 installed	Devices	☐	☐	edit
	Google Desktop	is installed	Devices	☐	☑	edit
	WeatherBug	is installed	Devices	☐	☑	edit
	Printer Supply Level	is < 30%	Printers	☐	☑	edit
	Any Software	is not compliant	Devices	☐	☑	edit
	IT Service End Date	< 30 Days	All IT Services	☐	☑	edit
	Warranty Expiration Date	expiring in < 45 Days	Devices	☐	☑	edit
Click here to add a new monitor						

1. ábra

Network Map (hálózati térkép)

A hálózat szerkezetéről ad képet. Feltünteti az eszközöket, a közöttük levő kapcsolatokat. Az alkalmazás lehetőséget ad többféle megjelenítésre: sugaras, hierarchikus, faszerkezetű. Kiválasztható, azaz szűrhető, hogy a térképen mely típusú eszközök jelenjenek meg.

Lehetőséget ad mind a hálózatban levő eszközök, mind az eszközök közötti kapcsolatok manuális szerkesztésére. Ezáltal felvihetünk új eszközt, létező eszközt szerkeszthetünk, avagy törölhetünk.

Magán a térképen az eszköz kiválasztása esetén az eszközről, kapcsolat kiválasztása esetén csomópontok közötti kapcsolatról szolgáltat információt. Ezáltal segítségünkre van a hálózat optimalizálásában is, mert megmutatja, hogy mely kapcsolatok, avagy eszközök vannak esetlegesen túlterhelve, illetve leterhelve. Valamint segítségünkre van a hálózat esetleges bővítésének a megtervezésében is. Ezáltal megelőzhető esetleges hiba kialakulása.

Help Desk

A csapatmunkát, valamint a munkavégzés nyomon követését segíti elő. Úgynevezett hibajegyeket lehet kiírni az egyes feladatok elvégzése céljából. Új feladat létrehozása esetén megadjuk, hogy ki írja ki a feladatot, egy címet adunk a feladatnak, majd leírjuk, hogy mi is a

feladat. Emellett lehetőség van arra, hogy megadjuk, hogy mely hálózati eszközhöz tartozik a feladat, ki végezze el a feladatot, milyen határidővel készüljön el, milyen típusú a feladat. Lehetőség van szükség esetén fájlok csatolására is, például úgynevezett screenshot-ok csatolására. A feladat elvégzése során a hibajegyhez hozzá lehet írni, hogy mennyi időt foglalkoztunk a feladat elvégzésével. Előnye, hogy a felügyelet menetét nyomon lehet követni vele, hogy ki milyen faladatot végez éppen, milyen feladatokat végzett már el. Láthatjuk, hogy még mely feladatok várnak megoldásra. Ezek megtekintéséhez segítséget nyújt a hibajegyekhez tartozó szűrés: el nem fogadott, nyitott, lezárt, határidőn túli, nemrég módosított, saját hibajegyek.

Reporting (jelentés)

Lehetőség van jelentés-tervezet készítésére. Ha már létezik egy jelentés-tervezet, akkor azt lefuttatva, a tervezetben leírt adatoknak és feltételeknek megfelelően egy jelentés generálódik ki. Lényege, hogy minden egyes lefuttatás esetén, az aktuális hálózati állapotról fogunk kapni egy jelentést. Jól alkalmazható hivatalos jelentés készítésére, valamint saját célú, a hálózat állapotát megfigyelő lekérdezésekre.

Előnyök:

- Ingyenes.
- Könnyen használható, jól dokumentált, és széles, bárhonnan elérhető súgóval rendelkezik.
- Gyors motorral rendelkezik.
- Viszonylag több gép monitorozását megfelelően ellátja.
- Nagyon jól paraméterezhető, egyéni monitorozási igények kielégítésére is alkalmas.
- A riasztások segítségével az esetlegesen előforduló hibák időben megelőzhetőek.
- Rendellenes működés esetén segít a hiba felderítésében, a hiba távoli kijavításában.

5.2. Munin

Általánosan

A Munin egy ingyenes hálózat és rendszermonitorozó alkalmazás, mely felméréseket készít az eszközökről, s a kapott eredményeket grafikonok segítségével, webes felületen keresztül jelenít meg. A hangsúly a „plug and play” képességén van. Számos „plugin” letölthető hozzá, ezzel még részletesebbé téve a megfigyelést, elemzést. Segítségével nyomon követhetjük a teljesítményét a megfigyelt számítógépeknek, hálózatnak. Szerver-kliens architektúrára épül, melyben a szerver kapcsolódik a kliensekhez meghatározott időközönként, s lekéri tőlük az adatokat. Az adatokat RRD fájlokban tárolja, szükség esetén frissíti a grafikonokat.

