pdate na 17/18?

Sup kids, chcete taky přístup? Už můžete.

ISA - Studentská skripta, ak. rok 2015/2016

---

http://wiki.fituska.eu/index.php/ISA:_Ot%C3%A1zky

Ústní zkoušení (otázky co se objevili)

1. Vše ohledně DNSSEC. Proč se používá, jak se používá, jakým útokům zabraňuje,...
2. Email - protokoly, jak probiha překódování do 7 bitů, jak vypadá výsledná zpráva, popsat zabezpečení, principy zabezpečí, jak funguje podepsaní ap.

Příprava k semestrální zkoušce

A2

1. VoD naco je ake protokoly pouziva, uvedte priklad naviazania spojenia
2. Tabulka popisat nastorje/ technologie ako dostiahnut toho co je v tabulke. Nieco na sposob ako zistit dostupnost weboveho serveru, ako zistit dostupnost tlaciarne ako zisiti pocet paketov prenesenych za nejaky cas….
3. DNSSEC, naco je co zabezpecuje zaznamy
4. Sietovy tok, agregace, vizualizace, filtrovanie u Netflow,
5. Token Bucket ako prvy termin
6. ako lokalizovat SIP server cez DNS
7. bitovy vektor a RFC

B1

1) Definujte pojmy:

SLA - Service Level Agreement => zaistenie kvality služieb z požiadavok používateľa danej siete definované v jeho zmluve - zmluva špeciikuje nároky na prenos zákazníckych dát, resp. požiadavky na fyzické pripojenie (typ sieťovej infraštruktpry - dostupnosť siete, či prenosová kapacita), požiadavky na sieťový prenost (strátovosť, celkové doby odozvy, rozptyl, oneskorenie)

traffic shaping => rozloženie provozu, služi k regulácii rychlosti a objemu provozu jednotlivých tokov či agregovaných tokov, ktorého primárnou úlohou je prispôsobiť prenos paketov danej rýchlosti špecifikácie

token buket - zásobník žetónov => uchováva v zásobníku jednotlivé žetóny, a dovoľuje prepúšťať určité množstvo zhlukov pri danej priemernej rýchlosti

RSVP (Resource reservation protokol) => rezervacni protokol je signalizačným protokolom, ktorý umožňuje skupine vysielajúcich staníc preniesť vysielané toky v požadovanej kvalite k prijímacím staniciam tak, aby prenosové pásmo bolo optimálne využité a predošlo sa zahlteniu prenosových zdrojov. požiadavok na rezerváciu posiela koncová aplikácia prostedníctvom RSPV najbližšiemu smerovaču, ktorý ju posielá dalej po ceste k zdroji vysielania

2) Rozdíl v multicastu u IPv4 a IPv6 - z těchto hledisek a) komunikace, b) přihlašování do skupiny, c) mapování adres L3 na L2
b)IPv4 - zariadenie pošle IGMP join paket so žiadosťou o pripojenie do multicastu
IPv6 - žiadosť o pripojenie do multicastu sa posiela v pakete ICMPv6
c) na L2:
IPv4: k prefixu 01:00:5e sa pridá jeden bit 0 a za to sa pripojí 23 spodných bitov z ipv4 adresy
IPv6: k prefixu 33:33 sa pridá 32 spodných bitov IPv6 adresy
na L3:
IPv4 - adresy zo skupiny D
IPv6 - adresa s prefixom 0xFF00::/8

3) Zadán obsah packetu, určit z toho co je to za komunikaci a k čemu slouží. Vybrat z toho důležité informace
Bolo to SIP/SDP - bolo treba napísať že to obsahuje ip adresu, port, kodeky...ale aj čo je kodek a aká adresa a aký port to je

4) Co můžeme sledovat u SNMP u následujících objektů (3 ke každému): interface, system, ICMP, UDP

interface - meno rozhrania, počet prenesených paketov/ bytov
system- typ OS, systemovy cas, kontakt
ICMP- typy icmp správ
UDP- pocet prenesených paketov, počet prenesených bytov

5 -9b) Připravit DNS záznamy i reverzní (A,NS,PTR,CNAME,MX) pro firmu s adresou /29, kde je 5 stanic n1-n5, www server www.nordik.cz, DNS server ns.nordik.cz, mail server mail.nordik.cz.

nordik.cz. IN NS ns.nordik.cz.

nordik.cz. IN MX 10 mail.nordik.cz.

nordik.cz. IN A 192.168.0.1

1.0.168.192.in-addr.arpa. IN PTR nordik.cz.

www.nordik.cz. IN CNAME nordik.cz.

Vedel by nekdo, co s tema 5 stanicema?
odpoved: IMHO (ak predpokladame stanicu n1 rovnaku ako server nordik.cz, ak nie tak adresy++)
`n1.nordik.cz. IN A 192.168.0.1`
...
`n5.nordik.cz. IN A 192.168.0.5`
mozeme pripadne doplnit aj arpa zaznamy ako
`1.0.168.192.in-addr.arpa. IN PTR n1.nordik.cz.`
...
`5.0.168.192.in-addr.arpa. IN PTR n5.nordik.cz.`

6) Rozdíl mezi TSL/SSL a S/MIME, na jaké vrstvě fungují a jak zabezpečují.
napísala som že S/MIME je kodovanie textu na strane mailového klienta a TSL/SSL je kodovanie na strane serveru (dal mi 2 body)

TLS/SSL je zabezpecenia komunikacie medzi klientom a serverom. TLS/SSL nezabezpecuje data. Cize na strane serveru nezabezpeci nebudu ozudene nou stranou. ????? NEBUDU OZUDENE NOU STRANOU? CO TO JE ČESKY? - autor tam ma typo. chcel tym asi povedat to, ze server nezarucuje ze data budu sifrovane aj pri prechode na dalsie smtp servery a teda mozu byt odcudzene/odcizene.

S/MIME pracuje na aplikanej urovvni s protokol SMTP. Pracuje na principe asymetrickej kriptografie. Zabezpecuje data.
Podrobnosti vid kniha od Matouska kapitola Mail. to je kapitola 4 ktorú nemáme k dispozícii nie?
Kapitola 4 asi k dispozici neni, ale pokud se podivate na prednasky, tak tohle probiral na nejake prednasce az pozdeji. Jeste bych zde zminil neco o Certifikatech k tomu S/MIME.

odpoved: **???**

7 -12b) Rozhodnout jaká je vhodnější (lepší nároky na šířku přenosového pásma) velikost vzorku u kodeku G711 s režijí 58B. Jestli 20ms nebo 30ms. Kodek vzorkuje na 8b.

odpoved:
rezia 58B, vzorkovanie 8b

pre 20ms:

0.02s*64000b/s = 1280b => 160 B 64000b/s / 1280b = 50 paketov za sekundu

pasmo:(58+160)*50*8 = 87.2 Kb/s

pre 30ms:
0.03s*64000b/s = 1920b => 240B
64000b/s / 1920b = 33.33 => 34 paketov za sekundu

pasmo: (58+240)*34*8 = 81.056 Kb/s

Na prednasce to pocital jinak: 30ms = 33,3 rámců (s/30ms) = 34; 34*58 = 1914B = 1914*8 + 64000 = 15776 + 64000 = 79 776 -> 79.776 Kb/s (prvni touto metodou vyjde stejne, ale druhej ne)

Rozdil je v tom, že v tom prvním se do toho posledního paketu dá celý obsah, tz. usek delsi nez jedna sekunda.. Podle mě záleží jak se to bere, takže teoreticky by mohly byt dobře oba způsoby

30ms velkost vzroku lepsie vyuziva prenosove pasmo.

Pro mou jistotu, tech 8nitu ktere zadal jsou vec kodeku G711, takze to neresim, ze?

zadání 6. 1. 2016
A1
1. otázka (8 bodů) byly 4 zprávy ICMPv6 (Multicast Listener Discovery, Router Solitation, Router Advertisement a něco) ... ... lehce každou popsat
Multicast Listener Discovery = Přihlašování do multicastu IPv6, informační zpráva.
Router Solicitation = výzva směrovači, informační zpráva
Router Advertisement = ohlášení směrovače, inf. zpráva
Neighbor Solicitation = výzva sousedovi, inf. zpráva

2. otázka (10 bodů) broadcast ... popsat, na příkladu ukázat IPv4 adresu, 3 aplikace kde se používá, jak se to implementuje pomocí BSD sockets

Broadcast

-všesměrové vysílání jehož cílem jsou všechny uzly v síti
-pouze v IPv4
-minimalizuje síťový přenos
-podporuje pouze transportní protokol UDP
-rozsah komunikace pouze na lokální síti
Broadcast pro síť 147.229.8.0/23
L3 vrstva : 147.229.9.255

L2 vrstva : ff:ff:ff:ff:ff:ff

3 aplikace: NTP, DHCP, ARP

pomocí BSD socket:

1. vytvoření schránky typu UDP
2. povolení přenosu broadcast
3. nastavení adresy a portu
4. posílání dat pomocí sendto()

3. otázka (8 bodů): napsat co znamená zkratka + lehce popsat

1. **RTSP**
   - **Real-time Streaming Protocol**
   - Tzv. **signalizační protokol - Slouží k navázání a ukončení spojení** – řízení jednoho nebo více časově synchronizovaných média streamů (funguje jako „síťový dálkový ovladač“ pro multimediální servery)
   - Je určený pro VoD a multimedia multicasting a broadcasting
   - **Out-of-band** protokol – data jsou doručována jiným protokolem (např. RTP)
   - Stateful protocol – server si udržuje informaci o stavu relace (ID relace, sekvenční číslo)

   zdroj (ISA-Videokonference-2015.pdf slide 24)
2. **SDP**
   - **Session Description Protocol**
   - Přenášen ve zpravach SIP typu INVITE a OK Informace potrebne pro prijimani hlasoveho toku

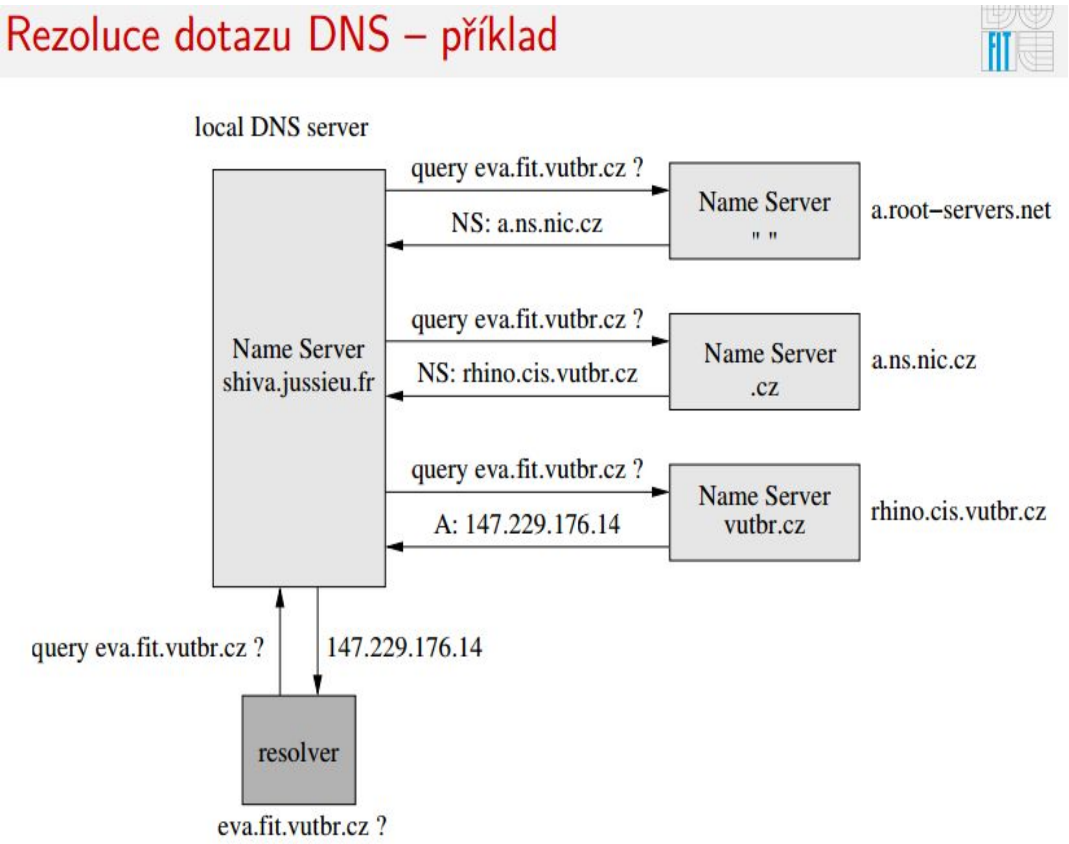
   zdroj (ISA-Videokonference-2015.pdf slide 27)
3. **MPEG TS**
   - **MPEG Transport Stream**
   - Standard, který popisuje, jakým způsobem jsou části multimediálního obsahu kombinovány do jednoho datového toku – multiplexing
   - Umožňuje v jednom datovém toku posílat jeden i více multimediálních streamů
   - Vhodné pro přenosy, při nichž je možné, že dojde ke ztrátě paketu nebo poškození dat
   - Sekvence transportních paketů o pevné délce obvykle 188 bytů + 4 bytovoa hlavička
   - První byte hlavičky obsahuje hodnotu 0x47 (sync byte)
   - Klíčovou položkou je 13 bitový Packet Identifier (PID), která odpovídá jednotlivým elementárním streamům

   zdroj (ISA-Videokonference-2015.pdf slidy 17-22)

4. **MCU**
   - **Multipoint Control Unit**
   - Umožňuje pořádat vícebodové videokonference, přijímá a přeposílá streamy
   - Přijímá a dekóduje příchozí streamy v různých formátech a vytváří příslušný výstupní stream pro koncové uzly
   - Streamy mohou být zasílány asymetricky

   zdroj (ISA-Videokonference-2015.pdf slide 48)

4. otázka (6 bodů) co je to DNS rezoluce, popsat rozdíl iterativní x rekurzivní



Prednaska 2015-10-23@52:50

←**Rekurzívny** nameserver (na obrázku) dostane dotaz na adresu a postupne zisťuje (najprv cz, potom vutbr potom eva) a vráti už hotovú adresu

Iteratívny nameserver urobí len jeden krok (teda zistí napr cz) a vráti adresu, čiže resolver sa musí znova dotázať dalšieho NS.

5. otázka (10 bodů) Netflow: napsat části + každou lehce popsat, popsat protokol, na jaké vrstvě to běží, jaké data zachytává

Části: - Exportér: sonda/router pro získávání statistik o tocích
- Kolektor: zařízení pro ukládání záznamů o tocích
- Nástroje pro zobrazení dat: grafy, statistiky apod.

Běží na vrstvách L2 a vyšších???

Protokol definován na aplikační vrstvě.

Zachytává toky dat - jednotlivé pakety, které se v síti vyskytují

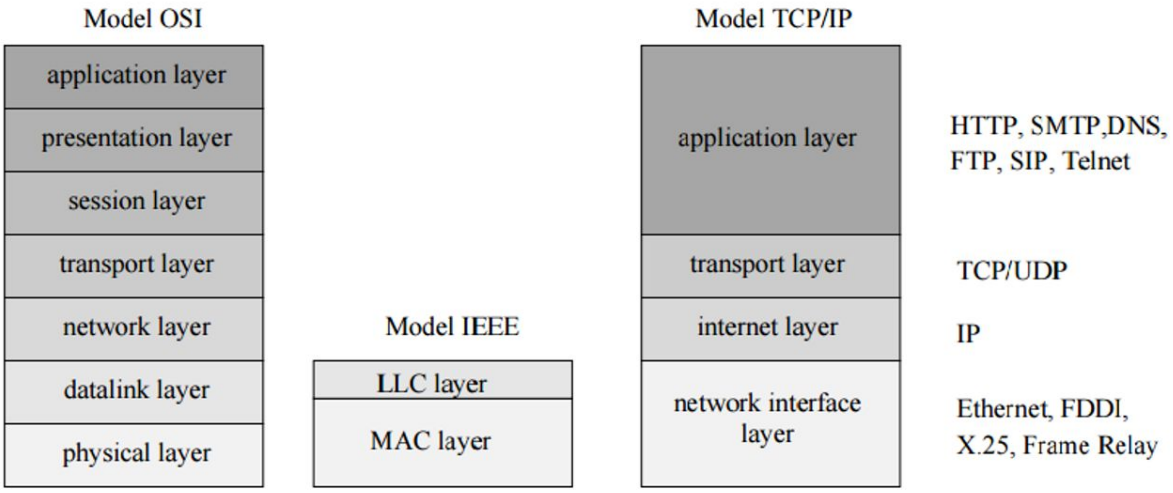
6. otázka (8 bodů) příklad na Tekoucí vědro (vesměs jsem to neviděl, ale stačilo mi umět příklad na Token Bucket z konce slajdů)

7. otázka (12 bodů) definovat vícebitový trie + udělat na 6ti pravidlech s krokem 3 a 5

Teorie

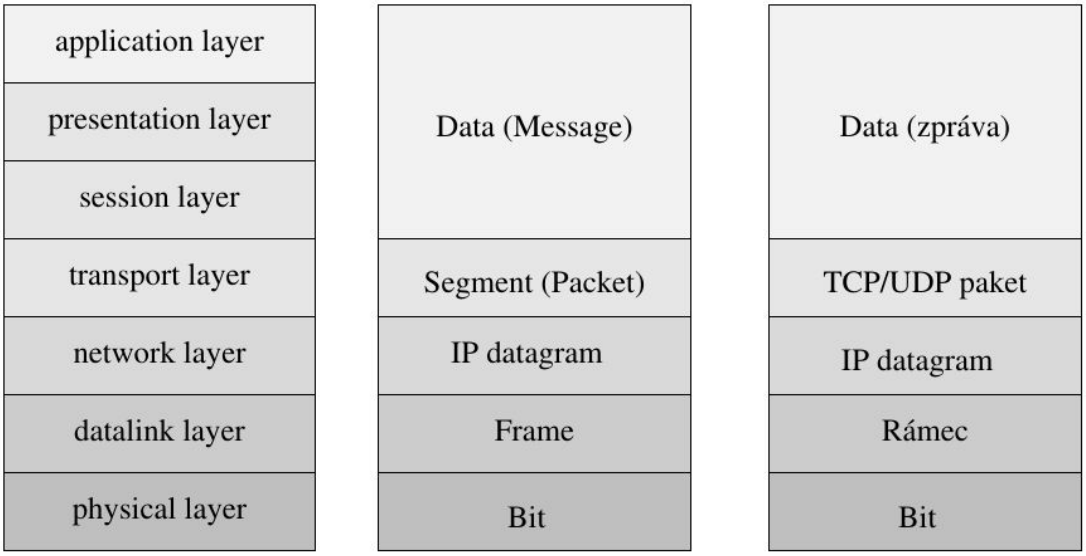
Model síťové architektury

- hierarchie vrstev, definujících služby na jednotlivých vrstvách
- každá vrstva je spojena se skupinou protokolů na ní působících
- pro služby vyšší vrstvy je činnost vrstev nižších transparentní
- komunikující koncové body leží vždy ve stejné vrstvě a využívají pro komunikaci služeb vrstev nižších



Obrázek 1.5: Porovnání modelů OSI, IEEE a TCP/IP

- OSI je konceptuální model původně navržen pro popis síťové architektury, reálně se používá TCP/IP
- application layer: DHCP, DNS, FTP, HTTP, IMAP, LDAP, POP, RTP, RTSP, RIP, SIP SMTP, SNMP, SSH, Telnet, TLS/SSL
- transport layer: TCP, UDP
- internet layer: IP (IPv4 IPv6), ICMP, ICMPv6, IGMP
- network interface (link) layer: ARP, MAC (Ethernet, Frame Relay)
- adresování:
- hw adresa mac - fyzické rozhraní
- ip adresa - ip vrstva
- port - transportní vrstva
- uri - aplikační vrstva



- datové entity v jednotlivých vrstvách

Socket

- schránky, přes které komunikují procesy aplikační vrstvy, jsou identifikovány IP adresou a číslem portu
- pro komunikaci se používají knihovny jako BSD sockets

Fyzické rozhraní

- linková vrstva L2
- 48-bit fyzická adresa např. 00:0c:6e:77:ce:22, jednoznačně identifikuje síťové rozhraní počítače, určena k adresování v lokální síti
- první 3 byty MAC jsou OUI (Organizational Unique Identifier) - identifikátor výrobce
- obsahuje informace o protokolu Protocol Type: 0x0800 (IPv4), 0x086DD (IPv6)

Síťová vrstva

- 32 bit IPv4 adresa např. 208.97.169.68
- 128 bit IPv6 adresa např. 2a00:da80:f::48
- jednoznačně identifikuje počítače v síti

Adresování IPv4

- adresa = adresa sítě + host adresa
- maska sítě udává kolik bitů (zleva) slouží k adresaci sítě
- podle masky dělíme adresy na třídy

- A - 8 bitů
- B - 16 bitů

- C - 24 bitů
- D - multicast
- E - experimenty
- pro lepší využití prostoru je zavedeno beztřídní adresování CIDR, maska sítě proměnlivá

NAT (Network Address Translation)

