Semestrálna práca z predmetu Y36SPS

Michal Sivák

Zadanie: `Vhodne zaznamenajte využívanie multicast streamov v lokálnej sieti, výsledok zobrazte pomocou MRTG do grafického výstupu.`

1. Problematika multicast streamu

Multicast je metóda preposielania IP datagramov z jedného zdroja skupine viacerých koncových staníc. Miesto odosielania datagramov ku každému cieľu je odoslaný jediný datagram.

Multicast je definovaný prostredníctvom protokola IGMP (v RFC 1112), používa adresy skupiny D.

Vysielač (zdroj) odosiela pakety na multicastovú adresu (neslúži k identifikácii príjemcu, ale skupiny), zdrojová adresa je normálna unicastová. Na routroch sa paket odosiela do všetkých smerov, kde je nejaký príjemca (paket sa duplikuje). Medzi routrami sa multicast prenáša pomocou multicast smerovacieho protokolu (najčastejšie PIM). Ten vytvára distribučný strom, a prevádza preposielanie paketov. Príjemci (koncové stanice) sa musia zaregistrovať do multicast skupiny pomocou IGMP.

2. Multicast na lokálnom subnete - switchi

V rámci bežného switchu sa chová multicast ako broadcast, teda sú zaplavené všetky porty. Vďaka tomu, že sa ale posiela práve na cieľovú IPadresu (vyžiadanú klientom, skupina D adries), môže klient filtrovať, či chce paket prijať, alebo nie, už priamo na tretej vrstve. Pri bežnom broadcaste filtruje až dátový obsah.

Dôvod, prečo sa multicast chová ako broadcast na L2 je práve v tom, že switch dokáže rozlišovať iba ethernet rámce, a multicast nerozlišuje adresáta na L2 vrstve. Takže switch nevie, kto chce multicast prijať. Aby mohol switch vykonávať lepšie rozhodnutia, buduje si na základe "join" a "leave" správ tabuľku, kam zapisuje, kto žiada o aký multicast, a kam má aký multicast smerovať. Dnes sa používa hlavne metóda *IGMP snooping*.

3. Multicast medzi subnetmi

Táto práca sa nezaoberá sledovaním multicast streamov mimo lokálnu sieť. Pre úplnosť však uvediem, že mimo lokálnu sieť rieši posielanie multicastov *PIM* protokol, ktorý smeruje multicasty na iné routre (iba multicasty z globálneho rozsahu sa dajú posielať von).

PIM môže fungovať v dvoch režimoch: *dense mode* (posiela multicasty na všetky interfacey, keď susedný router nechce prijímať, musí to oznámiť -> výsledný strom, kam sa posielajú multicasty sa tak "oreže") alebo *sparse mode* (neposiela traffic žiadnemu routru, ktorý si nepožiada).

4. Problematika tejto práce - IGMP snooping

IGMP snooping je optimalizačný mechanizmus, ktorý vytvorí tabuľku s portami kam posiela jednotlivé multicasty. Porty sú teda dynamicky konfigurované na príjem multicastu.

Ako to funguje? Switch najskôr rozosiela IP multicastový traffic iba pripojeným routrom, a počúva všetky IGMP správy. Keď zachytí IGMP Report, priradí si záznam o tomto porte do pamäti, a začne mu posielať dáta pre príslušnú multicastovú skupinu. Každá táto správa je za normálnych

okolností posielaná do siete iba jednou z prihlásených staníc, a bez IGMP snoopingu by bola preposlaná všetkým (lebo užívatelia na všetkých ďalších portoch by už vedeli, že aj im bude informácia preposlaná, IGMP Report by neposlali). Switch by tak nebol schopný rozoznať, na ktoré porty majú traffic posielať a na ktoré nie. Z toho dôvôdu switch neprepošle IGMP report do siete, takže IGMP Query donúti odpovedať všetkých účastníkov, pretože každy z nich si myslí, že je jediným príjemcom. Iba jeden IGMP report je poslaný routru, aby posielal ďalej dáta pre skupinu.

Keď sa chce niektorý PC odhlásiť zo skupiny, pošle IGMP Leave. Switch si nie je istý, či na jednom porte nie sú ďalší užívatelia, preto pošle na port IGMP General Query. Pokiaľ nie je ďalší záujemca o príslušnú multicast skupinu, switch prestane posielať dáta na tento port.

V bežnom provoze posiela router periodicky IGMP Query (približne raz za 1-2 minúty), switch odpovie jedným IGMP reportom (ale Query prepošle ďalej a obmieňa si tabuľku).

