1. Overview

Miért tekinthetjük a hálózatmenedzsmentet egy elosztott rendszernek? Érveljen!

A hálózatmenedzsment alkalmazások definíció szerint elosztottak, ugyanis vannak menedzselt és menedzselő rendszerek. Emellett a skálázhatóság, megbízhatóság és magas rendelkezésre állás eléréséért maga a menedzselő rendszer is elosztott kell, hogy legyen. Például a karbantartás csak akkor valósítható meg egy rendszeren, ha a feladatait át tudják venni tőle. Elosztott rendszerre a nap járását követő menedzsment-szolgáltatások is jó példa, ugyanis a munkaidőben mindig érdemes a helyileg közel lévő rendszereket használni.

Hogyan növelheti egy szolgáltató hálózatmenedzsmenttel a bevételeit?

Három különböző esetben növelheti a bevételeket a hálózatmenedzsment**. Az első** esetben maga a kiadások csökkentése nyit meg egy teljesen új piacot. Ez akkor lehetséges, ha a kezdő költségek a menedzsment nélkül túl nagyok lennének ahhoz, hogy az a felhasználónak megérje, azonban alacsonyabb költséggel új szolgáltatásokat lehet nyújtani. **A második** eset, amikor a menedzsment a szolgáltatás megrendelése és üzembe helyezése közti időtartamot rövidíti le, ezzel gyorsabban generál bevételeket. Bizonyos esetekben a gyors szolgáltatás hiánya az ügyfél teljes elvesztésével járhat. **A harmadik** esetben a szolgáltató menedzsment funkciókkal bővíti a szolgáltatásait, melyek így több ügyfelet vonzzanak.

Hálózatmenedzsment feladatok milyen alkalmazási működéssel valósíthatóak meg? Hogyan és milyen hálózatmenedzsment célokat szolgálnak?

Tranzakció-alapú (transaction based) működés során a hálózat kéréseket küld ki, válaszokat dolgoz fel, valamint folyamatokat menedzsel. Alapvetően a szolgáltatás felépítésénél (Provisioning) használt módszer.

Megszakítás-alapú (interrupt) működés kvázi valós idejű karakterisztikával a monitoring során használt.

Riasztások (alarmok) segítségével tudja a hálózat pontos állapotát megmutatni a menedzsernek, és megfelelő esetekben figyelmeztetni a hibákra.

Adatmennyiség-alapú (number cruncing) működés a statisztikai elemzések szempontjából fontos, hogy megfelelően tudjuk elosztani az erőforrásokat (bottleneck), hogy meghatározzuk az Accounting szempontjából a kapcsolódó bejegyzéseket, hogy tervezhessünk a jövőbeni bővítésekről.

Hogyan segítheti a hálózatmenedzsment a vállalati megtakarításokat?

With network testing and troubleshooting tools.
With systems that facilitate turn-up of services and automate provisioning.
With performance-reporting tools and bottleneck analysis.

A megtakarításokat a működési költségek csökkentésével tudja a hálózatmenedzsment növelni. Ez jelenthet hálózati tesztelő és hibaelhárító eszközöket,melyek megtakarítást jelentenek a rendszergazdák idejéből. Automatikus üzembe helyezést, mely az egyébként sok egymást követő redundáns információkat igénylő lépést helyettesíti, vagy teljesítmény-analizátorokat, melyek segítenek a pontosabb erőforrás-elosztásban.

Mi az a "swivel-chair syndrome" (görgős szék szindróma)? Jó ez vagy rossz? Miért?

A görgős szék probléma arra utal, amikor egy rendszergazda görgős székével közlekedik a több terminál között. A probléma lényege az integráció hiánya, melynek következtében a felhasználók kénytelenek rendszeresen, sok együtt nem működő alkalmazás között váltogatni. Ez rossz, és elkerülendő, mivel a felhasználóknak rendkívül sok különböző alkalmazás használatát kell

megtanítani, ami költséges és felesleges. Helyette az integrált menedzsment lenne a megoldás, melynek lényege, hogy közös megjelenítő felületen, közös adatbázisokban dolgozna a több különböző menedzsment-alkalmazás.

Hasonlítsa össze üzleti hálózatmenedzsment lehetőségek szempontjából a berendezés gyártót és a független hálózatmenedzsment alkalmazások készítőjét!

A **berendezés gyártói** általában minimális képességű menedzsment alkalmazásokat szeretnének nyújtani, hiszen nekik az elsődleges céljuk a „vas" eladása. Azonban az utóbbi évek trendje az, hogy a különböző gyártók eszközei között egyre inkább csak a nyújtott szolgáltatás minősége a különbség, és ezért egyre nagyobb figyelmet szentelnek a hálózatmenedzsment lehetőségek bővítésére.

A **független hálózatmenedzsment alkalmazások készítői** alapvetően azt az űrt próbálják betölteni, amely az eszközök gyártói és a hálózatüzemeltetők között van. Ez azt jelenti, hogy a gyártó által nyújtott minimális menedzsment felület helyett egy könnyebben kezelhető, több funkcionalitást is tartalmazó interfészt nyújtanak. További előnyük, hogy a független (harmadik fél által gyártott) alkalmazások gyártó függetlenek és több gyártó több eszközének is hasonló interfészt nyújt az üzemeltetők felé.

Milyen szereplők lehetnek a hálózatmenedzsment területen? Ismertesse ezek közül a hálózatmenedzsment felhasználóit!

Az összes felhasználó csoportra igaz: fontos számukra a sebesség, a QoS, a hibakezelés, a kihasználtság.

A **felhasználók első csoportja a távközlési vagy hálózati szolgáltatók.** Az ő bevételi forrásuk az általuk nyújtott szolgáltatás, és a hálózatmenedzsment létfontosságú számukra, mivel a költségeik túlnyomó része a működési költség. Ezen túl a bevételeiket is növelhetik ha gyorsan tudnak szolgáltatást üzembe helyezni (provisioning), esetleg új szolgáltatásokat nyújtani a menedzsment által csökkentett induló költségekkel (truck roll), vagy akár menedzsment szolgáltatásokkal bővített szolgáltatást nyújtani.

A **második csoport a céges IT részleg**. A fő különbségek a szolgáltatókhoz képest: az IT a cég számára kizárólag költséget jelent, nem céljuk a bevétel generálása; csak egy ügyfelük van, és az ügyfelüknek nincs igazán lehetőségük a leváltásukra (csak az outsourcing); mivel a cég számára nem alaptevékenység az IT, nem fontos a versenyképességük, ezért sokszor valóban outsourcingolják a szolgáltatásaik nagy részét; alacsony (vagy néhol semmilyen) törvényi szabályozás vonatkozik rájuk.

A **harmadik csoport a végfelhasználók**, akik itt a hálózatmenedzserek. Ide tartoznak a hálózati rendszergazdák, technikusok, help-deskesek, és a hálózattervezők is. Szerepük és működésük erősen a cég jellegétől függ.

Mit csinál a "hálózatmenedzsment"?

A hálózatmenedzsment azokat a cselekvéseket, eljárásokat, folyamatokat és eszközöket jelenti, amelyek egy hálózatban annak működtetésével (operation), adminisztrációjával (administration), karbantartásával (maintenance), és szolgáltatásával (provisioning) kapcsolatos. Ennek modellje az **OAMP.**

Mit takar az OAMP rövidítés? Röviden ismertesse az egyes betűkhöz rendelt feladatokat!

Az OAMP az Operation, Administration, Maintenance és Provisioning kifejezéseket takarja, melyek a hálózatmenedzsment alap funkciói.

O: a hálózat (és az általa nyújtott szolgáltatások) fennakadás nélküli működtetése. Beletartozik a hálózat folyamatos ellenőrzése (monitoring), hogy a hibákat a lehető leghamarabb észrevegyék,

lehetőleg mielőtt egy felhasználónak problémája származna belőle.
A: az erőforrások és azok kiosztásának számon tartása. Minden „házimunka” ide tartozik, mely ahhoz kell, hogy irányításunk alatt tartsuk a rendszert.
M: javítások és fejlesztések. Ide tartoznak a módosító és megelőző intézkedések is, a menedzselt hálózat jobbá tétele érdekében.
P: az erőforrások olyan átrendezése, hogy egy adott szolgáltatást lehetővé tegyenek .

Milyen szereplők lehetnek a hálózatmenedzsment területen? Ismertesse ezek közül a hálózatmenedzsment előállítóit!

A **berendezés gyártói** általában minimális képességű menedzsment alkalmazásokat szeretnének nyújtani, hiszen nekik az elsődleges céljuk a „vas” eladása. Azonban az utóbbi évek trendje az, hogy a különböző gyártók eszközei között egyre inkább csak a nyújtott szolgáltatás minősége a különbség, és ezért egyre nagyobb figyelmet szentelnek a hálózatmenedzsment lehetőségek bővítésére.

A **független hálózatmenedzsment alkalmazások készítői** alapvetően azt az űrt próbálják betölteni, amely az eszközök gyártói és a hálózatüzemeltetők között van. Ez azt jelenti, hogy a gyártó által nyújtott alkalmazások általában nem alkalmasak több gyártó berendezéseinek menedzselésére, vagy ha igen, akkor is csak korlátozottan. A független (harmadik fél által gyártott) alkalmazások azonban megpróbálnak több gyártó több eszközének közös interfészt nyújtani az üzemeltetők felé.

A **rendszer integrátorok** a különböző eszközök és alkalmazások egyéni testre szabását végzik, mivel általában az alkalmazások nem működnek gond nélkül együtt, és ezért külön menedzsment szolgáltatást tudnak nyújtani.

Milyen módon járulhat hozzá az üzleti sikerhez a hálózatmenedzsment? Hogyan?

Cost, Quality, Revenue

- magas rendelkezésre állás
- redundáns hardware, kommunikációs utak
- network management
- végpont-végpont szolgáltatás
- monitorozás
- korreláció analízis a riasztásokra
- ha nem tudják garantálni a szolgáltatás minőségét, ellenszolgáltatás.

3. Basics

Milyen hálózatot lehet használni a hálózatmenedzsmentre? Mik az egyes megközelítések előnyei és hátrányai?

Lehet használni a termelésre használt **hálózattal közös, megosztott hálózatot**, vagy **dedikált menedzsment hálózatot**. Előbbi olcsóbb megoldás, és sokszor az egyetlen, mivel nincs lehetőség egy másik önálló hálózat kiépítésére. Utóbbi előnyei a megbízhatóság, a biztonság, az interferencia elkerülése és a hálózattervezés egyszerűsítése.

Miért nem kezelhető a MIB egy adatbázisként?

A MIB számára az adatbázis-kezelők túl robusztus, túl költséges megoldást jelentenének, nagy overheaddel az eszközök számára. Mivel a MIB-ek információi általában hierarchikus elrendezésűek, valamint elosztottak, nehezen tudná egy adatbázis megragadni ezeket a különlegességeket. Fontos még, hogy míg egy adatbázisban sok hasonló struktúrájú adat van kevés táblában, addig egy MIB-ben sok fajta bejegyzésből van kevés. Lényegében a MIB nem lehet valódi adatbázis, hiszen folyamatosan változik a természeténél fogva.

Oldja fel az NOC akronimát. Mi történik az NOC-ban?

Network Operations Center: Az a hely, ahonnan egy nagy hálózatot menedzselnek, minden menedzsment funkciót innen indítanak. Kiterjedt hálózatokban nem elég egy NOC, hanem akár több, regionális NOC-ra is szükség van (akár a nap járása szerint).
A menedzsmenttel kapcsolatos tevékenységeket itt végzik:
- monitorozás,
- provisioning,
- hálózati eszközök konfigurációjának biztonsági mentése,
- felhasználók adatainak gyűjtése számlázás céljából,
- kommunikációs berendezéseket is tarthatnak itt.

Oldja fel a MIB akronimát. Fejtse ki mi az a MIB!

Management Information Base: A MIB egy virutális adattár, mely menedzsment nézetet tartalmaz az adott eszközhöz. Nem egy valódi adatbázis, hanem csak egy nézet, bejegyzései mögött valódi konfigurációs lehetőségek vannak.

Hasonlítsa össze a manager/agent és a client/server paradigmát hasonlóságok és különbségek alapján!

Ebben a kontextusban a manager a client és az agent a server, olyan értelemben, hogy a manager kér információkat az agent-től, és az kiszolgálja azt.
A különbség az, hogy míg alapértelmezésben a server kevés, client sok a felállás, itt a manager a kevés és az agent sok.

A ágens kifejezése milyen két kontextusban használatos? Hogyan kapcsolódnak ezek egymáshoz?

Az ágens kifejezés jelentése az egyik kontextusban a menedzser-ágens kapcsolatban a menedzselt eszközök szerepe.
A másik jelentése egy szoftver, amelyet a hálózati elemek nyújtanak, és megvalósítja a menedzsment interfészt.

Ha egy hálózatnak 99.999% (5 kilences) rendelkezésre állást kell biztosítania, akkor szükséges-e a menedzsment rendszernek is 99.999%-os rendelkezésre állást biztosítani? Indokolja válaszát!

A hálózat működéséhez nem szükséges a management. Az 5 kilences rendelkezésre állást nem szabad, hogy befolyásolja az, ha a management rendszer rendelkezésre állása alacsonyabb. Ezért ahhoz, hogy a hálózat 5 kilences rendelkezésre állást biztosítson, általában nem kell, hogy a management rendszer is ezt tudja biztosítani.

Azonban szükséges megkülönböztetni egymástól a management alkalmazások típusait: az olyan management rendszerek rendelkezésre állása, amely a hálózat monitorozásával foglalkozik, nem szabad, hogy sokkal az 5 kilences rendelkezésre állás alá essen, mert a monitorozás kihagyása jelentős hatással lehet a hálózat rendelkezésre állására. Másrészről, például egy olyan management rendszer rendelkezésre állása, amely azzal foglalkozik, hogy új felhasználókat adjon a hálózathoz, kevésbé kritikus.

Végül, az olyan management rendszer rendelkezésre állása, amely számlázási adatokat gyűjt, ugyanolyan magas rendelkezésre állást kell, hogy biztosítson, mint a hálózat. Nem azért, hogy a hálózat rendelkezésre állása nagyobb legyen, hanem azért, mert a számlázási információk elvesztése jelentős bevételi veszteség a szolgáltató számára.

Oldja fel az OSS akronimát. Mit csinál egy OSS?

Operational Support Systems: Az OSS maga a menedzsment rendszer, általában a szolgáltatók nevezik így azért, mert lényegében a működésüket támogatja a menedzsment rendszer.

