

Softvérový smerovač

Milan Ilavský

Obsah

1	Zadanie	2
2	Úvod do problematiky	3
2.1	Smerovač	3
2.2	Smerovanie	3
2.2.1	Smerovacia tabuľka	3
2.2.2	Smerovacie algoritmy a protokoly	4
2.2.2.1	Statické smerovanie	4
2.2.2.2	Dynamické smerovanie	4
2.2.2.3	Protokol RIP	5
3	Návrh riešenia	7
3.1	Vývojové prostredie	7
3.2	Užívateľské rozhranie	7
3.2.1	Implementácia	7
3.2.2	Protokol ARP	8
3.2.3	Smerovacie tabuľky	8
3.2.4	Smerovanie	8
3.2.5	Dynamické smerovanie	9
3.2.6	Filtrovanie prenosu	9
4	Záver	10

1 Zadanie

Analyzujte princípy a mechanizmy smerovania na skutočných smerovačoch a navrhните softvérový smerovač. Smerovač má byť iba dvojportový (dve sieťové karty, port 1, 2), ovládanie interfejsov realizujte príslušnými paketovými ovládačmi.

Smerovač navrhните a implementujte tak, aby umožňoval ako statické tak aj dynamické smerovanie a:

- zobrazoval smerovaciu tabuľku: IP destinácia – odchádzajúci interfejs
Smerovač sa svoju dynamickú smerovaciu tabuľku učí, t.j. implementujte svoj vlastný dynamický smerovací protokol.
- umožňoval užívateľovi zadať aj tzv. "statickú" cestu. Nech zo smerovacej tabuľky je zrejmé, čo je naučené dynamicky a čo staticky.
- smerovacia tabuľka je aktuálna.
- filtroval komunikáciu v rozsahu 3. - 4.vrstvy RMOSI, aplikácie (well known ports). Riešenie navrhните tak, aby bolo možné realizovať aj ľubovlnú kombináciu filtrov, napr. pre nejakú IP povoliť iba http komunikáciu a pre inú IP zakázať napr. "ping". Filtre rozlišujte v smere "IN" a "OUT". (Např. Host A sa nedostane von na web (HTTP), ale u neho bežiaci apache server bude dostupný)
- zobrazoval, nuloval štatistiky IN/OUT na interfejsoch soft. smerovačov.

2 Úvod do problematiky

2.1 Smerovač

Smerovač je uzol medzi dvoma alebo viacerými sieťami, ktorý zabezpečuje prenos paketov medzi nimi. Pracuje na tretej vrstve RMOSI, čo znamená, že pracuje s logickými adresami. Proces, ktorý zabezpečuje správny prenos paketov sa nazýva smerovanie. Smerovanie je vo veľkej miere spájané s protokolom IP, ale týka sa aj iných, menej používaných protokolov.

2.2 Smerovanie

Smerovanie môžeme nazvať tiež aj ako hľadanie ciest v počítačovej sieti. Jeho cieľom je dopraviť paket z jedného miesta na druhé. Cesta, ktorú musí paket prejsť, môže byť značne zložitá, preto sa vo väčšine prípadov postupuje po krokoch (smerovačoch). V každom uzle sa hľadá uzol, kam by sa paket poslal. Smerovanie je práca hlavne tretej sieťovej vrstvi RMOSI. Smerovanie zabezpečuje každé zariadenie, pracujúce na tretej vrstve, dokonca aj koncový počítač.

2.2.1 Smerovacia tabuľka

Ako základnú dátovú štruktúru, môžeme v smerovačoch považovať smerovaciú (routovaciú) tabuľku. Predstavuje "smery", kam je potrebné packet poslať. Samotná tabuľka je zložená z:

- cieľovej adresy, ktorej sa konkrétny záznam týka. Môže to byť adresa konkrétneho počítača, alebo siete. Záznam je vo väčšine prípadov udaný prefixom, ktorý ma

tvar napríklad 192.168.0.0/16. Hodnota pred lomítkom je cieľová adresa a hodnota za lomítkom je počet významových bitov adresy. V tomto konkrétnom prípade vyhovuje každá adresa, ktorá sa začína na 192.168.