- mapování a překlad IP adres z jedné skupiny do druhé
- NAT mapování N:1 je mapování N adres vnitřní sítě na jednu adresu vnější sítě, například z domácí sítě na internet
- NAPT je NAT + port, který značí ktonkrétní spojení

ARP

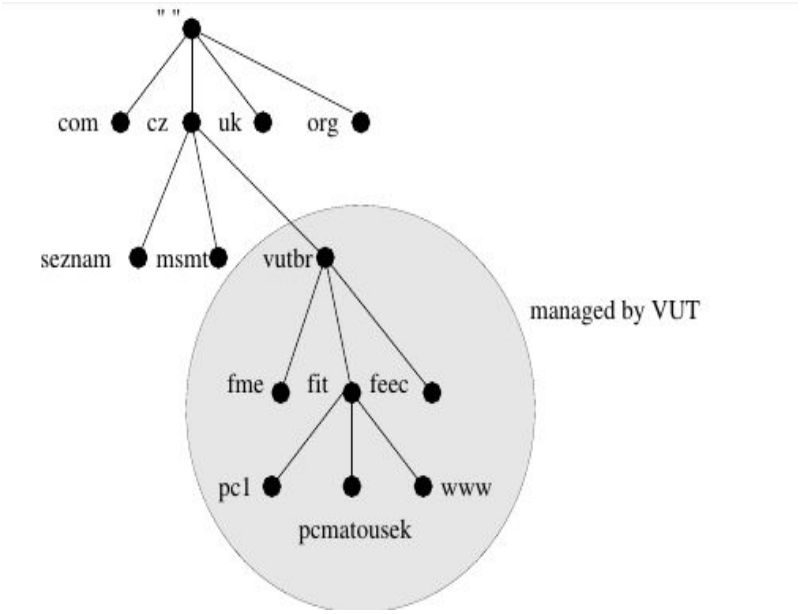
- Address Resolution Protocol
- slouží k získání linkové adresy síťového rozhraní ve stejné podsíti za pomocí IP adresy
- pomocí “arp flooding” lze zaplnit CAM tabulku switche a ten se začne chovat jako hub

Transportní vrstva

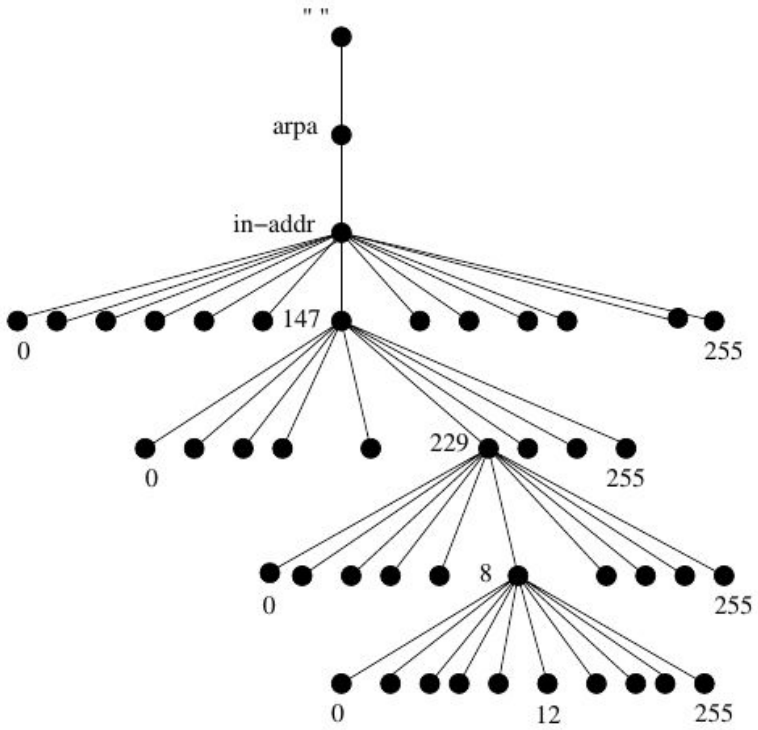
- adresování služby pomocí čísla portu 16 bit (jednoznačné id služby na daném pc)
- spojované TCP a nespojované UDP služby

DNS

- Domain Name System
- služba, která definuje mapování doménových jmen na IP, přístup k datům a uložení a správu dat
- pracuje s hierarchickým prostorem doménových adres
- používá DNS protokol, rezoluce
- využívá distribuované DNS databáze na DNS serverech
- prostor doménových adres reprezentuje stromová struktura



- zóna je fyzická část prostoru pod jednotnou správou, správa zón je decentralizovaná
- pro reverzní mapování (ip na doménu) je speciální podstrom arpa



například hledání 12.8.229.147.in-addr.arpa

- DNS systém koordinuje organizace ICANN (akredituje registrátory doménových jmen)
- registrátoři domén první úrovně - národní (cz,sk,uk...), pro česko CZ-NIC
- IP adresy spravují regionální registrátoři (RIR) na nejvyšší úrovni, pak lokální (LIR) a pak ISP

DNS server

- každý server obsahuje část prostoru doménových jmen - zónu
- primární (master) - úplné autoritativní záznamy o doménách, každá doména má jeden primární server
- sekundární (slave) - autoritativní kopie dat od primárních serverů (zone transfer)
- záložní (caching-only) - pouze přijímá dotazy a předává dalším serverům, ve vyrovnávací paměti má odpovědi uloženy, tak může poskytovat neautoritativní odpovědi (neúplné a neaktuální)

Resolver

- klientský program, který získává info od DNS serveru, součást OS
- rezoluce je proces vyhledávání odpovědi v systému DNS
- rekurzivní dotaz - pokud odpověď nedostane od serveru, tak se ptá dalších serverů
- iterativní dotaz - vrátí nejlepší možnou odpověď

DNS záznamy

- SOA - každá zóna má právě jeden SOA, obsahuje název primárního serveru a email na správce, sériové číslo (serial) - identifikuje změnu záznamu, refresh - interval na zjištění změn (u slave), retry - doba po které slave v případě neúspěšného přenosu zkusí aktualizovat zónu, expire - doba platnosti na slave
- NS - name server - určuje autoritativní server pro danou zónu.
- A - přímé mapování adresy na IPv4
- MX - Mail Exchanger - přesměruje poštu na daný poštovní server
- CNAME - Canonical Name - mapování aliasu na kanonické jméno pc, síťové zařízení má i více aliasů, jsou to symbolická jméno pro servery
- PTR - Domain Name Pointer - mapuje IPv4 na doménu, reverzní mapování přes in-addr.arpa, IPv6 také přes ip6.arpa.
- NAPTR - mapování řetězců na data, obsahuje informace o běžících službách, např SIP
- SRV - Service Record - lokalizace služeb a serverů, například na jakém serveru běží SIP z NAPTR
- TXT - textová data, dodatečné info o doméně atd, taky se používá na ověření vlastnictví, například Google Apps
- AAAA - přímé mapování domény na IPv6

Zone transfer (přesun zón)

- sekundární server využívá schéma vyzývání (polling), závisí na intervalu aktualizace (refresh v SOA)
- IXFR - přírůstkový přenos zón - sekundární server posílá s výzvou záznam SOA, primární zkontroluje databázi změn a pošle rozdílové změny

DNSSEC

- zabezpečení dat DNS pomocí asymetrické kryptografie, rozšíření klasického DNS
- podepisování záznamů DNS v zónách pomocí ZSK (Zone Signing Key)

- podepisování klíčů pro podepsání záznamů pomocí KSK (Key Signing Key)
- chain of trust - řetězec důvěry - každá zóna má veřejný a tajný klíč

Záznam	Význam	Standard
DNSKEY (DNS Key Record)	veřejný klíč pro ověření podpisu	RFC 4034
RRSIG (Resource Record Signature)	podpis pro daný záznam	RFC 4034
DS (Delegation Signer)	potvrzení pravosti klíče v DNSKEY	RFC 4034
NSEC (Next-Secure Record)	odkaz na další záznam v doméně	RFC 4034
NSEC3 (NSEC version 3)	viz NSEC bez procházení zóny	RFC 5155
NSEC3PARAM (NSEC3 parameters)	parametry pro NSEC4	RFC 5155

DPI (Deep packet inspection)

- pokročilé filtrování na úrovni aplikačních protokolů

Klasifikace paketů

- zařazení paketu do dané třídy podle množiny pravidel v tzv. klasifikátoru
- typy porovnávání v klasifikátorech:
- přesné (exact match)
- prefixové
- intervalové

Vyhledávání pravidel

- lineární: nejjednodušší, vhodné pro malý počet pravidel, postupně procházím pravidla a porovnávám
- stromové: binární stromová struktura trie, cesta z uzlu ke kořeni tvoří prefix, mohou být i vícebitové
- bitový vektor: označuje výskyt prefixu v množině pravidel, AND pro shodu se všemi prefixy
- kartézský součin: množina kart. souč. všech dimenzí, součiny pro všechny kombinace, hledáme vektor s nejlepším ohodnocením vůči hledanému

Architektura elektronické pošty

- poštovní klient (UA, User Agent)
- poštovní server (MTA, Mail Transfer Agent)
- komunikační protokoly SMTP, POP3, IMAP, HTTP (Webmail)

Formát zpráv

- původně 7bit ASCII, pak rozšíření MIME (8-bit Multipurpose Internet Mail Extension) pro netextová data a přílohy
    - každých 8 bitů se namapuje na jeden z původních 7
- obálka, hlavička, tělo

SMTP (Simple Mail Transfer Protocol)

- slouží k posílání zpráv, serveru jsou zaslány informace o mailu a ten jej pak přeposílá na další až do cílové schránky

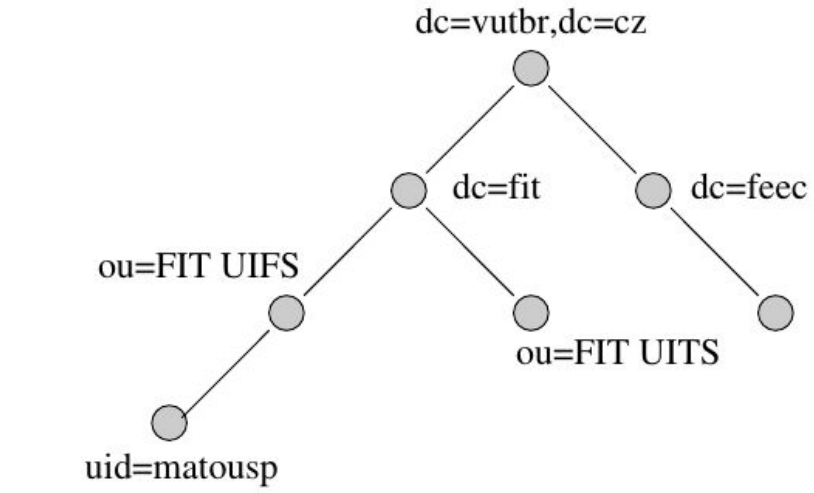
POP3 (Post Office Protocol)
- slouží ke stažení zpráv ze MTA k UA
- pouze jeden klient může přistupovat ke schránce
- obsah je přenesen ke klientovi a aktualizován až při ukončení práce (je smazán na serveru a pak vrácen)
- více schránek lze spravovat pouze lokálně u klienta

IMAP (Internet Message Access Protocol)
- vylepšený POP3
- vícenásobný přístup ke schránkám, možnost synchronizace s více zařízeními
- atributy (Seen, Answered, Recent, Deleted, Flagged)

Zabezpečení mailu
- přenos zpráv mezi servery: SMTP over SSL/TLS (Secure Socket Layer/Transport Layer Security)
- čtení mailu: IMAP over SSL, HTTPS
- zabezpečení obsahu a podepisování: PGP (Pretty Good Privacy) - program pro šifrování a podepisování, S/MIME

Adresářové služby
- elektronická databáze pro vyhledávání uživatelů
- původně podpora emailu, dnes se používá i pro autentizaci uživatelů, autorizaci, uložení údajů
- globální distribuovaný systém, jednotné adresování, různé typy dat (text, adresy, čísla, obrázky)
- filtrování přístupu
- snadná integrace se službami internetu

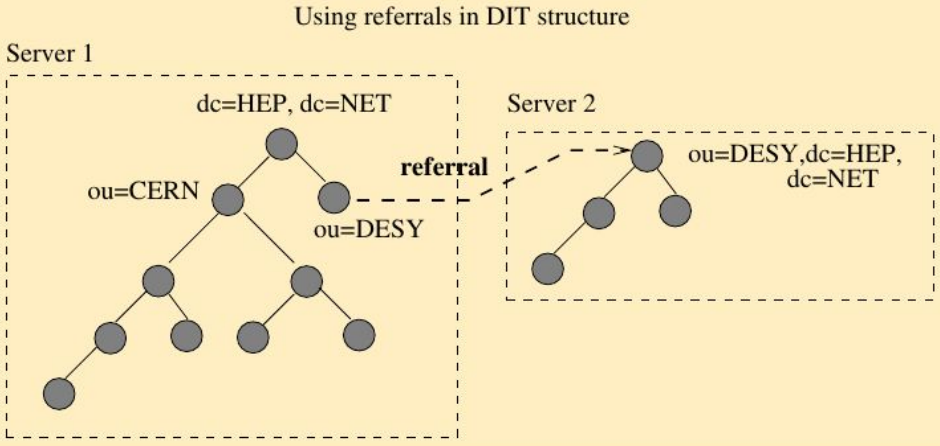
Služba LDAP
- alternativa ke staršímu X.500
- hledání tel čísel, emailových adres, autentizace uživatelů, tvorba certifikátů
- jednodušší implementace, jeden přenosový protokol LDAP
- pro uspořádání dat X.500 koncept: Directory Infromation Tree (DIT)
- záznam (entry) - základní jednotka uložení informace, popsán třídou objektů (např. person), obsahuje seznam atributů (typ, hodnota), jednoznačný identifikátor DN
- záznamy jsou organizovány (jmenný model) ve stromové struktuře DIT (Directory Information Tree) - grafová struktura: vrcholy jsou záznamy, hrany vztahy mezi nimi



DN:uid=matousp,ou=FIT UIFS, dc=fit, dc=vutbr, dc=cz

- referal záznam odkazuje na jiné místo (server) kde je daná sekce LDAP

dn: ou=DESY,dc=HEP,dc=NET
ref: ldap://ldap.desy.de/ou=DESY,dc=HEP,dc=NET



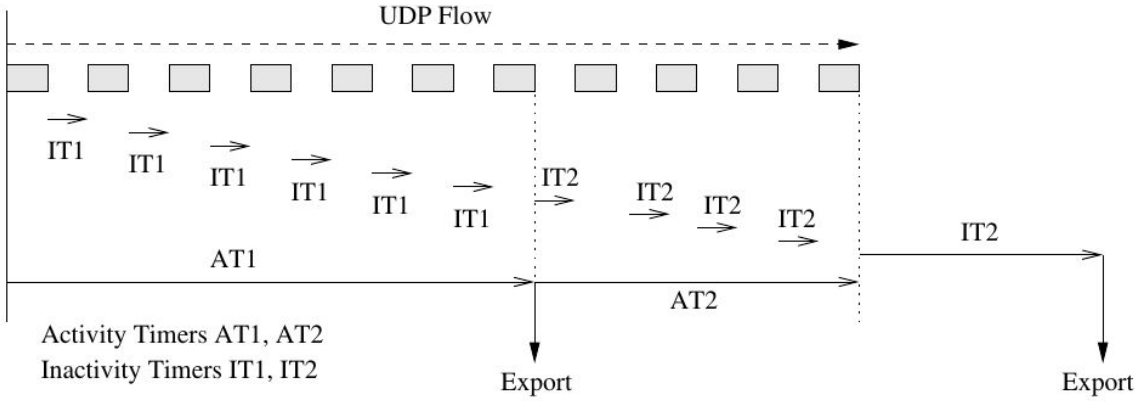
Protokol LDAP

- klient-server, klient tvoří dotaz a server odpoví jednou či více zprávami

Síťový tok

- posloupnost paketů majících společnou vlastnost a procházejících bodem pozorování za určitý časový interval, všechny pakety jednoho toku mají společné vlastnosti odvozené z jejich obsahu
- neaktivní tok - nepřišel žádný paket k danému toku v časovém limitu
- k exportu toku dochází
    - při detekci konce dat (třeba u TCP příznak RST či FIN)
    - neaktivita toku, neaktivní timeout

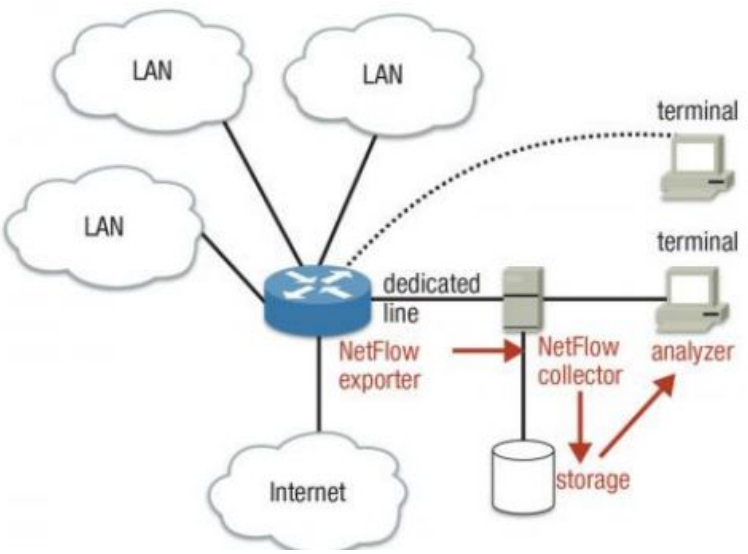
- příliš dlouhý tok, aktivní timeout
- zaplnění netflow cache



-

CISCO NetFlow

- vyvinut firmou CISCO, pro monitorování provozu na síti
- základní prvky:
    - exportér - sonda/router/software pro získávání statistik o tocích, vytváří záznamy flow records, také aktualizuje v netflow cache, expirace, agregace, export - UDP protokol, také využívá vzorkování dat (vybírá jen některé pakety, ne všechny -> menší nároky na hw) - deterministické: pravidelný interval vs náhodné, také filtrování na základě hodnot v hlavičce paketu
    - komunikační protokol NetFlow
    - kolektor - zařízení (software) pro ukládání dat o tocích, přijímá data z 1 či více exportérů, zpracování záznamů, možná agregace, uložení statistik na disk či do databáze, výsledek grafická reprezentace dat
    - nástroje pro zobrazení dat - grafy, statistiky apod.
- exportér a kolektor mají privátní linku, nevyužívají služeb sítě kterou monitorují (když se podělá tak by se to kolektor nedozvěděl)



- využití: monitoring sítě, plánování sítě, bezpečnostní rizika (detekce útoků, odchylky od normálu - virus nebo porucha), dlouhodobé ukládání o přenosech, sledování uživatelů (Skypujou v práci?), účtování (kontrola SLA, přenos mezi ISP - cena)

NetFlow protokol

- verze 5 měla předem určené hodnoty, ty musel exportér dodržet, speciálním datům kolektor nerozumí i když je může ukládat
- verze 9 zavádí šablony, které definují položky monitorování, nejprve se pošle šablona (template flowset) která definuje jak budou vypadat data, poté se posílají data flowset pakety vyplněné podle šablony (šablona se pak ještě občas pošle pro jistotu během přenosu)

Transit provider

- zajišťuje připojení do internetu, typicky pro menší ISP
- transit = konektivita

95/5 percentil

- technika pro měření datového přenosu a následné zpeněžení
- měření provozu není spojité ale vzorkuje se typicky po 5 minutách
- tyto vzorky se na konci měřeného období vezmou a zahodí se vrchních 5%
- ze zbytku se spočítá zatížení a cena přenosu
- špička přenosu se tedy zahazuje, takže uživatel může přesáhnout limit na nějaký čas

Multihoming

- připojení zařízení k více sítím (třeba u více ISP), může být voleno pro lepší spolehlivost sítě, nebo nižší ceny

Peering

- propojení sítí pro výměnu dat, propojením velkých sítí v měřítku planety Země tak vzniká Internet
- typicky bez poplatků
- snižuje náklady na tranzit

IXP (Internet Exchange Point)

- fyzická infrastruktura pro přenášení dat mezi ISP či CDN

Tier 1

- operátor, který má přístup ke všem sítím bez toho aby platil za tranzit (největší sítě propojené mezi sebou aby fungoval net)
- tier 2 platí jen za některé

Content provider

- typicky nepeeruje, jen kupuje tranzit, primárně tvorba obsahu
- velcí jako je Google mají vlastní sítě a peerují

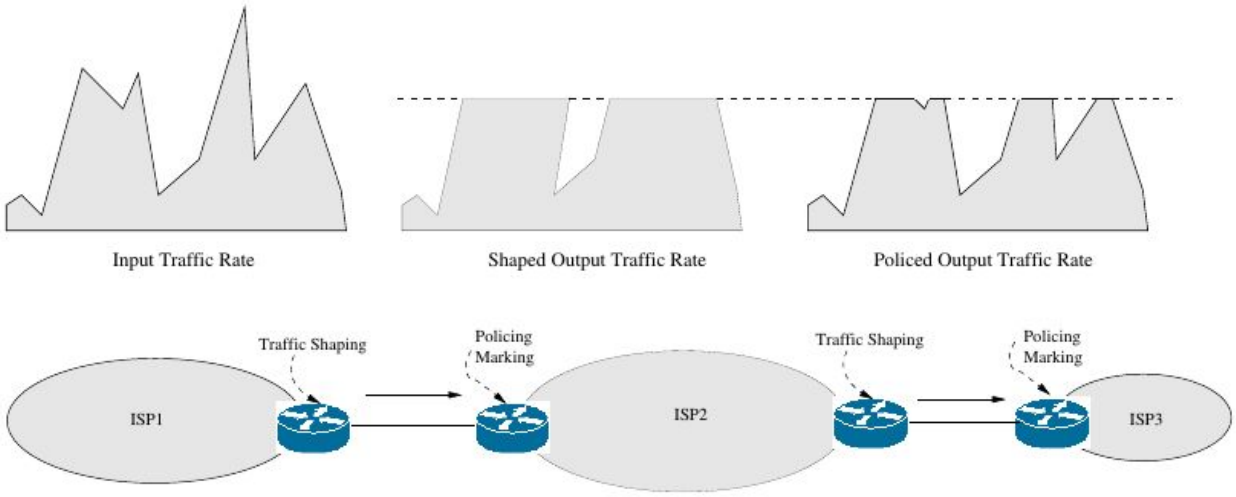
Content delivery network CDN

- sítě poskytující typicky velká data (datová centra a servery připojené k páteřní síti)
- content provideři platí CDN za správu jejich data poskytování těchto dat uživatelům

Zajištění požadavků SLA probíhá pomocí:

- značení paketů
- rozložení provozu (traffic shaping) - slouží k regulaci rychlosti přenosu, shluky paketů se rozprostřou v čase, přenosové pásmo je lépe využito (implementace pomocí leaky bucket či token bucket)
- ořezání provozu (traffic policing) - omezení max rychlosti přenosu, pokud pakety přesáhnou limity (moc paketů naráz, překročí rychlost ve smlouvě) tak je tok ořezán a pakety nad mezí se zahazují

Opora kapitola 8, strana 2:
Rozložení provozu (traffic shaping) … Rozložení provozu lze v síťových zařízeních implementovat pomocí modelu **tekoucího vědra** (Leaky Bucket, viz 8.2.5) či model **Token Bucket** (viz 8.2.6).
Ořezání provozu (traffic policing) … Podobně jako rozložení provozu (traffic shaping) lze i ořezání provozu implementovat pomocí modelu **tekoucího vědra.**
Znamená to, že shaping je token a leaky a policing pouze leaky?