Telepítés

Az alkalmazás telepítése az Ubuntu operációs rendszerre nem okoz nagy nehézséget. Az archívumában már megtalálható, a két részből áll a programcsomag. A megfigyelt számítógépekre elegendő csak a kliens oldali programot telepíteni. Mindkét esetben a telepítés egy egyszerű paranccsal végrehajtható. A webes felületen keresztül csak a beállított eszközökről származó adatoknak a grafikonjai láthatóak. Ahhoz, hogy beállítsuk a monitorozni kívánt eseményeket az egyes eszközökről, a program konfigurációs állományát kell módosítani.

Telepítés: *sudo apt-get install munin*

Telepítés után a konfigurációs állomány manipulálásával lehet beállítani, hogy mely események legyenek monitorozva, mely „plugin”-okat használja, esetleges hibák esetén kiket értesítsen. A program konfigurációs állománya a *munin.conf*.

Négy alapvető könyvtárat definiálnunk kell, mely a Munin működéséhez elengedhetetlenek:

- az rrd állományok tárolási helye: *dbdir* */var/lib/munin*
- a grafikonok, és a html állományok helye: *htmldir* */var/www/munin*
- naplózás helye: *logdir* */var/log/munin*
- aktuális állapot fájlok helye: *rundir* */var/run/munin*

Telepítés során a */var/www/munin* nem jön létre, azt nekünk kell létrehozni.

sudo mkdir –p /var/www/munin

Lehetőség a monitorozás eredményeinek a rendezett megjelenítéséhez előre definiált sablonok segítségével, melyek helyét a *tmpldir* segítségével adhatunk meg.

A monitorozott klienseket fel kell sorolni a konfiguráció során. Az egyes klienseket csoportokba, a csoportokat további csoportokban helyezhetjük el, ezáltal egy hierarchiát felépítve, mely megkönnyíti a kezelést és az áttekintést. A kliensek megadása több formátumban lehetséges.

[Halbi-teszt;Room-1;WS-1]

[Halbi-teszt;Room-1;WS-2]

[Halbi-teszt;Room-2;WS-1]

[Server.Halbi-teszt]

Mely eredménye:

Kliens megadása esetén az IP cím megadása kötelező. Minden klienshez hozzá van rendelve, hogy mely események legyenek monitorozva. Alapértelmezetten monitorozza az egyes eszközökön a merevlemez használatot, a Munin általi aktivitást, a hálózati aktivitást, a nyomtatási forgalmat, a folyamatokat, a rendszerjellemzőket. Az eszközről származó adatokból grafikont készít, melyet napi, illetve heti bontásban jelenít meg.

Beállítható, hogy az alkalmazás jelezzen, hogy ha egy érték a megfelelő tartományon kívülre esik, például kevés a lemezterület, vagy esetleg a hálózati kommunikáció során túl sok csomag veszik el. Ezen események egy részéhez van alapértelmezetten beállítva figyelmeztetési, valamint kritikus érték. A figyelt eseményekhez manuálisan is beállíthatjuk, hogy mikor figyelmeztessen az esetleges hibaforrásra, avagy hibára.

Engedélyezni kell a hibajelentési rendszert, valamint be kell állítani, hogy kinek, s mikor jelezzen.

contacts me

contact.me.command mail –s ”Hibabejelentés ${var:host}” halbi@freemail.hu

contact.me.always_send warning critical

Ez esetben a megadott e-mail címre küld egy levelet akkor, ha egy megfigyelt esemény figyelmeztetési, vagy kritikus értéket ér el. A levél tárgyába belekerült, hogy melyik kliensen történt a hiba. Az e-mailben további Munin-beli változóknak az értékét lehet megadni, hogy pontosabban megmutassa a szélsőséges értéket elérő esemény helyét. Ezáltal nagy segítséget nyújtva a hiba javításában.

A Munin fontosabb változói:

${var:group}	Csoport neve.
${var:host}	A kliens neve.
${var:graph_title}	A „plugin” neve.
${var:graph_category}	A „plugin” kategóriája
${var:label}	A mező neve.
${var:value}	A mező értéke
${var:extinfo}	Egyéb információ a mezőről.
${var:wrange}	A mező figyelmeztetési értékhatára.
${var:crange}	A mező kritikus értékhatára.
${var:wfields}	A figyelmeztetési értéket meghaladó mezők neve.
${var:cfields}	A kritikus értéket meghaladó mezők neve.
${var:ufields}	Ismeretlen értéket tartalmazó mezők neve.

Egyéb események figyelésének beállításához lehetőség van más által megírt „plugin” telepítésére, avagy lehetőség van saját „plugin” írására.

Előnyök:

- Ingyenes.
- Nem csak előre meghatározott események monitorozására van lehetőség, tetszés szerint bővíthető akár saját kóddal is.
- Gyors, jól paraméterezhető.
- Szélsőséges érték esetén jelez a megfelelő személyzetnek.
- Segít a hiba detektálásában.

