UNIVERSITATEA POLITEHNICA BUCURESTI

FACULTATEA DE ELECTRONICA, TELECOMUNICATII SI TEHNOLOGIA INFORMATIEI

MASTER INGINERIA INFORMATIEI SI A SISTEMELOR DE CALCUL

RETELE DE CALCULATOARE SI INTERNET

SNMP IN CAZUL DISPOZITIVELOR CE RULEAZA CU UN SISTEM DE OPERARE JUNOS

Profesor coordonator,
Conf. dr.ing. Stefan STANCESCU

Masterand,
ing. Cristina Mihaela TUDOR

BUCURESTI, 2016

CUPRINS

1. GENERALITATI JUNOS OS

Junos face parte din familia de sisteme de operare Unix si a fost dezvoltat de compania Juniper Networks pentru echipamentele de tip router, switch si dispozitive de securitate, create de aceeasi companie. Prima versiune a fost lansata in data de 7 Iulie 1998, iar ultima versiune a fost dezvoltata in data de 20 Noiembrie 2015. Juniper Networks ofera un SDK (Software Development Kit) pentru clienti ce permite o customizare a produsului.

De la prima lansare de Junos OS compania Juniper Networks a lansat versiuni ale sistemului de operare o data la paroximativ 90 de zile. Acest sistem de operare ofera o baza de cod functionala pentru majoritatea platofrmelor Juniper. [1]

Compania Juniper Networks oferta dispozitive de inalta performanta ce creaza un mediu favorabil pentru accelerarea dezvoltarii implementarii servicilor si aplicatiilor intr-o singura retea. Junos OS sta la baza acestor retele de mari performante. Acest sistem de operare este conceput astfel incat sa ofere o disponibilitate crescuta, eficienta operationala si flexibilitate in cadrul retelelor. Junos OS prezinta avantaje prin faptul ca: [2]

- exista un singur sistem de operare
- exista o singura lansare de software
- exista o singura arhitectura de software modular

In cazul Junos OS, spre deosebire de alte sisteme de operare pentru retele ce sunt impartite in mai multe programare restranse sub acelasi nume, acesta este un sistem de operare singular si corerent ce este distribuit pe toate dispozitivele. Acest lucru pentru o usurinta in a dezvolta noi caracteristici ale softwareului o singura data si de a le distribui simultan pe toata linia de productie. De asemenea, un alt avantaj este ca implementarea, facandu-se pe toata linia de productie, se reduc costurile de training necesare pentru a invata diferite unelte sau metode pentru fiecare dispozitiv in parte. Prin acest lucru se reduce si problema interoperabilitatii intre dispozitive. [2]

Juniper Network au elaborat un standard de lansare a noilor versiuni de Junos OS astfel incat acestea sunt lansate in acelasi timp pentru toata linia de productie dupa un calendar bine stabilit. Orice versiune noua trebuie sa includa caracteristicile fiabile ale versiunilor anterioare, iar erorile de regresie trebuie sa tinda catre zero. [2]

Arhitectura software Junos OS este gandita in asa mod incat chiar daca modulele individuale comunica intre ele prin interfete foarte bine definite, fiecare din module ruleaza in spatiul de memorie propriu, astfel prevenind intreruperea unui modul de catre altul, dar si restartarea individuala a fiecarui modul. In acest mod se asigura o performanta de nivel inalt, valabilitate si securitate crescuta, dar si scalabilitate, lucru ce este rar intalnit in cazul altor sisteme de operare similare. [2]

Dispozitivele furnizate de catre Juniper Networks prezinta Junos OS preinstalat. Atunci cand dispozitivele sunt pornite pentru prima data, software-ul porneste automat. Singura configurare necesara este aceea pentru a permite dispozitivului sa participe in retea. Desigur, se poate face upgrade de software atunci cand noi caracteristici sunt adaugate sau cand sunt rezolvate probleme ale versiunii existente. [2]

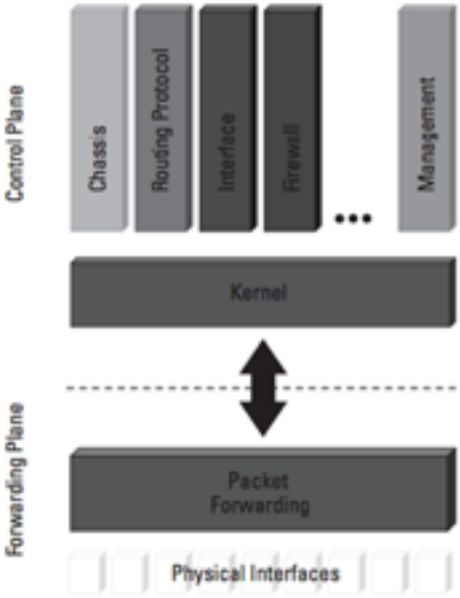

Figura 1.1 Arhictectura software modulara Junos OS.

Arhitectura sotfware a fost impartita in doua plane: planul de control si planul de transport, astfel incat intre cele doua plane sa nu existe impact. Planul de control ruleaza pe motorul de routare al dispozitivelor Juniper; el are urmatoarele functiuni: updatarea tabelului de rutare, raspunde la cerintele SNMP-ului, procesarea traficului SSH sau HTTP, updatarea protocoalelor de rutare, etc. Planul de transport ruleaza pe un motor separat de transport al pachetelor ce foloseste un kernel de dimensiuni reduse contruit special ce contine doar functiile necesare pentru rutare si schimbarea traficului. [3]

2. GENERALITATI SNMP

SNMP (Simple Network Management Protocol) este un protocol pentru layer-ul de aplicatii definit de Internet Architecture Board pentru colectarea si organziarea informatiei despre dispozitivele dintr-o retea. El face parte din suita de protocoale de control al transmisiei si protocoale de internet. Dispozitivele ce folosesc SNMP sunt de obicei routere, switch-uri, servere, statii de lucru, imprimante, modem-uri. [4]

SNMP este foarte des folosit in cazul sistemelor de management al retelelor pentru monitorizarea dispozitivelor atasate la retea. SNMP expune datele de management sub forma unor variabile ce descriu configuratia sistemului. Aceste variabile pot fi interogate de catre aplicatiile de management. [4]

Componentele de baza ale unui SNMP sunt: [4]
- SNMP manager
- dispozitivele administrate
- SNMP agent
- MIB