4. Dimenziók

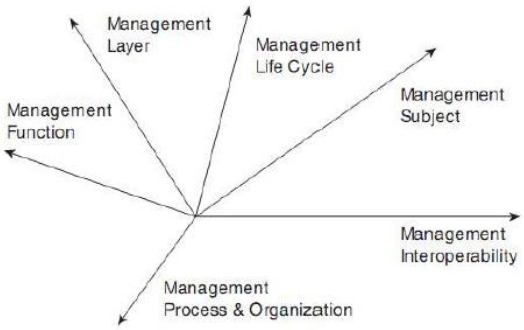
Milyen dimenziói vannak a hálózatmenedzsmentnek? Sorolja fel ezeket!

● **Interoperability** (együttműködés) – kommunikáció, funkciószolgáltatás, információszolgáltatás, közös dimenziók

● **Subject** (tárgy) – scope, technológia, szolgáltatások

● **Life cycle** (életciklus) – plan, deploy, operate, decommisson

● **Layers** (rétegek) – minden rétegekre van osztva, mngmnt layer, network elements -> element management -> network management -> service management -> business management



● **Functions** (funkciók) – layerenként különböző, ahhoz hasonló FCAPS funkciók (Fault, Configuration, Accounting, Performance, Security)

● **Process&Organization** (folyamat és szervezés)

Hálózatmenedzsment együttműködés szempontjából milyen szempontokat kell figyelembe venni?

Kommunikáció (communication viewpoint): A menedzsment kommunikációban résztvevő felek közti üzenetek típusait, az üzenetküldés módját, szabályait határozza meg. (fő részei: session establishment, authentication, request type,timestamp, encoding)

Funkció (function viewpoint): Meghatározza, hogy milyen funkciókat várunk el a kommunikáló felektől. Pl. passzív működés (monitoring), aktív működés (QoS beállítás); lehetséges-e az “event subscription”; meg tudja-e mondani, hogy milyen feladatokra képes stb.

Információ (information viewpoint): Definiálja, hogy a felek között kicserélt menedzsment információ milyen módon van reprezentálva.

Adja meg a menedzsment életciklust és röviden ismertesse egyes fázisait!

Az egyes modulok kialakulásának, a hálózat élettörténetének lekövetése.

Fázisai: (Prepare) -> Plan -> (Design) -> Deployment -> Operate -> (Optimization) -> Decommission

előadáson nem voltak a zárójelesek!

● Prepare: Az elérni kívánt cél meghatározása, technológiai döntések meghatározása, előnyök és hátrányok elemzése történik ebben a fázisban.

● **Plan:** Megvalósítással kapcsolatos kérdések megvitatása (pl. pontosan mit kell újítani, van-e hely az új eszköznek, bírni fogja-e az áramellátás az új eszköz beüzemelését)

● **Deployment:** Ez azt jelenti, hogy a berendezéseket el kell helyezni, majd beüzemelni. Egyedi management műveletek végrehajtását is magában foglalja (pl. IP cím beállítása)

● **Operation:** Az a fázis, amelyben a legtipikusabb hálózat management feladatok foglalnak helyet (pl. hálózat monitorozása, hibakeresés, teljesítmény növeléséért tett lépések, teljesítménnyel kapcsolatos statisztikai adatok gyűjtése, számlázási adatok gyűjtése stb.)

● **Decommission:** Különféle okok miatt szükség lehet az eszközök leszerelésére (pl. újabb eszközökre cseréljük a régieket). Az eszközök leszerelését úgy kell végrehajtani, hogy az esetleges hálózati kimaradás a lehető legkisebb fennakadást okozza.

Miért kellenek "sablonok" (cookbooks), dokumentáció és standard eljárások a hálózatmenedzsmentben?

A sablonok, a dokumentációk, a standard eljárások azért szükségesek, hogy a szervezés és minden management eljárás (vagyis maga a network management) a lehető leghatékonyabb legyen.

Milyen rétegekre (TMN) bonthatjuk fel a hálózatmenedzsmentet? Mi az egyes réteget feladata? Milyen előnyök és hátrányokkal járhat ez a rétegződés?

Network Elements: A hálózati berendezések által nyújtott menedzsment funkciók.
Element Management: A hálózati berendezések állapotának megfigyelése, változtatása és azok öntesztelésre utasítása illetve az általuk küldött riasztások figyelése.
Network Management: A hálózati berendezések közti kapcsolatok és függőségek megfigyelése és a hálózat működésének szempontjából helyes beállítása.
Service Management: A hálózat által nyújtott szolgáltatások menedzsmentje, és annak biztosítása, hogy a szolgáltatások megfelelően működnek és jól futnak.
Business Management: A szolgáltatásokkal kapcsolatos üzlet menedzsmentje, úgy mint számlázás, helpdesk menedzsment, üzleti előrejelzések, stb.
Előnye a modellnek, hogy egymásra épülő funkciókra lehet bontani a hálózatmenedzsmentet. Hátránya, hogy ha szó szerint próbáljuk megvalósítani, sokszor feleslegesen sok adminisztrációt igénylő és rugalmatlan rendszert kapunk.

5. Funkciók

A Performance (teljesítmény) és Accounting (könyvelés) menedzsment területek hasonló adatokkal dolgoznak. Mi mégis a lényeges különbség közöttük az adatok gyűjtésére és felhasználásra vonatkozólag?

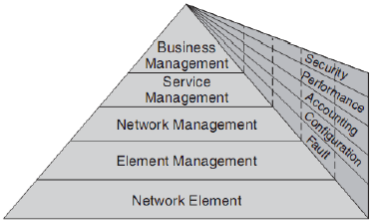
Accounting esetén általában kevés paraméter lényeges, mivel az szolgáltatásonként meg van határozva, hogy mi alapján kell számlázni. Ezek gyűjtése online, és felhasználásuk többnyire offline, kivéve az előre fizetett szolgáltatások esetén.

Performance esetén minden egyes eszközről több különböző adatot szeretnénk megtudni, ezért rendkívül költséges lenne ezek folyamatos figyelése. Ehelyett az eszközök saját tárukban gyűjtik inkrementálisan az adatokat, majd a lekérdezés egy alacsony forgalmú időszakban (éjszaka) történik, és az adatokat egy célalkalmazás (number crunching) dolgozza fel.

Az adatgyűjtés célja eltér egymástól a két esetben. Performance management során az adatokat statisztikai célból, elemzés céljára gyűjtjük, mint például a hálózat terhelése, kihasználtsága, stb. Az accounting management során az adatok alapján határozzuk meg az egyes felhasználók által fizetendő díjakat. Ebben az esetben jóval szigorúbb szabályoknak kell megfelelnünk, mivel fontos hogy az adatok teljesek és pontosak legyenek. Végérvényben minden kis elvesztett adattal a bevételből is veszítünk. Performance management esetén kis mértékű adatvesztés megengedett mindaddig, míg statisztikailag releváns marad a többi infó.

A TMN referencia modellben milyen menedzsment funkciók vannak? Mindegyiket definiálja 1 mondatban!

Fault, Configuration, Accounting, Performance, Security



Hibamenedzsment: monitorozás, hibákra való reakció. Általában Trouble Ticketing rendszerrel megsegítve.
Konfigurációm.: A hálózat beállításait, eszközeit követi. A változásokat dokumentálja. A menedzsment rendszer nyilvántartását szinkronizálja az eszközökben található beállításokkal. Más menedzsment rendszerek nagyon függnek a konfig. menedzsmenttől.
Accountingm.: A nyújtott kommunikációs szolgáltatások által szerzett bevétel menedzsmentje. Az erőforrások felhasználásának nyomon követése.
Teljesítménym.: A legfontosabb nem funkcionális tulajdonság. Pl. Átbocsátóképesség, késleltetés, megbízhatóság, jitter.
Biztonságmen.: A hálózat fenyegetésektől való védelme. A menedzsment biztonsága: csak az menedzseljen, akinek szabad. A biztonságosság menedzsmentje: védelem a vírusoktól, támadásoktól, behatolás detektálás.

Mi az a TOM, milyen módon közelíti meg a hálózatmenedzsmentet? Hogyan viszonyulnak ezek a területek az FCAPS funkciókhoz?

Telecoms Operations Map, amely a **Fulfillment, Assurance, Billing** megközelítést használja. A TOM középpontjában a menedzsment életciklus áll. Az OAMP helyett új megközelítéssel állt elő. A Fulfillment a Configuration feladatait látja el, a Billing az Accounting, az Assurance pedig a Fault, Performance és Security feladatait.

A konfiguráció menedzsment esetén milyen szinkronizációs megközelítéseket alkalmazhatunk? Melyik mely területeken terjedt jobban el? Miért?

Reconciliation (Egyeztetés) abban az esetben történik, ha a hálózat állapotát tekintjük helyesnek, és ezért a MIB alapján frissítjük az adatbázisunkat. Ez az elterjedtebb megoldás, főleg céges környezetben, ugyanis leginkább az a fontos, hogy tudjuk, mi van a hálózatban.
Reprovisioning akkor történik, ha a menedzsment rendszerben tárolt információkat tekintjük helyesnek, és a hálózati eszközöket újrakonfiguráljuk azok alapján. Ez a megoldás főleg a telekommunikációs vállalatoknál jellemző, ugyanis ott a tervezett hálózatnak meg kell felelnie a kiépítettnek, ha nem, akkor az utóbbi a rossz. Előfordul, hogy egy cégen belül különböző hatáskörrel mindkét megközelítést alkalmazzák.
Discrepancy reporting (Eltérés jelentés) esetén a felhasználó dönti el minden egyes esetre, hogy melyik megoldást alkalmazza.

Mi a különbség a hibák szűrése és a hibák korrelációs elemzése között?

Szűrés esetén csak bizonyos riasztások jutnak el a felhasználóig, az érintett alrendszer, szolgáltatás vagy súlyosság alapján. Szűrés az is, amikor a riasztások duplikálódását kívánjuk megakadályozni, és így a redundáns riasztásokat kiszűrni.
A korrelációs elemzés során több különböző riasztás helyett azokból egy következtetett összegző riasztást küldünk, mely értelmileg ugyanaz, mintha egyenként küldenénk el, de sokkal hatékonyabb módja a közlésnek.

Mi a különbség az Accountin és a Billing között?

Az Accounting csak egy része a Billingnek. Ahhoz, hogy egy szolgáltatást számlázzunk, nem elég a körülményeket ismerni (mit, mikor, hogyan használtak), hanem szükségesek a tarifák ismeretei is, melyek segítségével eldönthetjük, hogy a könyvelt adatokat hogyan számlázzuk. Billing = Accounting + tarifa

Ismertesse az OAMP feladatokat! Viszonyítsa ezeket az FCAPS funkciókhoz!

O: a hálózat (és az általa nyújtott szolgáltatások) fennakadás nélküli működtetése. Beletartozik a hálózat folyamatos ellenőrzése (monitoring), hogy a hibákat a lehető leghamarabb észrevegyék, lehetőleg mielőtt egy felhasználónak problémája származna belőle. (részben Fault, Performance)
A: az erőforrások és azok kiosztásának számon tartása. Minden „házimunka” ide tartozik, mely ahhoz kell, hogy irányításunk alatt tartsuk a rendszert. (főleg Accounting, részben Performance, Security)
M: javítások és fejlesztések. Ide tartoznak a módosító és megelőző intézkedések is, a menedzselt hálózat jobbá tétele érdekében. (főleg Fault, Performance, Security, részben Configuration)
P: az erőforrások oly átrendezése, hogy egy adott szolgáltatást lehetővé tegyenek (Configuration)

	F	C	A	P	S
O	(X)	—	—	(X)	—
A	—	—	X	(X)	(X)
M	X	(X)	—	X	X
P	—	X	—	—	—

Miről szól a Performance menedzsment?

Arról, hogy a hálózatot folyamatos monitoring segítségével úgy konfiguráljuk, hogy a lehető legjobb teljesítményt nyújtsa. Ezt pillanatnyi megfigyelések, és hosszú távú adatgyűjtés segítségével tudjuk megalapozni.
Performance is the most important nonfunctional property.

Mi a különbség az auditing, discovery és autodiscovery feladatok között?

Az auditing a rendszer állapotának kiolvasása, hogy meg tudjuk állapítani, hogy mi lett beállítva az eszközeinken**. A discovery** alapjelentése annak felfedezése, hogy milyen eszközök vannak a hálózatban, azonban sokszor az auditing helyett használják tévesen ezt a kifejezést. Az a**utodiscovery** kifejezést szokták ezért néha a discovery funkciók leírására használni.
Másképp:
auditing – azzal foglalkozik, hogy ténylegesen mit konfiguráltak
discovery – kideríteni, hogy mi van a hálózatban
autodiscovery – automatikusan méri fel, hogy mi van a hálózatban (például: előre definiált időközönként)

Milyen feladatai vannak a hiba menedzsmentnek? Milyen eszközöket használhatunk ennek segítésére?

Feladata a monitorozás és a megfelelő reakció az egyes hibákra.
● **Network monitoring**: célja a hálózat működésének megfigyelése. Segítségével választ kaphatunk az egyes hibajelenségek okára.
● **Root cause diagnosis**: Hibás működés, probléma esetén meg kell keresni a probléma eredetét.
● **Log history maintenance**: Az egyes esetek naplózása a későbbiekben hasznos lehet, ha hasonló hiba jelentkezik.
● **Trouble ticketing**: Tömör leírás a hiba jellemzésére. Kézről-kézre jár (eközben bővül a ticket), amíg valaki meg nem oldja a problémát.
● **Proactive fault management**: Megpróbáljuk megelőzni a hibát, vagy felkészülni annak bekövetkezésére.
● **Alarm handling**: Össze kell gyűjteni, meg kell jeleníteni és naplózni is kell az egyes riasztásokat.

Adjon egy-egy példát hibamenedzsmentre az eszköz, a hálózati és a szolgáltatási rétegekre vonatkozólag!

érdemes sajátot kitalálni…
“Egy VoIP szolgáltatásra vonatkozólag ezt végig lehet gondolni: Üzleti réteg: Help Desk (telefonos ügyfélszolgálat) Szolgáltatási réteg: egyik call controller (SIP proxy) magas hívás blokkolást jelez. Hálózati: adott útvonalon csomagvesztés, késleltetés vagy késleltetés ingadozás Eszköz szinten: adott interfészeken hibás csomagok vétele”

Milyen technikai indokok jöhetnek szóba a szolgáltatók flat-rate árazása mögött?

Flat rate: Előre meghatározott teljes ár, amely független a teljesítés alakulásától, a megjelenésektől, stb. A szolgáltatóknak nem minden esetben van technológiai megoldásuk a használtsági adatok pontos kigyűjtéséhez, hogy az ügyfelek részére tételesen biztosítsák a számlázási adatokat.