- akcie určujúce, čo urobiť s paketmi, ktorých adresa vyhovuje prefixu. Akcie môžu doručiť paket priamo adresátovi, alebo poslať paket ďalej.

Smerovanie prebieha samostatne pre každý paket. Vezme sa jeho cieľová adresa a porovná sa so smerovacou tabuľkou. Zo všetkých vyhovujúcich záznamov sa vezme "najpresnejší" záznam, tj. záznam z najdlhším vyhovujúcim prefixom.

2.2.2 Smerovacie algoritmy a protokoly

O tom, ako smerovacia tabuľka vzniká a ako je udržiavaná, sa starajú takzvané smerovacie algoritmy. Keď sú pre nejaký algoritmus presne definované pravidlá, formát komunikácie a podobne, vzniká smerovací protokol. Smerovacie algoritmy môžeme rozdeliť na statické a dynamické.

2.2.2.1 Statické smerovanie

Pri statickom (neadaptívnom) smerovaní sa smerovacia tabuľka nemení. Ja daná konfiguráciu smerovača a všetky zmeny treba robiť ručne. Táto možnosť siete vyzerá neprakticky, ale veľká väčšina smerovaní v sieti je statická.

Do tejto kategórie patria skoro všetky užívateľské počítače, kde je smerovacia tabuľka natoľko jednoduchá, že dynamické smerovanie nemá zmysel. Smerovanie prebieha len v lokálnej sieti, alebo cez jeden uzol do vonkajšej. Informácie o tomto uzle zadáme buď ručne, alebo sa nastaví automaticky vďaka protokolu DHCP.

2.2.2.2 Dynamické smerovanie

Dynamické smerovanie flexibilne reaguje na zmeny v sieti a prispôbuje im smerovaciu tabuľku. Podľa toho, akým spôsobom prebieha vzájomná výmena alebo kombinácia informácií jednotlivých zariadení, delíme dynamické algoritmy do niekoľkých skupín, a to:

- **Centralizované:** Všetky smerovače posielajú informácie do tzv. smerovacieho centra, ktoré podľa nich zostaví sieť a jednotlivé smerovacie tabuľky, ktoré pošle konkrétnym smerovačom. Z pohľadov smerovačov je to veľmi jednoduché riešenie a taktiež smerovacie centrum má komplexné informácie o sieti, podľa ktorých môže vybrať najvýhodnejšie smerovanie. Problém však nastáva pri posielaní informácií do smerovacieho centra a pri synchronizácii informácií poslaných centrom.
- **Izolované:** Tieto algoritmy pracujú čisto individualisticky. Nevymieňajú si žiadne dáta a o smerovaní sa rozhodujú úplne samostatne. Patrí tu napríklad algoritmus nazývaný flooding. Smerovaný paket sa pošle do všetkých rozhraní. Takto vyskúša všetky cesty, čiže aj tú najkratšiu. Problém ale zostáva v efektívnosti a pri slučkách, ktoré môžu v sieti vznikáť.
- **Distribúované:** Informácie o zmenách sa predávajú postupne medzi susednými smerovačmi, až kým sa nerozšíria po celej sieti. Takto navrhnuté algoritmy sú dostatočne pružné a stabilné a zvládajú aj rozsiahle siete, dôkazom čoho je aj sieť Internet, ktorá je na nich postavená. Konkrétne prístupy a mechanizmy sa výrazne líšia. Medzi najvýznamnejšie protokoly patria:
 - RIP
 - OSPF
 - BGP
- **Hierarchické:** Riešia problémy rozsiahlych sietí. Smerovanie sa rozdeľuje medzi menšie samostatné oblasti a zmeny v topológii sa šíria len v týchto oblastiach. O výmenu súhrnných informácií sa starajú takzvané hraničné smerovače. Jedna oblasť dostáva o druhej iba súhrnné informácie. Tento spôsob implementuje napríklad protokol OSPF.