Ako switch pozná, na ktorom porte je router? Podľa IGMP Query sa to nedá, pretože v lokálnej sieti posiela iba jeden router tieto správy, ostatné routry "mlčia". Prepínač tak sleduje ostatné správy z iných protokolov (PIM, OSPF, CGMP ...), prípadne môže mať informáciu nadefinovanú staticky na danom porte.

Na zobrazenie informácie, ktorý klient je napojený na aký multicast stream, sa na CISCO switchi používa príkaz *show ip igmp snooping multicast*. Informácie sú dostupné aj cez SNMP.

5. Postup práce

Je potrebný prístup k SNMP dátam, v prípade kolejnej siete Pod-O-Lee sú prístupné iba v serverovej VLANe. Vďaka lokálnemu kontu na stroji ip6gw.pod.cvut.cz som sa k týmto dátam dostal. Rovnako je treba mať informáciu o tzv. **SNMP Community name** (nejedná sa o presný názov, v tejto práci myslím označením "community name" práve verejné označenie domény SNMP, v ktorej je prístupný SNMP strom).

Z bezpečnostných dôvodov neuvediem tento názov v sieti Pod-O-Lee, pretože to predstavuje istý bezpečnostný risk (pomocou SNMP sa dajú napríklad vyzistiť MD5 hashe prístupových hesiel k sieťovým prvkom).

K vyhľadávaniu informácií potrebujem poznať multicast adresy streamov, ktorých pripojenie chcem sledovať. Cieľom je spraviť systém nezávislý na momentálnom dostupnom počete streamov, ale systém jednoducho rozšíriteľný (aby bolo jednoduché pridať ďalší sledovaný stream do celkovej štatistiky).

K získavaniu informácii zo SNMP som využil snmpwalk (apt-get install net-snmp net-snmp-libs net-snmp-utils), syntax príkazu je nasledovná:

snmpwalk -c COMMUNITY_NAME -v 2c IP_ADDR 1

-v špecifikuje verziu (*2c*)
IP_ADDR je adresa routra, ktorý preposiela multicasty v sieti
1 .... je časť podstromu, v ktorej chceme vyhľadávať (samotná "." špecifikuje celý strom)

Teraz stačí použitím *grep* vyselektovať informácie, ktoré potrebujeme (vyhľadat potrebné riadky, kde sú dvojice multicast_address - IP_host address). Keďže ale strom je príliš rozľahlý, vyhľadanie všetkých IP adries pre všetky zadané multicasty by trval príliš dlho. Preto je nutné strom bližšie špecifikovať. Pri vyhľadaní multicast streamu napr. 233.11.36.101, dostávame po dlhej chvíli čakania tento výstup (s bezpečnostných dôvodov som pozmenil výsledné dáta):

SNMPv2-SMI::experimental.59.1.1.2.1.3.233.11.36.101.20 = INTEGER: 2
SNMPv2-SMI::experimental.59.1.1.2.1.3.233.11.36.101.25 = INTEGER: 2
SNMPv2-SMI::experimental.59.1.1.2.1.4.233.11.36.101.20 = IpAddress: **147.32.88.200**
SNMPv2-SMI::experimental.59.1.1.2.1.4.233.11.36.101.25 = IpAddress: **147.32.93.100**
SNMPv2-SMI::experimental.59.1.1.2.1.5.233.11.36.101.20 = Timeticks: (28832662) 3 days, 8:05:26.62
SNMPv2-SMI::experimental.59.1.1.2.1.5.233.11.36.101.25 = Timeticks: (122490) 0:20:24.90

SNMPv2-SMI::experimental.59.1.1.2.1.6.233.11.36.101.20 = Timeticks: (12613) 0:02:06.13
SNMPv2-SMI::experimental.59.1.1.2.1.6.233.11.36.101.25 = Timeticks: (14346) 0:02:23.46
SNMPv2-SMI::experimental.59.1.1.2.1.7.233.11.36.101.20 = INTEGER: 1
SNMPv2-SMI::experimental.59.1.1.2.1.7.233.11.36.101.25 = INTEGER: 1
SNMPv2-SMI::experimental.59.1.1.2.1.8.233.11.36.101.20 = INTEGER: 0
SNMPv2-SMI::experimental.59.1.1.2.1.8.233.11.36.101.25 = INTEGER: 0
SNMPv2-SMI::experimental.61.1.1.5.1.3.233.11.36.101.147.32.2.245 = INTEGER: 1
SNMPv2-SMI::experimental.61.1.1.5.1.4.233.11.36.101.147.32.2.245 = Timeticks: (20378) 0:03:23.78
SNMPv2-SMI::experimental.61.1.1.5.1.5.233.11.36.101.147.32.2.245 = Timeticks: (429496729) 49 days, 17:02:47.29
SNMPv2-SMI::experimental.61.1.1.5.1.6.233.11.36.101.147.32.2.245 = INTEGER: 1