5.3. Nagios

Linux-os monitorozó eszköz. Elég jól ismert a hasonló alkalmazások körében. Ingyenes alkalmazás. Jelenleg a 3.2.3-as verzió a stabil. Elég jól használható mind a sajátgép, mind távoli számítógép monitorozására. Lehetőséget ad, hogy gépenként különböző monitorozási lehetőséget állítsunk be, szükség szerint különböző küszöbértékekkel. A Nagios különböző beállításait konfigurációs fájlok módosításával érhetjük el.

Telepítés:

Az előzőleg tesztelt programokhoz képest, a Nagios telepítése kicsit körülményesebb, mivel nem elegendő maga a program telepítése, szükség van webkiszolgáló alkalmazásra (Apache), php értelmezőre, C fordítóra és a hozzá tartozó fejlesztői könyvtárakra, valamint a GD fejlesztői könyvtárakra. Amennyiben valamelyik hiányzik, telepíthető a következő parancsok segítségével:

Apache: *sudo apt-get install apache2*

PHP: *sudo apt-get install libapache2-mod-php5*

Fordító: *sudo apt-get install build-essential*

GD könyvtárak: *sudo apt-get install libgd2-xpm-dev*

Ezután célszerű root-ként folytatni a telepítést (*sudo -s*).

Szükséges felhasználók, s csoportok hozzáadása:

/usr/sbin/useradd -m -s /bin/bash nagios

passwd nagios

/usr/sbin/groupadd nagioscmd

/usr/sbin/usermod -a -G nagcmd nagios

/usr/sbin/usermod -a -G nagcmd www-data

A Nagios honlapjáról szükséges letölteni a Core-t, valamint a Plugin-okat valamely saját mappánkba, majd ki kell tömöríteni. Ezt követően lépjünk bele a Core-t tartalmazó mappába, majd az alábbi parancsokat adjuk ki.

./configure --with-command-group=nagcmd

make all

make install

make install-init

make install-config

make install-commandnode

A webes eléréshez installálni kell a Nagios-t az Apache conf.d könyvtárába, majd létrehozunk hozzá egy felhasználót, jelszóval együtt. Ahhoz, hogy érvénybe lépjen újra kell indítani az Apache-t.

make install-webconf

htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

/etc/init.d/apache2 reload

A Core magában nem elegendő a monitorozáshoz. Lépjünk bele a Plugin-okat tartalmazó mappába.

./configure --with-nagios- user=nagios --with-nagios-group=nagios

make

make install

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

Ezzel vége a telepítésnek, s el lehet indítani a Nagiost, majd valamely böngészőn keresztül a *http://localhost/nagios* lehet megtekinteni a monitorozás eredményeit.

/etc/init.d/nagios start

Felület:

General (Általános információk)

Külső hivatkozást tartalmaz, mely a Nagios hivatalos honlapjára mutat. Valamint megtalálható a Nagios rendszerhez tartozó teljes dokumentáció, mely nagy segítséget nyújt abban, ha elakadnánk. Előnye, hogy nincs szükség internetes hozzáférésre, mert a dokumentáció a saját gépen is van jelen.

Current Status (Aktuális állapot)

Tactical Overview (Áttekintés)

Egy áttekintő képet ad arról, hogy összesen hány eszközt monitorozunk, valamint, hogy ezen eszközök épp milyen státusban vannak (aktív, inaktív, elérhetetlen, függőben). Összegzi az ellenőrzések számát, s ezen ellenőrzések állapotát (kritikus, figyelmeztető, ismeretlen, OK, függőben). Végül tájékoztatást ad arról, hogy milyenek a jelenlegi monitorozási beállításaink.

Map (Térkép)

Többféle megtekintési lehetőséget ad arra, hogy megjelenítse a számítógépek, nyomtatók, routerek, switchek kapcsolatát, státuszát, a rajtuk futó ellenőrzések állapotát. Ha már nagy hálózattal dolgozunk, akkor nagy segítséget nyújt az áttekintésben az, ha csak az egyes csoportokba tartozó eszközöket jelenítjük meg. Előnyös, ha alapból úgy építjük fel a konfigurációs fájlokban a monitorozandó eszközöket, hogy megfelelően csoportba rendezzük őket.

Hosts (Eszközök)

A konfigurációs fájlokban megadott eszközöket tekinthetjük meg. Információt szolgáltat az egyes eszközök státuszáról, az eszköz utolsó ellenőrzésének idejéről, sikerességéről. Az egyes eszközök státusza lehet:

- Up: az eszköz elérhető, s aktív.
- Down: az eszköz elérhető, de inaktív.
- Unreachable: az eszköz nem érhető el, fizikai hibára utal.
- Pending: az eszköz még nem volt ellenőrizve, ellenőrzésre vár.

Services (ellenőrzések)

A konfigurációs fájlokban megadott eszközökhöz megadott ellenőrzéseket lehet nyomon követni. Megtekinthető az egyes eszközökre külön bontva, hogy mely eszközhöz milyen ellenőrzéseket végez a rendszer, s hogy ezen ellenőrzések milyen állapotban vannak éppen, továbbá az ellenőrzések milyen kimenetet adtak. Ez nagy segítséget ad abban, ha egy ellenőrzés nem OK állapottal tér vissza, akkor mi okozza.