Managerul SNMP este entitatea responsabila de comunicarea cu agentul SNMP implementand dispozitivele din retea. Functiile de baza sunt: interogarea agentilor, preluarea raspunsurilor de la agenti, setarea variabilelor in agenti. [4]

Dispozitivele administrate sunt acele echipamente (routere, switch-uri, servere, statii de lucru, imprimante) dintr-o retea ce necesita monitorizare si management. [4] Agentul SNMP este un program integrat in elementul de retea. Activand agentul, se permite completarea bazei de date cu informatii despre managementul dispozitivelor locale si face valabile informatiile catre managerul SNMP atunci cand datele sunt cerute. Functiile de baza sunt: colectarea de informatii de management din mediul local, depozitarea si recuperarea informatiilor de management, semnalizarea unui eveniment catre manager. [4]

MIB (Management Information Base) este o baza de date distribuita intre agent si manager prin care managerul cere catre agent informatii specifice. De obicei aceste baze de date contin un set standard de valori statistice si de control definite pentru nodurile hardware intr-o retea. [4]

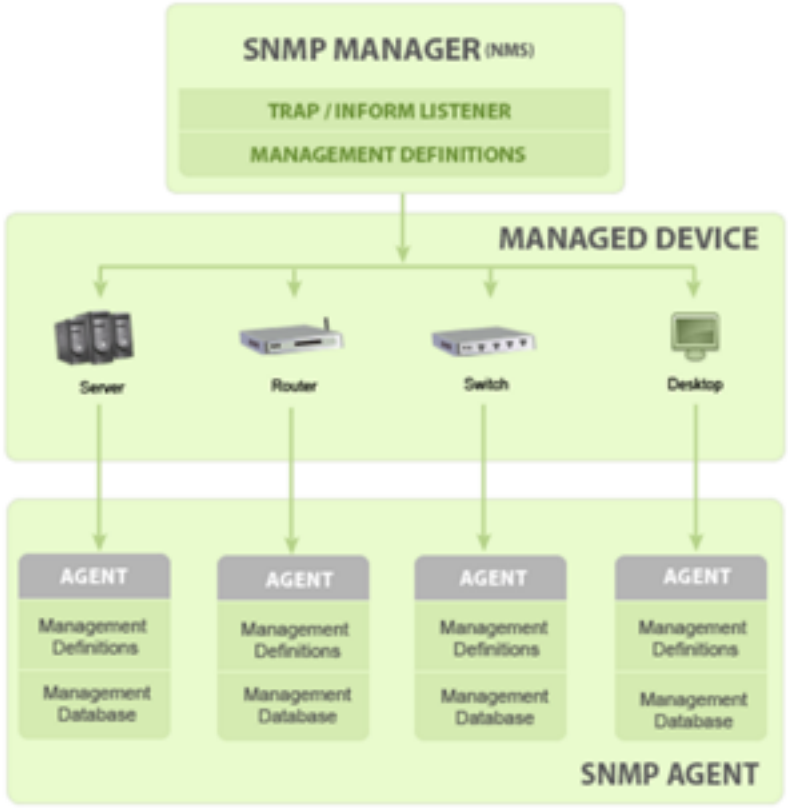

Figura 2.1 Diagrama de comunicare SNMP [4]

Comenzile de baza ale SNMP-ului sunt: [4]

- GET: aceasta operatie este o cerere trimisa de manager catre dispozitivele administrate si primeste inapoi una sau mai multe valori;
- GET NEXT: aceasta operatie este similara cu GET cu diferenta ca primeste valoarea urmatorului identificator de obiect din MIB tree;
- GET BULK: aceasta operatie este folosita pentru a returna date voluminoase din tabele MIB de mari dimensiuni;
- SET: aceasta operatie este folosita de manageri pentru a modifica sau asigna valori dispozitivelor administrate;
- TRAP: aceasta operatie este initiata de agentul SNMP si este un semnal catre manager atunci cand apare un eveniment;
- INFORM: aceasta operatie este similara cu TRAP, cu diferenta ca in acest caz este inclusa si o confirmare din partea managerului la primirea mesajului;
- RESPONSE: acesta operatie este o comanda folosita pentru a aduce inapoi o valoare sau un semnal de actiune directionat de catre manager.

Figura 2.2 Modul de transmitere al mesajelor in cazul comenzilor GET/GET NEXT/GET BULK/SET [4]

Figura 2.3 Modul de transmitere al mesajelor in cazul comenzilor TRAP [4]

Figura 2.4 Modul de transmitere al mesajelor in cazul comenzilor INFORM [4]

3. IMPLEMENTAREA SNMP PE UN DISPOZITIV CE RULEAZA CU SO JUNOS

SNMP-ul permite monitorizarea dispozitivelor din retea dintr-o locatie centrala. O implementare tipica SNMP include urmatoarele componente:[5]

- dispozitivele administrate
- agentul SNMP
- sistemul de management al retelei

Dispozitivele administrate sunt orice echipamente dintr-o retea cunoscute si sub numele de elemente de retea, ce sunt administrate de sistemul de management al retelei. Cele mai cunoscute astfel de sisteme sunt routerele si switchurile.[5] Agentul SNMP face schimb de informatii legate de managementul retelei cu software-ul manager, fie ruland intr-un sistem de management al retelei, fie prin hosting. Agentul raspunde la cererile de informatii si actiuni din partea managerului. Agentul mai este responsabil si de controlul accesului la baza de date de management, ce este o colectie de date ce pot fi vizualizate sau modificate de catre managerul SNMP.[6]

Managerul SNMP colecteaza informatii despre conectivitatea, activitatea, evenimentele retelei prin sondajul facut asupra echipamentelor administrate.[5] Sistemul de management al retelei (NMS) este o combinatie de echipamente hardware si tool-uri software, folosit pentru montorizarea si administrarea retelei.[5] Datele SNMP sunt stocate intr-un format bine structurat si ierarhizat cunoscut sub numele de MIB (Management Information Base). Structura MIB se bazeaza pe o structura de tip “tree”, ce defineste o grupare de obiecte in seturi relationale. Fiecare obiect din MIB are asociat un identificator de obiect (OID). “Frunza” din acest tip de structura este reprezentata de instanta obiectului administrat, adica o resursa, un eveniment sau o activitate ce are loc in cadrul echipamentelor din retea.[5]