6. Management infos

Objektum orientált modellezést használnak-e az SNMP MIB-ben? Indokolja a válaszát!

MIB objects lack features that are commonly associated with object orientation. One such feature is inheritance (the capability to derive specializations from existing object class definitions). Other object-oriented features that SNMP MIB objects lack but that were not mentioned in the chapter include polymorphism (the capability for instances of a subclass to be treated as if they are instances of a superclass) and the inclusion of methods as part of the object class definition, commonly associated with the property of encapsulation. SNMP MIB objects are essentially simply MIB variables.

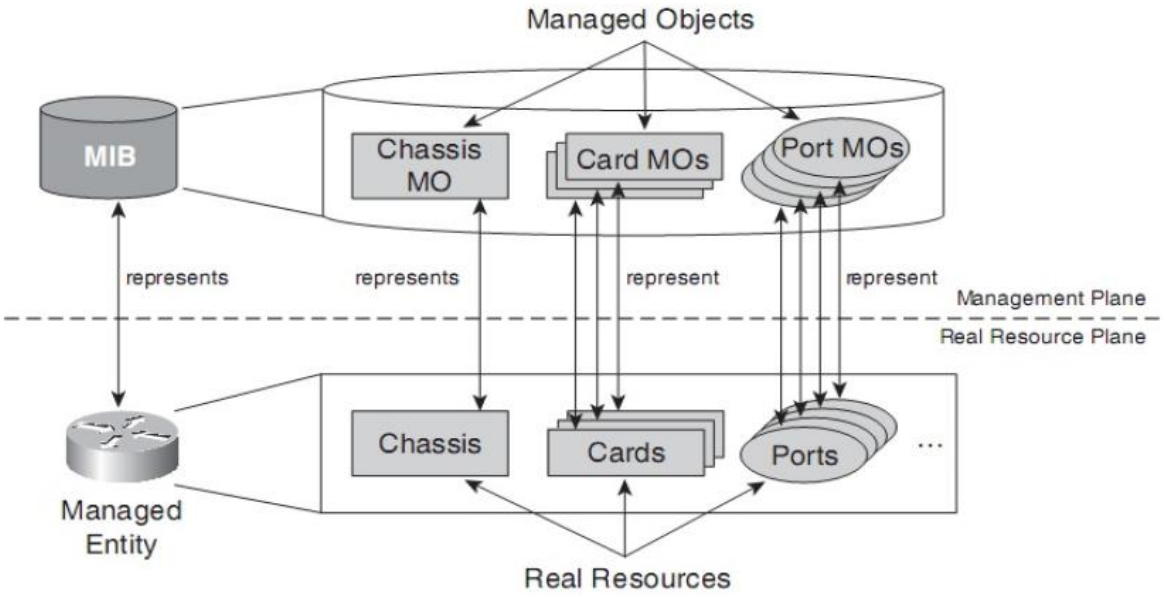
Nem, mert az MIB-ben levő objektumok nem rendelkeznek az objektum-orientáltságra jellemző tulajdonságokkal, pl. öröklés, polimorfizmus, tagfüggvények osztályhoz rendelése. Az SNMP MIB objektumok általában szimplán MIB változók.

Mi a lényeges különbség az objektum típus és az objektum példány között? Érveljen egy példán keresztül!

An object type is a definition of a kind of managed object; it is declarative in nature. In contrast, an object instance is an instantiation of an object type which has been bound to a value. For example, the notion of an entry in a routing table might be defined in the MIB. Such a notion corresponds to an object type; individual entries in a particular routing table which exist at some time are object instances of that object type.

Tehát az objektum típus egy menedzselt objektumot definiál, míg az objektum példány egy objektum típus példányosítása konkrét értékekkel. MIB = instance

Hogyan viszonyul egymáshoz a MIB, MO, Managed Entity, Real Resources? Rajzoljon!



Milyen kategóriái lehetnek a menedzsment információknak?

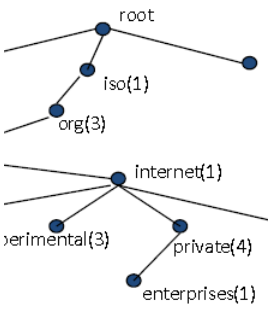
- **Állapot információk**
  - fizikai és logikai erőforrások
  - működési adatok
  - gyorsan változhatnak az adatok
  - management alkalmazás nem tudja módosítani az állapotinformációt csak lekérni
- **Fizikai konfigurációs adatok**
  - az eszközhöz tartozik
  - a manager nem tudja megváltoztatni
  - ritkán változik az információ
  - "Best to cache it at the managers" - dokumentálásra gondol szerintem
- **Logikai konfigurációs adatok**
  - tipikusan a management által kontrollált és megváltoztatható
  - "Cache it" - dokumentálás, csak akkor változik, ha a manager megváltoztatja
  - Típusai: startup és transient
- **Történeti információk**
  - nem tükrözi a jelenlegi állapotokat

Az SNMP MIB rendezésére használt fa miben tér el pl. egy fájl rendszer fától?

The naming tree in a file system represents a containment hierarchy between the objects. If you delete an object that contains other objects—that is, that are in a subtree underneath the object—those other objects will be deleted as well. The object identifier tree of SNMP MIBs, on the other hand, does not reflect a hierarchy between objects. Instead, it reflects the structure of the underlying MIB definition, or the way in which the definitions of the object types are grouped that are instantiated by objects in the MIB. The objects in the MIB themselves are flat; every one of them is a leaf node in the MIB object identifier tree.

A fájlrendszerben a könyvtár is egy elem tulajdonságokkal (pl. jogosultságok). MIB fában csak a leveleknek van tulajdonsága, értéke.

Milyen objektum típusok lehetnek az Internetes MIB-ekben?



Enterprises node: Companies to add their own proprietary MIB modules. No need to ask for permissions (from any authority)

Managed Objects are always leaf nodes of the tree (**az MO-k mindig a fa levelei**)

- Scalars
  - Instantiated only once in the agent (csak egyszer példányosított)
- Columnar objects
  - Instantiated multiple times (többször példányosított)
- Nevertheless all MO's are simple data types as defined in SMI and SMIv2
  (Mindazonáltal minden MO egyszerű adattípusként van definiálva az SMI-ben és az SMIv2- ben.)
  - i.e., no complex data types, so use creative ways to represent that information as simple object types

Milyen különböző paradigmák szerint lehet a menedzsment modellt megadni? Mutassa be a különbségeket!

- **object-oriented constructs** (objektum orientált konstrukció)
  - MO classes (MO osztályok)
    - Attributes (attribútumok)
    - Emits notifications (figyelmeztetés küldése)
  - derived classes (subclass / superclass; inheritance) (származtatott osztályok: subclass, superclass, örklődés)
- **tables and variables** that can be grouped in certain ways (táblák és változók különféle módon lehetnek csoportosítva)
  - Table == class of Mos
    - one particular aspect of the device (az eszköz egy bizonyos nézőpontja)
    - attributes == table columns and instances by the table rows (az attribútumok a tábla oszlopai, és a sorok által jelölt esetek)
- model everything as **commands and functions** and their parameters (Mindent parancsként, funkcióként, és ezek paramétereként modellez. Pl. CLI parancsok és ezek felparaméterezése. Itt a modell nincs expliciten megadva.)

Mi az a SMI? Mit ad meg?

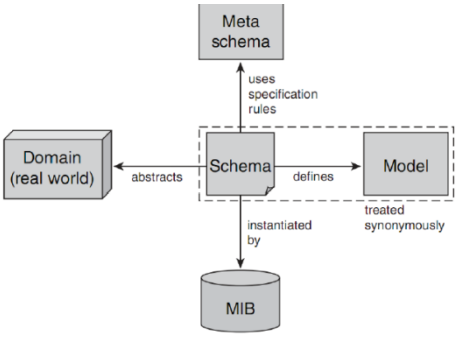
Structure of Management Information (SMI): A MIB definíciókat MIB modulokként specifikálja.
An MIB module generally serves a particular purpose, such as to define management information related to a device's communication interfaces or to a voice-mail server feature that is embedded on a particular type of device.

- Internet:
  - MIB-2: RFC 1213
  - SMI: RFC 1155
  - SMIv2: RFC 2578
- MIB definitions in MIB modules
  - Particular purpose
  - MIB of a devices instantiates multiple MIB modules
- SNMP MIB consists of a set of managed objects that instantiate object types that are part of a MIB module.

Együttműködés a mgmnt rendszerek között -> ennek érdekében a mgmnt adatokat strukturáltan érdemes tárolni.

Hogyan lehet MIB-eket definiálni? Hogyan viszonyul egymáshoz a való világ, a schema, a modell, a meta schema, és a MIB? Rajzoljon is!

- **Domain**: A domaint a modell foglalja össze (absztrahálja). A való világot domainnek hívjuk, mert ez jelképezi azt a területet, amelyet modellezünk.
- **Meta Schema**: specifikációs nyelv, ennek szabályait kell használni MIB definíció során. Meghatározza, hogy hogyan kell írni és értelmezni definíciókat.
- **Schema**: A modell, ami alapját képezi a MIB-ben lévő management információnak egy MIB definícióban.
- **MIB**: A séma, amely az eszköz MIB-jében van példányosítva.



- **Model:** A séma meghatározza a modellt, a sémával hasonló módon kezelik.

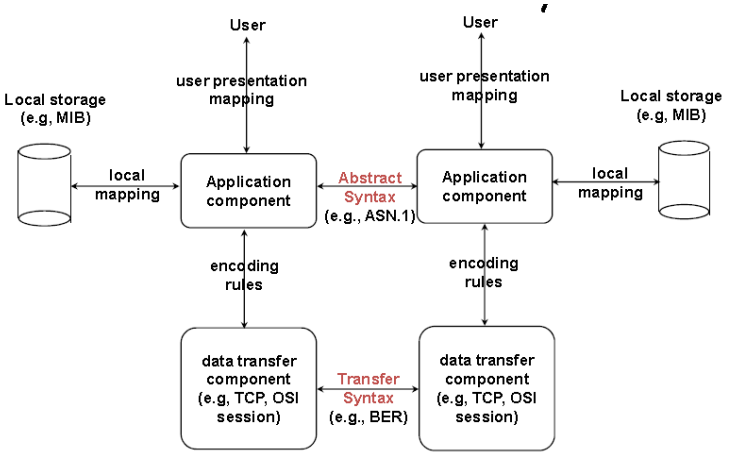
7. ASN1

Mi az ASN.1? Mire használják a menedzsmentnél? Hogyan köti össze a helyi adatokat, a megjelenítést, a transzport réteget? Rajzoljon!

SNMP-ben és OSI menedzsmentben használják, az ANS.1 kiterjedten használt alkalmazás adatok és PDU-k definíciójára.

Abstract Syntax Notation One (ASN.1)

- Eszköz független adat leíró nyelv
- CCITT (X.208) és ISO (ISO 8824) szabvány
- Absztrakt szintaxist ad meg alkalmazás adatoknak
- Alkalmazási és prezentációs struktúrát ad meg (protocol data units (PDUs))
- Az SNMP és OSI Management Information Base (MIB) megadására



Mi az a BER? Hogyan kapcsolódik az ASN.1-hez? Hogyan működik?

BER-t széles körben használják átviteli szintaxisok megadására.

Basic Encoding Rules (BER)

- Kódolási specifikáció
- CCITT (X.209) és ISO (ISO 8825) szabvány
- **Megadja az octet string kódolását minden ASN.1 típusnak**
- Típus-hossz-érték **type-length-value (TLV) struktúrát használ (|Type|Length|Value|)**
  - Az ASN.1-es TLV struktúra az **rekurzív** lehet
    - Minden érték további TLV értékeket tartalmazhat
  - Három metódus a kódolásra:a
    - Primitive, definite-length encoding
    - Constructed, definite-length encoding
    - Constructed, indefinite-length encoding
- A használt metódus függ az ASN.1 típustól és hogy a hossza ismert-e a típusból.

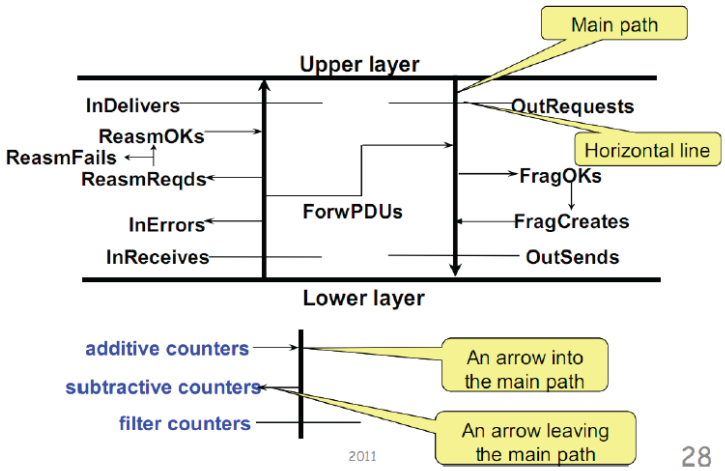
8. MIB-1

Ismertesse a CASE diagram koncepciót! Rajzoljon egy mintapéldát illusztrálásra, értelmezze a számlálókat!

A CASE diagram egy jól használható eszköz a MIB-ek fejlesztésénél

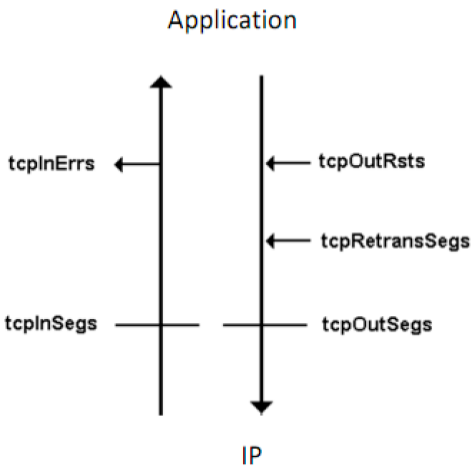
- a csomagtovábbítás folyamának leírása rétegenként
- általában szükséges a forgalmi összetétel számlálása protokoll rétegenként
- úgy, hogy minden valamelyik rétegből küldött vagy fogadott PDU valahová tartozik helyes és helytelen (hibás csomag) egyaránt csomagok áramlása az adott rétegben

ötlet: Jeffrey Case (1989)



Main path: a két réteget köti össze
Horizontal line: egy számlálót jelent, mely minden átmenő PDU-t megszámol
InReceives = InErrors + ReasmReqds + ForwPDUs - ReasmOKs + InDelivers
OutSends = OutRequests + ForwPDUs - FragOKs + FragCreates

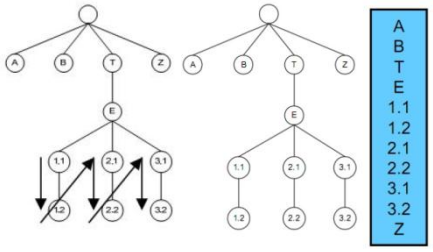
TCP:



Ismertesse a lexikografikus sorrendezést! Rajzoljon is! Mire és hol használják?