Tetszőleges ellenőrzést kiválasztva, több információt szolgáltat a kiválasztott ellenőrzés állapotáról, valamint további beállítási lehetőségeket kínál. A konfigurációs fájlokban lehet megadni újabb ellenőrzést, avagy létezőt módosítani.

Host Groups (Eszközcsoportok)

Csoportokba szedve jeleníti meg az eszközöket, ahogy azokat a konfigurációs fájljainkban megadtuk. Eszközcsoportok megadása nem kötelező, de megkönnyíti az azonos jellegű eszközök áttekintését, ha azok valamilyen szisztéma szerint csoportokba vannak osztva.

Service Groups (Ellenőrzéscsoportok)

Csoportokban jeleníti meg az ellenőrzéseket, ahogy a konfigurációs fájljainkban megadtuk. A csoportok megadása nem kötelező, de egyszerűbben áttekinthetjük az azonos csoportba tartozó ellenőrzések állapotát.

Problems (Problémák)

Azon ellenőrzéseket, eszközöket tudjuk megtekinteni, amelyek a legutolsó ellenőrzés során nem OK állapottal tértek vissza. Az ilyen jellegű ellenőrzések kiváltó okait is megtekinthetjük, ezzel elősegítve a hiba megoldását.

Reports (Jelentések)

A Nagios lehetőséget ad jelentések készítésére is, mely segítségével kimutatást tudunk készíteni a kiválasztott időintervallumban történt változtatásokról, figyelmeztetésekről, kritikus hibákról, értesítésekről. Hasznos hivatalos jelentés készítésére.

Konfigurációs állományokban használható parancsok, melyek segítségével megadhatjuk a figyelni kívánt eszközöket, valamint beállíthatjuk a monitorozni kívánt eseményeket. A megadás során minden sor egy-egy beállítási paramétert jelent. A paraméter nevével kezdődik, aztán a többi rész a paraméter értéke.

5.3.1. Eszköz megadása

Egy eszköz megadásához célszerű külön konfigurációs fájlt létrehozni, ezáltal elválasztani, már előzőleg definiált eszköztől. Így jobban elkülönülnek az eszközök, könnyebb őket menedzselni.

A *nagios.cfg*-ben ezt az új fájlt hozzá kell rendelni a Nagios-hoz.

A fájlban definiáljuk az eszközt a *define host {}* segítségével.

Kötelező paraméterek:

- host_name: azonosítja az eszközt. Konfigurációs állományokban ezzel a névvel tudjuk hivatkozni az eszközt.
- alias: az eszköz elnevezése.
- address: az eszköz IP címe.
- max_check_attempts: megadjuk, hogy hányszor próbálkozzon a Nagios (amennyiben nem OK a visszatérési érték), mielőtt hibaüzenetet generálna.
- check_period: olyan időszakot kell megadni, amikor az eszköz állapotát szeretnénk figyelni. Ha nincs olyan megírt időintervallum, ami megfelelne, tudunk sajátot is írni, (részletesebben később)
- contact_groups: probléma esetén az értesítendő csoportok vesszővel elváasztva.
- notification_interval: azt az időmennyiséget kell megadni, amely időközönként az eszköz változatlanságáról (állandóan inaktív, avagy elérhetetlen) értesítést küldjön a megadott csoportnak. Alapértelmezettként ezt percben kell megadni.
- notification_period: azt az időszakot kell megadni, értesítés küldése megengedett. Amennyiben olyan időszakot szeretnénk megadni, ami még nem létezik, akkor felvehetünk újat (5.3.4.).
- notification_options: akkor küld értesítést a személyzetnek, ha
  - d: az eszköz inaktív.
  - u: az eszköz nem érhető el.
  - r: az eszköz visszaáll aktív állapotba.
  - f: az eszköz gyakran vált állapotot megadott idő alatt.

Egyéb lehetséges paraméterek:

- parents: az eszköz valamilyen szinten, közvetve kapcsolódik a szerverhez. Ebben a kapcsolódási láncban az eszközhöz legközelebbi eszközt lehet megnevezni. Több

ilyen esetén, minden láncban, a hozzá legközelebbi eszközt lehet megnevezni. Ez segít a számítógépes hálózat térkép megjelenítésében.

- check_command: egy parancs megadása lehetséges. Célszerű egy egyszerű PING parancsot megadni az aktivitás ellenőrzésére. Új parancsot is meg lehet esetlegesen adni, ha szükség van rá. (Új parancs megadása)

Windows operációs rendszert futtató számítógép monitorozása esetén, szükséges a gépre telepíteni a NSClient++ telepítése, mert a Nagios ezen a programon keresztül szerzi meg a szükséges információkat.