In cazul dispozitivelor ce folosesc Junos OS comunicarea dintre agent si manager are urmatoarele forme: [6]

- cereri de tipul get, get bulk, get next
- cereri de tipul set
- notificari de tipul trap si inform

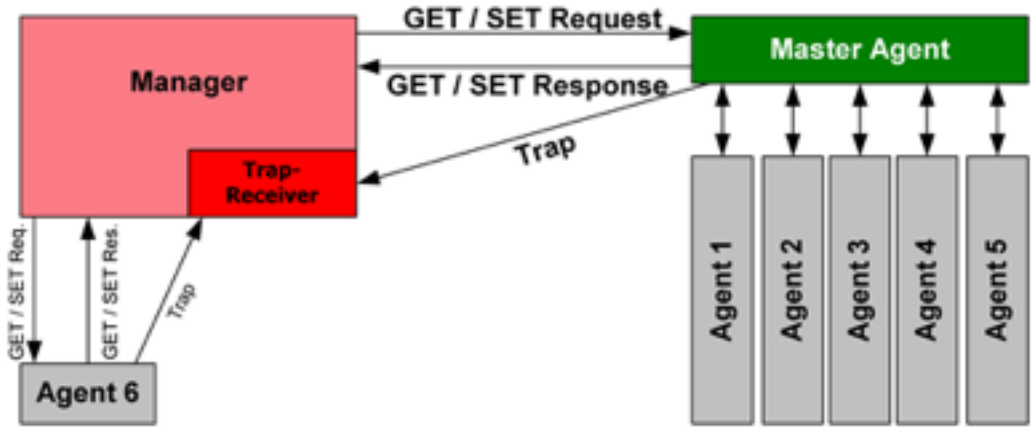

Figura 3.1 Principiul comunicarii SNMP intre manger, agentul master si subagentii [https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol]

Implementarea SNMP in Junos OS suporta doua tipuri de notificari: trap si inform. Trap sunt notificari neconfirmate, pe cand inform au confirmare. Cele din urma sunt suportate numai de echipamentele ce suport configuratie SNMPv3.[5] Junos OS suporta cozile de tipul trap pentru a se asigura ca aceste notificari nu sunt pierdute din cauza indisponibilitatii temporare ale ruterelor. Exista doua tipuri de cozi: cozi destinatie si cozi de accelerare, ce sunt formate pentru a asigura livrarea notificarilor de tip trap si de a controla traficul acestor notificari.[5]

Junos OS formeaza o coada destinatie atunci cand o notificare trap a unei destinatii particulare este returnata datorita faptului ca gazda nu este disponibila. Astfel se adauga in coada notificari trap ulterioare ale aceleasi destinatii.[5] Mecanismul de accelerare este responsabil de controlul numarului de notificari trap (valoarea predefinita este de 500 de notificari) si de a asigura consistenta traficului de notificari, in special in cazul in care un numar mare de notificari sunt generate datorita schimbarii statusului interfetei.[5]

Implementarea SNMP in Junos OS contine:[5]

- un agent SNMP master, cunoscut si sub numele de proces SNMP, aflat pe echipamente si administrat de NMS sau de gazda
- subagenti gazduiti pe diferite module Junos OS, cum ar fi motorul de rutare, administrati de agentul SNMP master
- Junos OS suporta urmatoarele versiuni de SNMP:[6]
- SNMPv1 - implementarea initiala de SNMP ce defineste arhitectura si cadrul de lucru pentru SNMP
- SNMPv2c - protocolul revizuit, cu un aport de imbunatatiri in ceea ce priveste performanta si comunicatiile manager-manager. Aceasta versiune implementeaza comunitatile de stringuri, ce functioneaza ca o parola in cazul accesarii datelor dintr-un agent SNMP de catre un client. Aceste comunitati sunt continute in comenzile get, get bulk, get next si set
- SNMPv3 - protocolul cel mai stabil din punct de vedere al securitatii. Acest protocol defineste un model de securitate USM (User-Based Security Model) si un model de acces al controlului VACM (View-based Control Model). SNMPv3 USM ofera integritatea datelor, autentificarea pe baza origini datelor, protectia reluarii mesajelor si protectie impotriva dezvaluirii incarcatii datelor. SNMPv3 VACM ofera control de acces pentru a determina daca tipul de acces (citire sau scriere) la informatiile de management sunt permise.

Software-ul agent SNMP Junos OS accepta adrese IPv4 si iPv6 pentru transport de tip IPv4 si IPv6. Pentru IPv6, Junos OS are urmatoarele caracteristici:[6]

- date SNMP prin retele IPv6
- date MIB specifice IPv6
- agenti SNMP pentru IPv6

Implementarea SNMP in Junos OS folosesc atat modelul standard (dezvoltat de IETF si documentat in RFC-uri) cat si modelul specific interprinderilor (dezvoltat de vanzatori specifici) de baza de date de management. In Junos OS, datele de management sunt mentinute de catre master SNMP la un nivel si de catre subagenti la nivelul urmator. Mai exista un nivel de date care nu este mentinut nici de agentul master, nici de subagenti. In aceste cazuri, aceste date sunt mentinute de procesele Junos OS ce partajeaza datele cu

subagentii atunci cand sunt interogati despre datele SNMP. Datele despre interfata si datele despre firewall sunt niste exemple de date mentinute de catre procesele Junos OS.[5]

In cazul in care sistemul de management de retea interogheaza agentul master despre date, acesta partajeaza imediat informatiile cu sistemul de management daca datele cerute sunt disponibile pentru agentul master sau pentru unul din subagenti. In cazul in care datele nu sunt disponibile agentului master sau vreunui subagent, atunci subagentii interogheaza kernel-ul Junos OS sau procesul care contine datele respective. Atunci cand datele cerute sunt primite, subagentul transmite raspunsul catre agentul master, care trimite mai departe catre sistemul de management al retelei.[5]