Használat: ismeretlen MIB (rész) szekvenciális bejárására (felderítésére).

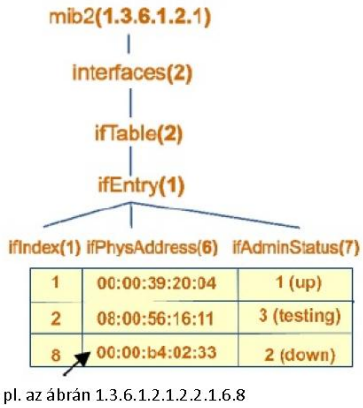
Mivel az objektumok attribútumait lexikografikus rendezésben tároljuk (mint pl. a szótárakban a szavakat), a táblázatok adatainak lekérdezése jelentősen egyszerűsödik a GetNextRequest kérés által, ami így az egész táblázatot oszlopfolytonosan adja vissza, mint egy (preorder) mélységi bejárás (ld. ábra).



Mondjon egy olyan MIB objektumot, ahol write only MAX ACCESS jogosultságnak lenne értelme! Miért?

Jelszavaknál van értelme, ugyanis fontos, hogy azokat senki se olvashassa, azonban a jelszó változtatására szükség van.

Hogyan épülnek fel a táblázatok a MIB-ben? Milyen ASN.1 típusként vannak definiálva? Mi a MIB konvenció? Hogyan férhetünk hozzá egy meghatározott táblázat elemhez? Rajzoljon és magyarázza!



pl. az ábrán 1.3.6.1.2.1.2.2.1.6.8

Kétdimenziós táblázatokat hozunk létre, amelyek nem tartalmazhatnak beágyazott táblázatokat. Az elemek egyértelmű azonosításáért egy vagy több index a felelős. A táblázat sorai **SEQUENCE típusként vannak definiálva**, a táblázat pedig ezen sorokból alkotott **SEQUENCE OF típusból** áll össze. Egyszerű objektumok példányaihoz az <objektum_azonosító>.0 azonosítóval, a táblázat példányaihoz pedig a <táblázat azonosító>.<oszlop>.<indexérték> azonosítóval férünk hozzá.

A MIB-ben mi a különbség az objektum típus és az instance között?

Egy objektum típusra vonatkozó OID globálisan egyedi. Egy instance-re vonatkozó OID csak az őt tartalmazó MIB-ben egyedi.

Válasszon 3 objektumot a MIB-II alatt és röviden ismertesse, hogy milyen menedzsment információkat szolgáltat az alatta lévő fa.

system: a rendszer általános információi (Uptime, Name, Location, stb.)
interfaces: információk a rendszer és a hálózat közti interfészekről
at: leírja a hálózat és az internet közötti címfordít
továbbiak: ip, icmp, tcp, udp, egp, transmission, snmp

Miért van szükség táblázatok indexelésére?

Mert az indexek azonosítják a táblázat sorait, elemei ezek segítségével érhetőek el. Indexeléssel a táblázat elemei gyorsabban, és nem csak lineáris kereséssel érhetőek el.

Hogyan lehet több indexet használni egy táblázatban? Hogyan lehet nem egyszerű típusú MO-val indexelni?

An SNMP table can be defined as an ordered collection of objects consisting of zero or more rows. Each row may contain one or more objects. Each object in a table is identified using the table index. A table can have a single index or multiple indices.
A scalar variable has a single instance and is identified by its ".0". On the other hand, a table object or the columnar variable can have one or more instances and is identified by its index value. To identify a specific columnar variable, the index of the row has to be appended to its OID.
For example for a table with OID .1.3.6.1.2.1.x.x.xTable, with the column name yy and the index value ind1, the value of the column yy can be got by appending the instance ind1 to the columnar OID .1.3.6.1.2.1.x.x.xTable.xEntry.yy. If the table has multiple indices namely ind1 and ind2 then the value of the column yy can be got by using the OID .1.3.6.1.2.1.x.x.xTable.xEntry.yy.ind1.ind2.
For example, consider tcpConnTable. It has four indices namely tcpConnLocalAddress, tcpConnLocalPort, tcpConnRemAddress, and tcpConnRemPort where the values of the table are as follows.

tcpConnState	tcpConnLocalAddress	tcpConnLocaPort	tcpConnRemAddress	tcpConnRemPort
listen(2)	0.0.0.0	21	0.0.0.0	0
listen(2)	0.0.0.0	23	0.0.0.0	0
listen(2)	0.0.0.0	3306	0.0.0.0	0
listen(2)	0.0.0.0	6000	0.0.0.0	0
established(5)	127.0.0.1	1042	127.0.0.1	6000
established(5)	127.0.0.1	6000	127.0.0.1	1042
closeWait(8)	192.168.1.78	1156	192.168.4.144	80

To get the value of the column tcpConnState for the last row, you have to query with the OID tcpConnState.192.168.1.78.1156.192.168.4.144.80 where 192.168.1.78 is the value of tcpConnLocalAddress for the last row, 1156 is the value of tcpConnLocalPort for the last row 192.168.4.144 is the value of tcpConnRemAddress for the last row 80 is the value of tcpConnRemPort for the last row.

Also if the index is of integer type, it can be in any order. For example in a table, if the values of the index column are {1,2,3,4}, it can have values in any order say {2,4,3,1}.

Lexikografikus rendezéssel?

9. SNMPv1

Milyen biztonsági mechanizmus van az SNMPv1-ben? Hogyan működik? Mi a hibája?

- Authentication service
  Ágens korlátozhatja a hozzáférést a MIB-hez.
- Access policy
  Különböző menedzsereknek különböző hozzáférési jogosultsággal rendelkeznek.
- Proxy service
  Ágens közvetíthet más menedzselt eszköz felé.
  A menedzselt eszközre vonatkozóan authentication service és access policy kellhet a proxyban.
- **Az SNMP csak egy primitív és korlátozott biztonsági képességgel rendelkezik**

!community

- Ágens és a menedzserek között: authentication, access control & proxy characteristics
- Ágens helyi értelmezése a fentieknek: Minden community egy egyedi community névvel azonosítva, Ágens több community-t is használhat**,** Minden parancsnak hordoznia kell a community azonosítót**,** Különböző ágensek használhatják ugyanazt a community azonosítót
- SNMP authentication service, Minden SNMP üzenetnek hordoznia kell a community nevet (mint valami jelszó)**,** Nagyon primitív megközelítés**,** Ezért a legtöbb ágens csak GET parancsot fogad el**, text alapú biztonsági mechanizmus** gyenge biztonság**,** community**,** ágens kérheti minden üzenetre **,**de ez egyszerű textként utazik az üzenetben

Milyen SNMPv1 PDU-k vannak? Ismertesse őket röviden!

Protocoll Data Unit
- Get Request : objektum értékének kinyerésére az ágensből
- Get-Next Request: hasonlóan a Get Request, de a MIB fában lexikografikus sorrendben következő objektum értékével tér vissza
- Set Request: objektum értékének beállítására
- Response: ágens válasza a Get Request, Get-Next Request és Set Request PDU –kra
- Trap: ágens jelentésekre**,** csak egyirányú, nincs válasz a menedzsertől

Hogyan kérdezne le egy ismeretlen táblázatot SNMPv1-ben?

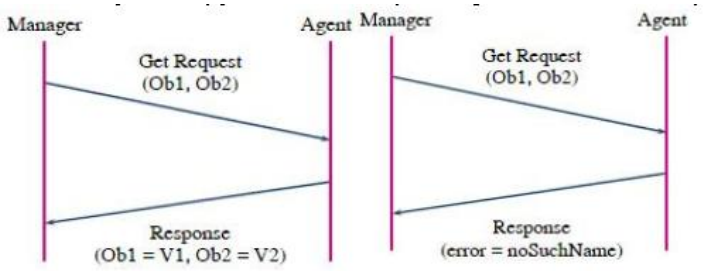
Lexikografikus rendezés -> sok getNextRequest

Mi az, hogyan és hol használják a lexikografikus sorrendezést?

- Soros MIB hozzáféréshez
- A fa struktúrából következően, bármely objektum OID-je meghatározható a gyökértől az objektumig vezető úton
- lexicographical ordering: preorder traversal (root, left, right) of a tree depth-first search (esetleg korábbi példa)

Mit jelent az SNMPv1 üzenetek atomikus volta?

Vagy minden értéket visszaad, vagy egy sem kerül visszaadásra.
Sikeres: (bal kép). Ha bármelyik objektum nincs implementálva, sikertelen: (jobb kép).



10. RMON 1

Hogyan működik az RMON szűrési csoportja?

Remote Network Monitoring

- csomagok kiválasztására
- Kétféle szűrő (filter)
    - Adat szűrő: bitmintán alapuló illesztés
    -Státusz szűrő: csomagstátusz alapján(crchiba, jó csomag, stb)
- szűrő logika

input = bejövő csomag rész, szűrésre

filterPktData = tesztelendő bitminta

filterPktDataMask = bitmaszk, releváns bitek

filterPktDataNotMask = teszt egyezés vagy nem egyezésre

Példa:

filterPktDatOffset = 0

filterPktData = 0x0000000000A50000000000BB

filterPktDataMask = 0xFFFFFFFFFFFFFFFFFFFFFFFF

filterPktDataNotMask = 0x000000000000FFFFFFFFFFFF

Milyen információt szolgáltat az RMON statistic csoport?

MAC (alhálózat) szintű kihasználtság és hiba statisztika., hálózat terheléséről,

R/W objektumok: etherStatsSource, etherStatsOwner, etherStatsStatus

counterek: packets, octets, broadcasts, multicasts, collisions, errors, csomagméret eloszlások

Hogyan vezérelhetjük a távoli monitorozást? Mik ennek jellegzetes problémái? Hogyan kezelik ezt az SNMP keretben?

Cél a távoli monitorozás, adatgyűjtés, alhálózati viselkedés megfigyelése, rendszer és ágens tehermentesítés. probléma a felismerés, és a jelentés, együttműködés több menedzserrel.

Az **RMON MIB**-ben kerültek meghatározásra **a vezérlési funkciók** (Configuration Control, Action Invocation), ezek funkcionális csoportokba vannak rendezve, csoportonként kontrol- és adattáblákkal (utóbbiak csak olvashatók). A kontrol táblában határozzuk meg az adatgyűjtés paramétereit (forrás, adat típusa, gyűjtés időtartama), az adattáblákból pedig kiolvashatjuk az összegyűjtött adatokat.
A kontrol táblában néhány paramétert csak úgy módosíthatunk, hogy invaliddá tesszük (ezzel töröljük a hozzá tartozó adattábla bejegyzéseket is), majd a módosított paraméterekkel újra létrehozunk egy bejegyzést.

Milyen információ szolgáltnak az RMON matrix csoport?

Kihasználtság, és hiba statisztika host párokra.

Hoszt párokra tárol forgalmi információkat az alhálózatra vonatkozóan : packets, octects, errors

mátrix formájú tárolás

1 kontroll tábla

2 adattábla

- ugyanaz az információ
- de
    - forrás és cél szerint
    - cél és forrás szerint indexelve
- az össze hosztból kifelé irányú és az összes a hosztba befelé irányú forgalom kigyűjthető

Hogyan működik a csomagelkapás az RMON-ban?

Elfogó (capture) csoport segítségével. A csatornán (szűrőkön) átjutó csomagokat puffereli két táblázattal:

_ bufferControlTable
- pufferelési funkciók

_ captureBufferTable
- adatok tárolására

Egy ControlTable bejegyzéshez tartozik n db. BufferTable bejegyzés.

Hogyan valósítja meg az RMON probe az erőforrások hatékony megosztását? Mire kell figyelnie a menedzsereknek?

Minden monitorozási funkcióhoz tulajdonost rendel (owner, aki létrehozta), így a menedzser felismerheti saját foglalásait (ha nincs már rá szüksége, felszabadíthatja). Egyeztetés és megfelelő jogok birtokában más erőforrását is felszabadíthatjuk (pl. ha a foglalást birtokló menedzser időközben összeomlott). A monitor rendelkezik saját funkciókkal is, ezekben a legnagyobb a bizalma. Egy funkció kérés beérkeztekor a kontroll tábla végigpásztázása hasonló, már kért funkció után.

Hogyan kezeli az RMON probe a konkurens vezérlő tábla sor hozzáadást? Mi az RMON polka?

RMON polka:

1. menedzser sor létrehozása createRequest(2) EntryStatus mezőértékkel
2. ha az ágens végrehajtotta a műveletet, akkor a sor státuszt underCreation(3) értékre állítja (mindaddig ez az értéke, amíg a menedzser az összes sorát be nem állítja)
3. ha a menedzser kész az összes sorával, azok státuszát valid(1) értékre állítja
4. ha a sor már létezik, vagy createRequest végrehajtása alatt van, akkor hibával tér vissza

Gyakorlatilag, amíg nem konzisztensek a táblában található adatok, addig más nem olvassa őket, mivel jelzi, hogy éppen létrehozás alatt állnak.

Hogyan épül egymásra az RMON adat és statisztika gyűjtés? Milyen kapcsolódó csoportok vannak?

Az RMON az adatokból vett minta alapján statisztikákat generál, majd ezen statisztikák periodikus mintavétele alapján top- és táblázatos history-kat készít.

RMON MIB csoportok

1. **statistics:** MAC szintű kihasználtság és hiba statisztika
2. **history:** periodikus statisztikus minták a statistics csoportra
3. **alarm:** mintavételi idők és riasztási küszöbök
4. **host:** hoszt forgalomak az adott alhálózaton
5. **hostTopN:** „top" jellegű sorrendezett statisztika valamely paraméterére a hoszt táblának
6. **matrix:** kihasználtság és hiba statisztika hoszt párokra
7. **filter:** adott szűrési kritériumnak megfelelő csomagok vizsgálatára
8. **capture:** hogyan küldjük az adatot a menedzser állomáshoz
9. **event:** RMON által generált jelentések meghatározása
10. (**tokenRing:** vezérlési és adattáblák token ring alhálózatra)

Mi az RMON koncepció? Milyen előnyökkel jár(hat) a távoli monitorozás?