Obrázek 8.2: Rozložení a ořezání provozu v sítích

QoS a fronty pro plánování

- FIFO - řazení paketů jak přicházejí, zpoždění paketů <= velikost fronty/rychlost linky, žádné priority
- PR (priority queues) - více front s různou prioritou (přednost při zpracování), klasifikátor rozděluje pakety do front, riziko vyhladovění že pojede jen více prioritní fronta a zbytek bude na ocet
- RR (round robin, cyklické fronty) - každý tok paketů má svou frontu, cyklická obsluha všech front (stejné objemy přenesených dat), rychlost = rychlost linky/počet toků
- WFQ (weighted fair queues, váhové fronty) - každý tok svoje fronta, váha určuje počet vybraných paketů/bytů z fronty, rychlost fronty i = rychlost linky*(váha/suma vah všech)
- Leaky Bucket (tekoucí vědro) - použití pro shaping, omezí rychlost na maximální hodnotu r, obsahuje FIFO s počítadlem bytů X, každou sekundu X += r, paket o délce Pi se pošle pokud Pi < X a dojde k X -= Pi, čeká dokud se X nezvětší, pokud se paket nevleze do fronty tak je zahozen
- Token Bucket (zásobník žetonů) - paměť pro uložení nevyužitých oprávnění k odeslání dat, narozdíl od LB nereguluje rychlost ale povolí občasné špičky pro velké množství dat, 1 žeton = můžeš poslat 1 byte, CIR - průměrná rychlost, CBS - max velikost shluku dat, PIR - rychlost ve špičce

  Maximální délka trvání špičky T

  Token rate r = CIR

  Bucket size b = CBS (omezuje špičky na maximum bytů v toku za čas T)

  Max. počet bytů na vstupu: $A(t) = b + t \times r = CBS + t \times \frac{CIR}{T}$

Integrované služby

- poskytují integrovanou společnou službu jisté množině požadavků, zajišťují kvalitu služeb nad IP
- rezervace zdrojů RSVP, a vytvoření spojení

RSVP

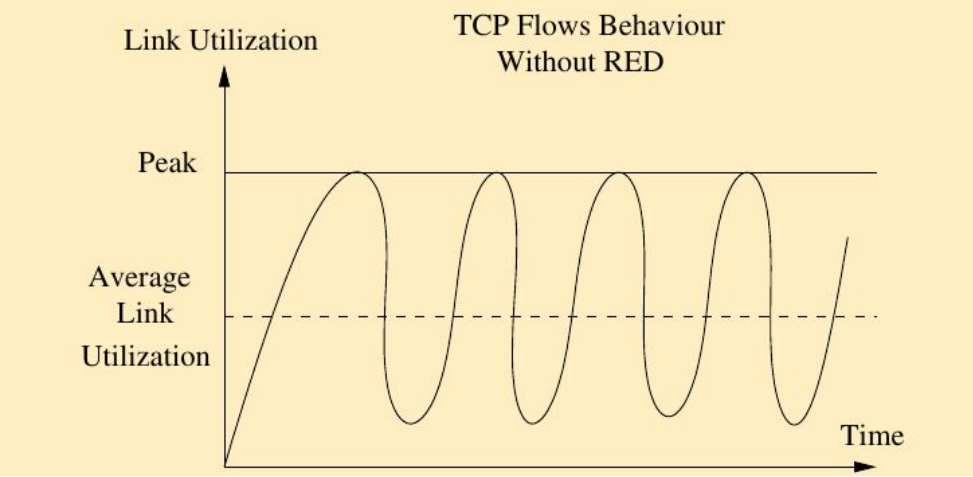
- signalizační protokol transportní vrstvy pro rezervaci zdrojů na síťových prvcích, žádost o rezervaci provádí koncová stanice (receiver oriented)

Diferenciované služby

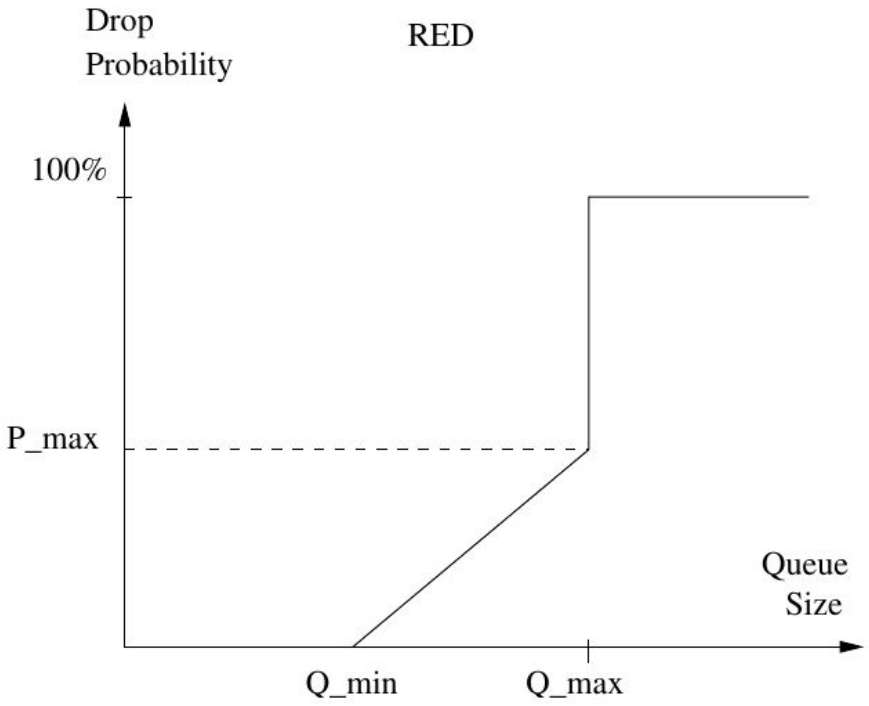
- využívají označení paketů a jejich prioritní přeposílání, rozšiřitelné a flexibilní, vhodné pro páteřní síť
- klasifikace paketů např. podle filtrovacích pravidel, využívá položky IPv4 protokolu ToS (type of service) pro označní priority

RED a WRED

- pokud dojde u TCP k zahlcení linky tak začnou zařízení na síti regulovat provoz, zahazovat pakety atd, provoz zase klesne a jakmile je volno tak zase jde nahoru, takto se to opakuje



- RED (random early detection) - náhodně zahazuje pakety ve vstupní frontě (pst zahození závisí na aktuálním zaplnění fronty)



  - $P_a = P_{max} \frac{Q_{avg} - Q_{min}}{Q_{max} - Q_{min}}$,
- WRED přidává priority

Analýza paketů (packet sniffing)

- proces zachycení dat v reálném čase a jejich interpretace za účelem zjištění, co se děje na síti (charakteristika provozu, identifikace špiček a útoků, nebezpečné aplikace)
- sběr dat -> převedení binárních dat do srozumitelné podoby -> analýza
- wireshark, tcpdump, knihovna libpcap

Možnosti zjišťování stavu sítě

- monitorování pasivní: sledování logů, čekání na události (asynchronní SNMP zprávy,NetFlow záznamy)
- monitorování aktivní: pravidelné testování prvků sítě (ICMP, SNMP, telnet)
- analýza provozu - sledování v reálném čase: wireshark
- analýza provozu - dlouhodobé statistiky: NetFlow

ICMP (Internet Control Message Protocol)

- tento protokol slouží k hlášení chybových stavů po síti
- ping - posílá echo zprávy na cílovou adresu, zpátky dojde echo reply
- traceroute posílá pakety se ~~snižujícím~~ zvyšujicím (na přednášce říkal že je chyba ve slidech) se TTL, zpátky chodí ICMP Time Exceeded Message

Zpráva ICMP	Typ	Popis
Destination Unreachable	3	Cílová síť je nedostupná.
Time Exceeded Message	11	TTL dosáhlo hodnoty 0.
Parameter Problem Message	12	Chyba při zpracování IP hlavičky.
Redirect Message	5	Informace o přesměrování datagramu.
Echo, Echo Reply	8,0	Data posílaná pomocí Echo se vrátí.
Timestamp, Timestamp Reply	13,14	Posílání časového razítka.

FCAPS

- ISO model pro rozdělení činností na síti
- F - fault - správa poruch, test konektivity, odezvy, integrity dat, oprava a izolace poruch
- C - configuration - sledování připojených zařízení, databáze konfigurací těchto zařízení, aktualizace a zálohování
- A - accounting - správa uživatelských poplatků
- P - performance - monitoring odezvy, propustnosti, využití, plánování zdrojů
- S - security - bezpečnost, distribuce klíčů a certifikátů, správa přístupu a oprávnění

SNMP (Simple Network Management Protocol)

- základní prvky:
    - řídíci stanice NMS (network management station)
    - agent MA (management agent)
    - databáze MIB (management information base)
    - protokol SNMP
- protokol pro práci (nad UDP) s monitorovanými objekty, nestavový - dotaz/odpověď
- umí zjistit stav zařízení ale ne celé sítě
- používá objekty v jazyce SMI

- BER - basic encoding rules - definuje reprezentaci hodnot při přenosu, formát TLV (type, length, value)

Podstrom	OID	Položky
system	1.3.6.1.2.1.1	OS name, system time, admin
interface	1.3.6.1.2.1.2	interface status
at	1.3.6.1.2.1.3	address translation (rarely used)
ip	1.3.6.1.2.1.4	IP address, routing information
icmp	1.3.6.1.2.1.5	ICMP statistics
tcp	1.3.6.1.2.1.6	TCP connections: closed, listen, synSent
udp	1.3.6.1.2.1.7	UDP statistics – sent/received packets
egp	1.3.6.1.2.1.8	EGP statistics
transmission	1.3.6.1.2.1.10	medium-dependable objects
snmp	1.3.6.1.2.1.11	sent/received SNMP packets

RMON (Remote Monitoring)

- navrženo pro monitoring LAN, komplexní pohled na celou síť
- offline zpracování nezávislé na NMS, analýza dat, nepřetržité logování a diagnostika
- složky: sonda RMON, řídící stanice NMS, protokol SNMP
- sonda se naučí MAC všech zařízení na svém rozhraní a pro každou dělá záznam

Syslog

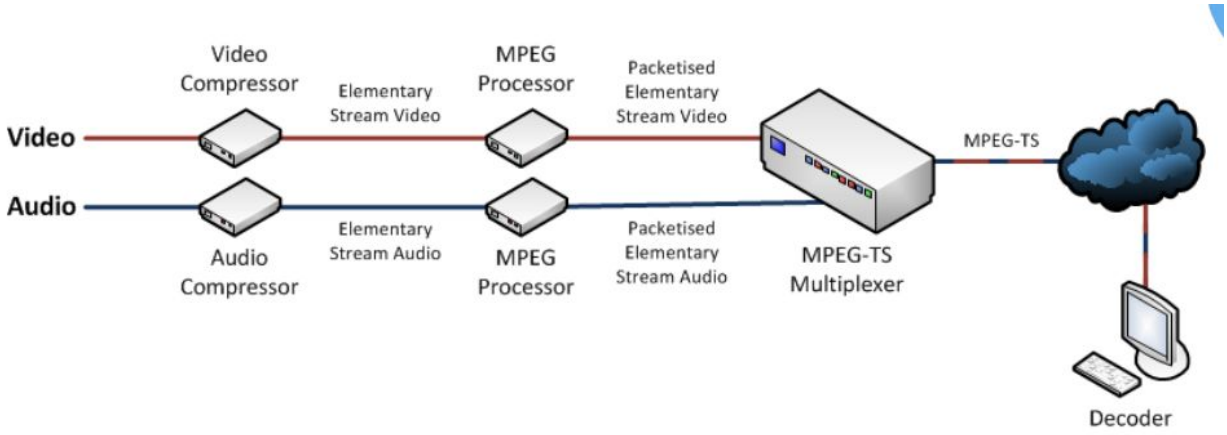
- nástroj pro logování aplikací a zařízení
- zapisuje aktivitu, chyby atd
- konzole (agent) - unix program, nebo sw na síťovém zařízení, posílá události serveru syslog
- zpráva syslog: čas, typ zařízení, důležitost, textový popis

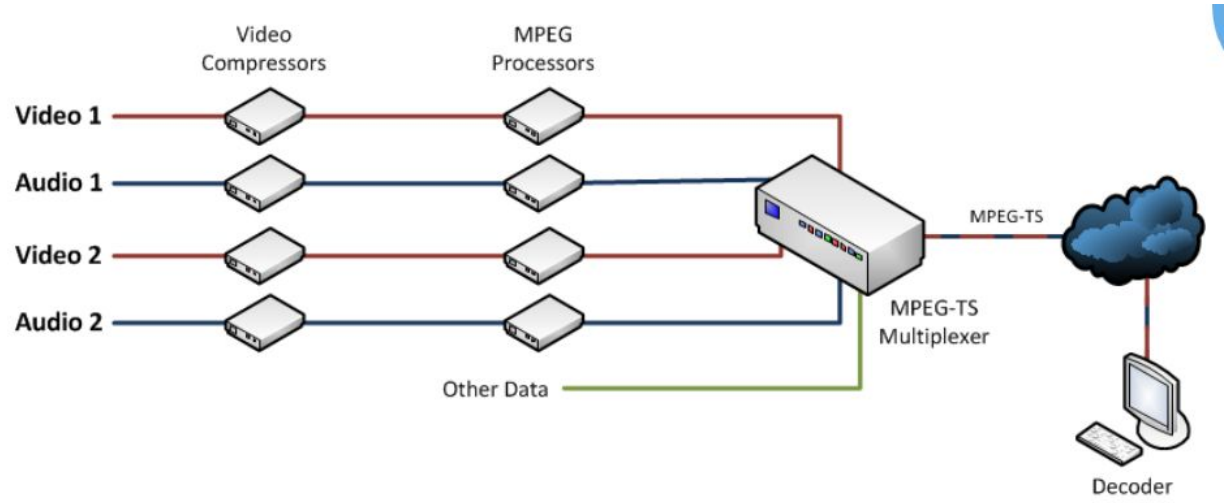
Streaming multimédií

- přenos multimediálních dat po sieti v reálném čase přehrávání
- živý (live) - všem přehrávačům v jeden okamžik stejná data, nelze řídit
- na vyžádání (on demand) - klient přijímá předem vytvoření obsah, může řidit

MPEG Transport System

- standard popisující jak jsou části multimediálního obsahu kombinovány do jendoho toku
- vhodné za předpokladu, že se často ztrácení data (po síti)





- sekvence transport paketů o pevné délce (188 B - 4 hlavička, 1. byte - sync byte, PID - packet identifier - označuje jednotlivé streamy (audio, video atd))
- také jsou zasílána metadata PAT - program association table (tabulka programů na TV) a PMT - program map table (na jakém PID je jaký stream)

RTSP (Real TIme Streaming Protocol)

- signalizační (jako SIP, FTP…), navazuje a končí spojení, řídí toky (dálkový ovladač)
- textový, podobný http
- ouf of band - samotná data doručuje jiný protokol (RTP)
- stateful - udržuje si stav
- používá se v on-demand spojení

RTP (Real-time Transport Protocol)

- standard pro přenos audio/video v realtime
- podpora unicastu i multicastu
- UDP
- každý media stream má svůj RTP tok

RTCP (Real-time Control Protocol)

- řídíci protokol RTP, info o datovém toku, synchronizace a info o kvalitě přenosu

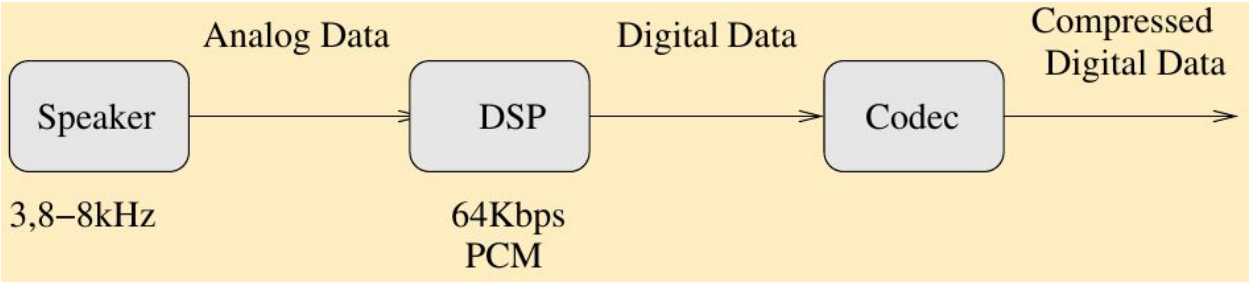
H.323

- doporučení definující protokoly pro multimediální komunikaci v paketově orientovaných sítích (kde nemusí být garantována QoS)
- součásti:
- terminal - koncové zařízení, zajišťuje obousměrnou komunikaci v realtime (videokamera, obrazovka, mikrofon, repráky, kodek, rozhraní)
- MCU - multipoint control unit - podpora vícebodových konferencí, přeposílá streamy
- gateway - koncové zařízení v síti, přepíná okruhy atd
- gatekeeper - překlad adres, řízení velikosti toků, přístupu

IP Telefonie

- převod hlasu na ip datagramy
- komunikace řízena přes ústřednu (registrace, adresování, směrování, udržování spojení a vytváření hovorů)
- aplikační protokoly: DHCP, DNS, IM, WWW, LDAP
- přenosové: H.323, SIP, H.248, RTP/RTCP

Kódování hlasu



- vzorkování signálu - kvantifikace vzorků - kódování do binary - komprese

RTP hlavička (12 B), UDP (8 B), IP (20 B)
Ethernet (18 B), Frame Relay (6 B)
IPSec transport (30-53 B), IPSec tunel (50-73)

ACR (Absolute Category Rating)

- pětibodová stupnice podle poslechu

MOS	Kvalita řeči	Stupeň zkreslení
5	vynikající	nepostřehnutelné
4	dobrá	postřehnutelné, málo znepokojující
3	průměrná	postřehnutelné, více znepokojující
2	slabá	znepokojující, ještě přijatelné
1	neuspokojivá	velmi znepokojující, nepřijatelné

SIP

- signalizace VoIP, vytváření a udržování relace, adresace pomocí URI, registrace uživatele, směrování hovorů, navazování spojení
- server UAS a klient UAC

1. registrace
2. ustavení spojení invite

Příklady

Vypočtěte potřebnou šířku přenosového pásma pro jeden telefonní hovor VoIP při použití kodeku G.711. Telefon posílá hlasové rámce každých 20 ms. Režie dat je 58 bytů.