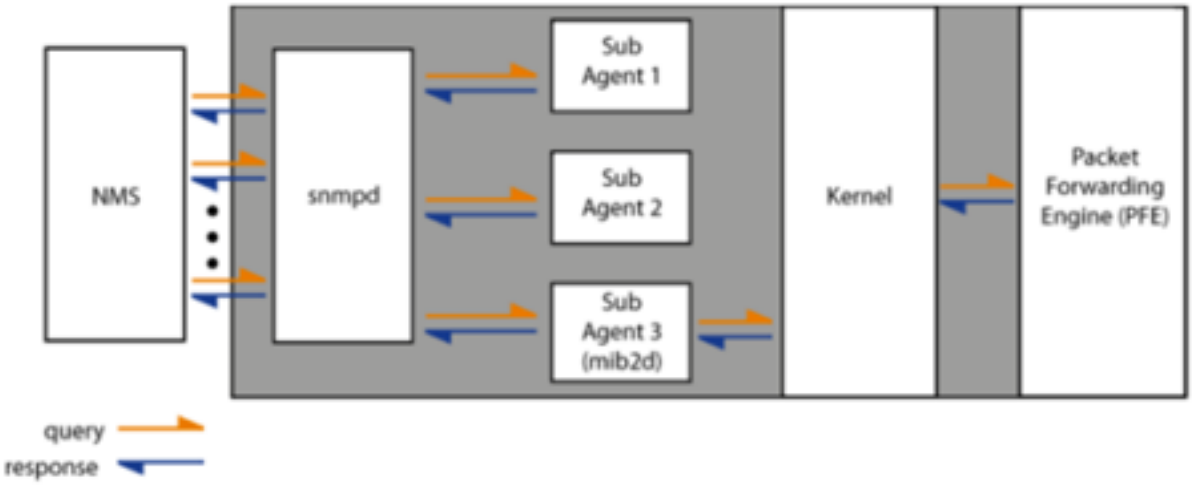

Figura 3.2 Fluxul de comunicatii intre sistemul de management al retelei (NMS), agentul master (snmpd), subagentii SNMP, kernel si motorul de transmitere al pachetelor.[5]

4. SNMP MIB PE UN DISPOZITIV CE RULEAZA CU SO JUNOS

Baza de informatii de management este o ierarhie de informatii folosita pentru a defini obiectele administrate intr-o retea de echipamente. Aceste baze sunt fie de tip standard fie de tip specific. Cele standard sunt create de catre Internet Engineering Task Force si documentate in numeroase RFC-uri. In functie de distribuitor multe MIB-uri standard sunt livrate impreuna cu software-ul de sistem de management al retelei. MIB-ul standard se poate downloada de pe website-ul IETF si compilat in NMS. MIB-urile specifice sunt dezvoltate de fabricanti de echipamente specifice. Daca reteaua contine echipamente ce au astfel de MIB-uri, acestea trebuiesc obtinute de la fabricant si complicate in software-ul de management al retelei.[7]

4.1. Incarcarea fisierelor MIB intr-un sistem de management al retelei

Pentru ca sistemul de management al retelei sa poata identifica si intelege obiectele MIB folosite de Junos OS trebuiesc incarcate mai intai fisierele MIB folosind compilatorul MIB. Acest compilator este un utilitar ce parseaza informatia MIB cum ar fi numele obiectului MIB, ID-urile, tipurile de date, etc. [7]

Pachetul Junos MIB contine doua foldere: StandardMibs si JuniperMibs. Primul folder contine MIB-uri standard si RFC-uri care sunt suportate pe dispozitive unde ruleaza Junos OS, pe cand cealalalt folder contine MIB-urile specifice create de Juniper Networks. [7]

Pentru a incarca fisierele MIB necesare monitorizarii si administrarii echipamentelor ce folosesc Junos OS se parcurg urmatorii pasi: [7]

1. Se acceseaza pagina Junos OS Enterprise MIBs
2. Se descarca fisierul ZIP
3. Se decompreseaza fisierul
4. Se incarca fisierele MIB standard intr-o ordine prestabilita, deoarece este posibil sa existe dependente ce cer un MIB particular prezent in compilator atunci cand se incarca un nou MIB
5. Se incarca fisierele MIB specifice

MIB standard suportate pe echipamentele ce ruleaza cu Junos OS

MIB/RFC	Platforma								
	ACX	M	T	MX	EX	PTX	SRX		
							Low End	Mid Range	High End
IEEE 802.1ab section 12.1, Link Layer Discovery Protocol (LLDP) MIB				x	x				
IEEE, 802.3ad, Aggregation of Multiple Link Segments		x	x	x	x	x	x	x	x
IEEE, 802.1ag, Connectivity Fault Management				x					
IEEE,802.1ap, ManagementInformationBase (MIB) definitions for VLAN Bridges				x					
RFC 1155, Structure and Identification of ManagementInformation forTCP/ IP-basedInternet	x	x	x	x	x	x	x	x	x
RFC 1157, A Simple Network Management Protocol (SNMP)	x	x	x	x	x	x	x	x	x
RFC 1195, Use of OSI IS-IS for Routing in TCP/IP and Dual Environments	x	x	x	x	x	x	x	x	x
RFC 1212, Concise MIB Definitions	x	x	x	x	x	x			x

RFC 1213, ManagementInformationBase for Network Management of TCP/IP-Based Internets: MIB-II	x	x	x	x	x	x			x

Tabelul 4.1 Cateva exemple de MIB standard suportate de echipamente de ruleaza cu Junos OS[7]

MIB specifice suportate de Junos OS

Junos OS suporta urmatoarele MIB-uri specifice: [7]