5.3.2. Eszközcsoport megadása

Valamely létező eszközleíró konfigurációs fájlba célszerű írni. Megadása a *define hostgroup {}* parancs segítségével történik.
Kötelező paraméterei:

- hostgroup_name: az eszközcsoport azonosítója. A konfigurációs fájlokban ezen az azonosító alapján tudjuk hivatkozni az eszközcsoportot.
- alias: az eszközcsoport megnevezése.
- members: az eszközcsoportba tartozó eszközök azonosítói vesszővel elválasztva.

5.3.3. Parancs megadása

Az parancs megadásához célszerű a *commands.cfg* fájlt módosítani, így egy helyen marad minden parancs.
Parancs megadása a *define command {}* parancs segítségével történik.
Kötelező paraméterei:

- command_name: a parancs azonosítója. A konfigurációs fájlokban ezen az azonosító alapján tudjuk hivatkozni ezt a parancsot.
- command_line: maga a parancs, amit le szeretnénk futtatni. A $-k közé írt szavak a paraméterek, avagy az előre rögzített változók.

5.3.4. Időszak létrehozása

Az időszak beállításához célszerű a *timeperiods.cfg* fájlt módosítani, így egy helyen lesz minden időszak.
Időszak beállítása a *define timeperiod {}* parancs segítségével történik.
Kötelező paraméterei:

- timeperiod_name: az időszak azonosítója. A konfigurációs fájlokban ezen az azonosító alapján tudjuk hivatkozni a beállított időszakot.
- alias: az időszak megnevezése

Amennyiben nem adunk meg mást, csak ezt a két paramétert, akkor az időszak hoz nem tartozik idő, s ami ezt hivatkozza, az sose lesz figyelve. Tervezés esetén viszont jól használható, mikor még nem tudjuk, hogy mely időszakra vonatkozzon az eszközmegfigyelés, értesítésküldés, az ellenőrzés, s majd utólag adunk neki értéket.
Egyéb lehetséges paraméterei:

- hétköznapok angolul (sunday, monday, …, saturday): megadhatjuk, hogy az adott nap mettől meddig tartson az időszak. Több időintervallum megadása esetén vesszővel választjuk el azokat.
- hónap, nap: beállítható vele, hogy adott hónap valamelyik napján hogy legyen az időszak. Például: *May 1 00:00-00:00*
- use: segítségével származtatni tudunk már meglévő időintervallumból.
- Beállítható nem fix dátum is, ami évente változik.
  Például (május első vasárnapja): *sunday 1 may 00:00-00:00*

5.3.5 Kapcsolattartók, s csoportok beállítása

Beállításukhoz célszerű a *contacts.cfg* fájlt módosítani. Új konfigurációs fájl használata esetén a *nagios.cfg*-ben az új fájlt hozzá kell rendelni a Nagios-hoz.
Új kapcsolattartó létrehozása a *define contact {}* parancs segítségével történik.
Kötelező paraméterei:

- contact_name: a kapcsolattartó azonosítója. A konfigurációs fájlokban ezen az azonosító alapján tudjuk majd hivatkozni.
- alias: a kapcsolattartó neve.

- host_notification_period: meg kell adni egy időszakot, amikor a megadott személy értesíthető, ha egy eszközben valamilyen hiba történik. Amennyiben olyan időszakot szeretnénk megadni, ami még nem létezik, akkor felvehetünk újat (időszak felvevős azonosító).
- service_notification_period: meg kell adni egy olyan időszakot, amikor a megadott személy értesíthető, ha egy ellenőrzés során valamilyen kritikus esemény történik. Amennyiben olyan időszakot szeretnénk megadni, ami még nem létezik, akkor felvehetünk újat (időszak felvevős azonosító).
- host_notification_options: egy, vagy több értékei is lehet. Több érték esetén vesszővel választjuk el őket.

  Értesítést küld a személynek, ha a megfigyelt eszköz:
  - d: inaktív állapotba kerül.
  - u: nem elérhető.
  - r: visszaáll aktív állapotba.
  - f: gyakran vált állapotot egy időszakon belül.
  - n: Nem küld értesítést.
- service_notification_options: egy vagy több értéke lehet. Több érték esetén vesszővel választjuk el őket.

  Értesítést küld a személynek, ha egy ellenőrzés:
  - w: figyelmeztető állapotba kerül.
  - u: ismeretlen állapotba kerül.
  - c: kritikus állapotba kerül.
  - r: ha visszaáll megfelelő állapotba.
  - f: ha túl sokszor változtat állapotot egy időszakon belül.
  - n: Nem küld értesítést.
- host_notification_commands: a megadott parancs segítségével értesíti a személyt, ha szükséges.
- service_notification_commands: a megadott parancs segítségével értesíti a személyt, ha szükséges.

Egyéb lehetséges paraméterei:

- email: a személy e-mail címe.
- pager: a személy telefonszáma.

- address*x*: a személy további elérhetőségei.

Ezek megadására azért lehet szükség, hogy a személyt, akár több lehetséges úton is informálni lehessen egy esetleges hibának a kialakulásáról.