A dedikált (vagy egyéb funkciókat is ellátó) RMON eszközök (monitorok vagy probe-ok) a menedzser és az ágensek tehermentesítése érdekében promiscuous módban **figyelik az adott alhálózat forgalmát, csomag(részleteket) gyűjtve**, **analizálva** (proaktív monitorozás). Így nem csak az egyedi eszközökre, hanem az adott LAN-ra vonatkozóan is gyűjthetünk információkat. Csökkenti az SNMP forgalmat, több menedzserrel is együttműködhet (növekszik a megbízhatóság). Az aktív analízisek alapján gyorsabb diagnosztika és jelentés az NMS felé, valamint lehetőség van offline monitorozásra is (ha a menedzser épp nem elérhető).

Milyen információt szolgáltat az RMON history csoport?

Periodikus statisztikus minták a statistics csoportra. (kivétel a csomagvesztés-eloszlás) Körkörös puffert (bucket) használ, melynek méretét a menedzser igényli. Vezérlés: historyControlTable (melyik szegmensre, milyen mintavétellel – javaslat: 30mp és 30p) Adatok: etherHistoryTable

Milyen riasztási sémákat alkalmazhatunk RMON-nál?

- monitor vagy menedzser új sor létrehozásával hozhat létre új riasztást

- figyelt változó, mintavételi intervallum, küszöb soronként egyedi
- **emelkedő (rising)** küszöbátlépés

  ha az aktuális minta nagyobb egyenlő, mint a felső küszöb és az utolsó érték kisebb a küszöbértéknél
- **eső (falling**) küszöbátlépés

  ha az aktuális minta kisebb egyenlő, mint az alsó küszöb és az utolsó érték nagyobb a küszöbértéknél

- kétféle érték a riasztásokhoz
  - **absoluteValue** : mintavételi időpontokban
  - **deltaValue**: rate of change

Mi az általános kapcsolat az adat és a vezérlési táblák között?

A vezérlési tábla indexei (pl. rm1ControlIndex) segítségével az adattáblában azonosíthatjuk az azonos bejegyzéshez tartozó adatsorokat, mivel a saját indexükön kívül tartalmazzák a hozzájuk tartozó kontrolltábla-bejegyzés indexét is (pl. rm1DataControlIndex).

Milyen információt szolgáltnak az RMON host csoportok?

Hoszt-forgalmak az adott alhálózaton. Ki-/bemenő csomagok, oktettek, kimenő multicast, broadcast üzenetek, hibák. A táblázat indexelhető MAC cím (hostTable), létrehozási idő (hostTimetable) vagy valamely paraméter (hostTopN) alapján.

Mely rétegekben működhet az RMON-2? Hogyan adhatjuk meg, hogy mely protokollokat vizsgáljuk?

- Protokoll azonosító (ProtocolDirID)
  - protokollonként egyedi byte sztring
    N x 32 bites azonosítók [a.b.c.d]
  - hierarchikusan, MIB szerű fába rendezve
    ether 2 = 1 [0.0.0.1]
    llc = 2 [0.0.0.2]
    snap = 3 [0.0.0.3]
    vsnap = 4 [0.0.0.4]
    ianaAssigned = 5 [0.0.0.5]
  - IP az Ethernet felett: ether2.ip
  - UDP az IP és Ethernet felett: ether2.ip.udp
  - SNMP az ...: , ether2.ip.udp.snmp
- hálózati réteg
  Type mező az Ethernet MAC keretben
  16bites protokoll azonosító
  [0.0.a.b], ahol a és b tartalmazza a 16 bites értéket
  IP esetén è [0.0.8.0]
  - transzport réteg
  IP PDU-ban Protocol mező tartalmazza a felsőbb réteg protokollokat
  8bites prot. azonosító
  [0.0.0.a]
  pl.: UDP esetén 17 è [0.0.0.17]
  - alkalmazási réteg
  UDP PDU Port mezőben tartalmazza a felsőbb réteg protokollokat
  16bites prot. azo.
  [0.0.a.b], ahol a és b tartalmazza a 16 bites értéket, PORTra
  pl.: SNMP-re 161 è [0.0.0.161]

11. RMON 2

Hogyan adhatunk meg protokollokat RMON2-ben?

Az RMON2 MIB **protokoll könyvtárcsoportja (protocolDir)** valósítja meg a protokollokra vonatkozó típusinformációk központi tárolását, így a menedzser megismerheti, hogy mely protokollok értelmezésére képes az ágens. Az RMON2 protocol directory group egy protokoll könyvtár tábla, amely minden kezelendő protokollra egy bejegyzést tartalmaz. (A többi az előző kérdésben.)

Mi a különbség az RMON-1 és RMON-2 között?

- RMON MIB kiterjesztése a MAC réteg fölé OSI rétegek 3 – 7
- hálózati réteg protokollok és hálózati címek szerinti monitorozás
- alkalmazás szintű monitorozás: email, file transzfer, WWW protokollok

RMON1
- MAC címek szerint / LAN szegmens
- a forgalom forrása nem meghatározható pl. routeren érkező forgalom esetén

RMON2
- MAC keretben a magasabb szintű fejrész értelmezése, tipikusan IP csomag
- megállapítható a forgalom „igazi” forrása / célja

Hogyan működik a TimeFilter az RMON2-ben?

Time filter indexing: időre történő szűrés

Mivel periodikus lekérdezések esetén a változások érdekesek, csak azokat az objektumokra történik szűrés, amelyeknél változás történt az utolsó lekérdezéshez viszonyítva.

```
fooTable ...
        INDEX { fooTimeMark, fooIndex }

        FooEntry {
           fooTimeMark  TimeFilter
           fooIndex     INTEGER,
           fooCounts    Counter
        }
GetRequest(fooCounts.t.1, fooCounts.t.2):
        fooCounts.0.1  5
        fooCounts.0.2  9
        fooCounts.1.1  5
        fooCounts.1.2  9
        fooCounts.2.1  5
        fooCounts.2.2  9
        fooCounts.3.1  5
        fooCounts.3.2  9
        fooCounts.4.1  5
        fooCounts.4.2  9
        fooCounts.5.1  5
        fooCounts.5.2  9
        fooCounts.6.1  5
        fooCounts.6.2  9
        fooCounts.7.2  9
        fooCounts.8.2  9
```

Idő	Index=1	Index=2
0	...	...
1		

note that row 1 doesn't exist for times 7 and 8"

6	5	13
7	X	65
8	X	9

timeStamp	fooIndex (fooTable.1.2)	fooCounts (fooTabel.1.3)
6	1	5
8	2	9

Mivel a timeStamp6-ban állítjuk az 1. index értékét 5-re, a 8.-ban pedig a 2.-ét 9-re (alsó táblázat), ezért az 1. indexnek nincs értéke a 7-8-ban (felső táblázat), hiszen akkor már nem történt változás. Órán vmi olyasmit mondott, h a timefilter nem egy history, ezért az utolsó értéket látjuk a GetReq-tel, ezért van minden t-ben az 1. indexre 5, a 2.-ra 9-es érték.

Hogyan lehet egy alkalmazási réteg protokoll statisztikáit gyűjteni RMON2-ben?

RMON2-ben 2 csoport foglalkozik hosztonkénti statisztika gyűjtéssel:
- hálózati réteg hoszt csoport (nlHost)
- alkalmazási réteg hoszt csoport (alHost)

Mindkét csoport adattábláját az nlHost kontroll táblája (nlHostControlTable) vezérli.

alHost: Minden egyes hálózati címen, minden egyes felismert alkalmazási réteg protokollra egy bejegyzés keletkezik az nlHostTable-ben, ha a protocolDirAlHostConfig==supportedOn. Mindkét irányú forgalmat számolja, de csak helyes MAC keretekre. Pl. adott hoszton az MS Mail forgalom

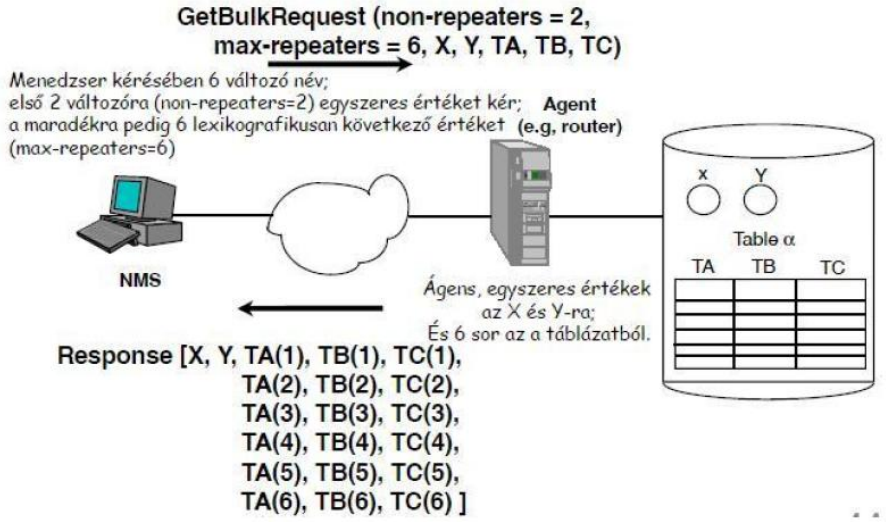
12. SNMPv2

Ismertesse a GetBulkRequest üzenet működését! Rajzoljon példát is!

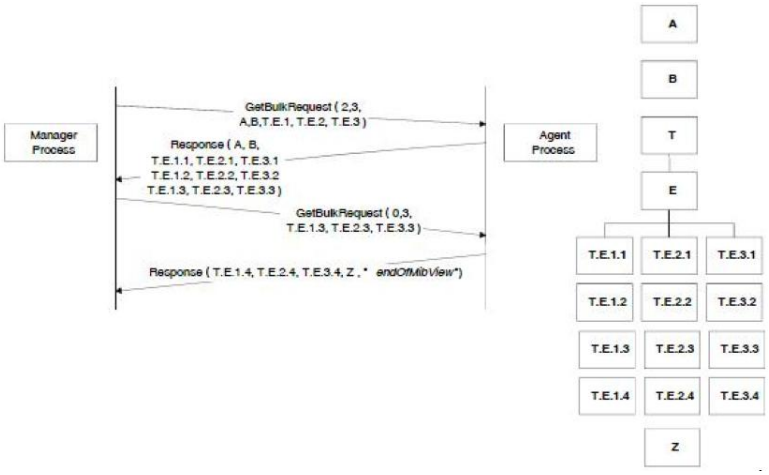
A kérés-válaszok minimalizálásra, ha „nagy” mennyiségű adatot kell lekérdezni

- Az OID hozzárendelés hasonló a GetNextRequest-hez, Lexikografikus következő elemek
- N + R változó név, N egyedi érték lekérdezésre, R többszörös érték lekérdezésére
- non-repeaters és max-repetition, Az N és R jelzésére

Pl.:



Pl2.:



Mi az az Augments? Mire használják? Hogyan?

Oszlopokat lehet meglévő táblákhoz adni és ezzel bővíteni.

Sor hozzáadás, követelmények:

-PDU-nál nagyobb sor létrehozása
-Ágensben nem implementált oszlopok megismerése a menedzsernek
-Az ágensben a menedzser számára el nem érhető oszlopok felderítése
-Menedzserek konkurens sor kezelése
- Sor létrehozásának védelme (sorrendiség)
-tooBig észlelése a végrehajtás előtt
-Egyazon sorban read-only és read-create objektumok létezése
-Ezek mind: fontos vagy hasznos tulajdonságok

Hasonlítsa össze az SNMPv1 és v2 üzenet típusokat! Miben különböznek ezek? Miért?

SNMPv1 PDU	SNMPv2 PDU	Direction	Description
GetRequest	GetRequest	Manager to agent	Request value for each listed object
GetNextRequest	GetNextRequest	Manager to agent	Request next value for each listed object
–	GetBulkRequest	Manager to agent	Request multiple values
SetRequest	SetRequest	Manager to agent	Set value for each listed object
–	InformRequest	Manager to agent	Transmit unsolicited information
GetResponse	Response	Agent to manager or manger to manager (SNMPv2)	Repond to manger request
Trap	SNMPv2-Trap	Agnet to manager	Transmit unsolicited information

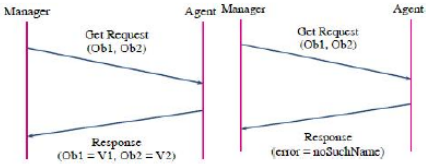
GetRequest: SNMPv2-ben nem atomikus (v1-ben igen).
v2:
Ha hiba történik, és az alábbi listával kezelhető, az OID megkapja ezek értékét: noSuchObject, noSuchInstance, endOfMibView.
A lista (a válasz response PDU előállítása):

1. ha a közölt prefix OID nem illeszkedik egyik hozzáférhető változó prefixre sem, akkor noSuchObject
2. egyébként, ha az OID neve nem illeszkedik egyik változóra sem, akkor noSuchInstance
3. minden más esetben pedig a változó értékével tér vissza

Ha az összeállított válasz meghaladja a lokális maximális méretet, akkor a válasz eldobásra kerül és egy új response PDU gyártódik le:

- error-status=tooBig
- Error index = zero
- variable-binding={}

v1 (emlékeztető egy korábbi kérdésből):
Vagy minden értéket visszaad, vagy egy sem kerül visszaadásra.
Sikeres (bal kép); ha bármelyik objektum nincs implementálva, sikertelen (jobb kép):



GetNextRequest: ugyan az igaz rá, mint a GetRequestre
v2 response PDU szabályok:

1. a lexikografikus sorrendezés alapján OID és érték pár kerül a variable-bindings részbe
2. ha nem létezik lexikografikus következő elem, akkor a válaszban a lekérdezett eredeti változó szerepel endOfMibView értékkel.

GetBulkRequest: SNMPv1-ben még nem volt.
SetRequest: az SNMPv2-ben a v1-hez képest csak a válaszok kezelésében van különbség. v2-ben:

- az ágens meghatározza a válasz méretét a variable-bindings lista és érték alapján
- ha a méret meghaladja a lokális maximálisan küldhető PDU-méretet, akkor a response PDU-ban: error-status=tooBig, error-index=zero v. variable-binding üresre állítva.
- variable-bindings 2 fázisban feldolgozva:
  - 1. lépésben minden OID és érték pár ellenőrizve
  - ha a fenti sikeres, akkor kerül az értékadás megvalósításra

SNMPv1-ben és v2-ben is atomikus.