- AAA Object MIB - ofera suport pentru monitorizarea autentificarii userilor, autorizatia si contabilizare prin RADIUS, LDAP, SecurID si serverele de autentificare locala;
- Access Authentification Object MIB - ofera suport pentru monitorizarea autenficarii firewall;
- Alarm MIB - ofera suport pentru alarma de la router;
- Analyzer MIB - contine un analizator si date de analiza;
- Antivirus Object MIB - ofera informatii despre motorul antivirus si scanarea antivirus;
- ATM Class-of-Service MIB - ofera suport pentru monitorizarea modului asincron de transfer;
- BGP4 V2 MIB - contine obiecte folosite la monitorizarea calculelor BGP;
- BFD MIB - ofera suport pentru monitorizarea sesiunilor BDF;
- Configuration management MIB - ofera notificari pentru configurarea schimbarilor prin trap-uri SNMP. Fiecare trap contine timpul la care configuratia s-a schimbat, numele user-ului care a facut schimbarea si metoda prin care configuratia s-a schimbat. Se pot memora cel putin 32 de modificari;
- Destination Class Usage MIB - ofera suport pentru monitorizarea numaratorilor de pachete bazandu-se pe punctele de intrare si iesire pentru traficul care tranziteaza reteaua. Punctele de intrare sunt identificate prin interfetele de intrare. Punctele de iesire sunt identificate prefixele destiantiilor grupate in unul sau mai multe seturi, cunoscute sub numele de clase de destinatie. Numaratorul este administrat prin interfata prin clasa destinatie, pana la un numar de 16 numaratoare per interfata;
- DHCP Objects MIB - ofera suport SNMP pentru server local DHCP si configuratiile aferente;
- DHCPv6 MIB - ofera suport SNMP pentru server local DHCPv6 si configuratiile aferente;
- digital optical monitoring MIB - ofera suport pentru cererile SNMP Get pentru statistica si notificarile SNMP Trap pentru alarma;
- DNS Object mib - ofera suport pentru monitorizarea cozilor, cererilor, raspunsurilor si esecurilor de tip proxy DNS;
- Dynamic Flow Capture mib - ofera suport pentru monitorizarea statusului operational al capturilor dinamice de flux;
- Event mib - defineste un trap generic ce poate fi generat folosind un script on sau o polita de eveniment. Acest mib ofera posibilitatea de a specifica un string log de sistem si de a genera un trap daca acel string log de sistem este gasit.
- firewall mib - ofera suport pentru monitorizarea numaratoarelor de filtre firewall;
- flow collection services mib - ofera statistica pentru fisiere, inregistrari, memorie, FTP si declaratii de erori pentru o interfata de monitorizare a servicilor. De asemenea mai oferta trap-uri SNMP pentru destinatii ce nu sunt valabile, transferuri de fisier ce nu au avut succes, supraincarcare de flux si supraincarcare de memorie;
- IDP object mib - oferta suport pentru monitorizarea cozilor, cererilor, raspunsurilor si esecurilor SNMP IDP;
- interface mib - extinde standardul ifTable (RFC 2863) cu statistica aditionala si informatii specifice juniper Networks;

- IP forward mib - extinde stanrdardul IP Forward table MIB (RFC 4292) pentru a include informatii CIDR;
- IPv4 MIB - ofera informatii suplimentare despre adresa IPv4 si suporta asignarea unor adrese IPv4 identice pe interfete separate;
- License mib - extinde suportul SNMP la informatiile despre licenta si introduce trap-uri SNMP ce alerteaza clientul cand licenta este aproape de a expira sau cand numarul total de useri depaseste numarul maxim de useri specificat in licenta;
- logical system mib - extinde suportul SNMP la profilele de securitate ale sistemului prin diverse mib-uri defiite sub jnxLsysSecurityProfile;
- MVPN MIB - contine obiecte ce activeaza managerul SNMP pentru a monitoriza conexiunile MVPN;
- NAT objects mib - ofera suport pentru monitorizarea traducerilor adreselor de retea;
- OTN interface management mib - defineste obiecte pentru administrarea interfetelor retelelor optice de transport pe dispozitivele ce ruleaza cu junos OS;
- Packet forwarding engine mib - ofera statistice despre notificari pentru motorul de transport al pachetelor;
- ping mib - extinde standardul ping mib control table (RFC 2925); obiectele din acest mib sunt create atunci cand sunt create intrari in pingCtltable din Ping MIB;
- PPPoE MIB - ofera suport SNMP pentru informatiile legate de PPPoE cum ar fi tipul autentificarii folosite, caracteristicile interfetei, status si informatii statistice;[7]

5. CONFIGURATIE SNMP PE UN DISPOZITIV CE RULEAZA CU SO JUNOS

In mod uzual, SNMP nu este activat pe dispozitivele ce ruleaza cu SO Junos. Pentru a activa SNMP-ul pe un router sau un switch trebuiesc incluse declaratii de configurare SNMP la nivelul ierarhic [edit snmp].[8]

Pentru a configura cerintele minime SNMP, trebuie inclusa urmatoarea declaratie: [8]

```
[edit]
snmp {
community public;
}
```

Comunitatea declarata ca si publica permite accesul de a citi toate datele MIB de catre orice client. Astfel se activeaza operatiile SNMPv1 si SNMPv2 Get si GetNext. [8] Pentru a activa operatiile de tip SNMPv1 si SNMPv2 Set pe un dispozitiv, trebuie inclusa urmatoarea declaratie: [8]

```
[edit snmp]
view all {
oid .1;
}
community private {
```

```
view all;
authorization read-write;
}
```

Exemplul urmator arata configuratia minima necesara pentru SNMPv1 si SNMPv2 pentru trap-uri:[8]

```
[edit snmp]
trap-group jnpr {
targets {
192.168.69.179;
}
}
```

5.1. Configurarea contactului de sistem pentru un echipament ce ruleaza cu Junos OS

Se poate specifica un contact administrativ pentru fiecare sistem administrat de SNMP. Acest nume va fi plasat in obiectul MIB|| sysContact. Pentru a configura numele contactului, se include declaratia “contact” la nivelul ierarhic [edit snmp]: [8]

```
[edit snmp]
contact contact;
```

Daca numele contine spatiu, acesta se va scrie intre ghilimele (“ “), spre exemplu: [8]

```
[edit]
snmp {
contact "Juniper Berry, (650) 555-1234";
}
```

5.2. Configurarea locatiei sistemului pentru un echipament ce ruleaza cu Junos OS

Locatia poate fi specificata in parte pentru fiecare sistem administrat de catre SNMP. Acest string va fi localizat in obiectul MIB || sysLocation. Pentru a configura locatia sistemului, trebuie inclusa declaratie de locatie la nivelul ierarhic [edit snmp]: [8]

```
[edit snmp]
location location;
```

Daca locatia contine spatii, atunci ea se va scrie astfel: [8]

```
[edit]
snmp {
location "Row 11, Rack C";
```

```
}
```

5.3. Configurarea descrierii sistemului pentru un echipament ce ruleaza cu Junos OS

Descrierea sistemului poate fi specificata separat pentru fiecare sistemul din SNMP. Acest string va fi plasat in obiectul MIB|| sysDescription. Pentru a configura descrierea trebuie folosita urmatoarea declaratie: [8]

```
[edit snmp]
description description;
```