Csoport kialakítása a *define contactgroup {}* parancs segítségével történik.
Kötelező paraméterei:

- contactgroup_name: a csoport azonosítója. A konfigurációs fájlokban ezen az azonosító alapján tudjuk majd hivatkozni.
- alias: a csoport neve.
- members: a csoport tagjai vesszővel elválasztva.

A csoportok előnye, hogy az azonos eseményekre reagáló személyek esetén csak a csoportra kell hivatkozni. Ha egy új személyt fel akarunk venni egy adott esemény figyelésére, akkor egyszerűen csak a csoportba kell felvennünk.

5.3.6. Ellenőrzések létrehozása

Az ellenőrzés beállítását célszerű abban a konfigurációs fájlban elhelyezni, amely az eszközre vonatkozik a monitorozás.
Új ellenőrzés beállításához a *define service {}* parancsot kell használni.
Kötelező paraméterei:

- host_name: az eszköz neve, amin futtatni szeretnénk ezt az ellenőrzést
- service_description: az ellenőrzés leírása.
- check_command: meg kell adni a parancsot, ami lefusson. Új parancsot is lehet írni, ha szükséges (új parancs írása rész).
- max_check_attempts: be kell állítani, hogy maximum hányszor próbálkozzon a rendszer az adott ellenőrzéssel, mielőtt az nem OK állapotba kerülne.
- normal_check_interval: azt az időegységet kell megadni, amely időközönként futtassa a rendszer ezt az ellenőrzést, akkor ha az ellenőrzés OK állapotban van, avagy nincs OK állapotban, de már a *max_check_attempts* értékét meghaladta a próbálkozások száma. Alapértelmezettként percben kell megadni.
- retry_check_interval: azt az időegységet kell megadni, amely időközönként futtassa a rendszer ezt az ellenőrzést, ha az ellenőrzés nincs OK állapotban, s még a

próbálkozások száma nem érte el a *max_check_attempts* értékét. Alapértelmezettként percben kell megadni.

- check_period: meg kell adni az időszakot, amikor ez lefusson. Megadhatunk új időintervallumot is, ha szükséges (új időintervallum rész)
- notification_interval: azt az időegységet kell megadni, amely időközönként értesíteni kell a személyzetet ha az ellenőrzés változatlanul nem OK állapotban van.
- notification_period: meg kell adni azt az időszakot, amikor szükség esetén értesíthető a megfelelő személyzet.
- notification_options: akkor küld értesítést a személyzetnek, ha
  - w: figyelmeztető állapotba kerül az ellenőrzés
  - u: ismeretlen állapotba kerül az ellenőrzés
  - c: kritikus állapotba kerül az ellenőrzés
  - r: visszaáll OK állapotba az ellenőrzés
  - f: az ellenőrzés gyakran vált állapotot megadott idő alatt.
- contact_groups: fel kell sorolni az értesítendő csoportokat

Egyéb lehetséges paraméterei:

- notification_enabled: engedélyezhető / letiltható, hogy értesítést küldjön a rendszer.

5.4. Kipróbált alkalmazások néhány szóban

A Windows-os szoftver fixen megadja, hogy mit lehet monitorozni, esetleg csak a küszöbértékeket állíthatjuk be, de nem támogatja, hogy a felhasználó bővítse saját kódjával. Ettől eltekintve viszont könnyű feltelepíteni, könnyű a használata, automatikusan felismerte a hálózatra kötött eszközöket.

A Linux-os szoftverek feltelepítése komplikáltabb – számomra, aki inkább a Windows-os környezetben érzi otthon magát –, általában több fájlon keresztül lehet beállítani a paramétereket. Nem ismeri fel automatikusan a hálózatra kapcsolt eszközöket, manuálisan kell beállítani a hálózati eszközöket, s kapcsolataikat. Így viszont nem fordulhat elő olyan, hogy a rendszer rosszul ismerné fel a kapcsolatokat, hiszen a felhasználó állít be mindent. A monitorozást webes felületen keresztül lehet nyomon követni. Diagramok segítségével könnyebben értelmezhetőek az események. Nagy előnye az, hogy a felhasználó, ha ért hozzá, saját kódjával, bővítheti a már meglévő rendszert.

6. Összefoglalás

A diplomamunkám első felében bemutattam a hálózati monitorozás kialakulásához vezető alapokat. Részletesebben kifejtettem, hogyan fejlődött az Simple Network Management Protocol (SNMP). Jellemeztem a hálózat menedzselését szolgáló adatok tárolására szolgáló adatbázist.