SNMPv2-Trap: a 2 verzióban a funkciójuk megegyezik (a v1-ben simán "Trap"-nek hívják) és egyikre sem várunk választ.
A formátum változott: egység-formátum a többi SNMPv2 üzenettel (=>egyszerűbb feldolgozás)

InformRequest: a v1-ben még nem volt, a v2-től vezették be.

Response: a v1-ben GetResponse a neve. (Az SNMPv2-nél ez van leírva:) Ezeket a PDU-kat csak a biztonsági kiterjesztések használták, és mivel ezeket törölték a végleges szabványból, ezen PDU használatára nincsenek szabványok. <-- **Ez a Report, nem a Response**

Mire használják az InformRequest PDU-t?

Az InformRequest v2 menedzsertől másik v2 menedzserhez **küldött kérés**, a menedzsment alkalmazás nevében valamilyen információ kinyerésére.
Célja: hierarchikus és elosztott menedzsment támogatása, ahol egynél több menedzsert használnak.

Hogyan lehet új sort létrehozni egy táblázatban SNMPv2-ben?

RMON-nal.

2 fajta tábla van az SNMPv2-ben:

- menedzser által sor hozzáadható és törölhető
- menedzser által nem vezérelhető táblák (teljesen az ágens által vezérelve)

Hogyan alakult ki az SNMPv2-es szabvány csoport? Milyen v1 hiányosságokat javít?

Kialakulás:
- SNMP előnyök
  - egyszerűség (SMI és MIB)
  - gyors implementálhatóság
  - Simple Gateway Monitoring Protocol (SGMP) alapokon, amire rengeteg gyakorlati tapasztalat volt.
- 1988-as alapelvek
  - kettős megközelítés
    - rövidtávra: SNMP
    - hosszútávra: OSI alapú megoldás
      - CMIP over TCP/IP
- kettős megközelítés nem működött
  - SMI és az SNMP MIB-nek az OSI menedzsment részhalmazának kellett volna lennie
    - egyszerű átállást elősegítendő, viszont a komplex, objektum orientált OSI megközelítés nem volt kompatibilis (nem volt használható) a gyors implementációs elvárásoknak megfelelni igyekvő SNMP számára
  - Az OSI alapú megvalósítások késtek, sőt még előrelátható stabil szabványok sem voltak
    - ezzel ellentétben az SNMP-t széles körben használták és támogatták

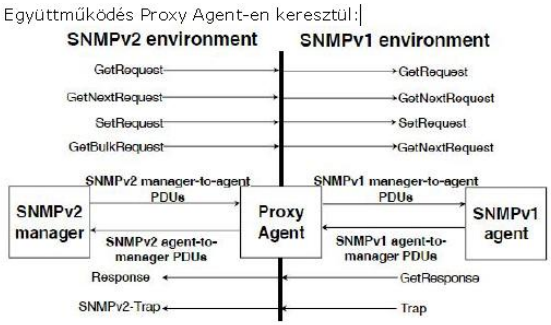
Hiányosságok:
- hálózati méret és komplexitás növekedésből adódóan azonban az SNMP életciklusának végéhez ért
  - SNMP „javítása", hogy további használata biztosítható legyen
- biztonsági támogatás hiánya a
  - menedzser hitelesítése és az üzenetek lehallgathatóságának területén
  - SNMP védtelen az illetéktelen konfigurálás ellen
  - è biztonságos (secure) SNMP, javaslat 1992 július
- teljesítménybeli hiányosságok
  - SMP (Simple Management Protocol) fejlesztése

Fejlesztések 4 kategóriában
- *Scope* :
  - Bármely erőforrás menedzselésére, nemcsak hálózati erőforrásra.
  - SMP alkalmazások menedzsmentje, rendszer menedzsment, menedzser-menedzser kommunikáció
- *Size, speed, and efficiency* :
  - Nagy méretű adatok mozgatására (bulk transfer)
- *Security and privacy* :
  - SMP-be beleágyazni a secure SNMP javításokat
- *Deployment and compatibility* :
  - SNMP-vel együttműködés az SMP funkciók részhalmazán

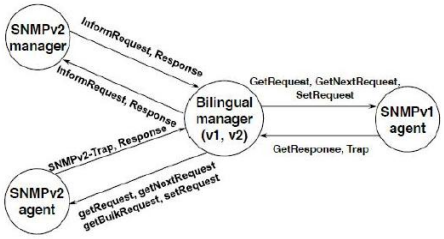
Hogyan működhet együtt egy SNMPv1 és v2-es rendszer? Milyen üzeneteket kell és hogyan átalakítani?

Evolúciós szempontból a visszamenőleges kompatibilitás szükséges: v2 menedzserek v1 ágensekkel. 2 kategóriát kell tekinteni: menedzsment információkat és protokoll működést. SMI különbségek: object definitions, trap def., compliance def., capabilities def. Együttműködésre **3 mód van:**

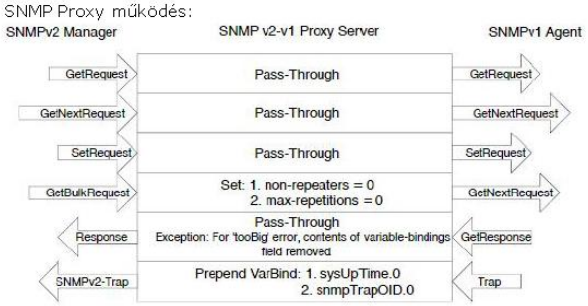
Együttműködés Proxy Agent-en keresztül:



Együttműködés Bilingual Manager:



SNMP Proxy működés:



13. SNMPv3

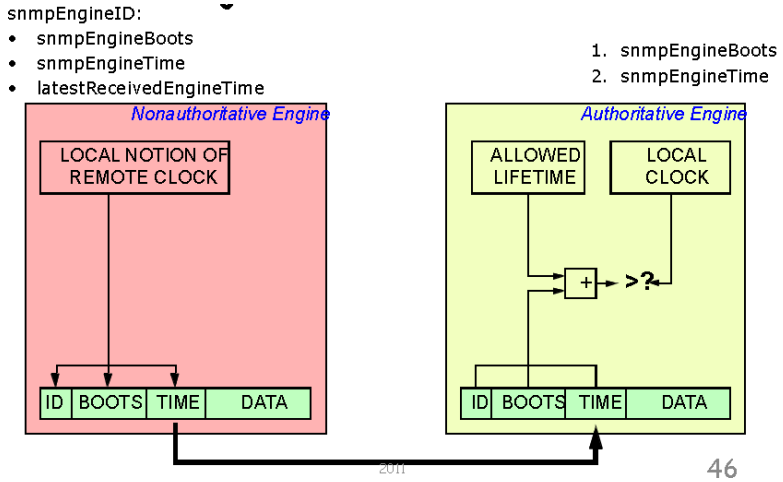
Ismertesse a View-based Access Control Model-t! Milyen elemekből áll (értelmezze)?

Read és write view-k tárolják csoportonként, hogy az adott csoport tagjai miket olvashatnak, ill. írhatnak.

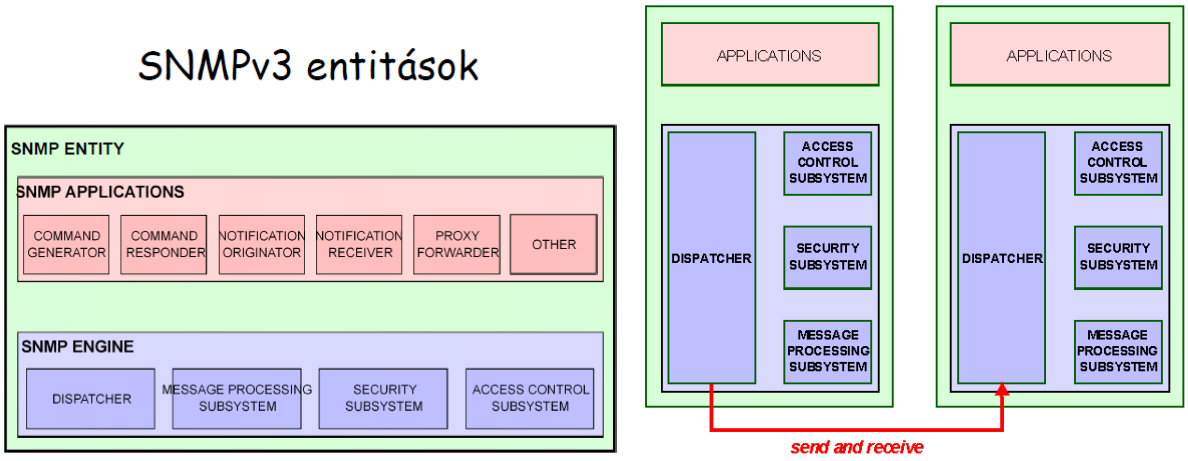
- Groups: felhasználói csoportok
-Security levels: üzenetbiztonság szintje
- Context:

Hogyan működik a visszajátszás védelem (replay protection)? (milyen változókat tartanak nyilván? rajzoljon is!)

Az authoritative engine-nek megfelelő ID, BOOTS, TIME étékek mennek át, ezzel időszinkront teremt.



Rajzolja le az SNMPv3 architektúrát! Részletezze az egyes modulok funkcióját!



Hogyan építhet ki egy manager SNMPv3 biztonságos és titkosított kapcsolatot egy ismeretlen ágenssel? (Discovery)

Két lépésben

1. msgAuthoritativeEngineID meghatározása

non authenticated Request

msgUserName= initial

msgAuthoritativeEngineID=null

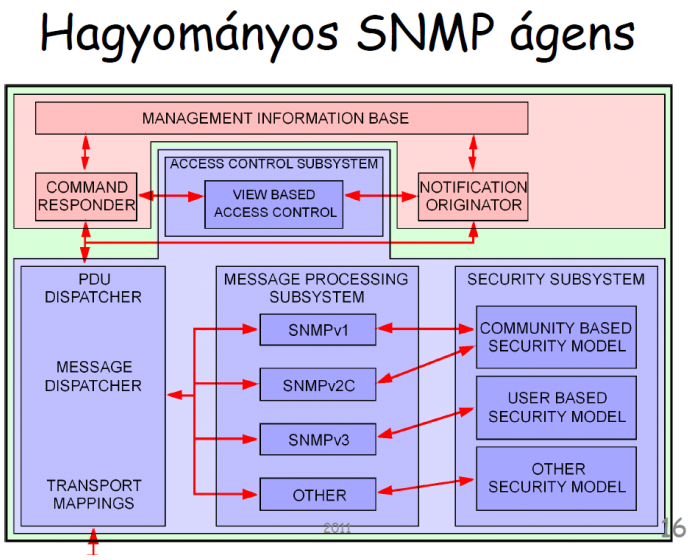
2. establishing time synchronization

authenticated Request

Milyen biztonsági módjai vannak az SNMPv3-nak?

NoAuthNoPriv, AuthNoPriv, AuthPriv

Rajzolja le egy SNMPv3 ügynök (agent) moduljait! Egy-egy mondatban ismertess funkcióikat!



Hogyan kezelik a felhasználói kulcsokat SNMPv3 architektúrában?

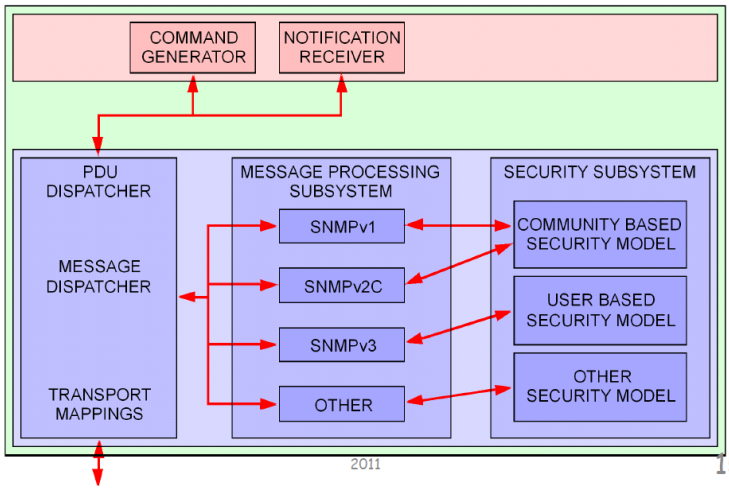
- Követelmények
  - Minden érintettre egyedi
    - § authentication key
    - § encryption key
  - A kulcsok nem elérhetőek az SNMP / MIB rendszerben
  - Kulcsok jelszavakból generálódjanak
- Jelszó – kulcs párosítás
  - concatenation and hashing (MD5)
- Kulcsok
  - Minden felhasználónak minden ágensen külön kulcs
  - Kulcsok a különböző ágenseknél különbözőek
  - A hálózatban bárhonnan lehet menedzselni
  - Jelszóból kulcs előállítása
- Az egyszeres felhasználói kulcsot nem visszafordítható egyirányú függvényekkel (secure hash) különböző helyi kulcsokká transzformáljuk
- Kulcsok frissítése
  - Nem SNMP feladat

Kié az authoritative Engine ID egy SNMPv3 kommunikációban? Miért van rá szükség?

The snmpEngineID of the authoritative SNMP engine involved in the exchange of this message. Thus, this value refers to the source for a Trap, Response, or Report, and to the destination for a Get, GetNext, GetBulk, Set, or Inform.

Rajzolja le egy SNMPv3 menedzser moduljait! Egy-egy mondatban ismertess funkcióikat!

Hagyományos SNMP menedzser



14. MIB modellezés

Milyen osztályokba kategorizálhatjuk a menedzsment információkat? Hogyan viszonyulnak ezek egymáshoz?

Components Menedzselt logikai és fizikai eszközök vagy szolgáltatások - motor

Attributes Jellemezően statikus tulajdonsága a modellezett objektumnak – Lóerő

Actions Vezérlés – ki, be, gyorsít

Statistics Hasznos információ a rendszer múltbéli állapotáról –max fordulatszám

State - Aktuális rendszer állapot RPM, olajnyomás, kibe

Milyen megfontolásokat célszerű figyelembe venni egy MIB model készítésénél? Milyen egyszerű leképzésekkel lehet a modellből MIB-et készíteni?

Fentről-lefelé részletezve

Fizikai/logikai beágyazás? (Mely fizikai eszköz tartalmazza, mely komm. rendszer része)

Kardinalitás: hány helyen fordul elő az adott elem a rendszerben.

Statikus/dinamikus?

Csoportosítható/hierarchiába szervezhető?

Azonosítás módja: véletlenszerű azonosítók/attribútumból származtatva

Nincsenek explicit akciók: Célokkal és lépésekkel oldjuk meg

15. Patterns

Hasonlítsa össze az esemény alapú és a lekérdezéses menedzsmentet előnyök és hátrányok szempontjából? Mikor melyiket használhatjuk? A gyakorlatban hogyan használjuk ezeket együttesen?