2.2.2.3 Protokol RIP

Routing Information Protocol (RIP) umožňuje komunikáciu medzi smerovačmi. Tento protokol patrí medzi najstaršie a doposiaľ stále používané protokoly. Uplatnenie nachádza najmä v menších sieťach, vďaka svojej jednoduchosti a nenáročnej konfigurácii.

RIP je protokol typu distance-vector (vzdialenostný vektor). Využíva Bellman-Fordov algoritmus pre určenie najkratšej cesty v sieti. V jeho prípade je metrikou na určenie najkratšej cesty, počet skokov (hop count). Smerovače si vymieňajú tabuľky periodicky, každých 30 sekúnd. Veľkosť siete je obmedzená na najdlhšiu vzdialenosť medzi uzlami v skokoch, na 15 skokov.

3 Návrh riešenia

3.1 Vývojové prostredie

Smerovač som sa rozhodol implementovať v operačnom systéme Linux, za pomoci knižnice pcap. Ako kompilátor budem používať gcc, resp. jeho C++ verziu g++.

Operačný systém Linux som sa rozhodol používať vzhľadom k lepšej použiteľnosti, čo sa týka sieťových riešení.

3.2 Uživatelské rozhranie

Aplikácia obsahuje jednoduché konzolové prostredie, ktoré pokladám pri riešení takéhoto problému za najvýhodnejšie, nakoľko nie je závislé na žiadnom grafickom systéme.

Po štarte sa objaví klasický prompt očakávajúci príkazy. Smerovač sa snaží o používanie podobnej štruktúry a formátu príkazov, ako sa používa v klasických smerovačoch. Snaží sa rozpoznať aj bežné skratky týchto príkazov a ponúknuť k nim stručnú nápovedu.

3.2.1 Implementácia

Pri implementácii je potrebné riešiť viacero problémov, rozdelil by som ich do týchto základných častí.

- Implementácia klasického protokolu ARP
- Smerovanie podľa smerovacích tabuliek
- Komunikácia medzi smerovačmi (Dynamické smerovanie)

- Filtrovanie prenosu

3.2.2 Protokol ARP

Pre správne smerovanie, je potrebné implementovať protokol ARP. Pri implementácii tohoto protokolu by som vyzdvihol najmä implementáciu ARP tabuľky (ARP cache).

ARP tabuľku som sa rozhodol riešiť pomocou spájaného zoznamu, v ktorom budú uložené MAC adresy, k nim prislúchajúce IP adresy, zariadenie, ktorému záznam patrí a životnosť konkrétneho záznamu. Životnosť záznamu bude štandardne 30 sekúnd, ale bude ju možné nastaviť cez príkazy. ARP tabuľku je možné implementovať aj staticky, pričom by sa položky mohli pridávať taktiež cez príkazy. Statickú ARP tabuľku, ale nebudem implementovať, nakoľko riešenie nášho problému od nej nie je nejak závislé.

ARP budem implementovať ako triedu a hlavný program jej bude podsúvať ARP pakety. Trieda sa bude nastavovať podľa ich obsahu. V prípade potreby, bude môcť odoslať ARP paket (request, reply).

3.2.3 Smerovacie tabuľky

Smerovacie tabuľky som sa rozhodol udržiavať v spájaných zoznamoch. Pre každú tabuľku jeden zoznam. Pri smerovaní treba dbať na prioritu, pričom dynamická tabuľka má nižšiu prioritu ako statická. Ak sa nájdu dve možnosti smerovania, vyberie sa presnejšie (tá ktorá má väčšiu masku)

V smerovacej tabuľke si budem udržiavať IP adresu siete, pre ktorú viem smerovať, jej masku, zariadenie, na ktoré sa má smerovať a v prípade komunikácie so smerovačom aj IP (ip next hop), ktorá presne špecifikuje cieľový smerovač.