Diplomamunkám második felében bemutattam néhány monitorozásra szolgáló programot. Választottam mind Windows, mind Linux operációs rendszeren futó alkalmazásokat. Ezen programok mindegyike, nem csak a sajátgép monitorozására szolgál, hanem hálózatban lévő számítógépek, nyomtatók, routerek, switchek monitorozására is alkalmasak. Ebből kifolyólag a szerepük a mai világban jelentős, mivel egy nagyobb hálózat egyszerűen megfigyelhető egy központi helyről. Én csak ingyenesen használható alkalmazásokat mutattam be, melyek főleg monitorozásra alkalmasak. Viszont a fizetős alkalmazások fejlettebbek az általam választottaknál. Általában tartalmaznak döntéssegítő szolgáltatásokat, mint például, hogy hálózat bővítése esetén milyen mértékben változik a hálózat terheltsége, ezáltal hol érdemes bővíteni. Tartalmazhatnak automatikus javítómechanizmusokat. Természetesen ezen szoftverek árát szinte csak a nagyvállalatok képesek megfizetni. Egy kisebb vállalkozás, vagy akár középvállalkozás számára, amíg csak néhány száz géppel dolgozik elegendő az általam is említett alkalmazások használata.

Ezen programok jelentősége egyre inkább növekedni fog, hiszen a mai világban a számítógépes hálózatok mérete is növekedőben van. Mi se bizonyítja jobban, hogy napjainkban folyik az IPv4-ről történő átállás az IPv6-ra. Valamint egyre jellemzőbb az is, hogy nem csak a számítógépek kapnak saját IP címet, hanem látható, hogy már a nyomtatók is, valamint egyes elektronikai berendezések is. Ebből kifolyólag, hogy átlássuk az egyre növekvő hálózati rendszert, elengedhetetlenné fog válni olyan szoftverek használata, ami nyomon tudja követni, hogy az egyes eszközök épp milyen állapotba vannak, milyen várható hibák alakulhatnak ki, avagy kialakult hiba esetén hol alakult ki a hiba, s ezt hogyan lehet a lehető leghamarabb orvosolni.

A diplomamunkám során segítségemre volt az internet, az RFC dokumentumok, valamint a programok első lépéseihez, mind a szoftver saját dokumentációja, mind az erre létrejövő fórumok sokasága, valamint a témakörökben megjelent szakirodalom.

7. Irodalomjegyzék

Könyvek

[1] Alexander Clemm: Network Management Fundamentals
Cisco Press, 2006

[2] J. Richard Burke: Network Management: Concepts and Practice, A Hands-On Approach
Prentice Hall, 2003

[3] Andrew S. Tanenbaum: Számítógép hálózatok
Panem, 2004

[4] Stephen B. Morris: Network management, MIBs and MPLS: Principles, Design and Implementation
Prentice Hall, 2003

[5] Sidnie M. Feit: SNMP: A Guide to Network Management
McGraw-Hill Companies, 1993

RFC dokumentumok

[6] U. Blumenthal, B. Wijnen: User-Based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3), 1999, RFC2574

[7] B. Wijnen, R. Presuhn, K. McCloghrie: View-based Access Control Model (VACM) for then Simple Network Protocol (SNMP), 1999, RFC2575

[8] J. Case, D. Harrington, R. Presuhn, B. Wijnen: Message Processing and Dispatching for the Simple Network ManageMent Protocol (SNMP), 2002, RFC3412

[9] M. Rose, K. McCloghrie: Structure and Identification of Management Information for TCP/IP-based Internets, 1990, RFC1155

[10] K. McCloghrie, M. Rose: Management Information Base for Network Management of TCP/IP-based internets: MIB-II, 1991, RFC1158

[11] U. Warrier, L.Besaw: The Common Management Information Services and Protocol over TCP/IP (CMOT), 1989, RFC 1095

[12] J. Case, M. Fedor, M. Schoffstall, J.Davin: A Simple Network Management Protocol (SNMP), 1990, RFC 1098

Megtekintett oldalak

[13] 2010. március
http://kisspepe.atw.hu/Letoltes/Rendszeradminisztratori%20ismeretek/Rendszeradminisztratori%20ismeretek.doc

[14] 2010. március
http://www.lincoln.edu/math/rmyrick/ComputerNetworks/InetReference/108.htm

[15] 2010. április
http://oreilly.com/perl/excerpts/system-admin-with-perl/twenty-minute-snmp-tutorial.html

[16] 2010. október
http://www.obj-sys.com/asn1tutorial/asn1only.html

[17] 2010. október
http://www.hiradastechnika.hu/data/upload/file/2004/2004_08/HT0408-5.pdf

[18] 2010. október
http://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf

[19] 2010. november
http://www.linuxvilag.hu/content/files/cikk/32/cikk_32_45_47.pdf

[20] 2010. november
http://www.linuxvilag.hu/content/files/cikk/71/cikk_71_57_61.pdf