5.4. Configurarea string-ului de comunitate SNMP

Stringul de comunitate SNMP defineste relatia dintre un sistem de server SNMP si sistemele de client. Acest string se comporta ca o parola de control al accesului clientilor la server. Pentru a configura stringul de comunitate intr-o configuratie Junos OS, trebuie inclusa urmatoarea declaratie: [8]

```
[edit snmp]
community name {
authorization authorization;
clients {
default restrict;
address restrict;
}
view view-name;
}
```

Nivelul predefinit de autorizatie pentru comunitate este doar de citire. Pentru a seta permisiunea cererilor de tip Set in cadrul unei comunitati, trebuie definita acea comunitate ca autorizata pentru scriere-citire. De asemenea, este nevoie sa fie incluse obiecte specifice MIB ce sunt accesibile cu privilegii scriere-citire folosind declaratia “view”. Declaratia “view” predefinita include suport pentru toate obiectele MIB ce sunt accesibile cu privilegii doar de citire; nici un obiect MIB nu este accesibil cu privilegii de scriere-citire. [8]

Declaratia client listeaza adresele IP pentru clientii care au permisiunea de a folosi aceasta comunitate. Daca nu este prezenta declaratia client, atunci toti clientii au permisiune. Pentru campul adresa trebuie specificata o adresa IPv4 sau IPv6, nu un hostname. [8]

Junos OS permite adaugarea de unul sau mai multe grupuri de clienti la o comunitate SNMP. Pentru a defini o lista de clienti se include urmatoarea declaratie urmata de adresele IP ale clientilor: [8]

```
[edit snmp]
```

```
client-list client-list-name {
ip-addresses;
}
```

De asemenea, se poate configura o lista de prefix la nivelul ierarhic [edit policy options]. Suportul pentru listele de prefix in configuratia de comunitate SNMP face posibila folosirea unei singure liste pentru a configura politicile SNMP si cele de routare. [8]

5.5. Configurarea unui agent proxy SNMP

Incepand cu versiunea 12.3, Junos OS permite asignarea unui dispozitiv in retea ca fiind agent proxy SNMP prin care sistemul de management al retelei poate interoga alte echipamente din retea. in cazul unei configurari proxy, se pot specifica numele echipamentului care va fi agent proxy SNMP: [8]

```
     proxy proxy-name{
device-name device-name;
logical-system logical-system {
routing-instance routing-instance;
}
routing-instance routing-instance;
<version-v1 | version-v2c> {
snmp-community community-name;
no-default-comm-to-v3-config;
}
version-v3 {
security-name security-name;
context context-name;
}
}
```

Declaratia proxy permite specificarea unui nume unic pentru configurarea proxy. Declaratiile version-v1, version-v2c si version-v3 permit specificarea versiunii de SNMP. Declaratia e no-default-comm-to-v3-config este optionala, iar atunci cand este inclusa ea trebuie configurata manual. Daca ea nu este inclusa, atunci declaratia [edit snmp v3 snmp-community community-name] si d [edit snmp v3 vacm] vor fi initializate automat. [8]

5.6. Configurarea timer-ului commit delay

In momentul cand un router sau un switch primesc pentru prima data o cerere non-volatila Set SNMP, o sesiune de protocol XML Junos OS se deschide si previne alti utilizatori sau

aplicatii sa schimbe configuratia candidata. Daca routerul nu primeste noi cereri SNMP Set timp de 5 secunde, configuratia candidata este operata si sesiunea de protocol XML se inchide. Daca routerul primeste totusi noi cereri SNMP Set in timp ce configuratia candidata este operata, cererile SNMP Set sunt respinse si se genereaza o eroare. Daca routerul primeste cererile in timp de 5 secunde, atunci timer-ul se reseteaza la 5 secunde. [8]

Timer-ul este predinit setat la 5 secunde. Pentru a il configura de include urmatoarea configuratie: [8]

```
[edit snmp nonvolatile]
commit-delay seconds;
```

5.7. Configurarea optiunilor si grupurilor SNMP Trap

Unele dispozitive au mai mult de un receiver trap ce transmite trap-urile catre un NMS central. Un dispozitiv ce ruleaza cu junos OS poate fi configurat pentru a trimite aceeasi copie a trap-ului SNMP catre toate receiverele trap configurate in grupul trap. [8] Adresa sursa din header-ul IP al fiecarui pachet trap SNMP este setata la adresa interfetei de iesire. Atunci cand un receiver trap transmite un pachet catre NMS-ul central, adresa sursa este pastrata. NMS-ul (verificand doar adresa sursa a fiecarui pachet trap SNMP), considera ca fiecare trap SNMP vine din surse diferite. [8]

In realitate, trap-urile SNMP vin din acelasi router, dar fiecare pleaca pe o interfata de iesire diferita din router. [8] Pentru configurarea optiunilor trap-ului SNMP si a grupurilor trap se includ declaratiile trap-options si trap-group: [8]

```
[edit snmp]
trap-options {
agent-address outgoing-interface;
source-address address;
}
trap-group group-name {
categories {
category;
}
destination-port port-number;
targets {
address;
}
version (all | v1 | v2);
}
```

Adresa sursa a pachetelor trap se poate configura in unul din formatele: [8]

- adresa IPv4 valida cofigurata pe una din interfetele routerului

- un nume logic de sistem
- un nume instanta de rutare

Adresa agent este valabila doar in pachetele trap SNMPv1. [8]

6. CONFIGURATIE SNMPV3 PE UN DISPOZITIV CE RULEAZA CU SO JUNOS

Spre deosebire de versiunile de SNMP 1 si SNMP 2, versiunea SNMP 3 ofera suport pentru autentificare si criptare. Versiunea 3 foloseste modelul de securitate bazat pe client (USM) pentru securitatea mesajelor si modelul de control al accesului bazat pe vizualizare(VACM) pentru controlul accesului. [8]

USM foloseste conceptul unui user pentru care parametrii de securitate sunt cnfigurati atat pentru agent cat si pentru manager. Mesajele trimise folosind USM sunt mai bine protejate decat mesajele trimite prin string-uri de comunitate, unde parolele sunt vizibile. Prin USM schimbul de mesaje intre manager si agent poate avea verificare de integritate a datelor si autentificare a originilor datelor. USM protejeaza impotriva intarzierii mesajelor si reluari mesajelor folosind indicatori de timp si ID-uri de cereri. [8]