Zaujímajú ma práve uvedené adresy. Vidím, že koniec stromu SNMP je práve *59.1.1.2.1.4.IP_address*, čo mi ale pri prehľadávaní nepomôže. Nasleduje teda zdĺhavejšia procedúra, ktorá určí podstrom presnejšie. Treba tak voliť postupne jednotlivé vetvy hlbšie a hlbšie, a hľadať, či sa vo výstupe nájde práve táto známa vetva. Ak áno, zanoriť sa ďalej a ďalej,,,

Postupne som našiel konečnú vetvu: **1.3.6.1.3.59.1.1.2.1.4.IP_address**

Z výstupu je tak potrebné vyselektovať práve IP adresu hostov, pre každý zadaný multicast. Aby bola práca rozšíriteľná, všetky zaujímavé multicast. adresy treba dať do súboru (vo formáte multicast_adresa + názov streamu), následne prejsť tento strom while cyklom, a pre každý stream zavolať dotaz cez *snmpwalk*. Na výstupe môžem prehľadne zobraziť všetky dvojice adries a pozeraných streamov.

Príklad uloženia dát v súbore *stream.dat:*

233.10.47.22 Joj_plus
233.10.47.70 Markiza
233.10.47.71 STV1
233.10.47.72 STV2
233.10.47.76 STV3
---

6. Finálne riešenie

Potrebujem: súbor stream.dat, kde umiestnim dvojice multicast_adresa + meno (oddelené medzerou).

Následne využijem tento skript (streamscript):

```
#!/bin/bash

while read IP MENO
do

for i in `snmpwalk -c COMMUNITY_NAME -v 2c 147.32.88.1 1.3.6.1.3.59.1.1.2.1.4.$IP | cut -d : -f 4`; do echo "$IP($MENO) $i"; done | grep -v ^$ | sort | uniq

done < stream.dat
```

Skript na výstupe vypíše všetky dvojice mutlicast(názov), host_IP_address oddelené medzerou, ktoré sú aktuálne v tabuľke routra. Pre presnejšie informácie je vhodné ešte vygrepovať adresy streamovacieho PC, jeho adresu tiež neuvádzam, v každej sieti bude rozdielna.

Pre zobrazenie konkrétnych užívateľov, ktorí sledujú daný program nám teraz stačí použitím grep selektovať priamo podľa názvu streamu. Najvhodnejšie je pracovať s ďalším skriptom, ktorému na vstupe zadám názov streamu ako parameter. Uvedený skript prikladám až po vysvetlení MRTG.

7. MRTG a zobrazenie výsledkov

Multi router traffic grapher, je nástroj primárne určený na monitorovanie sieťového provozu na jednotlivých interfaceoch routru. Jeho modularita však dokáže to, že do grafu dokáže zapísať takmer čokolvek (teplotu, záťaž procesora, voľné miesto na disku...).

Grafy sú vytvorené pomocou PNG obrázkov, kedy sa uchováva staršia a novšia verzia (dokreslená o ďalší záznam).

MRTG robí štatistiku každých 5 minút, a následne počíta priemer za 30minút, dve hodiny, a deň. Výsledok sa potom zobrazuje v dennom, týždennom, mesačnom, prípadne ročnom súhrne.

Výhoda MRTG je, že mu postačujú 4 parametre na vstupe pri základnej konfigurácii (primárne nastavený na zobrazenie INcoming a OUTgoing trafficu). Keď potrebujem zobraziť iba jeden údaj, sú prvé dva parametre rovnaké. Tretí parameter zobrazí čas, ktorý mu pošlem ako vstup vo formáte 0:0 (túto funkciu som do mojej práce neimplementoval), Štvrtý parameter je názov monitorovaného stroja (ja som nechal miesto názvu stroja vypisovať multicast adresu sledovaného streamu).