Šířka přenosového pásma kodeku

- Kódování G.711 (PCM): 8000 vzorků/s, každý vzorek 8 bitů
- Požadované pásmo: 8 kHz $\times$ 8 bitů $=$ 64 kb/s

Velikost vzorku v paketu

- Jeden rámec se vzorkem (PDU) poslán každých 20 ms
- Velikost takového vzorku: 20 ms $\times$ 64 kb/s $=$ 1 280 bitů $=$ 160 Bytů

Potřebné přenosové pásmo pro PDU

- Zapouzdření: RTP (12), UDP (8), IP (20), Ethernet (18), tj. režie 58 B
- Paketů za sekundu (PPS): 64 kb/s / 1280 bitů $=$ 50
- Celkové přenosové pásmo: (58+160) $\times$ 8 $\times$ 50 $=$ 87 200 b/s $\doteq$ 87 kb/s

Pls proč se na konci násobí osmi ? rezie i velikost vzorku je v bytech, takze je to prevod na bity

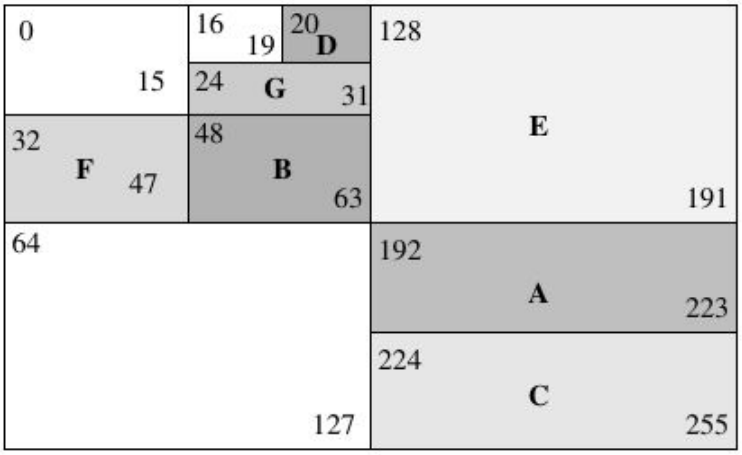
Subnetting

Navrhněte rozdělení počítačů v lokální síti s adresou 192.168.1.0 do 7 podsítí aby podsíť A - 21 pc, B - 9 pc, C - 29 pc, D - 2 pc, E - 47 pc, F - 10 pc, G - 5 pc.

Každá podsíť musí být schopna adresovat daný počet pc + adresu podsítě a broadcast.
Zarovnám na mocniny 2 (formát ip adresy).

Název podsítě	*A*	*B*	*C*	*D*	*E*	*F*	*G*
Počet počítačů	*21*	*9*	*29*	*2*	*47*	*10*	*5*
Zarovnání	*32*	*16*	*32*	*4*	*64*	*16*	*8*

Vytvořím tabulku pro prostor. Vezmu největší podsíť, pokud v tabulce je okno této velikosti tak vložím jinak půlím a to samé.



Název podsítě	A	B	C	D
Počet počítačů	21	9	29	2
Adresa nové podsítě	192.168.1.192/27	192.168.1.48/28	192.168.1.224/27	192.168.1.20/30
Adresa broadcastu	192.168.1.223/27	192.168.1.63/28	192.168.1.255/27	192.168.1.23/30
Začátek adres.prostoru	192.168.1.193/27	192.168.1.49/28	192.168.1.225/27	192.168.1.21/30
Konec adres.prostoru	192.168.1.2228/27	192.168.1.62/28	192.168.1.254/27	192.168.1.22/30

Název podsítě	E	F	G	
Počet počítačů	47	10	5	
Adresa nové podsítě	192.168.1.128/26	192.168.1.32/28	192.168.1.24/29	
Adresa broadcastu	192.168.1.191/26	192.168.1.47/28	192.168.1.31/29	
Začátek adres.prostoru	192.168.1.128/26	192.168.1.31/28	192.168.1.25/29	
Konec adres.prostoru	192.168.1.190/26	192.168.1.46/28	192.168.1.30/29	

Vytváření podsítí, 192.168.0.1 s maskou 22.

Rozepíšu si IP binárně a pod ni masku binárně.

11000000 10101000 000000**00** 00000001

11111111 11111111 111111**00** 00000000

Kde jsou 1 u masky, tam jsou bity adresy sítě (červeně).
Adresa sítě je tedy 192.168.0.0/22
Minimální adresa zařízení je o jedno větší: 192.168.0.1
Broadcast je nejvyšší adresa (samé 1 na konci (tam kde není maska 1)): 192.168.3.255
Maximální adresa zařízení je o jedno menší: 192.168.3.254

Jakou MAC adresu má multicastová IP 227.138.0.1?
Při převodu IP multicastu na MAC je nutné zachovat následující pravidlo:
Nová MAC adresa multicastu se skládá ze tří částí

- prefix definovaný standardem IEEE: 01:00:5e
- jednoho 0 bitu za prefixem
- zbylých 23 bitů sebraných od konce příslušné IP

Takže převedu IP na binární:
11100011 10001010 00000000 00000001
Vezmu posledních 23 bitů
Doplním o 0 na 24. místo od konce
00001010 00000000 00000001
Sestavím z prefixu a vybraných bitů s 0 adresu (hexadecimálně)
01:00:5e:0A:00:01
btw broadcast je mapovaný na MAC ff:ff:ff:ff:ff:ff

Úkoly k procvičení

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

1. Pro zadaná filtrovací pravidla vytvořte dvoudimenzionální binární strukturu trie (a) se zpětným vyhledáváním a bez duplicit, (b) se zpětným vyhledáváním a s ukazateli.

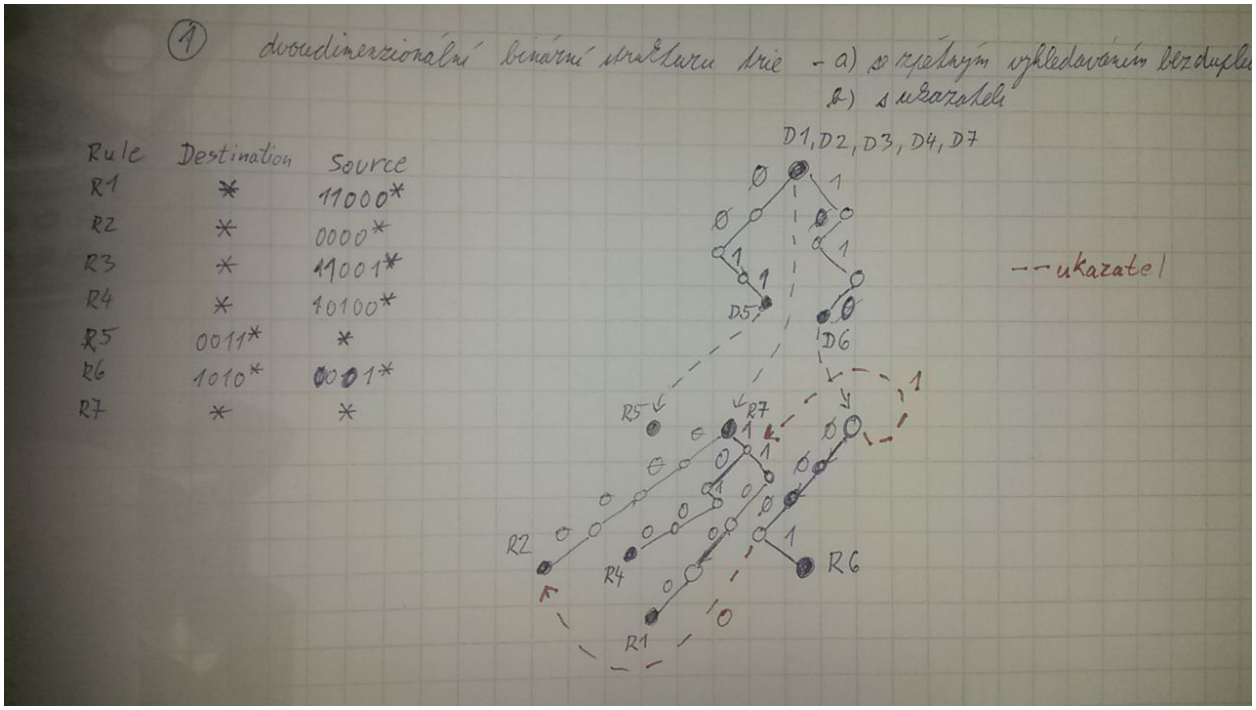
```
R1: deny ip from 194.0.0.0/5 to any
R2: deny ip from 0.0.0.0/4 to any
R3: deny ip from 200.0.0.0/5 to any
R4: deny ip from 160.0.0.0/5 to any
R5: permit ip from any to 48.0.0.0/4
R6: permit ip from 160.1.0.0/4 to 16.0.0.0/4
R7: deny ip from any to any
```

2. Implementujte níže uvedená pravidla firewallu (a) pomocí bitového vektoru, (b) pomocí kartézského součinu rekurzivní klasifikace RFC s třídami ekvivalence (srcIP-dstIP) a (SrcPort-DstPort).

```
R1: permit TCP from 147.229.0.0 to any dst-port 80
R2: permit UDP from 147.229.0.0 to any dst-port 53
R3: permit UDP from any to 147.229.0.0 src-port 53
R4: permit ICMP from 147.229.0.0 to any
R5: deny ICMP from 147.229.1.15 to any
R6: deny IP from any to any
```


Poznámka: ty pravidla v binárním formátu se získají zápisem adresy do binárního formátu a useknutí daným prefixem.

prosím o kontrolu prvního příkladu (před R6 se jedná o prázdné kolečka a ty zahnuté čárkované čáry z větve D6 jsou ukazatele)



chybí R3

otázka: není R6 prohozené? Destination = 0001* Source = 1010*? - taky se mi zda 3x

otázka: je chyba pokud bych to udělal naopak? v 1. dimenzi dělal source a ve druhé destination? (na pořadí dimenzí nezáleží 1x)

CO znamena prosim to ze bez duplicit azpetnym vyhledavanim (akoze normalne bez ukazatelov?)?

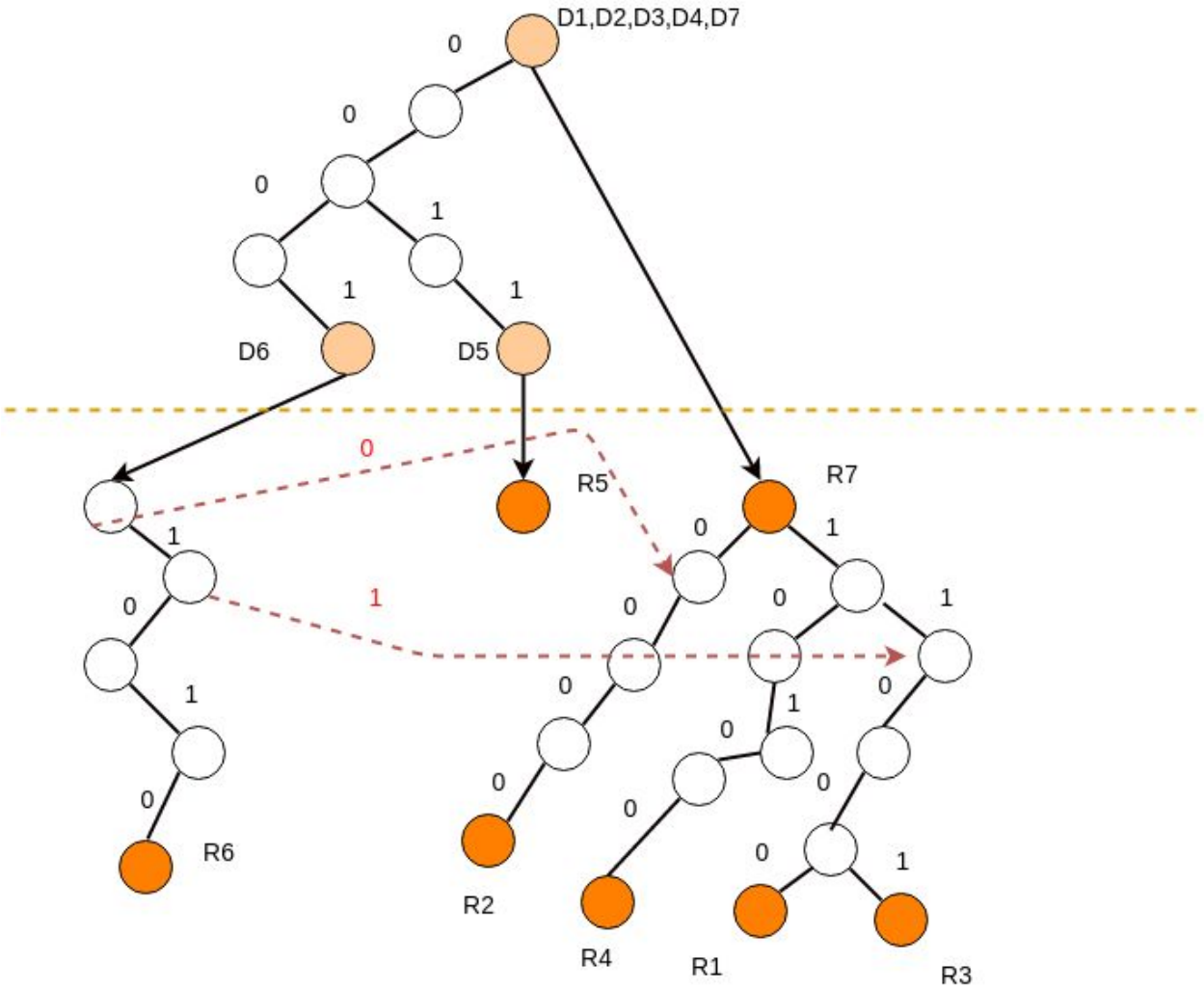
zpětné vyhledávání je že by ses vracel zpátky nahoru ve stromu a pak šel znova, ukazatele to ruší

duplicity jsou myslím že by se pro každý podstrom musel dělat celý, jakoby tam nebyla ta druhá úroveň ale rozepisovalo by se to zvlášť pro každé pravidlo DIIIIIIKYYYYY

Takto? (Mám to stejně 3x)

RULE	DST	SRC
R1	*	11000*
R2	*	0000*
R3	*	11001*
R4	*	10100*

R5	0011*	*
R6	0001*	1010*
R7	*	*



Podle čeho je ta délka prefixů? Když se třeba liší na 5. bitu chápu, ale jinak pokud tam není žádná změna, tak by R4 a R6 source mohla mít jen 101* ne? Pokud teda není definované, že se má zarovnávat na 3 a 5.
Délka prefixů je ze zadání výše (R4 má /5 a R6 má /4).

Proč ten ukazetel s ohodnocemím 1 jde tam kam jde a nejde o jednu úroveň víš (do té 1)?
no protože on může jít vždy jen o úroveň níž, to bych se vracel kdybych šel víš
jakoby ta levá strana (začátek šipky) je v té src sekci v posloupnosti 1, ta pravá je taky 1, proto další prvek bude 1 1 (ten chybí nalevo)
ten ukazatel ukazuje tam kam musím jít DÁL když nemůžu v tom konkrétním podstromu

IP Lookup - longest prefix match

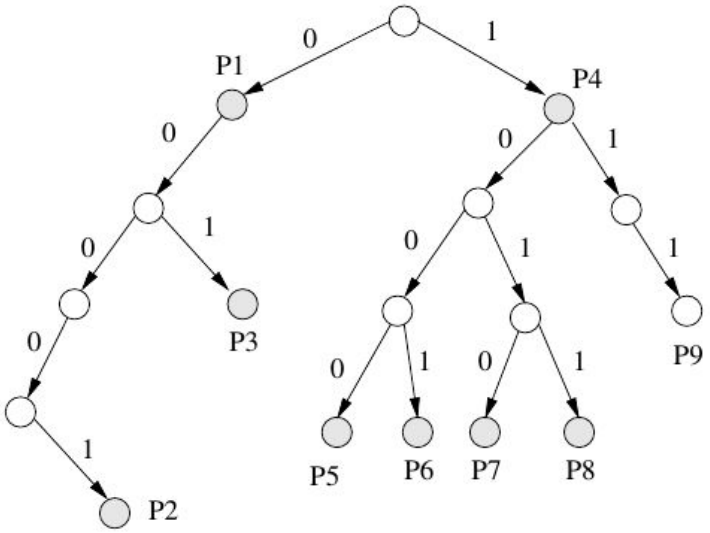
Destination	Gateway	Flags	Refs	Use	Netif	Expire
default	147.229.176.1	UGS	0	202426574	em1	
10.10.10.0/24	link#1	U	0	23333292	em0	
10.10.10.1	link#1	UHS	0	0	lo0	
10.10.11.0/24	link#1	U	0	8865	em0	
10.10.11.1	link#1	UHS	0	11	lo0	
10.10.12.0/24	link#1	U	0	8865	em0	
10.10.12.1	link#1	UHS	0	0	lo0	
127.0.0.1	link#4	UH	0	559933	lo0	
147.229.176.0/23	link#2	U	0	12730	em1	
147.229.176.18	link#2	UHS	0	2904	lo0	

- Otázka: Kam se přepošle datagram s cílovou adresou 147.229.8.12?

Jediné pravidlo které dá match je default.
Hledám nejdelší prefix, který sedí na danou adresu. Například kdyby byla adresa 147.229.176.0/16 tak to sedí. U longest prefix match nezáleží na pořadí pravidel. U obyčejného filtrování beru první vhodné pravidlo.
otázka: S čím 147.229.176.0/16 sedí? vždyť to je uplně jiná síť než 147.229.176.0/23 ne? Tzn. pokud bychom vyhledávali třeba pomocí trie, tak bychom k té předposlední možnosti ani nedošli, protože /16 by měla jen 16b a stejně bychom skončili na default.
asi máš pravdu, hm a kdyby tam bylo 147.229.176.0/x kde x=>23 tak to seí ne?
u 147.229.176.0/16 by s tou původní adresou 147.229.8.12 sedělo těch prvních 16b, protože se liší až na 17. místě, takže kdyby bylo předposlední pravidlo /16. tak by to prošlo

Jaká je doba zpracování paketu o velikosti 40 B na síti s rychlostí 40 Gb/s.
vzpomínky na fyziku s = v*t :D
t = s/v = 40*8/40*10^9 = 8 ns

Sestavte trie pro zadanou množinu pravidel a vyhledejte paket 01001100 a 0001.



P1: 0*
P2: 00001*
P3: 001*
P4: 1*
P5: 1000*
P6: 1001*
P7: 1010*
P8: 1011*
P9: 111*

0 jdou doleva, 1 doprava, cesta z kořene do listu dává pravidlo
Pokud dostanu paket například 01001100 tak začnu hledat od kořene. Jdu nalevo (první 0), pak máme 1 ale taková cesta není, končím hledání a pravidlo které aplikuji je P1.
Dále paket 0001, jdu vlevo,vlevo,vlevo, jednička nikde není v aktuálním uzlu, vrátím se a hledám nejbližší pravidlo (kterým jsem prošel), tedy P1.

Rozšíření prefixů na krok 3 a 5 bitů a sestavení vícebitového trie.

Pravidlo	Původní prefix	Rozšířené prefixy
P_1	$0*$	$000*, 010*, 011*$
P_2	$00001*$	$00001*$
P_3	$001*$	$001*$
P_4	$1*$	$100*, 101*, 110*$
P_5	$1000*$	$10000*, 10001*$
P_6	$1001*$	$10010*, 10011*$
P_7	$1010*$	$10100*, 10101*$
P_8	$1011*$	$10110*, 10111*$
P_9	$111*$	$111*$

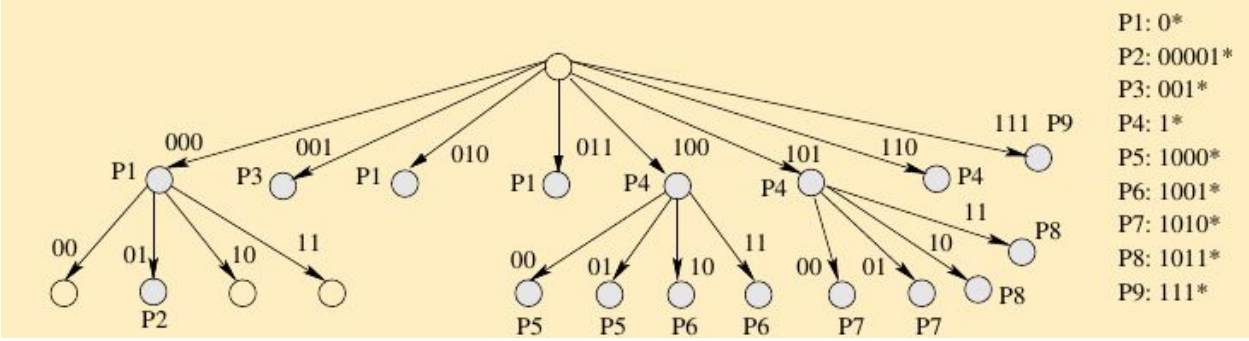
Podle počtu bitů v původním pravidle rozhodnu na jakou mez zarovnáme. V situaci zarovnání na 3 voda 5 zarovnám všechny x<3 na 3 a všechny 3<x<5 na 5. Pokud pravidlo již má 3, nebo 5 tak ho nechám být.
Vezmu vždy původní a rozepíšu všechny možné rozšíření na 3 a 5 bitů, například P1:
0* můžu rozšířit na 3:

- 000*, 001*, 010*, 011*

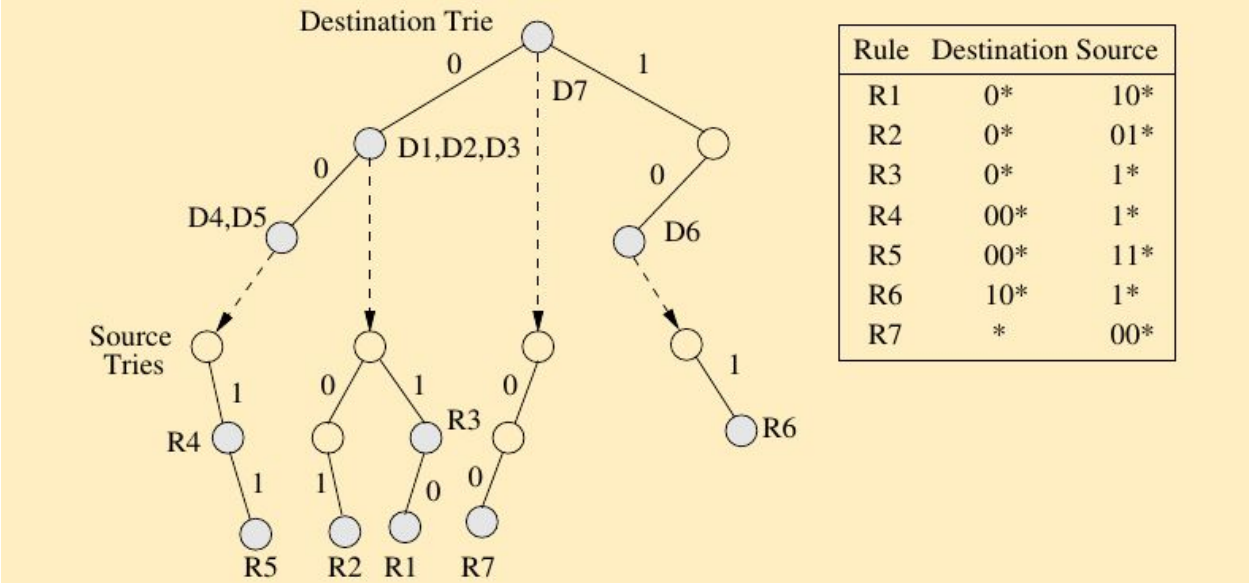
V tabulce původních ale už máme 001*, takže to vyhodíme.