3.2.4 Smerovanie

Samotné smerovanie bude prebiehať nasledovne:

Pri odchytení paketu zistím, či to nebol paket, ktorý som práve odosielať (pcap odchyťava aj tie pakety, ktoré sám odoslal). Túto skutočnosť zistím podľa zdrojovej MAC adresy. Potom zistím typ paketu (Ethernet II, atď.), pokračujem vnoreným protokolom, pričom ma zaujíma iba protokol ARP (tieto pakety budem podsúvať ARP triede) a protokol

IP, ktorý budem smerovať. V prípade protokolu ARP, celý paket posuniem ARP triede, ktorá ho spracuje. V prípade IP ako prvé zistím IP adresu a prehľadám smerovacie tabuľky. Ak sa záznam v smerovacích tabuľkách nenachádza, paket zahodím. Ak sa v nich záznam nachádza, vyberiem ten najvhodnejší, podľa priorít opísaných vyššie. Keď má záznam nastavenú ip next hop, zistím MAC adresu, ktorá prislúcha tejto IP a prepíšem ňou cieľovú adresu v pakete. Ak sa záznam IP next hop nenachádza, tak vyšlem ARP request s cieľovou IP adresou na zariadenie, ktoré je nastavené v tabuľke. Keď budem vedieť príslušnú MAC adresu, nahradím ňou cieľovú MAC adresu v pakete. Okrem toho zmením ešte zdrojovú MAC adresu za MAC adresu zariadenia, z ktorého paket posiadam.

3.2.5 Dynamické smerovanie

Na komunikáciu smerovačov medzi sebou navrhнем jednoduchý protokol. Na začiatku sa vyšle broadcast na všetky zariadenia. Broadcast sa bude posilať vždy pri spustení alebo po zadaní príkazu. V momente, keď príde smerovaču tento paket na vybraný port, uloží si odosielateľa (susedný smerovač) do tabuľky (spájaného zoznamu) a smerovaciu tabuľku bude posilať len smerovačom, ktoré má v tejto tabuľke. Pri pripojení ďalšieho smerovača sa automaticky prvým broadcastom pripojí k ostatným. Následná komunikácia bude prebiehať periodicky v danom intervale, ktorý sa bude dať nastaviť. Smerovač bude zo svojej smerovacej tabuľky posilať len číslo siete a masku. Dynamická tabuľka smerovača sa po každom prijatí vymaže a zostaví od začiatku, aby mohla reagovať aj na zrušené záznamy.

3.2.6 Filtrovanie prenosu

Filtrovanie prenosu som sa rozhodol riešiť cez štandardný filter, ktorý obsahuje pcap, pričom sa bude tento filter dať nastaviť na každé zariadenie a na každom zariadení na príchodzie a odchodzie pakety zvlášť. Rozlišovanie príchodzích a odchodzích paketov sa dá nastaviť v jednom filtri, ale pre zjednodušenie som sa to rozhodol implementovať to dvoma, pričom rozlíšenie do filtra dogenerujem automaticky. Filter bude možné nastavovať cez príkazy, pričom štandardná politika bude všetko povolené.

4 Záver

Naša implementácia softvérového smerovača, bude spĺňať základné požiadavky smerovania a bude rozšírená o dynamické smerovanie a filtrovanie prenosu. Jeho využitie bude vhodné na demonštratívne účely, kde sa budú dať sledovať základné princípy smerovania a prekladania IP adres na MAC adresy (protokol ARP). Taktiež bude možné sledovať dynamické smerovanie a výmenu tabuliek. Účel projektu, hlbšie pochopiť problematiku smerovania a prácu smerovačov na praktickom príklade, bol podľa môjho názoru splnený, nakoľko pre funkčnú implementáciu bolo potrebné naštudovať a pochopiť všetky základne princípy smerovania.

Literatúra

[1] *Odkazy na wikipédií*

<http://en.wikipedia.org/wiki/Router>

<http://en.wikipedia.org/wiki/Routing>

http://en.wikipedia.org/wiki/Address_Resolution_Protocol

http://en.wikipedia.org/wiki/Routing_Information_Protocol

<http://cs.wikipedia.org/wiki/Router>

<http://cs.wikipedia.org/wiki/Routov%C3%A1n%C3%AD>

[2] *Dokumentácia knižnice Pcap*

<http://www.tcpdump.org/>

[3] *Ostatné použité odkazy*

<http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html>