Inštalácia MRTG:

Je potrebné mať nainštalovaný apache. Inštalácia MRTG cez *apt-get install mrtg*
to vytvorí aj podadresár vo */var/www/mrtg*
teraz je potrebné zmeniť globálne nastavenia v konfiguračnom súbore (*/etc/mrtg.cfg*), doležité je hlavne zadať správny workdir (generujú sa v ňom stránky a logy).

```
RunAsDaemon: yes
WorkDir: /home/miq/public_html/mrtg
```

Konfiguračný súbor sa môže prekopírovať aj inam (napr. pod užívateľa, ktorý nemá root práva), potom sa *RunAsDaemon* použije iba v hlavnom konfiguračnom súbore v */etc/mrtg.cfg*).
Generovanie stránok cez úpravu crontabuľky (v postupe neskôr).

```
Options[_]: growright, bits
Target[233.x.x.x]: `/home/miq/sps/usercount_mrtg 147.32.*`
MaxBytes[233.x.x.x]: 1000
Title[233.x.x.x]: Users watching TV (all users)
YLegend[233.x.x.x]: Users
ShortLegend[233.x.x.x]: [-]
Legend1[233.x.x.x]: Users watching TV
LegendI[233.x.x.x]:  
LegendO[233.x.x.x]:  
Options[233.x.x.x]: growright,gauge,nopercent,unknaszero
PageTop[233.x.x.x]: <H1>Users watching TV (all users)</H1>
  <TABLE>
   <TR><TD>System:</TD><TD>233.x.x.x</TD></TR>
   <TR><TD>Maintainer:</TD><TD>miq</TD></TR>
  </TABLE>
```

V prvom riadku je práve skript s parametrom, ktorý vráti počet všetkých užívateľov. K detailu implementácie sa dostanem v ďalšom bode.

8. Detaily implementácie a problémy

Je problémom, že v prípade spustenia skiptu sa zobrazí iba aktuálny stav hostov, ktoré si udržuje router. V prípade, že niekto pozeral TV predchádzajúce 4 minúty, a následne sa odpojil, záznam sa neobjaví. Preto som implementoval postup, ktorý zistí počet užívateľov všetkých streamov každú minútu, a zapíše ich vždy do unikátneho súboru. Následne práve už spomínaný skript usercount_mrtg iba vezme posledných 5 vytvorených súborov, zotriedi ich, a výstupom budú unikátne záznamy už so zadaním požadovaného parametra, cez ktorý sa zistí ich celkový počet.

Zápis všetkých užívateľov každú minútu mi zabezpečí tento skript (*cronscript1*), ktorý vložím do crontabuľky.

```
----- cronscript1
#!/bin/bash
/home/miq/sps/streamscript > ~/sps/tmp_files/vystup`date +%H:%M:%S`
-------
```

Tento skript spustím každú minútu. Pre šetrenie miesta na disku som vytvoril ďalší skript, ktorý premaže každé dve hodiny nepotrebné súbory z adresára /tmp_files/.

```
----- cronscript120
#!/bin/bash
find /home/miq/sps/tmp_files/* -mmin +120 -exec rm -f {} \;
-----
```

Výsledná crontabuľka vypadá nasledovne: (editujem cez *crontab -e*)

```
-----
# m h  dom mon dow   command
*/1 * * * * /home/miq/sps/cronscript1 >/dev/null
2   * * * * /home/miq/sps/cronscript120 >/dev/null
-----
```

Takže momentálne sa mi logujú každú minútu všetci užívatelia do unikátneho súboru. Skript *usercount_mrtg* je teda nasledovný (vracia 4 parametre pre MRTG, prvé dva sú rovnaké):

```
----- usercount_mrtg
#!/bin/bash

prem=$1

for i in `seq 1 2`;
do  cat `find /home/miq/sps/tmp_files/* -mmin -6 -mmin +1` | sort | uniq | grep $prem | wc -l;
done

echo "0:0"
echo $1
-----
```

Skript prechádza posledných 5 súborov (posun o jeden súbor skôr, aby nedošlo k prípadnému konfliktu, prístup k súboru tak nemusí byť synchronizovaný).

Keď chcem vyhľadať všetkých užívateľov, zadám to, čo majú spoločné: napríklad 147.32. ... V prípade, že je jedna stanica vysielaná na viacerých multicastových adresách, grepom zistím počet

cez jej názov, a dokážem tak dynamicky zisťovať počet užívateľov nie v závislosti od multicast adresy, ale v závislosti od názvu (sledujem počet tých, čo pozerajú zadanú stanicu).