Výsledek pro P1 na 3 = {000*,010*,011*}

…



Sestavení 2D trie. (bez duplicit se zpětným vyhledáváním)

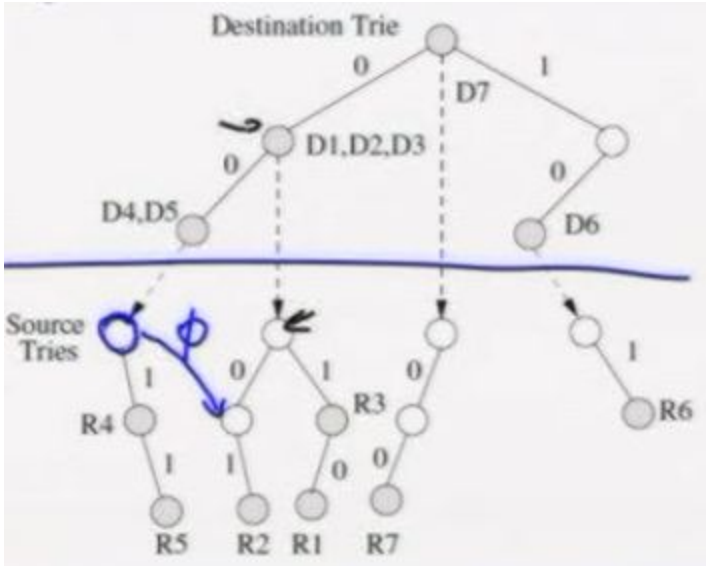


Rule	Destination	Source
R1	0*	10*
R2	0*	01*
R3	0*	1*
R4	00*	1*
R5	00*	11*
R6	10*	1*
R7	*	00*

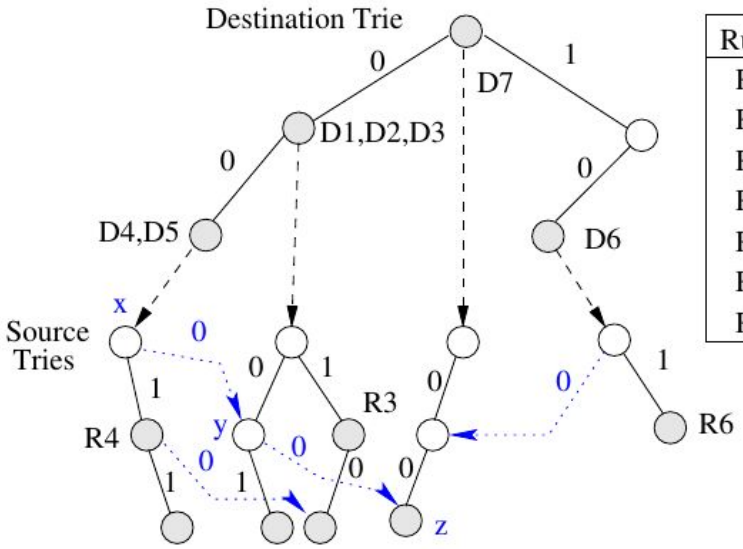
Nejprve vytvořím strom v 1D podle první sady pravidel pro destination.
D označují destination část a R už celé pravidlo, D+S
Potom dělám jednotlivé stromy pro druhou dimenzi, tyto stromy navážu na D uzly podle tabulky.
Při vyhledávání se používá backtracking, nenajdu pravidlo tak se vracím.
Paket 001,001:
jdu vlevo,vlevo, dál nemůžu takže první dimenze dává D4 nebo D5 (00*)
jdu do 2. dimenze (přerušovanou čarou z D4 a D5), žádná 0 tam není, musím se vrátit o krok zpátky -> D1,D2,D3 uzel (0*, sice zkrátím o jednu 0 ale je tam * takže to je platné, snažím se najít nejdelší prefix)
zde jdu vlevo ale další 0 není tak se zase vracím -> D7
tady mám match na 0,0 -> R7

U trie se zpětným vyhledávání a ukazateli přidám přechody, které by vznikly, kdybych se vracel
Když se dostanu do uzlu v druhé dimenzi tak se ptám zda existuje obecnější pravidlo.

Například v bodě za D4 a D5 (za přerušovanou šipkou) vím že bych se vrátil do D1,D2,D3 a od tama pokračoval dolů a doleva , tím pádem tam přidám ukazatel abych se nemusel vracet, ale mohl jít přímo.

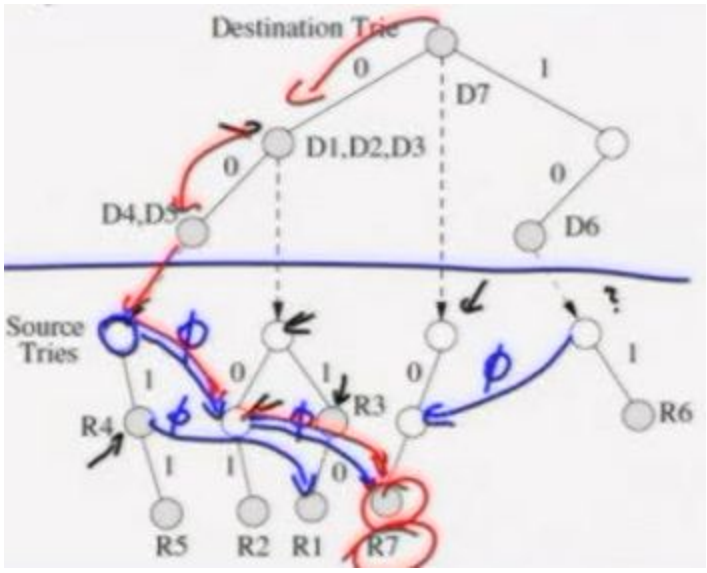


To udělám pro všechny uzly v druhé dimenzi. Řeknu si třeba u R4, je tu hrana označující pravidlo 10? Není a je jinde? Jo je, u R1, tak tam udělám šipku s 0.
Testuju prostě všechny uzly, kde chybí nějaká hrana.
Nemůžu například spojit R3 s R5 jako 11, protože pravidlo se kterým spojuji musí být odvozeno vždy z obecnějšího (v první dimenzi musí být blíže kořen), taky nelze spojovat uzly zpětnými ukazateli v rámci jednoho podstromu druhé dimenze
Výsledek je:



Rule	Destination	Source
R1	0*	10*
R2	0*	01*
R3	0*	1*
R4	00*	1*
R5	00*	11*
R6	10*	1*
R7	*	00*

Vyhledávání 001,001 by vypadalo takto:



Lineární vyhledávání pomocí bitového vektoru
Pravidla
..

	Dst IP	Src IP	Dst port	Src port	Proto	Action
1	147.229.*.*	*	25	*	*	permit
2	147.229.*.*	*	53	*	UDP	permit
3	147.229.*.*	*	22	*	TCP	permit
4	147.229.5.1	153.13.2.5	123	123	UDP	permit
5	*	117.16.*.*	*	*	IP	permit
6	*	*	*	*	*	deny

Chci vyhledat H=(147.229.5.1, 147.228.15.1, 53, 1029, UDP)
Každá dimenze (položka jako ip adresa, port atd) bude mít svůj bit vektor, svoji tabulku. Do této tabulky si umístím jedinečné položky z pravidel.
Například Dst IP: tři různé hodnoty:

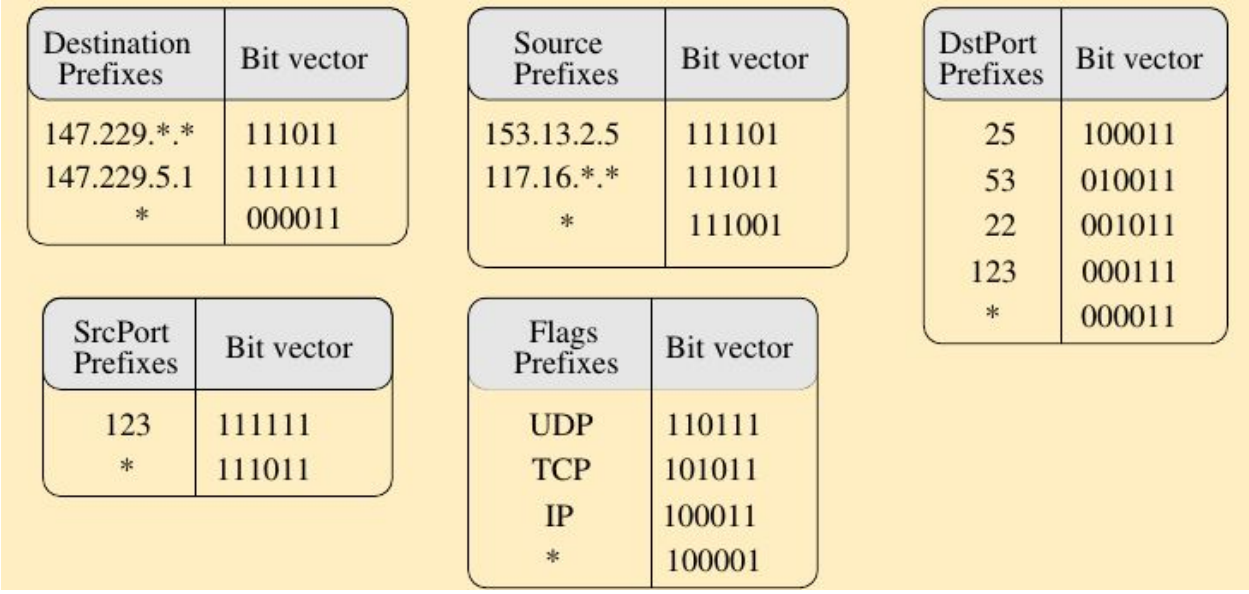
Destination Prefixes
147.229.*.*
147.229.5.1
*

Teď si udělám bitový vektor (který má délku podle počtu pravidel, tedy 6) pro každý řádek tabulky. Jinými slovy vezmu hodnotu destination prefix z tabulky a zkusím ji napasovat na pravidla. Hodnota dest. prefix musí být podmnožinou daného pravidla.

P1: 147.229.*.* srovnávám s 147.229.*.* => jo => 1
P2: 147.229.*.* srovnávám s 147.229.*.* => jo => 1
P3: 147.229.*.* srovnávám s 147.229.*.* => jo => 1
P4: 147.229.5.1 srovnávám s 147.229.*.* => ne => 0 (naopak ano ale my chceme aby hodnota byla podmnožinou pravidla)
P5: * srovnávám s 147.229.*.* => jo => 1
P5: * srovnávám s 147.229.*.* => jo => 1
Bit vektor je tedy 111011

Takto postupuju pro všechny:

Bacha na chyták, UDP a TCP se bere jako podmnožina IP.
Toto se vytvoří ještě před klasifikací samotnou.
Teď vezmu zadání: **H=(147.229.5.1, 147.228.15.1, 53, 1029, UDP**



Destination Prefixes	Bit vector
147.229.*.*	111011
147.229.5.1	111111
*	000011

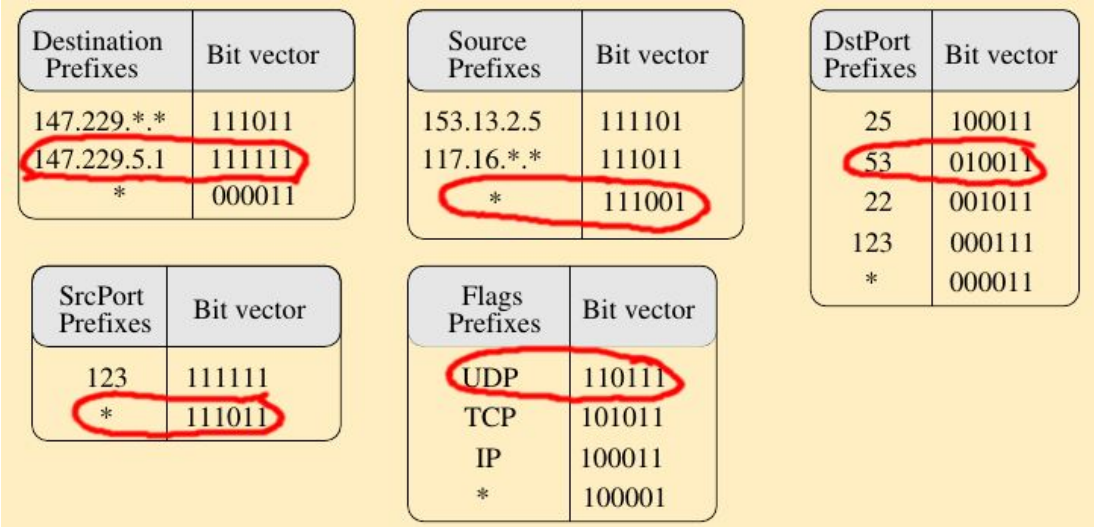
Source Prefixes	Bit vector
153.13.2.5	111101
117.16.*.*	111011
*	111001

DstPort Prefixes	Bit vector
25	100011
53	010011
22	001011
123	000111
*	000011

SrcPort Prefixes	Bit vector
123	111111
*	111011

Flags Prefixes	Bit vector
UDP	110111
TCP	101011
IP	100011
*	100001

)
Technika divide and conquer
Vezmu první dimenzi 147.229.5.1 a najdu bitový vektor odpovídající této hodnotě (**nej prefix)**



Destination Prefixes	Bit vector
147.229.*.*	111011
147.229.5.1	111111
*	000011

Source Prefixes	Bit vector
153.13.2.5	111101
117.16.*.*	111011
*	111001

DstPort Prefixes	Bit vector
25	100011
53	010011
22	001011
123	000111
*	000011

SrcPort Prefixes	Bit vector
123	111111
*	111011

Flags Prefixes	Bit vector
UDP	110111
TCP	101011
IP	100011
*	100001

Z nich vezmu příslušné bit vektory a poANDuju
111111 AND 111001 AND 010011 AND 111011 AND 110111 = 010001
=> použiju pravidlo 2, nebo 6 (podle toho zda vybírám podle best prefix, nebo první vhodné)

Implementujte níže zapsaná pravidla firewallu:

```
R1: permit TCP from 147.229.0.0 to any dst-port 80
R2: permit UDP from 147.229.0.0 to any dst-port 53
R3: permit UDP from any to 147.229.0.0 src-port 53
R4: permit ICMP from 147.229.0.0 to any
R5: deny ICMP from 147.229.1.15 to any
R6: deny IP from any to any
```

a) **pomocí bit vektoru** **prosím o kontrolu**
mám to stejně 4x

SourceIP	
147.229.0.0	111101
*	001001
147.229.1.15	001011

Nemělo by být 147.229.1.15 jako 111111 ? Protože 147.229.0.0 je to samé jako 147.229.*.* ne? A do toho 147.229.1.15 spadá.
myslím že to neni to samé, to by tam musela být maska prefixu 147.229.0.0/16
Takže na toto pravidlo bude sedět jenom packet který má Source IP = 147.229.0.0?

DestIP	
*	110111
147.229.0.0	111111

DstPort	
80	101111
53	011111
*	001111

SrcPort	
*	110111
53	111111

Protocol	
TCP	100001
UDP	011001
ICMP	000110 neni ICMP IP?ne
IP	000001

b) pomocí kartézského součinu rekurzivní klasifikace RFC s třídami ekvivalence (srcIP-dstIP) a (srcPort - dstPort)

pokud to chápu dobře z opory tak se vezmou tabulky bit vektorů a prostě se podle zadání mezi nimi udělá kartézský součin a vektory se zANDujou, podle výsledných vektorů tak máme jakési třídy pravidel
takže:

///+-

srcIP-destIP	rule vector	class
147.229.0.0 x *	110101	1
147.229.0.0 x 147.229.0.0	111101	2
* x *	000001	3
* x 147.229.0.0	001001	4
147.229.1.15 x *	000011	5
147.229.1.15 x 147.229.0.0	001011	6

srcPort-destPort	rule vector	class
53 x 80	101111	1
* x 80	100111	2
53 x 53	011111	3
* x 53	010111	4
53 x *	001111	5
* x *	000111	6

a pak se ma podle vseho udelat kartezsky soucin trid obou tabulek + potom ještě kartézský součin s tabulkou protokolů.
ale i když je v zadání napsané s třídami ekvivalence (srcIP-dstIP) a (srcPort - dstPort)? neznamená to že stačí toto? pevně doufám že ano… may the force be with us

btw jde to i bez předchozích bit vektorů (v některých zadáních z minula je jako poslední příklad bit vektor a jinde kartezský součin, takže nejspíš korektní je nedělat oboje ale jen jedno a přímo), jakože v podstatě je to to samé ale bez nutnosti to tabulkovat na dvakrát přímý postup:
vypíšu si tabulku všech (případ portů) portů src a dst
src: 53, *
dst: 80,53,*

teď součin obou skupin: 53x80,53x53,53x*,*x80….
pro každou dvojici se podívám pro první člen v jakých je pravidlech (bit vektor) a pro druhý a dám jim and
takže
53x80: 111111 AND 101111 = 101111
...

Dva toky s limitem průměrné rychlosti 100 paketů/s u prvního a 6000 paketů/minutu u druhého. Který je více omezen?
První, protože u druhého mohou být i špičky nad 100 p/s.
Tok s rychlostí 100 paketů za sekundu je více omezen než tok 6 000 paketů za minutu, i když oba toky mají v delším časovém intervalu stejnou rychlost. Je to tím, že druhý limit umožní přenést tisíc paketů během minutového intervalu (což může být i tisíc paketů za sekundu), zatímco v případě prvního limitu dojde po dosažení limitu 100 paketů za sekundu k zahazování.

Mějme systém pro rozložení provozu (traffic shaping) typu Token Bucket s parametry CIR 2.4 Mb/s a CBS je 3000 B.

- Uvažujte pakety přicházející v následujících časech (od času 0): paket A (příchod 1 ms, délka 1 kB), paket B (příchod 5 ms, délka 1.5 kB), paket C (příchod 8 ms, délka 700 B), paket D (příchod 12 ms, délka 2 kB) a paket E (příchod 14 ms, délka 1.5 kB).
- Uveďte, v kterých časech se pakety A – E objeví na výstupu?
- Uvažujte toto zadání při omezení provozu (traffic policing). Jaké pakety a kdy se objeví na výstupu? Jaké budou zahozeny?