VACM este un model de mare granularitate de control al accesului pentru aplicatii SNMPv3. Bazandu-se pe conceptul de aplicare politici de securitate numelui grupului ce interogheaza agentul, agentul decide daca grupul are sau nu permisiunea de a vedea sau a schimba obiectele MIB. VACM defineste colectiile de date, grupurile de useri de date, si declaratiile de acces ce definesc care grup poate citi, scrie sau primi un trap. [8]

Trap-urile in SNMPv3 sunt create configurand notificarile, filtrul de notificari, adresa target, si parametrii target. [8] Pentru a configura cerintele minim pentru SNMPv3, trebuie inclusa urmatoarea declaratie la nivelul ierarhic [edit snmp view-name]: [8]

```
[edit snmp]
view view-name {
oid object-identifier (include | exclude);
}
[edit snmp v3]
notify name {
tag tag-name;
}
notify-filter profile-name {
oid object-identifier (include | exclude);
}
snmp-community community-index {
security-name security-name;
}
target-address target-address-name {
address address;
target-parameters target-parameters-name;
}
target-parameters target-parameters-name {
notify-filter profile-name;
parameters {
```

```
message-processing-model (v1 | v2c | v3);
security-level (authentication | none | privacy);
security-model (usm | v1 | v2c);
security-name security-name;
}
}
usm {
local-engine {
user username {
}
}
}
vacm {
access {
group group-name {
(default-context-prefix | context-prefix context-prefix){
security-model (any | usm | v1 | v2c) {
security-level (authentication | none | privacy) {
notify-view view-name;
read-view view-name;
write-view view-name;
}
}
}
}
}
security-to-group {
security-model (usm | v1 | v2c) {
security-name security-name {
group group-name;
}
}
}
}
```

In mod predefinit, ID-ul motorului local foloseste adresa IP a ruterului. Pentru fiecare user SNMPv3 se poate specifica username, tipul autentificarii, parola, tipul de privacy, si parola de privacy. In mod predefinit, criptarea nu este deloc setata. [8]

7. EXEMPLE COMENZI SNMP

Comanda SNMPGET [9]

snmpget -mALL -v1 -cpublic snmp_agent_Ip_address sysName.0

Aceasta comanda returneaza un nume administrativ asignat nodului administrat. In mod conventional, acesta este numele de domeniu complet calificat pentru acel nod. In cazul in care numele nu este cunoscut, valoarea returnata este un string de lungime zero. [9]

Exemplu: [9]

snmpget -v2c -cprivate -mALL snmp_agent_Ip_address sysName.0 sysObjectID.0 ilomCtrlDateAndTime.0

SNMPv2-MIB::sysName.0 = STRING: SUNSPHOSTNAME
SNMPv2-MIB::sysObjectID.0 = OID: SUN-ILOM-SMI-MIB::sunILOMSystems
SUN-ILOM-CONTROL-MIB::ilomCtrlDateAndTime.0 = STRING: 2007-12-10,20:33:32.0

Pe langa obiectul sysName.0, aceasta comanda afiseaza si continutul obiectelor sysObjectID. 0 si ilomCtrlDateAndTime.0. [9]
sysObjectID este identificatorul autoritativ al subsistemului de management de retea continut in entitate. Valoarea este alocata in cadrul sub-arborelui SMI si ofera un inteles usor si fara ambiguitate asupra a sistemului ce este adnimistrat. [9]
ilomCtrlDateAndTime seteaza data si timpul dispozitivului ce este administrat. [9]

Comanda SNMPWALK

Atunci cand dorim sa scrie o aplicatie de monitorizare SNMP, indiferent de exista sau nu a unui MIB, cea mai buna modalitate de pornire este prin a "plimba" dispozitivul si de a intreba agentii ce valori pot oferi ei. Astfel, putem stii sigur ce anume se va raporta unde si cum vrem sa integram acest lucru in cod. [9]
Unealta SNMPWALK poate plimba arborele OID bazandu-se pe un OID de inceput sau fara nici un OID care doar returneaza OID-urile MIB-II. [9]

Exemplu: [9]
snmpwalk -v1 -c public 10.10.1.224

SNMPv2-MIB::sysDescr.0 = STRING: APC Web/SNMP Management Card
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.318.1.3.7
SNMPv2-MIB::sysUpTime.0 = Timeticks: (47372422) 5 days, 11:35:24.22
SNMPv2-MIB::sysContact.0 = STRING: Ben Rockwood
SNMPv2-MIB::sysName.0 = STRING: APC-3425
SNMPv2-MIB::sysLocation.0 = STRING: 3425EDISON
SNMPv2-MIB::sysServices.0 = INTEGER: 72
IF-MIB::ifNumber.0 = INTEGER: 1
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifDescr.1 = STRING: veya
.....
SNMPv2-MIB::snmpOutGetResponses.0 = Counter32: 338
SNMPv2-MIB::snmpOutTraps.0 = Counter32: 0
SNMPv2-MIB::snmpEnableAuthenTraps.0 = INTEGER: 0

Aceasta comanda returneaza informatii despre sistem, interfete si grupuri SNMP. [9]

Exemplu comanda fara MIB: [9]

snmpwalk -v1 -c public 10.10.1.224 .1.3.6.1.4.1.318

SNMPv2-SMI::enterprises.318.1.1.1.1.1.1.0 = STRING: "Silcon DP340E"
SNMPv2-SMI::enterprises.318.1.1.1.1.1.2.0 = STRING: "UPS_IDEN"
SNMPv2-SMI::enterprises.318.1.1.1.1.2.1.0 = STRING: "314.10.D"
...

In acest exemplu se cere utilitarului SNMPWALK sa plimbe toate OID-urile pornind de la OID-ul de baza .1.3.6.1.4.1.318. SNMPWALK va returna toate OID-urile. [9]

Exemplu comanda cu MIB: [9]

snmpwalk -v1 -c public -m "./APC-POWERNET.txt" 10.10.1.224 apc

PowerNet-MIB::upsBasicIdentModel.0 = STRING: "Silcon DP340E"
PowerNet-MIB::upsBasicIdentName.0 = STRING: "UPS_IDEN"
PowerNet-MIB::upsAdvIdentFirmwareRevision.0 = STRING: "314.10.D"
.....