V konfiguračnom súbore teraz stačí duplikovať nasledujúce riadky so zmenenými parametrami (názvom, parametrom k skriptu usercount_mrtg, ...):

----- (pokračovanie súboru mrtg.conf)
Options[_]: growright, bits

Target[233.11.36.92]: `/home/miq/sps/usercount_mrtg Nova`
MaxBytes[233.11.36.92]: 100
Title[233.11.36.92]: Users watching Nova TV
YLegend[233.11.36.92]: Users
ShortLegend[233.11.36.92]: [-]
Legend1[233.11.36.92]: Users watching Nova TV
*LegendI[233.11.36.92]: *
*LegendO[233.11.36.92]: *
Options[233.11.36.92]: growright,gauge,nopercent,unknaszero
PageTop[233.11.36.92]: <H1>Users watching Nova TV</H1>
<TABLE>
<TR><TD>System:</TD><TD>233.11.36.92</TD></TR>
<TR><TD>Maintainer:</TD><TD>miq</TD></TR>
</TABLE>

Po uložení konfigurácie potrebujeme vytvoriť index.html, použijem píkaz indexmaker:

```
indexmaker --title "TV watching account" --columns=1 /home/miq/sps/mrtg.cfg  >
/home/miq/public_html/mrtg/index.html
```

a upravíme záznam do crontabuľky (cez crontab -e, a vložíme nasledujúci riadok):

**/5 * * * * /usr/bin/mrtg /home/miq/sps/mrtg.cfg --lock-file /tmp/mrtg.lock --confcache-file /tmp/mrtg.cfg.ok >/dev/null*

MRTG pracuje s predchádzajúcimi verziami generovaných PNG súborov. Preto v prvých dvoch periódach (2x5min) crontab zahlási chybu, a prepošle chybový výstup ako mail do súboru (napr.: */var/mail/miq*). Následne už k žiadnym chybám nedochádza, a grafy sa začnú vykresľovať.

Výstup mojej práce: *https://ip6gw.pod.cvut.cz/~miq/mrtg/*

24.5.2009, 22:45
Po kliknutí na príslušný stream sa zobrazia podrobnosti (štatistika za posledné obdobie). To že je graf taký "zubatý" je preto, že ke´d užívateľ prepína programy, zobrazí sa na viacerých miestach v logu (akoby pozeral viacero staníc). Obmedzovať túto skutočnosť by malo za následok uprednostnenie niektorých TV staníc pred inými, prípadne nezobrazenie informácie, kedy užívateľ skutočne pozerá viacero programov naraz (napríklad počas športových stretnutí).

9. Sumarizácia postupu

Je potrebné:

- nainštalovať apache, mrtg, snmpwalk, indexmaker

- mať prístup k SNMP informáciám, poznať CommunityName
- spísať dvojice MulticastAddr Name do súboru (stream.dat)
- skriptom zistiť užívateľov všetkých streamov (streamscript)
- skriptom logovať každú minútu stav na routri (cronscript1), zapísané do crontab
- skriptom mazať nepotrebné súbory (cronscript120)
- zadefinovať do konfiguračného súboru MRTG sledované streamy (duplikácia uvedeného záznamu)
- vytvoriť index.hmtl (indexmaker)
- naplánovať cez crontab spúšťanie MRTG (každých 5 minút)

10. Záver

MRTG (multirouter traffic grapher) je silný nástroj, ktorý prehľadne a jednoducho zobrazuje dáta do grafického výstupu. Pomáha tak správcovi sieťe mať prehľad o teplote na aktívnych prvkoch, vyťaženosti jednotlivých liniek a podobne. Problematika sledovania siete sa v dnešných dňoch objavuje dosť často, táto práca pomôže štatisticky vyhodnocovať záujem užívateľov o príslušné TV programy. Pomocou ďalšieho softwareu je tak možné napríklad sledovať podiel na trhu u jednotlivých TV staníc v prípade istej cieľovej skupiny užívateľov (študenti VŠ).

Bola splnená úloha? Áno, ale iba v rámci možností, ktoré pozná Cisco router (a ďalej ich prezentuje cez SNMP). Router však vie aktuálne iba o niektorých užívateľoch (podľa IGMP snoopingu dostane odpoveď od switchu v podobe jedného IGMP reply, takže ak je na switchi viacero používateľov toho istého streamu, vracia vždy náhodného z nich). Vzhľadom k pravdepodobnostnému rozloženiu náhodnosti odpovede je počet objektívny práve vtedy, keď je na switchi maximálne 6 užívateľov toho istého streamu, takže skutočný počet zobrazený cez MRTG je 100% iba pri grafoch, ktoré neprekračujú počet 18 užívateľov (s výnimkou celkového počtu). Takže uvedený postup dokáže relevantne vyhodnocovať štatistiku divákov hlavne menej využívaných programov.

11. Použité zdroje k teoretickému úvodu

http://www.cs.vsb.cz/grygarek/SPS/projekty0405/IGMPsnooping-Buzek.pdf

http://www.debianadmin.com/mrtg-installation-and-configuration-in-debian-based-distributions-2.html

http://www.samuraj-cz.com/clanek/tcpip-skupinove-vysilani-ip-multicast-a-cisco/