Doporučuji mrknout do opory na správné řešení:

https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/ISA-IT/texts/Traffic_conditioning.pdf?cid=10337dík

První bod
Z CIR si přepočítám rychlost generování žetonů.
CBS berte jako maximální kapacitu bucketu.

toto je podivny policing (doplnil som shaping spravne nizsie) (protoze zacina s prazdnym bucketem? ano to je jeden z dovodov, takisto tu autor nezahadzuje pakety ktore by mal - tak to neni policing ale shaping ne? v teoretickej rovine ano ale pri shapingu by mala byt perioda pripocitania do bucketu 10ms a nie 1ms, (citacia doplnena)

- CBS/CIR ti dava periodu, a CBS dalej znaci aky je prirastok) (**CBS neni prirustek ale max hodnota tokenu, udavajici kolik dat najednou muze prolezt, viz zelena citace nize)**

uplna citacia preco je tento pseudo shapelicing kompletne odveci uz len pre spatnu periodu

“Assume that the traffic needs to be sent into the network at a mean rate CIR of 2.4 Mbps. If a burst for a duration of 10 millisec (= 0.01 sec) needs to be sustained, the CBS can be calculated using the token bucket definition as CBS = 2,400,000 bits/sec × 0.01 sec 8 bits/byte , which yields 3000 bytes. Thus, the token rate is 300,000 (=2,400,000/8) bytes per sec, the CBS is 3000 bytes, and the token interval (T) is 10 millisec. Therefore, the token generator credits the token bucket with 3000 bytes worth of tokens every 10 millisec.” (zdroj opora link vyssie)

Pro ujasneni, CBS je kolik maxmalne muze odejit dat najednou, takze maximalni hodnota tocken bucketu, podle: *Committed burst size (CBS) specifies the maximum number of bytes that can be transmitted into the network in an extremely short interval of time. In theory, the committed burst size, as the time interval tends to zero, represents the number of bytes that can be instantaneously transmitted into the network.*

Co jsem pochopil, tak predesla citace (cerne) vypocitava CBS z predpokladu, ze chci byt schopen prenest najednou data co prijdou behem 10ms, ale ze by CBS (kolik muze najednou odejit = max hodnota bucketu) mela ovlivnovat jak casto se tokeny generujou mi z toho nevychazi.

ten priklad je v linku hore cely opisany a aj vypocet periody je tam opisany, graf (*FIGURE 23.6 Shaping traffic using the token bucket algorithm.*) tiez sedi s nizsie doplnenym riesenim. Su tam riesenia ako pre shapin tak pre policing, toto riesenie nesedi ani na jedno z nich, dokonca kombinuje prvky z oboch (rad by som sa o tom dalej bavil ale do skusky je uz len par hodin :( ) OK, vidim to tam, jenom mi to nesedelo dokupy s tema definicema

tl;dr: Podle grafu v dokumentu [23.6 | link nad prikladem] **se kazdych CIR/CBS doplni tocken bucket do maxima (coz je CBS)**

CHYBA: v case 20-21 na obrazku pribyde 400B, kdyby pribylo spravne 300, tak E odejde az v t=23
Otázkou je, zda je možné aby paket E v 16ms předběhl paket D. - myslím, že ne - od toho je to FRONTA (3x)

V kterých časech se tedy objeví na výstupu:
A - 3,33ms (nie 1(prichod)+1000/300 = 4.33(odesel)??)
B - 8,33ms (5 + 1500/300 = 10?) ..
C - 10,67ms
D - 17,33ms
E - 22,33ms
SOUHLAS??? (2x) NESOUHLAS (2x) jak jste prisli na tyhle cisla? - logicky - pokud do kbelíku teče 300B/ms, trvá 3,33ms než nateče 1KB na odeslání A. Obdobně pro další…

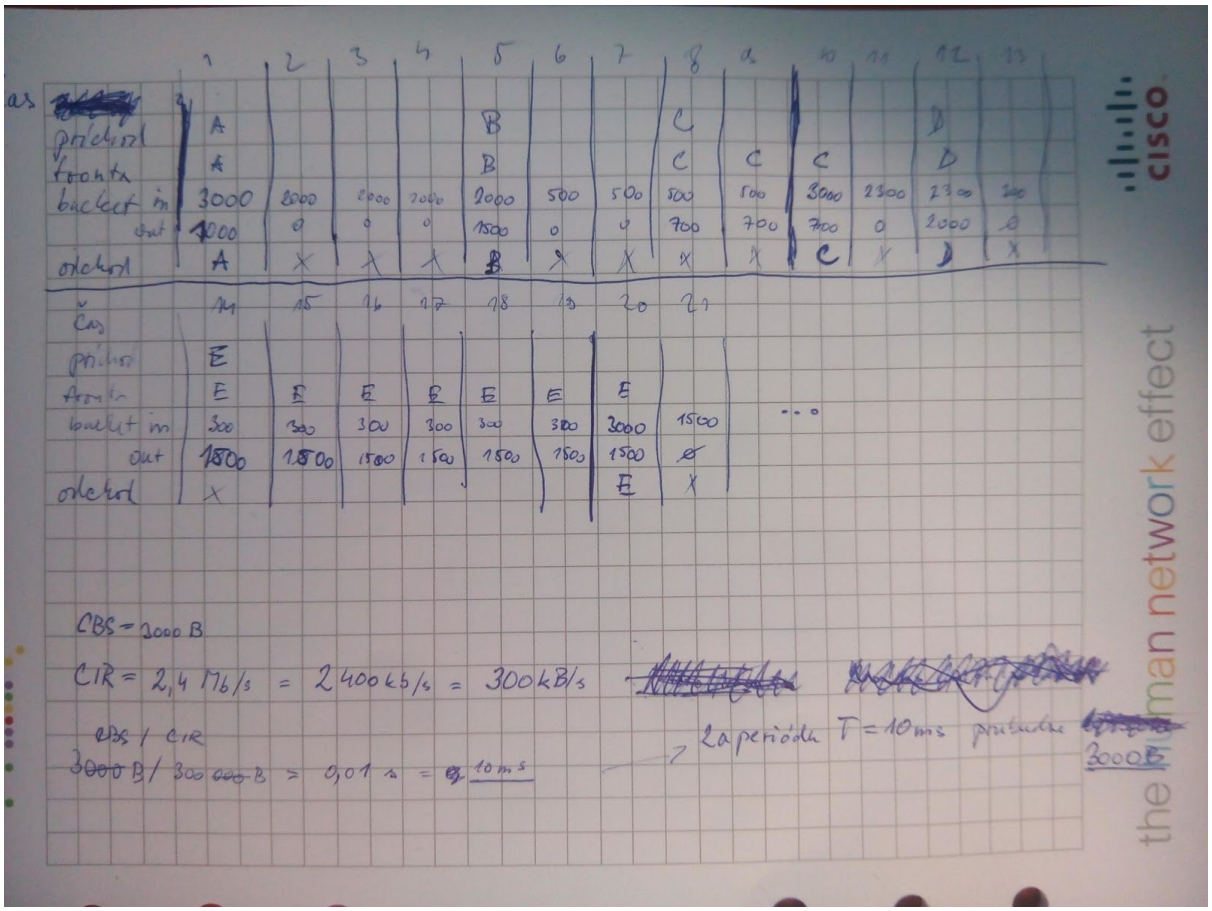
CBS/CIR -> 3000 jednou za 10ms žiadne logicky 300/ms

Sprave riesenie MODRYM vid OPORA
Neměly by ty časy na výstupu být takto:
A - 1ms
B - 5ms
C - 10ms
D - 12ms
E - 20ms
????

Nema to byt spis tak, ze si z hodnot CBS a CIR ziskam periodu T = 0,01s, a ty tokeny budu generovat 3000 jednou za 10 ms? - Podle opory to tak je. Z přednášek si nic o tomto napamtuju. ANO 4x souhlas
zaroven CBS je maximum, takze se kazdou periodu CIR/CBS doplni do maxima (= CBS),
v pripade policingu se tokeny generujou v dobe prichodu packetu, v mnozstvi podle toho, kolik casu ubehlo (ale zase jenom do maxima danem CBS)

spravne riesenie k tu uvedenemu prikladu (SHAPING) zhodne s link hore

(*FIGURE 23.6 Shaping traffic using the token bucket algorithm.*)



*

U SHAPINGU DOCHÁZÍ K NÁRŮSTU TOKENŮ KAŽDÝCH 10MS
NÁSLEDUJÍCÍ PLATÍ PRO TRAFFIC POLICING (nárust každou 1 ms):
podle těch opor výše je na začátku plný zásobník tokenů
správně tedy:
CIR = 2.4 Mb/s => rychlost nárůstu tokenů = 2.4/8 = 0.3 MB/s = 300 kB/s
příchody jsou v ms takže chceme rychlost taky tak => 300 kB/s * 10^-3 s = 300 B/ms

čas	zásobník	akce
0ms	3000B (za 1 ms nárůst 300 ale zahodí se, max je 3 kB)	
1ms	3000 - 1000 = 2000B	příchod A (1 kB), odchod A
2ms	+300B	
3ms	+300B	

4ms	+300B	
5ms	2900 + 300 (ořezáno na 3000) 3000 - 1500 = 1500	příchod B (1.5 kB), odchod B
6ms	+300B	
7ms	+300B	
8ms	2100 + 300 = 2400 2400 - 700 = 1700	příchod C (700 B), odchod C
9ms	+300B	
f10ms	+300B	
11ms	+300B	
12ms	2600 + 300 = 2900 2900 - 2000 = 900	příchod D (2 kB), odchod D
13ms	+300B	
14ms	1200 + 300 = 1500 1500 - 1500 = 0	příchod E (1.5 kB), odchod E

v opoře je ale variace, kde paket E má velikost 1.7 kB jinak je to stejné, v takovém případě by se stalo toto:

14ms	1200 + 300 = 1500	příchod E (1.7 kB), nemůže odejít (1500<1700)
15ms	1500 + 300 = 1800 1800 - 1700 = 100	odchod E

I v opore mas prichod v ms (str 291/ pr 8.4 - ma prichody v ms a periodu 10ms, a po techto periodach pridava celkovych 3000 zetonu). šak jo, proto jsem to převedl. ale stejně přidáváš kapacitu po 1ms :-) ajo takto to myslíš, no šak perioda 10ms znamená že 300 za 1ms ne? :D takto to je správně, je to tak i v opoře (odkaz nahoře)
navíc o periodě není v zadání řeč
Nevím odkud to berete, ale i v tom materiálu, co je nahoře je napsané, že pokud dostanu 3000 žetonů jednou za 10ms, tak si to nemůžu přepočítat tak, že dostanu 300 každou ms….
Pokud v čase t = 2ms přijde paket o velikosti 3kB, tak okamžitě odejde, ale pokud přijde v čase t=3ms paket o velikosti 1kB, tak čeká do času t=10ms, než dojdou žetony, aby prošel… zpozdí

se tedy o 7ms na výstupu, podle vaší teorie by prošel už po 4ms bo by dostal 1200 žetonů za tu dobu (300 žet za 1ms) -- takto to dělá POLICING ne SHAPING
jo už vidím máš pravdu, nevšiml jsem si že v zadání je shaping, takže ten poslední by se nezdržel ale zahodil ne (1.7kB)? v případě policingu ano => však jsi psal, že děláš policing a ne shaping, tak jakto, že se nezahodil?

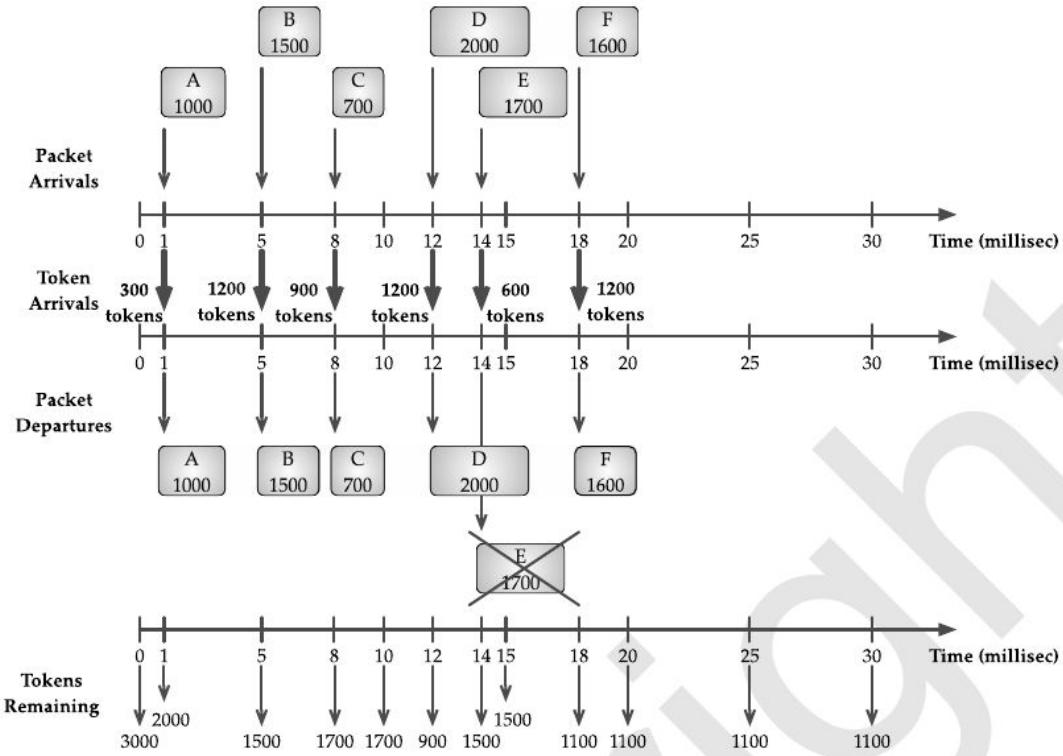


FIGURE 23.7 Policing traffic using the token bucket algorithm.

?

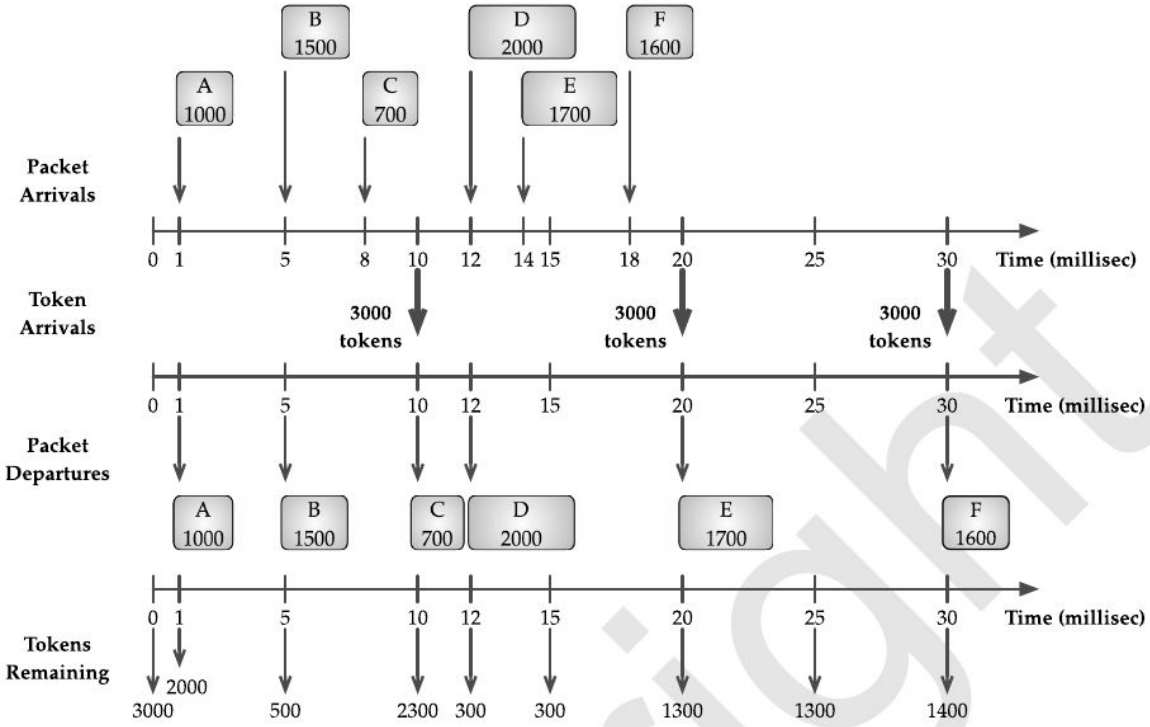


FIGURE 23.6 Shaping traffic using the token bucket algorithm.

Druhý bod

Osobně bych postupoval stejně, jen pokud bych v bucketu neměl dost žetonů pro příchozí paket, tak jej na místo uložení do fronty zahodím. Je to ale jen domněnka. Může mi ji někdo potvrdit/vyvrátit prosím?

Souhlasím, výsledek by tedy byl:

A - se zahodí

B - se odešle v čase 5ms

C - se odešle v čase 8ms

D - se zahodí

E - se odešle v čase 14ms (a zbude 500 žetonů)

To, jestli je na začátku bucket plný nebo ne, to by mělo být v zadání definované ne?

SOUHLAS??? (4x)

Myslienka je dobra ale asi tak ako je to hore

A - posle

B - posle

C - zahodi (v zasobniku je malo zetonov)

D - posle

E - zahodi

Uvažujte tři fronty WFQ s váhami 40%, 30% a 30%. Předpokládejte, že pakety ve frontě Q1 mají velikost 100 B, v Q2 velikost 150 B a v Q3 velikost 250 B. Jak bude vypadat rozložení paketů na výstupu?

Q1 = 100B *0.4 = 40 B
Q2 = 150B *0.3 = 35 B
Q3 = 250B *0.3 = 75 B

^ IMHO kravina, viz nize

Fronty sa budu vyprazdnovat v tomto pomere (z Q1 40B, Q2 35B, Q3 75B). // prosim o kontrolu to nejde, nemůžeš poslat data o velikosti menší než 1 paket. A ptají se na rozložení paketů, ne velikosti dat. **A taky by to znamenalo ze rychlost fronty s mensi vahou (Q3) bude vetsi nez rychlost fronty s vetsi vahou (Q1)!** To bude Q1 = x, Q2 = x/2, Q3=3x/10? => 4*8jste si 100% jistí?
prednaska z 20.11. cas cca 42:00, rychlost fronty = rychlost linky * váha linky, tedy pro rychlost *X* je rychlost Q1 = *X*0.4*, pakety musi odchazet z front tak, aby ta rychlost u kazde fronty odpovidala

---
MYSLIM ZE JE TO JINAK:
podle prednasky se podle vah urcuje rychlost jednotlivych front, takze **pokud by byla rychlost 100B/ms, tak** (stejne poradi by bylo pro jakoukoliv rychlost, ale at nepisu 0.4*x tak jsem zvolil nejake cislo)
Q1s = 40B/ms - jeden paket (100B) se prenese za 100/40 = 2,5ms
Q2s = 30B/ms - jeden paket (150B) se prenese za 150/30 = 5ms
Q3s = 30B/ms - jeden paket (250B) se prenese za 250/30 = 8.3ms *(- tady je skarede cislo, mozna ze u mensi rychlosti by to mohlo trochu ovlivnit vysledek, dost to asi ovlivni zaokrouhlovani na ms)*
takze z fronty Q1 pujdou pakety v casech - 3ms, 5ms, 8ms, 10ms...
takze z fronty Q2 pujdou pakety v casech - 5ms, 10ms, 15ms…
takze z fronty Q3 pujdou pakety v casech - 9ms, 17ms, …
poradi bude tedy: Q1, Q1, Q2, Q1, Q3, Q1, Q2, …atd (je to upravena cyklicka fronta, takze prioritu beru ze ma Q1 kdyz je stejny cas)
---
^ muze nekdo potvrdit ze to tak je?

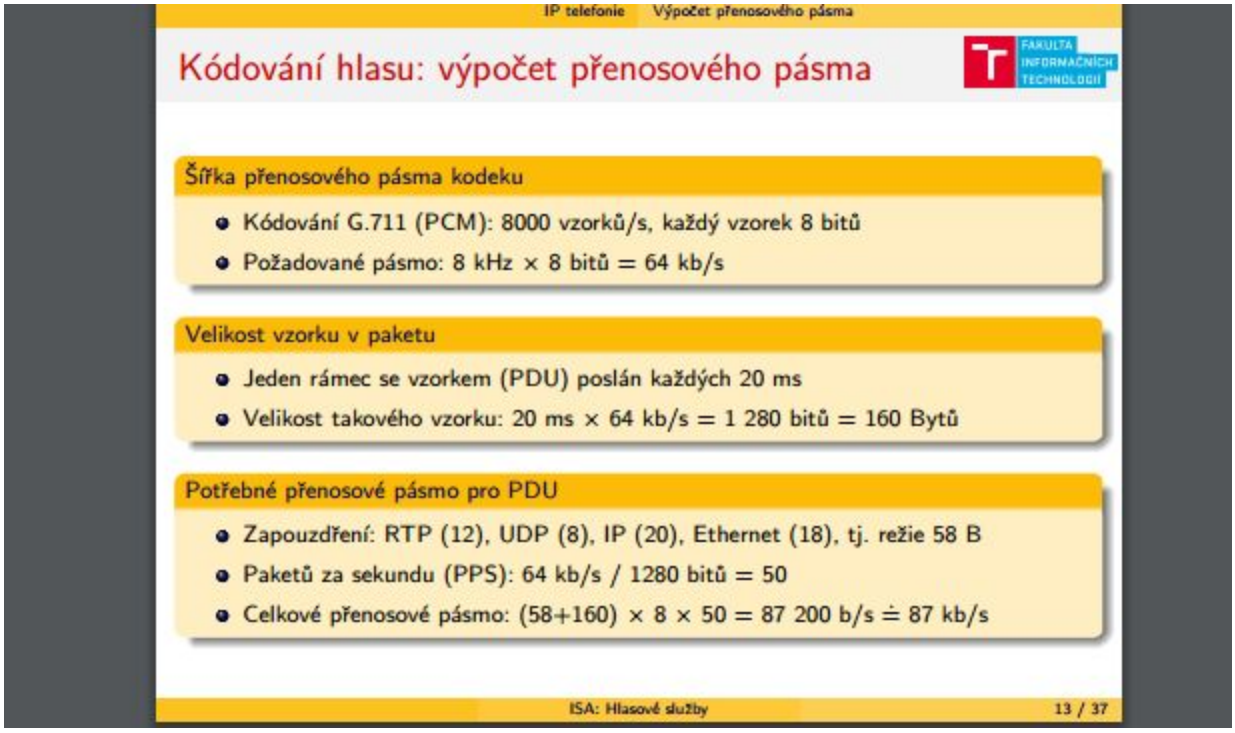
Jak jsi zjistil ten přírustek prosím? Ten vzoreček nedává moc smysl, nijak se tam nepoužívají ty procenta. prednaska z 20.11. cas 42:00, rychlost fronty = rychlost linky x procenta. A rychlost linky je dle těch vzorečků výš? myslim ze vzorecky vys jsou hovna :) :D a jak jsi zjistil tu rychlost linky tedy? :D :) tu jsem si zvolil, ono pro jinou rychlost by byly jine cisla, ale pta se na poradi a

to by zustalo stejne vzdycky....takže sis prostě zvolil nějakou konstantu a tu jsi považoval za rychlost linky...? :) jop, v tomhle pripade 100kb/s - tak ju, to se mi líbí :)

Uživatel používá IP telefon s kodekem G.726r32, který převádí vstupní analogový signál o max. frekvenci 4 kHz na 4-bitový vzorek. Hlasový rámec obsahuje kromě dat režii 58 B. Jaké přenosové pásmo uživatel potřebuje, pokud se data posílají každých 20 ms? Jaké protokoly obsahuje odchozí rámec, který zapouzdřuje hlasová data?