Polling (lekérdezés): a menedzser kérésére a kliens válaszol a MIB alapján (konfiguráció begyűjtése, kondíció periodikus lekérdezése).
Előnye: robosztus, hibatűrő. Hátránya: jelentős forgalmat generálhat, időkritikus.
Event reporting (jelentés): kliens által akár periodikusan, akár eseményhez kötve.
Előnye: az eseményhez kötődően azonnali jelzést kínál. Hátránya: a jelentés célba jutásáról vagy annak meg nem történtéről nem értesülünk (nem megbízható)

Lehet-e lekérdezés alapú (polling) hibamenedzsmentet csinálni? Ha igen, mégis milyen indokok miatt használnak inkább jelentés alapút?

Lehet pollinggal is, de **a jelentés alapú kevésbé terheli meg a hálózatot**, kevesebb fölösleges információt (állandó lekérdezések; még akkor is ha sokáig semmi hiba nincs a hálózatban) küld a hálózaton. Ráadásul a **polling „drága**", azért mert sok hálózati elemtől kell lekérdezgetni az állapotukat. Az event alapú további előnye hogy az esemény (hiba) bekövetkeztekor azonnal generálódik a jelentés, nem kell megvárni a következő lekérdezési periódust. (Ráadásul elképzelhető a legrosszabb eset, hogy a probléma mindig a lekérdezések között jelentkezik. Ekkor nyilván a megoldás a lekérdezések sűrítése, de ez csak fokozza a terhelést.)

Mik a problémák a tranzakció kezeléssel menedzsment rendszereknél? Mit csinálnak helyette? Mik ezen alternatívák előnyei és hátrányai?

(**Probléma:** Hálózati management esetén a management műveletek elég nehézkesek, ha tranzakciókezelésről van szó, még akkor is, ha egyetlen eszközről van szó. A **nehézséget az okozza**, hogy számolni kell az időbeni késleltetéssel, a network-control protokollok zavarhatják egymást, a fizikai komponensek meghibásodhatnak, stb.)

Megoldás: A management alkalmazásoknak a fenti esetek előfordulását számításba kell venniük, ezért eléggé bonyolultak lehetnek.
● **Verification** (ellenőrzés): ellenőrzési lépésekre van szükség a konfigurációs műveletek alkalmazása előtt. Az ellenőrzés szintaktikai és szemantikai ellenőrzéseket is magába foglal. (Ezzel növelhető annak a valószínűsége, hogy a működés az elvártnak megfelel majd.)
● **Validation** (érvényesítés): a konfiguráció után ellenőrizni kell, hogy a műveletek hatásai az elvártaknak megfelel-e.)

//Szerintem inkább az alábbira kíváncsiak, de nem tudtam egyértelműen eldönteni: **Probléma**: A rollback műveletek elvégzése a management alkalmazások felelőssége. A rollback sajnos nem (jól) kivitelezhető, mert elbukhat. Vannak olyan műveletek, amelyeket nem lehet visszavonni, mert már van visszavonhatatlan hatásuk.

Megoldás: Amikor a rollback nem lehetséges az a legjobb lépés, ha "roll forward"-ot hajtunk végre, ami azt jelenti, hogy a hálózatot egy jól definiált konfigurációs állapotba visszük.
Előnyök: A hálózat konzisztenciáját helyre lehet állítani.
Hátrányok: A nem az elvártnak megfelelő működés hatásai nem vonhatóak vissza.

Milyen fundamentális kommunikációs minták lehetnek a menedzser és az ágens között?

Manager-initiated request and response, and agent-initiated events.

16. Management protocolls

Hogyan működik a CLI? Miért nem lehet könnyen menedzsment rendszerekhez illeszteni?

Command Line Interface: Humán bevitelre, és érzékelésre van tervezve, így nem olyan robosztus, és jól kereshető, mint azt egy menedzsment interfésztől elvárnánk. Nehézkes, vagy nem lehetséges a parancsok eredményeinek újra-felhasználása, struktúrába rendezése. A megjelenítés is problémát okoz, főleg nagy mennyiségű output adat esetén (screen scraping - MÁTRIX :)) Kérdés-válasz párbeszédeket tesz csak lehetővé, az eseménykezelés alapból nem támogatott.

Mi az a netconf? Mire használható? Milyen technológiát használ? Hogyan? Mi az a datastore?

Network Configuration Protocol. Hálózati konfiguráció-menedzsmentet tesz lehetővé, monitorozást nem! XML alapú, RPC hívásokkal végez műveleteket MIB-eken, vagy azok részein. A datastore egy fájlhoz hasonlóan tárolja az adott MIB-re vonatkozó konfigurációs beállításokat. Műveleteket datastore-okon végezhetünk.

Mi az a syslog? Hogyan működik? Hogyan illeszthető egy menedzsment rendszerbe? Előnyök és hátrányok?

A syslog a UNIX rendszerekből származó mechanizmus, ami lehetővé teszi, hogy a menedzselt objektumok esemény üzeneteket adjanak ki, amik aztán logolva lesznek (minden üzenet egy új bejegyzés), és később menedzsment célokra felhasználhatóak. Előny, hogy így végigkövethetjük egy menedzselt eszköz összes tevékenységét, a CLI-hez hasonlóan “emberi nyelven.” Hátránya, hogy gyengén strukturált (header-body) plaintext, így komoly feladat lehet a parse-olása.

17. Scaling

Hogyan kezelhető a menedzsment rendszer felépítésének komplexitása?

A horizontális (taszkok), és vertikális (rétegek mentén). Taszkok modulokra bontása és egymásra mappelése a bulid komplexitás (menedzsment rendszert fenn kell tartani, fejleszteni, bővíteni) miatt. Rétegek kialakítása felelősségek és megvalósítandó funkciók igényei alapján (számításigényes, időkritikus, tárhely igényes stb.) a runtime komplexitás (lépést kell tartani a hálózat növekedésével).

Miért készítenek hierarchikus menedzsment rendszereket? Milyen módszerekkel lehet hierarchikusan kiszervezni funkciókat?

Lokálisan releváns, kisebb sávszélesség igényű, kompaktabb adathalmazokkal dolgozó rendszerek hozhatóak így létre, valamint előny, hogy a rétegek csak az alattuk, ill. felettük levő rétegekkel kommunikálnak, így a menedzsment információk is hierarchizálva lesznek. **a rétegek kialakításával levehetjük a menedzsment szoftverről az egyszerű, ámde számítás-, és/vagy sávszélesség-igényes feladatokat,** amiket így az alsóbb rétegek szolgáltatnak majd számára, strukturált információként.

Milyen menedzsment stílusokat ismer? Röviden ismertesse ezeket!

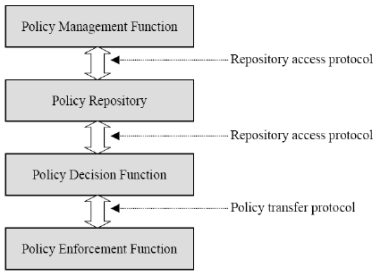
Delegációs menedzsment Feladatok kiosztása a beosztottaknak. Pontosan meghatározzuk, hogy mit tegyenek, és lehetővé tesszük, hogy megtehessék.

Célorientált menedzsment Célokat tűzünk ki a beosztottak elé, és rájuk hagyjuk, hogyan érik el azokat.

Kivételkezeléses (eszkalációs) menedzsment A beosztottak a felelősök/döntéshozók, de amennyiben valami szokatlan történik, és eszkaláció szükséges, akkor mi is belépünk a megoldási folyamatba.

Ismertesse az IETF-es Policy keretrendszer komponenseit, köztük használt protokollokat!

This document articulates the requirements and basic framework of a policy-based management system for IP networks. It focuses on the storage and retrieval of Policy Rules from a repository, for use in the management and operation of IP networks. This framework document describes functional components and operational characteristics of a system that is intended to be device and vendor independent,
interoperable and scalable.
Policy goals
Policy rules
If [some conditions],
then [some actions]
Policy goals can be refined to
policy : rules
More concrete
Policy hierarchies
abstract and higher-
level policies ——————————— implemented by lower-level policies



- IETF
  - key policy components, but not implementation details!
- Architecture
  - Management Tool
  - Policy Repository
  - Policy Decision Point
  - Policy Enforcement Point
- Levels of Abstraction
  - SLA (Service Level Agreements)
  - SLO (Service Level Objectives)
  - Policy (Rules)
  - Configured Device Level

Milyen rétegekben lehet mediálni menedzsment rendszerekben? Mutassa be ezeket példákon keresztül!

- Átviteli réteg UDP → SSH

A netconf SSH UDP keretben nem érkezett meg, a kapcsolat nem épül ki, amin UDP keretek tunneleznének.

- Menedzsment protokoll réteg CLI vs. SNMP get **,** syslog vs. SNMP trap

Szabályok, és template-ek használatával nem mosható össze a két mediált mgmt protokoll.

- Menedzsment információ réteg SNMP V3 trap PDU mappelése syslog üzenetté (szintaxis)**.,** Egyedi fordítási szabályok(szemantika).

Az a gond, hogy a feldolgozás sajnos a menedzsment rendszerre hárul.

18 LDAP

Lightweight Directory Access Protocol

Hogyan kapcsolódik az LDAP a policy alapú menedzsmenthez? Elemezze, hogy mely tulajdonságai miatt esethetett a választás az LDAP-ra?

Policy: szabályok, hogy mit körténnek, amikor bizonyos conditions megtörténnek. Van egy általános egyezmény az eladói communitiyn belül, hogy a policy szabályokat LDAP-könyvtárakban tárolják.

X,500 volt a másik standard, de az túl nagy, bonyolult, nincs benne API. A könyvtár – directory- alapú szolgáltatás network accessible database: kevés infó minden üzenetben, limitáált funkcionalitás adatábzisrendszerekhez képest, és lassabban updatel- változik mint a lekérdezések. A könnyű kezelhetőség, és a tény hogy képes x500 könyvtárakat megnyitni. ezért hozták létre, később önállósult.
LDAP catches on independently of X.500 in the Internet community adding standardized:
Information Model (how the information looks)
APIs (how applications get the information)
Replication (how servers share information)
Access Control (who can see what information)

There is a general agreement within the vendor community that policy information should be stored in a directory LDAP as a lightweight directory access protocol to access X.500 directories. LDAP catches on independently of X.500 in the Internet community adding
standardized:
- Information Model (how the information looks)
- APIs (how applications get the information)
- Replication (how servers share information)
- Access Control (who can see what information)

Van egy általános megegyezés a gyártók között, hogy az irányelvekre vonatkozó információkat címjegyzékekben kell tárolni. A LDAP egy kevés erőforrást használó protokoll a szabványos X.500 címjegyzékek elérésére. Az X.500-tól függetlenül terjedt el az Internetes közösségben, mert szabványosítja az alábbiakat:
- információs modell (hogy néz ki az információ)
- alkalmazásprogramozási felületek (az alkalmazások hogyan szerzik be az infót)
- válaszok (kiszolgálók hogyan osztják meg az infót)
- hozzáférésirányítás (ki láthatja az infót)

Ismertesse az LDAP protokollt (data model, data information tree, attributes, objectClass)!

IDAP egy információs modell , vagy **adatmodell**. elnevezési modell, funkcionális moell, biztonsági modell.
The Lightweight Directory Access Protocol is an application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network. LDAP is defined in terms of ASN.1 and transmitted using BER.
Data modell: objektum hierarchia, operációs primitívek. Objektum = bejegyzés. Root létezik, mindenkinek egy szülője, végtelen gyereke lehet.
Data Information Tree: van egy root, azok alatt bejegyzések, sibling, parent, stb.
Attribute: objectClass tagja, lehet: név, data(data type), optional-mandatory, single-multi
objectClass: attribút halmazok, Must-may attribútumok. Hierarhikus elrendezés. Örököl mindent a szülő osztályából.

Mi az a DN és az RDN az LDAP protokollban?

DN: Distinguished Name: pl example.com
RDN: Relative distinguished name. pl: people, stuff, stb.

Distinguished Name: A DN RDN-ek sorozatából áll, így írja le a kívánt bejegyzéstől a gyökérig az elnevezési tulajdonságokat felfele a Címjegyzék fában. Magyarosabban egy másik doksiból: DN (megkülönböztető név): A DN a bejegyzés nevéből áll, megtoldva a név elérési útjával vissza a címtár hierarchia csúcsáig (mint egy fánál).
Relative Distinguished Name: Egyedi név egy attribútumnak, lehet egyszerű, vagy összetett, az utóbbi esetben egy '+' jellel választjuk el egymástól az attribútumokat. Csak akor van értelme, ha egy DN része.

19 SLA

Service Level Agreement

Ismertesse egy-egy mondatban a SLA szerződések résztvevőinek típusait! Milyen viszonyban lehetnek ezek egymással?

Every Service provider for every customer
Service Provider, Customer, Integrator, Customer Group, User Role, Value Chain

Internet Service Provider:

- Service Provider - Szolgáltató
- Customer - Vevő
- Integrator - Rendszerintegrátor (Cég, aki a rendszer kiépítését végzi, itthon pl TSystems, Delta, Synergon)
- Customer Group - Vevőcsoport
- User Role
- User Group Member Role
- Value Chain

Milyen szolgáltatásokat érdemes megkülönböztetni egy ISP-nek? Miért?

Szolgáltatások:

- Composite service
- Custome Facing Services
- Resource Facing Services
- Service Access Point

Mi a különbség a KPI és KQI között? Miért van ezekre szükség, hogyan használják őket?

KPI: Key Performance Indicator, technikai mértékegység, közvetlenül megmérhető
KQI: Key Quality Indicator: összesített performancia a termének-vagy szolgáltatásnak, jelentősségteljes az ügyfelek számára, KPI-kből áll. Significance, Relevance, Measurability.

Key Performance Indicator

- A KPI is a great tool to measure and control the performance of any given process.
- technical metric
- measured directly

Key Quality Indicator

- Overall perf. of product / service
- Meaningful to customers
- Mix of KPIs

Milyen szempontok szerint válasszunk KQI paramétereket az SLA-hoz? Részletezze a szempontokat! Ahol lehet példával is illusztrálja a gondolatmenetet!

Significance – ugyanabból a rétegből származzon, amiből a szolgáltatást nyújtuk? pl voice szervisnél a tonetime, end-to-end delay, Mean opinion score, stb.
Relevance – mennyire fontos, ne legyen túl sok
Measurability – Tiszta, és megszámlálható definíció az elfogadható szolgáltatás ismérve.