Hasznos linkek

Angol wikipedia: http://en.wikipedia.org

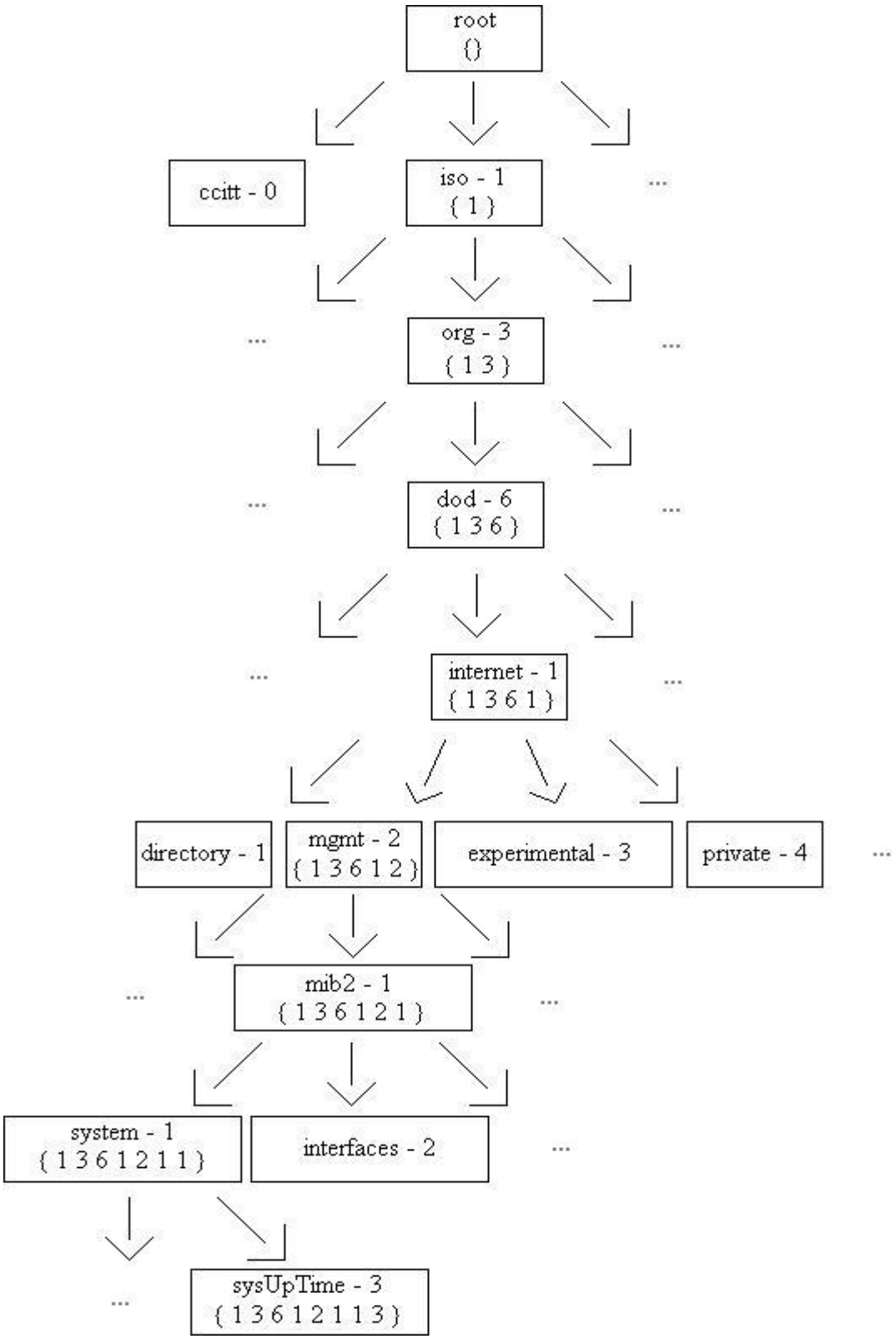
Magyar wikipedia: http://hu.wikipedia.org

SolidWorks hivatalos honlapja: www.spiceworks.com

Munin hivatalos honlapja: http://munin-monitoring.org

Nagios hivatalos honlapja: www.nagios.org

8. Függelék



Kép 1: A sysUpTime helye a hierarchiában

0	BER speciális érték
1	BOOLEAN
2	INTEGER
3	BIT STRING
4	OCTET STRING
5	NULL
6	OBJECT IDENTIFIER
7	ObjectDescriptor
8	INSTANCE OF, EXTERNAL
9	REAL
10	ENUMERATED
11	EMBEDDED PDV
12	UTF8String
13	RELATIVE-OID
16	SEQUENCE, SEQUENCE OF
17	SET, SET OF
18	NumericString
19	PrintableString
20	TeletexString, T61String
21	VideotexString
22	IA5String
23	UTCTime
24	GeneralizedTime
25	GraphicString
26	VisibleString, ISO646String
27	GeneralString
28	UniversalString
29	CHARACTER STRING
30	BMPString

Táblázat 1: Univerzális típusok

(forrás: http://www.obj-sys.com/asn1tutorial/node124.html)

9. Köszönetnyilvánítás

Mindenképp szeretném megköszönni konzulensemnek, Dr. Almási Béla tanár úrnak, hogy a félévek folyamán útmutatást adott nekem, valamint a kérdéseimre válaszolva elősegítette a diplomamunkám elkészítését.

Valamint szeretnék köszönetet mondani mindenkinek az egyéb támogatásokért.