1. max vstupní frekvence kodeku je 4kHz. Ta se musí navzorkovat (podle [lanquistova] teorému na dvojnásobek, tedy 8kHz. Ze zadání použije kodek 4bity na vzorek tudíž šířka pásma kodeku je celkem **32kbps**
2. Každých 20ms (0.02s) se posílá rámec. Z toho odvodím že za 1s se odešle **50krát**
3. Ke každému posílání dat se přičte i režie, tedy 50*58 = 2900B. Převedu na bity, aby mi seděly jednotky => **23 200b**
4. Sečtu odeslané rámce za sekundu a data z kodeku, tedy **55.2kbps - správně :)**

imho to nie je dobre. Príklad je vypočítaný v prednáške. Je to dobře. (V prednaske sa pocita s INYM KODEKOM kt. prenasa 64kb/s tu je kodek s 32/kbs)
jak přijdete na to, že se to má vynásobit 2 na začátku? pamatuju na Shanona ale kde na to přijdu?
proč se to nedělalo i u příkladu výš? - ve slidech je u všech kodeků 8k vzorků/s (2 x 4kHz), takže bych se podle toho řídil :-D
díky :D
řešení: potřebujeme vzorky/s … pokud máme zadáno frekvenci - násobíme 2 a získáme vzorky/s. Pokud máme již zadané vzorky/s, nenásobíme.
počkej a pro ve slidech nic takového nedělá, má přece frekvenci 8 kHz, vzorky za s je to stejné jako frekvence ne? ne, frekvenci tam má 4kHz a vzorky/s = frekvence * 2



zdroje

- přednáška ISA_2015-11-06 čas 50:00
- http://wiki.fituska.eu/index.php/ISA:_P%C5%99%C3%ADklady

???

tři pobočky (Byšice, Roudnice, Všetaty)
IP prostor: 194.212.89.88/29
Napsat DNS záznamy pro pobočky, WWW, email a DNS server, záložní DNS a poštovní server u ISP na main.isp.com (81.0.233.80)?

<url pobocky, asi> IN MX main.isp.com

???

Netflow. 40Gb/s, 10% zatizenost, 20 packet ve flowu, 1000 byte/packet, Netflow data je 50B/tok, kolik data uklada netflow za hodinu.

neco zkusim :)
40Gbit/s linka, zatizenost 10% → tece tam 4Gbit/s
kazdy flow ma 20 packetu, packet ma 1kbyte = 8kbit, flow = 160kbit
flowů za sekundu: 4 000 000kbit/160kbit = 25 000 flows/s
netflow má 50B na jednom toku, toků je 25000 za sekundu, netflow tedy generuje 1250kB kazdou sekundu, za hodunu to je 1250*3600 = 4 500 000kB, **to je 4,5GB**
---
^ muze nekdo potvrdit?

Otázky, vypracované na fitwiki -

http://wiki.fituska.eu/index.php/ISA:_Ot%C3%A1zky

Minuloročné otázky

1. termin (8.1.2014), sk A1

1. MIB - sledovani site (prednaska sprava siti, slajd: 21 [8b]
byly zadane 4 zaznamy - popsat jednotlive zaznamy (hodnota, typ, vyznam zaznamu)
napr.:
Kód:

1. IF-MIB::ifSpeed.1 = Gauge32: 1000000000
2. IF-MIB::ifPhysAddress.1 = STRING: 0:1c:c0:6c:61:60
3. IF-MIB::ifInOctets.1 = Counter32: 2383522562
4. IF-MIB::ifOutUcastPkts.2 = Counter32: 0

```
snmpwalk -v 1 -c public host.fit.vutbr.cz interfaces
IF-MIB::ifNumber.0 = INTEGER: 12
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifDescr.1 = STRING: em0
IF-MIB::ifType.1 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifMtu.1 = INTEGER: 1500
IF-MIB::ifSpeed.1 = Gauge32: 1000000000
IF-MIB::ifPhysAddress.1 = STRING: 0:1c:c0:6c:61:60
IF-MIB::ifAdminStatus.1 = INTEGER: up(1)
IF-MIB::ifOperStatus.1 = INTEGER: up(1)
IF-MIB::ifInOctets.1 = Counter32: 2383522562
IF-MIB::ifInUcastPkts.1 = Counter32: 289122241
IF-MIB::ifInNUcastPkts.1 = Counter32: 0
IF-MIB::ifInDiscards.1 = Counter32: 0
                  ...
IF-MIB::ifInErrors.1 = Counter32: 0
IF-MIB::ifInErrors.2 = Counter32: 0
IF-MIB::ifOutOctets.1 = Counter32: 122577245
IF-MIB::ifOutOctets.2 = Counter32: 0
IF-MIB::ifOutUcastPkts.1 = Counter32: 101179266
IF-MIB::ifOutUcastPkts.2 = Counter32: 0
```

Například výše uvedený příklad vypisuje údaje o rozhraních na zařízení `host.fit.vutbr.cz`. Toto zařízení má 12 rozhraní, z nichž první je rozhraní č.1 s názvem `em0`. Vidíme, že jde o ethernetovou síťovou kartu s rychlostí 1 Gb/s, MTU 1500 B a fyzickou adresou `0:1c:c0:6c:61:60`. Rozhraní je ve stavu `up`. Doposud tam bylo přijato 2,383 miliard oktetů (bytů) v celkem 289 mil. unicastových paketů. Dále vidím, že na rozhraní se neobjevily žádné chybné pakety. Můžeme též vidět počet odchozích bytů a paketů.

2. Priklad na vypocet sirky pasma (kodek pracuje 8kbps, 50 paketu/sekundu, rezie 62B) [10b]
32,8 kbps ???(ano 2 x)
Nepotřebuju ještě velikost vzorku?
není těch 8kbps už spočítané? jako vynásobená délka vzorku s pásmem?

3. VOIP telefonie: popsat vyznam a souvislost s prenosem hlasu [8b]

- PTSN - veřejná telefonní síť
- Jitter - rozptyl zpoždění paketů (každý paket může jít jinou cestou)
- MoS - mean opinion score (hodnocení kvality hovoru uživatelem, 5 nejlepší, 1 je na nic)
  ENUM - mapování tel čísel na uri

4. Ktere operace jsou blokujici a proc? Jaké jsou možnosti se tomu vyhnout? [6b]
recv(), recvfrom(), read(), accept(). Všechny vlastně čekají než jim dojde zpráva.
Vyhnutí: Použití neblokujících operací nebo použití funkcí select() a poll().
díky
Otázka: Funkce connect() je taky blokující?
myslim ze funkce connect ti vyhodi chybu kdyz se nepodari pripojit a neceka

5. Vyznam TTL u DNS. Jakou roli v tom ma zalozni server? V jake radove hodonte se pohybuje TTL [8b]

- TTL u DNS urcuje maximální dobu (v sekundách), po kterou je možné data uchovat v lokální paměti cache DNS serveru. (Uchovava neautoritativny DNS server v svojej cache pre zrychlenie odvpovedania).
- Po uplynutí doby TTL je musí server DNS ze své paměti cache odstranit a načíst si aktuální data z autoritativního serveru.
- Nastavenie TTL vyrazne ovplyvnuje pomer medzi konzistenciou databaze a zvysenou zatazou komunikacie.
- Doporucuje sa ako min hodnota 1den pre servery kde dochadza k castym zmenam. Pre stabilne servery 3-5 dni.

6. Popsat LDAP. Jak se sifrujou data pri prenosu, autentizace. Vyhledavani v LDAP. Priklad adresy; jak se adresuji data [8b]

- Architektura LDAP pochadza z standardu ITU-T x.500. A definuje sposob usporiadania zaznamov do stromovej struktury(informacny model)
- definuje strukturu zaznamov a operacii nad nimi (menny model)
- komunikaciu server/client a sposoby vyhladavania (funkcny model)
- a zabezpecenie informacii (bespecnostny model)
-
- veřejné servery s daty pouze pro čtení - mohou povolit anonymní přístup
- servery podporující autentizaci heslem - musí implementovat MD5 SASL (RFC 2831)
- servery podporující kryptování a autentizaci - musí implementovat operace pro TLS

7. Dvoudimenzionalni strom se zpetnym vyhledanim: vytvoreni stromu na zaklade zadanych pravidel. Byla zadana dest. a src. IP a melo se popsat jaka je trasa vyhledani a urcit konecne pravidlo + kolikrat se provedl backtracking[12b]

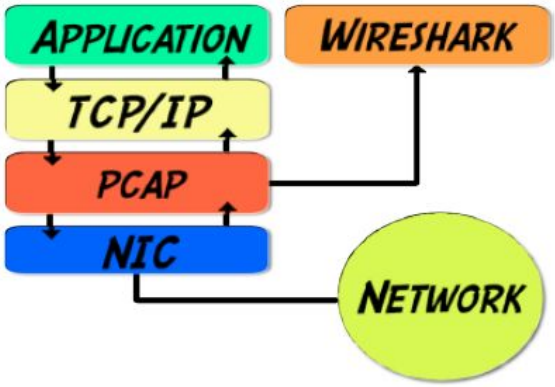
A2

1. 4 hodnoty z MIB popište význam , typ a hodnotu proměnné
2. K čemu je u DNS systému NS záznam, napište jeho příklad, kolik NS záznamů může být v jedné zóně.
NS - jméno doménového serveru. 2 záznamy v jedné zóně? (Primární a sekundární)
Příkaz, jak zjistit SOA záznam pro VUT
nslookup -type=SOA vutbr.cz?
3. Diferenciované služby - popsat co to je, napsat základní třídy klasifikace
4. 8 Zkratek, napsat na které vrstvě ISO/OSI modelu pracují
TCP,RPC,X.500,XDR,802.3,ASN.1
5. Uživatel má IP telefon, vzorkuje 4kHz na 4bit, datový packet se posílá každých 20ms, velikost hlavičky paketu - 58B. Jak velké pásmo telefon potřebuje, v jakých protokolech je jsou samotná data zapouzdřena.

6. Popište architekturu NetFlow, na jaké vrstvě TCP/IP NetFlow pracuje. Jaké informace o tocích NetFlow sleduje.
7. Vytvořit dvoudimenzionální strom trie Zdrojová-Cílová IP adresa, se zpětnými ukazateli a bez duplicit, podle zadaných filtrovacích pravidel. Nakreslit jak půjde určitý packet skrz strom.

Druha skupina, prvni termin, asi B1

1. Zjistit nekolik informaci z ifconfigu
2. K cemu slouzi knihovna libpcap? Na ktere vrstve OSI modelu bezi?
    - vytvari programove rozhrani pro zachytavani dat na sitovem rozhrani
    - čtení dat na linkové vrstvě OSI modelu
    - cte data ze sitoveho rozhrani nebo ze souboru ve formatu PCAP
    - implementuje funkce pro zpracovani ramcu a jejich analyzu
    - implementovana v jazyce C pro unix (libcap), windows (winPcap)
    - vyuziti: tcpdump, wireshark, snort, nmap, bro, scapy a pod.



dík

3. Netflow - vysvetlit agregaci, vzorkovani a filtrovani
4. K cemu slouzi zaznamy PTR? Co se stane, pokud takovy zaznam chybi? Napsat priklad takoveho zaznamu a jeho vyhledani.
5. Priklad na token bucket. Urcit kdy se jednotlive pakety objevi na vystupu + jake bude nejvetsi zpozdeni na lince
6. Vysvetlit princip RED + jak ho ovlivnuji hodnoty Qmin a Qmax
7. Podle zadanych pravidel vytvorit bitove vektory. Popsat jak to funguje a urcit jake pravidlo se pouzije pro zadany paket.

skupina B2

1. Popsat:
IF-MIB::ifNumber.0 = INTEGER: 3

IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifDescr.1 = STRING: em0
IF-MIB::ifType.1 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifMtu.1 = INTEGER: 1500
*
*
Byly tam dvě rozhraní pro ethernet a jedno pro loopback.

2. Autentizace 802.1x v LDAP - jak probíhá, jaké prvky v ní figurují.
Uživatel se přihlásí přes EAP => aktivní prvek přepošle autentizační data na Radius server => Radius server využije LDAP server pro vyhledání uživatele => po autentizaci se připojí do sítě
3. Vysvětlit pojmy z oblasti QoS: SLA, traffic shaping, CIR, RSVP.
SLA - service level agreement - smlouva o tom jaké velikosti/rychlosti dat uživatel dostane
traffic shaping - vyhlazení síťového toku, používá se k regulaci rychlosti a objemu provozu
CIR - průměrná rychlost provozu(jak přibývají žetony)
RSVP - signalizační protokol transportní vrstvy pro rezervaci zdrojů na síťových prvcích, žádost o rezervaci provádí koncová stanice
4. Jak se lokalizuje SIP server pomocí DNS.
- zjištění služby pro SIP doménu - NAPTR
- vyhledání příslušného SIP serveru - SRV
- určení IP adresy SIP serveru - A
- komunikace se serverem (odeslání požadavků)

5. Popsat adresování, formát dat a kódování v SMNP.
6. Vysvětlit útok cache poisoning, jak se mu dá zabránit. (to tak bývá, když se učí člověk názvy česky a potom mu to nic neříká. :))
- vložení nesprávné informace do vyrovnávací paměti.
- zneužití sekce Additional v odpovědi.

7. Ze zadaných pravidel udělat rekurzivní klasifikaci toků RFC - 2 tabulky pro DestIP-SourceIP a DestPort SourcePort, vytvořit třídy ekvivalence

Praktický test (půlsemestrálka)

Praktické otázky a odpovědi: http://fit.ipoul.cz/isa/

Otázky, skupina čtvrtek 13:00:

1. **Položka z ldap, která obsahuje poštovní adresu (postalAddress)**

   postup:
   ldapsearch -h ldap.fit.vutbr.cz -x -b “dc=fit, dc=vutbr”, dc=cz”

2. **maximální rychlost, kterou může komunikovat síťové rozhraní serveru eva.fit.vutbrcz (10MB -> údajně 10 000Mb)**

   postup:
   # ifconfig
   ix0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500

   options=8407bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,
   VLAN_HWCSUM,TSO4,TSO6,LRO,VLAN_HWTSO>
   ether 00:25:90:63:12:14
   inet 147.229.176.14 netmask 0xfffffe00 broadcast 147.229.177.255
   inet6 fe80::225:90ff:fe63:1214%ix0 prefixlen 64 scopeid 0x3
   inet6 2001:67c:1220:8b0::93e5:b00e prefixlen 64
   nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
   media: Ethernet autoselect (**10Gbase-Twinax <full-duplex>**)
   status: active

3. **na kterou MAC adresu se namapuje multicast adresa (zapište ve tvaru 0a:01:02:03:04:05)** (isa-sockets.pdf, slide 28)

   postup: pro 149.0.0.9 se MAC adresa namapuje na 01:00:5e:00:00:09
   - 01:00:5e - je pevne dany prefix pro IPv4
   - dalsi bit je 0
   - a zbylych 23 bitu se vezme z multicastove IP adresy (nizsich 23 bitu)

pro **ff02::1:ff3b:bd1c** se MAC adresa namapuje na 33:33:ff:3b:bd:1c
- 33:33 - je pevne dany prefix pro IPv6 a nizsich 32 bitu ze zkopiruje

4. **pomocí dotazů na školní mailový server napište příkaz (tuším) jakým probíhá autorizace (myslím že to bylo identifikace odesilatele) (nejspíš AUTH, ale v testu byl výsledek MAIL - bude se řešit)(kde jste našli, že jde o AUTH ? myslím postup -bylo povoleno použití dalších souborů z wisu, ze sekce "others" ...a tam bylo i smtp)**

   postup: ???

   asi MAIL

5. **test1.pcap a z tama určit na jaké SIP doméně se komunikuje. (cesnet.cz ,ale v testu výsledek IP adresa cesnetu - bude se řešit)**

   postup:
   otevřít ve Wiresharku a vyfiltrovat jen SIP
   je to napsané ve všech - paketech? - kromě těch OK
   konkrétně Session Initiation Protocol > Message Header > From / To
   což je cesnet, IPadresa, kterou pak vyhodí nslookup je 69.172.201.208

Najděte poslední 4 čísla z telefonního čísla docenta Víta Novotného z FEKTu.

postup:
ldapsearch -h ldap.fit.vutbr.cz -x -b "dc=feec, dc=vutbr,dc=cz" "(cnascii=Novotny Vit)" cnascii telephoneNumber
(Odpověď: 6972)

Zjistěte z konfigurace DNS na serveru eva.fit.vutbr.cz, jaká je doba expirace
A záznamů ukazujících na kořenové servery DNS? Hodnotu uveďte vsekundách?

postup:
No podla mna to urobime tak, ze napiseme **nslookup -type=soa eva.fit.vutbr.cz**
a je tam cislo 691200. to právě není pravda potřebuješ A záznam sakra :D

Ja si myslim, ze to sedi:
origin = guta.fit.vutbr.cz
expire = 691200

dig guta.fit.vutbr.cz ->
guta.fit.vutbr.cz. 12015 IN **A** 147.229.9.11

/etc/namedb/named.root
3600000 (default sekundy)

Zjistěte adresu la podle záznamů v DNS:

> **postup:**
> seznam subdomén: dig @rhino.cis.vutbr.cz vutbr.cz axfr
> nslookup -type=srv _ldap._tcp.vutbr.cz

Zjistěte doménové jméno sekundárního DNS serveru VUT.

> **postup:**
> nslookup -type=NS vutbr.cz <- zjisti DNS servery (jsou dva pipit a rhino)
> nslookup -type=SOA vutbr.cz <- zjisti primarni DNS (primarni je rhino (origin=rhino.cis.vutbr.cz), takže sekundární by měl být pipit)

Zjistěte IPv6 adresu určitého serveru (třeba http://www.google.com).