A felhasználó igényéhez kell igazítani őket, az egyes paraméterek valós fontosságát és mérhetőségét figyelembe kell venni, pontos mérést meg kell határozni, nem mindenki tud mindent mérni. A lényeg, hogy a usert nem az érdekli, hogy mekkora a max késleltetés hanem, hogy szakadozik-e a beszélgetése.

Hogyan lehet KQI paramétereket mérni? Mik a jellegzetességei a különböző rétegekre jellemző KPI/KQI paramétereknek mérés és használhatóság szempontjából? MOS

Mit tartalmaz egy SLA szerződés?

Tartalmazza, hogy milyen szolgáltatást, milyen minőségben kell biztosítaniuk. Specifikálja, hogy ezeket milyen módon kell ellenőrizni, valamint nem megfelelő minőség esetén a retorziókat.

- What will be delivered == service level objectives (SLO)
  - Customer SLO
    - What is needed / critical?
    - Qualifications (e.g. Time)
  - Provider SLO
    - Maintenance and upgrades
    - Definition of terms
    - Assumptions (e.g. call holding times)
    - Realistic SLAs
- How to track and verify?
- What if not delivered?

Hogyan lehet SLA szempontjából menedzselni a hálózatot?

Planning:

- Reserve resourcesw for SLAs: dimensioning problem
- Oversubscription Risk

Monitoring Parameters:

- source: mgmt information, SNMP MIB, syslog, neglow, ipfic stb.
- Passive and Active measurments

Preventing Failures:

- Service level forecasts, good tools: Threshold-crossing alerts,

Keeping Record:

- important as a proof of service
- Sometimes service level report are part of the service package
- statistics témakör…

● Packet classification: packet marking needed for router to distinguish between different classes; and new router policy to treat packets accordingly
● Isolation(scheduling and policing): provide protection (isolation) for one class from others
● High resource utilization: While providing isolation, it is desirable to use resources as efficiently as possible
● Call admission: flow declares its needs, network may block call (e.g., busy signal) if it cannot meet needs

Milyen 4 pilléren támaszkodik a szolgáltatás minőség biztosítása a csomagkapcsolt hálózatokban? Hogyan kapcsolódnak ezek egymáshoz? Fejtse ki és érveljen!

- **Packet classification:** packet marking needed for router to distinguish between different classes; and new router policy to treat packets accordingly
- **Isolation(scheduling and policing):** provide protection (isolation) for one class from others
- **High resource utilization:** While providing isolation, it is desirable to use resources as efficiently as possible
- **Call admission:** flow declares its needs, network may block call (e.g., busy signal) if it cannot meet needs

20. COPS

Common Open Policy Service Protocol

Mi az a COPS protokoll? Mire lehet használni?

Kliens-szerver protokoll. Client = PEP, server= PDP
TCP felett fut, megbízható
Policy Információs Bázis, message format
Security: authentication, replay protection and message integrity
Definiál objektumokat, context-objecteket, (eventek) döntéési objektumok, műveletek,
Bináris formátumot használ.

COPS specifies a simple client/server model for supporting policy control over Quality of Service (QoS) signaling protocols (e.g. RSVP). Policies are stored on servers, and acted upon by Policy Decision Points (PDP), and are enforced on clients, also known as Policy Enforcement Points (PEP).

A lényeg, hogy ez egy policy kiosztó/szállító protokoll, ami a PDP és a PEP között van.

21 BGP

Border Gateway Protocol
útvonalválasztási tartomány = **Autonomous System** AS

Részletezze, hogy hogyan működik a BGP protokoll? Miért hívják útvonal vektor protokollnak?

De Fakto szabvány az Interneten, egyszerű protokoll komplex használattal. Policy (szabályrendszer) alapú protokoll. Nem metrika alapú, minden útvonal egyenlő.

Működésekor routerpárok kicserélik a routing információkat BGP sessionök alkalmával. (logikai útvonalakkal foglalkozik, fizikaiakkal nem)

AS-ek közötti elérhetőségi információt terjeszt.
Azért hívják **útvonal-vektor protokollon alapulónak, mert**
- minden útvonalhoz tulajdonságokat, attribútumokat rendel
- explicit útvonalhirdetést és visszavonást visz végbe
- csak változásokat hirdet, állapotokat tart fent, ezáltal lecsökkenti az overheadet.

Prefixek (útvonal) terjesztése. Mindenhez tulajdonságok, csak a változásokat hirdeti. Azért vektor, mert megmondja, hogy milyen AS-eken fog átmenni. Ezalapján a MIN út lesz a preferált.

A BGP 3 listát (Route Information Base, RIB) tart nyilván, egyben azokat az utakat tárolja, melyeket szomszédaitól hallott (RIB-In), a másikban azokat, melyeket terjeszt (RIB-Out), a harmadikban azokat, melyeket az AS használ (local-RIB). Az első két listából minden szomszédos AS-hez tartozik egy-egy, azoknak az utaknak amit onnan hallott és oda terjeszt.
A célpontok prefixek. A BGP implementáció nem IP cím + maszk párosával írja le a prefixeket, hanem egy 1 byte hosszúságú hossz mezővel, mely a prefix bitben mért hosszát adja meg és magának a prefixnek értékes bitjeivel.
Az út-vektorok egyes szakaszai AS-ek sorozatát, mások pedig halmazát írják le. A sorozat egymás után következő AS-eket sorol fel, melyek mindegyikén keresztülhalad a csomag a célpont felé. A halmaz viszont azt jelenti, hogy a felsorolt AS-ek egyikén-másikán halad csak végig a csomag.

Mi a különbség az iBGP és az eBGP között? Melyiket hol használjuk? Melyik milyen attribútumokat változtat meg?

externalBGP
- különböző AS-ek között
- alapértelmezésben **csak közvetlen kapcsolatokon** keresztül kommunikál
- **egy, legjobb útvonalat** terjeszt minden célhoz
- attribútumokat is elküldi**, kivéve** a local preference-t
- **AS-PATH hozzáfűzi** a hirdető AS azonosítóját (ASN)
- **felülírja a next-hop** attribútumot.

internalBGP
- AS-en belül
- nem csak **közvetlenül** összekötött útvonalválasztók között
- **csak saját útvonalakat**, vagy direkt (eBGP hallott útvonalak terjeszthet – full mesh)
- **minden** attribútumot továbbküld
- nem változtatja az AS-PATH és NEXT-HOP attribútumokat

Mutasson egy példát olyan elrendezésre, amikor a küldő nem azt az AS útvonalat látja BGP szinten mint amelyen a csomag ténylegesen halad? Indokolja, hogy miért fordulhat ez elő!

Egy közbenső AS összefogja a prefixeket (1 és 4 --> 1), és egy cím mondjuk a 4. AS-ben van, de az összefogás miatt a feladó azt látja, hogy az 1-esbe megy.

Mi az a forró krumpli (hot potato) útválasztás a BGP esetében? Miért probléma ez? Hogyan lehet ezt kezelni a BGP-ben? Milyen körülmények között jelent ez megoldást?

Hot potato: minél gyorsabban továbbítjuk a csomagot, elég csak a következő állomás címét ismernünk. Azért probléma, mert nagy sávszélességű szolgáltatói gerinchálózatról befuthat a cucc alacsony sávszélességű, előfizeti gerinchálózatba, kis kérésre nagy válasz esetén ez probléma.
Multi-Exit Discriminatior (MED) preferált út az AS-be, ilyenkor azon az úton jön a válassz, **ahol ez az érték kicsi. Csak iBGP esetén!**

Ahogy lehet a csomag menjen át máshoz. Arra továbbítja a csomagot, ahova a leggyorsabban tovább tudja adni. (Égeti a kezét a krumpli! :D) Nem jó a skálázódásnak, a másik oldalon ott pont egy hosszú vagy gyenge link lehet.
Kifelé: MED beállítása, mit preferálunk. Local preference szintén erre. Kifelé AS_PATH hack: saját magunk többször ahol nem akarunk befele akkora forgalmat. Nem kötelező figyelembe venni. Community: szolgáltatók közti megállapodás alapján mehet csak, ez adhat skálázódást, gyakorlatilag azt mondja meg a másik AS-nek, hogy milyen local preference értéket állítson be az adott útvonalra..

Hogyan működik a BGP útvonal kiválasztási mechanizmusa? Milyen lehetőségünk van ezen keresztül befolyásolni a peering és tranzit AS relációkat? A kimenő vagy a bemenő forgalomra van nagyobb befolyása egy AS üzemeltetőjének?

Prioritás az útvonalválasztáskor:
1. Legnagyobb lokális preferencia
2. Legrövidebb ASPATH
3. legkisebb MED
4. iBGP < eBGP
5. Legkisebb IGP cost
6. Legkisebb router ID

Bejövő forgalomra szabályok: kifelé menő útvonalak szűrése, attribútumok módosítása által lehet befolyásolni az útvonalat.
Kimenő forgalomra: bejövő útvonalakat zárni, attribútum állítás szintén.

Egy AS-nek több befolyása van a kimenő forgalomra, mint a bejövőre.

Local pref > legrövidebb PATH > min MED > eBGP az iBGP felett > min költség a next-hop felé > router ID, az előbb leírva a befolyásolás, a kimenőt tudja jobban szabályozni

A BGP döntési folyamat a következő:
● A RIB-In-ből egy helyi függvény alapján kiszámoljuk minden útvonal preferenciáját, a legjobbnak ítélt utakat a belső BGP kapcsolatokon keresztül terjesztjük.
● A belső szomszédainktól ily módon kapott információt hozzácsapjuk sajátunkhoz és ebből az összegzett úthalmazból meghatározzuk minden célponthoz az AS számára rendelkezésre álló legjobb útvonalat, ezeket elhelyezzük a local-RIB-be.
● A local-RIB-ben levő információt aggregáljuk és a helyi politikának megfelelően, a terjeszthető részét áthelyezzük a RIB-out-ba.

22. Peering

Mi a különbség a peering és tranzit AS relációk között? Rajzoljon is!

Peeringért nem fizetsz, tranzitért fizetsz. A peering 2 kb. azonos (egymás hálózatába) irányuló forgalom esetén jöhet létre. Kiépítenek pl. egy dedikált linket. Így kölcsönösen megspórolnak egy csomó tranzit költséget. – nem volt rajz szerintem.

Hogyan állapítják meg a tranzit szolgáltatás díját? Indokolja miért ezt a megoldást választották! Lehet-e "játszani" ezzel az előfizetői oldalról? Ha igen, akkor hogyan?

Sorba rakják a sebesség-adatokat, amiket 5 percenként mérnek. A felső 5%-ot kidobják és ami ekkor marad 95-dik százalék azután kell fizetni. A fel-le forgalom közül a nagyobb után.
Lehet játszani: tudod hogy mikor van általában nagy börsztöd, kötsz több szerződést és szétosztod közöttük a forgalmat, így mindenhol keveset fizetsz.

Miért választanak peer kapcsolatot a szolgáltatók egymással? Mik az előnyei a peeringnek különböző szereplők számára?

Ingyenes adattovábbítás lehetősége.
Jobb QoS lehetőség biztosítása → pl. egy tartalomszolgáltató felé vonal kiépítése.

Hogyan lehet megállapítani, hogy kik a lehetséges peering partnerek? Miért?

Azok akik a forgalmi mennyiség alapú listában a középső részen vannak. A nagyobbak úgysem akarnak peerelni, a kicsikkel meg mi nem akarunk.

Milyen technikai megoldások vannak peering létesítésére? Ismertesse ezeket!

- Közvetlen kapcsolat kiépítés → általában költségek felezése
- Betelepülés egy exchange központba (BIX), ott már közel vannak a szolgáltatók
    - nagy switchen kapcsolat
    - ottani közvetlen “kábel kihúzás”

Hogyan számítják ki, hogy melyik fél mennyit fizet a másiknak a peering kapcsolatban?

Ha aszimmetrikus a forgalom, akkor aki többet forgalmaz az fizet, vagy például ő építi ki és tartja karban a vonalat.

Mi a köze a BGP-nek a peering kapcsolatok létesítéséhez?

Figyelni kell arra, hogy a peer partnertől kapott eBGP adatokat ne adjuk tovább másnak, mert akkor rajtunk keresztül fog routolni hozzá, és így ingyen adtunk tranzit szolgáltatást. Ami nem valami jó :)

23. Yang

Mi az a YANG? Mik az indokok a közelmúltbeli szabványosítása mögött? Mire lehet használni menedzsment környezetben? Hogyan lehet használni? Röviden ismertesse!

Yang: Modeling language. Szemantikát, és adatokat modellez. Konfigurációs adatok, állapotadatok, RPC és jelzések modellezése, leírása. modell: set of rules how to access this data

modell-alapú infrastruktúrája van. Bővíthető, XML tartalmak.

Netconf hibái: state vs config, kulcsértékek, default értékek, szemantikák, RPC-k, jelzések, error üzenetek, túl bonyolult.

Yang értékek: olvasható, limited scope, egyszerű szöveges formátum.

Állításokból áll, modulokból pontosabban, azok állítások lehetnek: Leaf, leaf-list, container, must, augment, stb.

YANG is a data modeling language for the NETCONF network configuration protocol. The YANG data modeling language was developed by the NETMOD working group in the IETF and was published as RFC 6020 in October 2010.
Indokok: Modeling languages such as SMI (SNMP), UML, XML Schema, and others already existed. However, none of these languages were specifically targeted to the needs of configuration management. They lacked critical capabilities like being easily read and understood by human implementers, and fell short in providing mechanisms to validate models of configuration data for semantics and syntax.
Használat: The data modeling language can be used to model both configuration data as well as state data of network elements. Furthermore, YANG can be used to define the format of event notifications emitted by network elements and it allows data modelers to define the signature of remote procedure calls that can be invoked on network elements via the NETCONF protocol.

YANG egy adatmodellezésre használt programnyelv, amelyet a NETCONF hálózati konfigurációs protokoll számára fejlesztettek ki, 2010-ben. A NETCONF által készített adatokat és konfigurációkat tudjuk vele modellezni és szemléltetni.
Indokok: Bár léteznek modellező nyelvek (SMI, UML, XML), egyiket sem speciálisan konfigurációmenedzsmentre találták ki. Nem olvashatók könnyen, és konfigurációs adatok szemantikai és szintaktikai leírásának érvényesítésére sem adnak elég jó módszert.
Használat:

- konfigurációs adatok modellezése
- hálózati elemek adatainak leírása
- hálózati elemek eseményeinek a jelzési üzeneteiknek a formátumát adhatjuk meg vele
- NETCONF-ban készített elemekre definiálhatunk eljáráshívásokat