Se suplimenteaza comanda cu optiunea -m ce specifica MIB-ul. [9]

Comanda SNMPBULKWALK

snmpbulkwalk -mALL -v2c -cprivate snmp_agent_Ip_address entPhysicalTable>time7

Aceasta comanda foloseste trasaturile protocolului GETBULK SNMP pentru a interoga un arbore intreg despre informatii legate de entitatea retea. Aceasta comanda poate impacheta mai multe obiecte in pachete speficicand "repetorii". Aceasta comanda este mai rapida decat SNMPWALK. [9]

Comanda SNMPTABLE

snmptable -mALL -v2c -cprivate snmp_agent_Ip_address sysORTable [9]

SNMP table: SNMPv2-MIB::sysORTable
sysORID
sysORDescr
sysORUpTime
IF-MIB::ifMIB
The MIB module to describe generic objects for network interface sub-layers.
0:0:00:00.01
SNMPv2-MIB::snmpMIB
The MIB module for SN MPv2 entities.
0:0:00:00.02
TCP-MIB::tcpMIB

The MIB module for managing TCP implementations.
0:0:00:00.02
RFC1213-MIB::ip
.....

Aceasta comanda returneaza continutul unui tabel SNMP si afiseaza continutul in format tabular, adica cate un rand de tabel astfel incat afisarea sa fie asemanatoare tabelului. [9]

Comanda SNMPSET

Desi din punct de vedere sintactic comenzile SNMPSET si SNMPGET sunt similare, ele sunt totusi diferite. Comanda SNMPGET citeste valoarea ID-ului de obiect specificat, pe cand comanda SNMPSET scrie valoarea specificata ID-ului de obiect. Pe langa valoarea ce va fi scrisa in ID-ul obiectului, trebuie specificat si tipul datei al ID-ului in comanda SNMPSET deoarece obiectele snmp suporta mai mult de un tip de date. [9]

Exemplu SNMPSET impreuna cu SNMPGET [9]

% snmpget -mALL -v2c -cprivate snmp_agent_Ip_address ilomCtrlHttpEnabled.0
SUN-ILOM-CONTROL-MIB::ilomCtrlHttpEnabled.0 = INTEGER: false(2)
% snmpset -mALL -v2c -cprivate snmp_agent_Ip_address ilomCtrlHttpEnabled.0 i 1
SUN-ILOM-CONTROL-MIB::ilomCtrlHttpEnabled.0 = INTEGER: true(1)
% snmpget -mALL -v2c -cprivate snmp_agent_Ip_address ilomCtrlHttpEnabled.0
SUN-ILOM-CONTROL-MIB::ilomCtrlHttpEnabled.0 = INTEGER: true(1)

Prima comanda SNMPGET verifica valoarea curenta a obiectului MIB. A doua comanda, SNMPSET, schimba valoarea obiectului MIB. A doua comanda SNMPGET verifica daca obiectul MIB a fost intr-adevar schimbat la valoarea ceruta. [9]
Daca se incearca executia unei comenzi SNMPSET folosind o comunitate publica in locul uneia private, aceaasta comanda nu va merge. Acest lucru se intampla din cauza faptului ca comunitatile private au permisiuni de scriere, dar cele publice nu au. [9]

Comanda SNMPTRAPD

SNMPTRAPD este o aplicatie SNMP ce primeste si inregistreaza trap-uri SNMP si mesaje Inform. Inainte ca sistemul sa poata sa primeasca astfel de mesaje, trebuie configurat trap daemon-ul pentru a putea asculta astfel de mesaje. [9]

Pentru configurarea trap daemon-ului trebuiesc urmariti pasii: [9]
1. Configurarea unei destinatii SNMP trap
2. Pornirea aplicatiei de receptie trap, snmptrapd
3. Generarea unui trap de test pentru a verifica daca trap-urile sunt trimise de agent si receptionate de receiver-ul trap.

Comanda de configurare a daemon-ului snmptrapd: [9]

snmpset -mALL -v2c -cprivate snmp_agent_Ip_address **ilomCtrlAlertSeverity.1 i 2 ilomCtrlAlertType.1 i 2 ilomCtrlAlertDestinationIP.1 a** dest_Ip_address

SUN-ILOM-CONTROL-MIB::ilomCtrlAlertSeverity.1 = INTEGER: critical(2)
SUN-ILOM-CONTROL-MIB::ilomCtrlAlertType.1 = INTEGER: snmptrap(2)
SUN-ILOM-CONTROL-MIB::ilomCtrlAlertDestinationIP.1 = IpAddress: dest_Ip_address

Comanda de pornire a trap-ului daemon: [9]

snmptrapd -mALL -Lo -f -t -OvQ -e -F "%H.%J.%K:%W:%w %q from %A:%V,% %v\n"
2007-11-29 13:21:07 NET-SNMP version 5.2.3 Started.

Comanda de testare a daemon-ului trap: [9]

set /SP/alertmgmt/rules testalert=true
SUN-ILOM-CONTROL-MIB::ilom.103.2.1.20.0 = STRING: "This is a test trap"

8. CONCLUZII

In aceasta lucrare a fost analizata configuratia SNMPv1 si SNMPv3 pentru dispozitivele ce ruleaza cu sistem de operare Junos OS. SNMPv1 a fost prima versiune de protocol suportata pe astfel de dispozitive. SNMPv3 este ultima versiune de protocol dezvoltata ce aduce cateva imbunatatiri semnificative: securitate imbunatatita, ofera suport pentru autentificare si criptare. Cu ajutorul autentificarii se poate asigura faptul ca trap-urile sunt citite doar de acei clienti care trebuie sa citeasca, astfel se previne citirea neautorizata.

BIBLIOGRAFIE

[1] https://en.wikipedia.org/wiki/Junos

[2] Junos ® OS System Basics Configuration Guide, Published: 2012-05-08, Juniper Networks, Inc., USA

[3] Michael Bushong, Cathy Gadecki, Aviva Garrett, JUNOS® FOR DUMMIES, Wiley Publishing, Inc., 2008, USA

[4] https://www.manageengine.com/network-monitoring/what-is-snmp.html#managed-devices

[5] SNMP-Based Network Management on Devices Running the Junos OS, 2012-11-14, USA

[6] http://www.juniper.net/documentation/en_US/junos13.2/topics/concept/understanding-snmp-junos-nm.html#id-10353669

[7] Junos ® OS SNMP MIBs and Traps Reference, 2015-05-27, USA

[8] Junos ® OS Network Management Configuration Guide, 2012-12-08, USA

[9] https://docs.oracle.com/cd/E19201-01/820-6413-13/SNMP_commands_reference_appendix.html