> **Proč nejde použít ping6? Proč dává jinou adresu než host?**
> **postup:**
> host -t AAAA www.google.cz stačí host www.google.com (najde to IPv4 (4 různé) i IPv6)
> *… nebo …*
> nslookup -type=AAAA www.google.cz
> … nebo … **(kebyze chcu IPv4ku, tak ktory by som mal zadat, vystupom su hned 4 adresy..)**
> dig google.cz AAAA (případně "+short" parametr pro méně ukecaný výstup)

Pomocí služby LDAP (server ldap.fit.vutbr.cz) zjistěte, kolik studentů je zapsáno ve 3. ročníku magisterského studia oboru MMI.
Odpověď: 8 (numEntries)

> **postup:** ldapsearch -h ldap.fit.vutbr.cz -x -b "dc=fit,dc=vutbr,dc=cz" "(ou=*MMI 3r*)"
> *Jak poznám, kdy píšu hvězdičky za ou= ?*
> hvezdicky pises kdyz ocekavas ze tam muze byt neco pred nebo za
> (třeba když hledáš všechny Dvořáky tak tam bude cnascii=Dvorak*)

Prvni se ptá na DOMÉNU druhá na ADRESNÍ PROSTOR

- odpověd: **TUKE nebo RIPE ?? .. asi spíš TUKE, RIPE je pro celou Evropu, to by bylo moc jednoduché**
- **POSTUP??**
  nslookup inetacad.net => tam precteme IP a pouzijeme ji do whois
  whois 147.232.55.68 => zde najdeme spravce prostoru (RIPE) a domeny (TUKE)

Napište kód OUI výrobce síťových karet na serveru merlin.fit.vutbr.cz. Odpověď zapište v hexadecimálním tvaru po bytech oddělenými dvojtečkou, např. 01:0c:13.
Odpověď: 00:25:90
Postup: ssh xlogin00@merlin.fit.vutbr.cz
zadat heslo
ifconfig

```
eth2      Link encap:Ethernet  HWaddr 00:25:90:C8:3F:1B
          ....
eth2:0    Link encap:Ethernet  HWaddr 00:25:90:C8:3F:1B
          ....
```

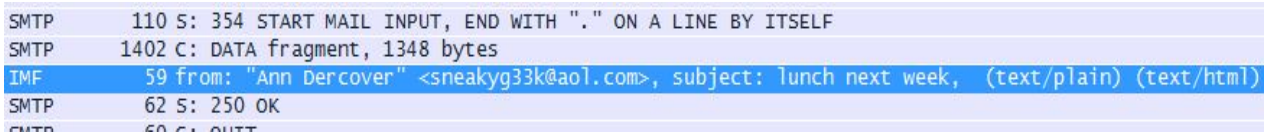
OUI - prvních 24 bitů fyzické adresy (Organizational Unique Identifier)
- zbytek je číslo síťového rozhraní přidělené výrobcem

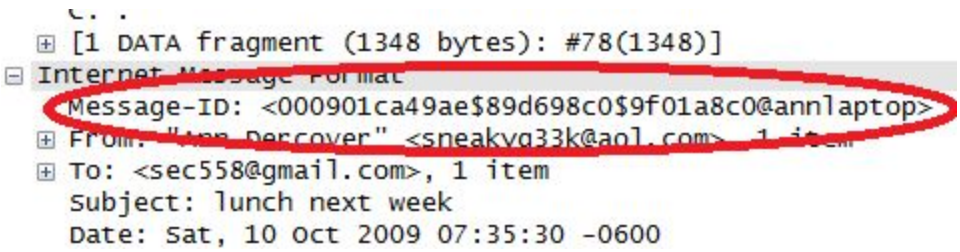
Uveďte telefonní číslo na správce adresového prostoru, kam patří adresa 147.251.45.10 (pouze posledních 9 číslic bez mezer, např. 603801113).
Odpověď: 549491801
Postup: whois 147.251.45.10

```
...

% Abuse contact for '147.251.0.0 - 147.251.255.255' is 'abuse@muni.cz'

...

role:         Masaryk University Network Administrators
address:      Masaryk University Network Administrators
address:      Institute of Computer Science
address:      Botanicka 68a
address:      Brno
address:      602 00
address:      The Czech Republic
phone:        +420 549 491 801
fax-no:       +420 541 212 747
```

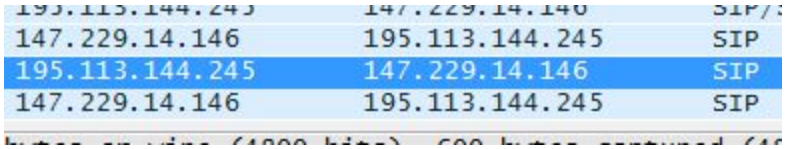
```
abuse-mailbox:  abuse@muni.cz
org:          ORG-MU9-RIPE
admin-c:       CA6687-RIPE
tech-c:        CA6687-RIPE
nic-hdl:       MUNA1-RIPE
mnt-by:        TENCZ-MNT
created:       2013-09-16T13:31:30Z
last-modified:  2015-09-25T11:56:27Z
source:        RIPE # Filtered
```

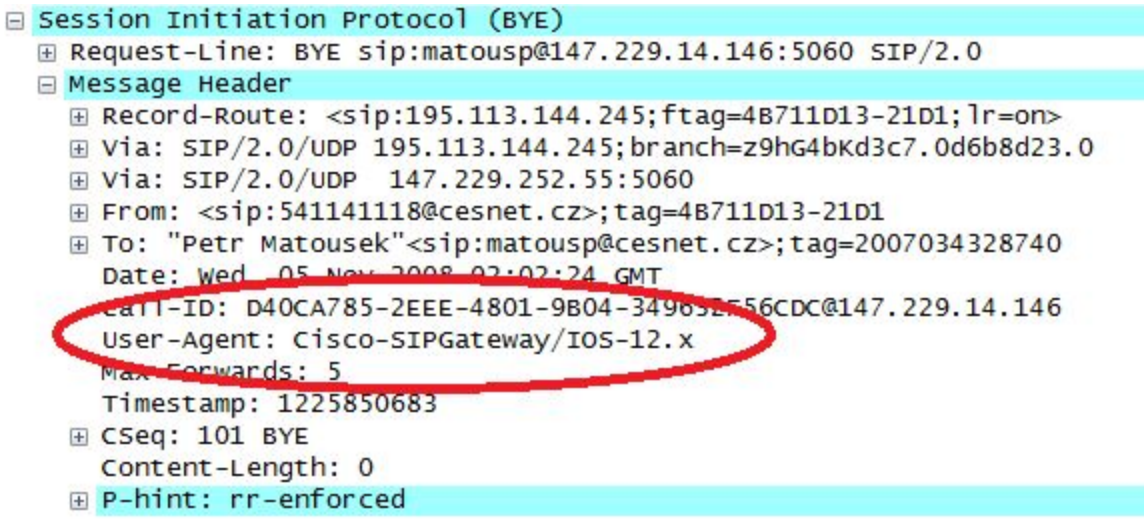
Otevřete si soubor smtp2.cap, který je uložen ve WISu u předmětu ISA. Pomocí síťového analyzátoru Wireshark najděte zachycenou emailovou komunikaci. Uveďte první čtyři znaky jednoznačného identifikátoru první emailové zprávy ze zachycené komunikace.
Odpověď: 0009
Postup: Najít přes SMTP filtr záznam kde jsou samotná data mailu





Otevřete si soubor test2.pcap, který je ve WISu u předmětu ISA. Z uvedené komunikace zjistěte, jaký software používá koncový klient SIP komunikace (VoIP brána)? V odpovědi uveďte pouze první slovo označující výrobce daného softwaru, například Microsoft.
Odpověď: Cisco
Postup:





Příklad z přednášek:
Lokalizace serveru SIP pomocí DNS (nepíšu výsledky těch dotazů, podle nich se pak hledá v tom dalším)

- zjištění služby pro SIP doménu: nslookup -type=NAPTR cesnet.cz (ukáže nám jaké služby poskytuje)
- vyhledání příslušného SIP serveru: nslookup -type=SRV _sip._tcp.cesnet.cz (podle služby najdu její server)
- určení IP: nslookup cyrus.cesnet.cz (nebo dig cyrus.cesnet.cz) (nebo host cyrus.cesnet.cz)

Užitečné nástroje:
(čerpáno z příkladů z přednášek a cvičení, prosím o doplnění, pokud máte podezření, že by se něco mohlo vyskytnout v testu)
pro níže uvedené příkazy doporučuji v případě problémů na testu vyvolat nápovědy, například “man <název příkazu>”
ipconfig/ifconfig

- informace o aktuálním síťovém nastavení, ip adresy, mac adresy apod.

ip a

- zobrazí seznam přiřazených ip adres

ip route

- zobrazí výchozí bránu

ping <address>

- zkouška konektivity

traceroute <address>

- -||- + výpis přes co všechno paket projde

netstat -r

- výpis směrovací tabulky

arp -a

- výpis ARP tabulky
- na základě ip můžeme najít patřičnou MAC

telnet <address> <port>

- test dostupnosti nebo připojení se na zařízení

sockstat/netstat -a

- sledování otevřených spojení

nslookup <address>

- dotaz na DNS server (překlad IP na www a naopak)
- -type=NS - určí autoritativní servery pro danou zónu (primární pak zjistíme pomocí host -v <adresa>, nebo nslookup -type=NAPTR <adresa>)
- -type=SOA - autoritativní server atd
- -type=MX - poštovní servery
- -type=TXT textová data, informace o doméně, správci atd
- -type=any vypíše všechno
- podrobnější údaje vrací **dig** nebo **host**

host <adresa>

- vrátí ipv4 a 6 adresy podobně jako nslookup
- s -v vrací podrobný výpis i o SOA (název primárního serveru a email správce)

ldapsearch -h nazevServeru -x -b "definice stromu" "(záznam=neco)"

- server např: ldap.fit.vutbr.cz
- strom podle adresy: dc=fit,dc=vutbr,dc=cz
- záznam např: ou=*MMI 3r*
- hvězdička znamená “cokoliv”
- ou je název položky v záznamu, pokud nevíme tak neuvedem (vynecháme celé “(...)”) a mělo by se nám vypsat několik záznamů, kde ty položky uvidíme, je to např jméno, adresa apod.
- pokud chceme kombinovat filtry například pomocí fce AND “(&(a=neco)(b=nvm))”

show cdp neigbours

- pokud jsme připojeni na zařízení, tak vypíše info o sousedech (směrovačích CISCO)

show cdp entry

- podle ID z neigbours vypíše info o tom zařízení, hlavně IP adresu (pak se lze na něj připojit telnetem)

show ip interface brief

- vypíše seznam rozhraní na zařízení

show ip route

- u směrovačů, výpis směrovací tabulky

show mac address-table

- u přepínačů, mac adresy připojených zařízení

show ip arp

- u směrovačů, zjištění IP na základě MAC, je nutné aby předtím proběhla komunikace se zjišťovaným zařízením

whois <adresa>

- info o majiteli domény či adresového prostoru

less /etc/protocols (unix)

- tento soubor obsahuje čísla protokolů, podle kterých je možné je identifikovat na IP vrstvě

less /etc/services (unix)

- tento soubor obsahuje čísla portů pro jednotlivé služby

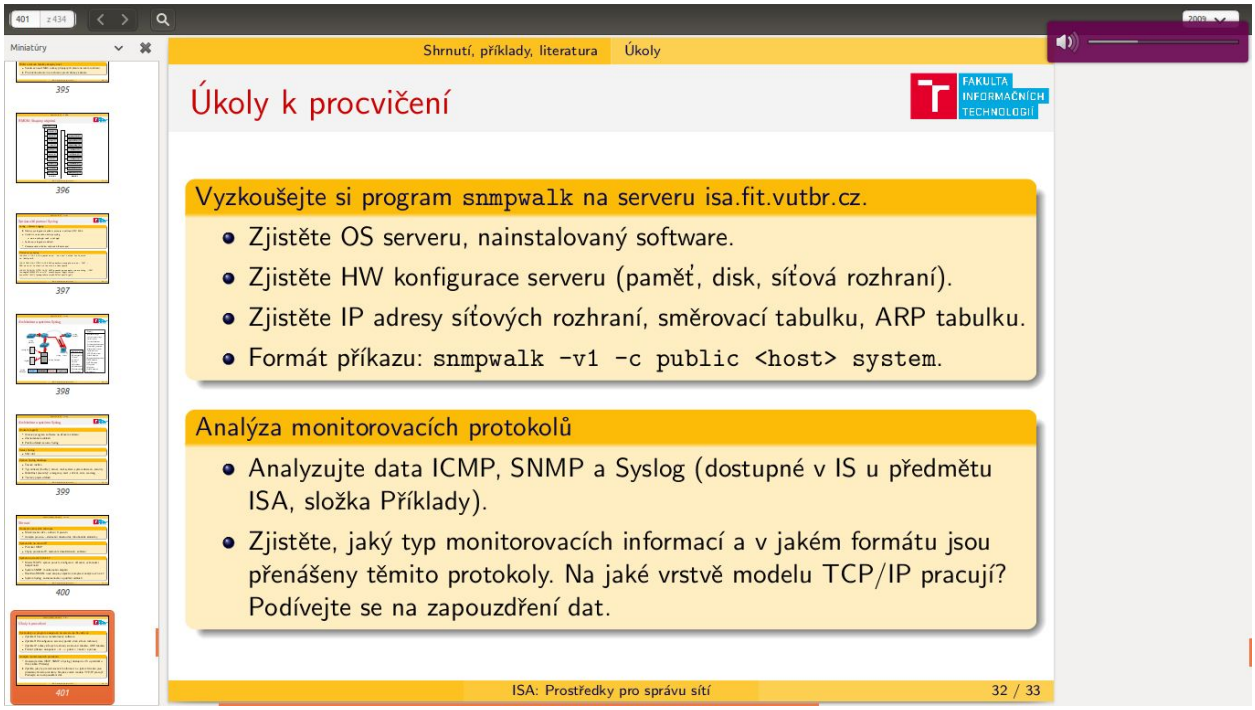
snmpwalk -v1 -c public isa.fit.vutbr.cz system

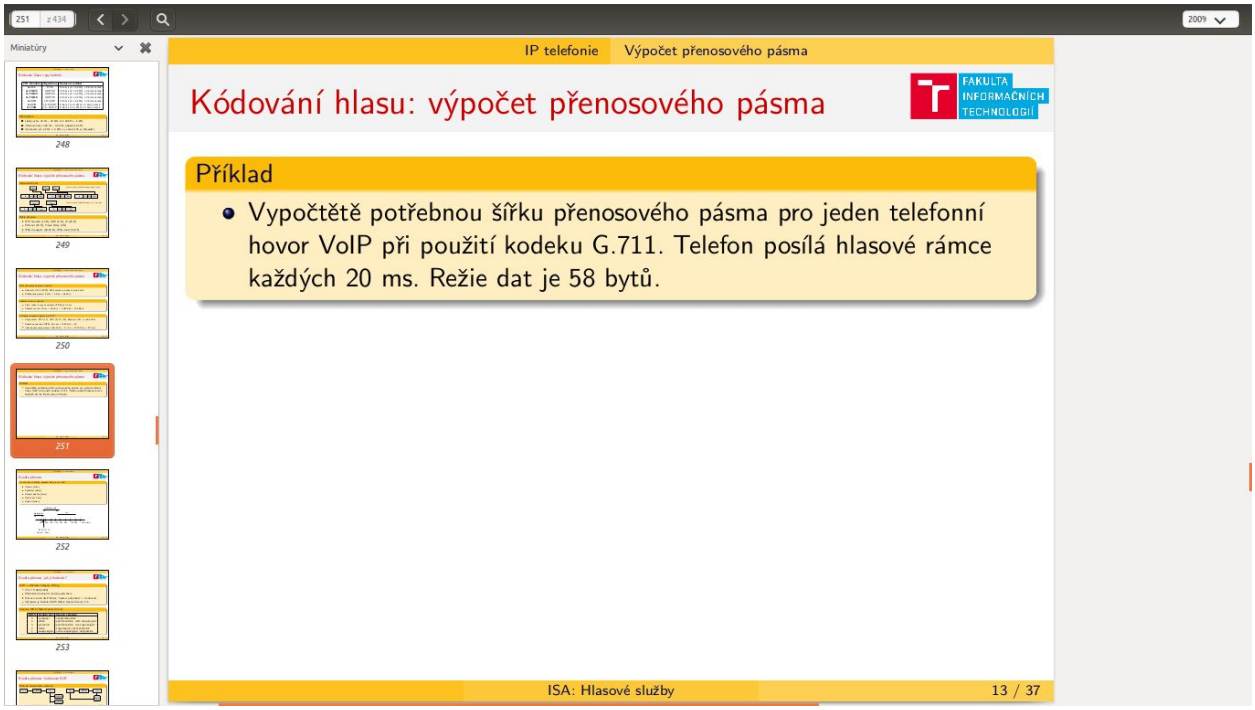
- info o systému (pomocí snmp)

minulorocni soupis otazek
https://fituska.eu/viewtopic.php?f=1404&t=24288
https://fituska.eu/viewtopic.php?f=1404&t=24315

1) Jak se nazývá organizace, která spravuje ip adresy pro celou Evropu?
   RIPE NCC
2) Co platí o zabezpečení DNSSEC?
   Používá asymetrickou kryptografii
3) Co znamená že odpoved´ DNS serveru je neautoritativní?
   http://kb.wedos.com/cs/dns/response.html
   Dotazovaní server je sekundární
4) Jaké parametry mohou být nastaveny v Omnet++?
   propagation delay, bit error rate, data rate
5) Co platí o LDAP?
   Formát je popsán ANS.1, a přenáší pomocí BER
6) Standardní port IMAP?
   143, protokol TCP

http://fit.ipoul.cz/isa/#1091



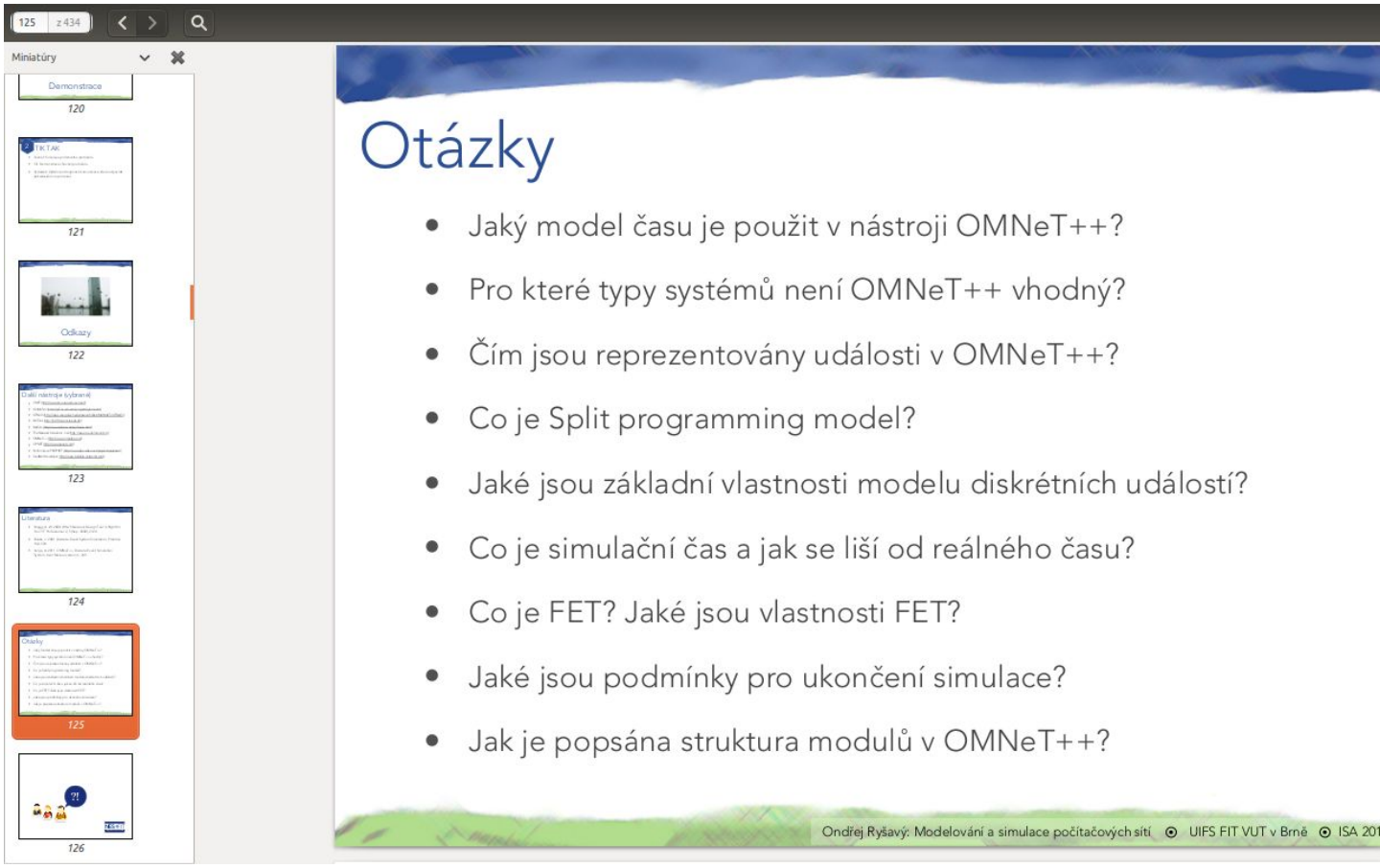


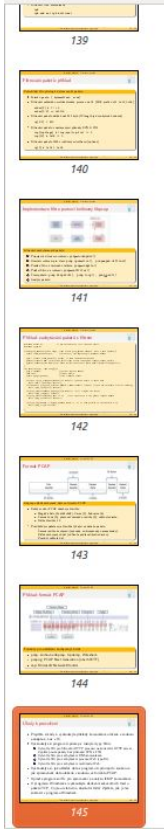
1s / 20 ms = 50 50*58bytů = 2900bytů = 2,9KB souhlas? NE - postup je v přednášce na dalších slidech

podobný příklad v aktuálních přednáškách i s řešením:
8000vz/s, po 8b -> 64kb/s, jeden rámec každých 20ms
20ms*64kb/s = 1280b = 160B

1s/20ms = 50p/s

(58B+160B) * 8 (bity) * 50p/s = 218*400 = 87200b/s zaokr. 87kb/s


125 z 434
Miniatury
Demonstrace
120
121
Odkazy
122
123
Literatura
124
Otázky
125
126
Otázky
Jaký model času je použit v nástroji OMNeT++?
Pro které typy systémů není OMNeT++ vhodný?
Čím jsou reprezentovány události v OMNeT++?
Co je Split programming model?
Jaké jsou základní vlastnosti modelu diskrétních událostí?
Co je simulační čas a jak se liší od reálného času?
Co je FET? Jaké jsou vlastnosti FET?
Jaké jsou podmínky pro ukončení simulace?
Jak je popsána struktura modulů v OMNeT++?
Ondřej Ryšavý: Modelování a simulace počítačových sítí
UIFS FIT VUT v Brně
ISA 201

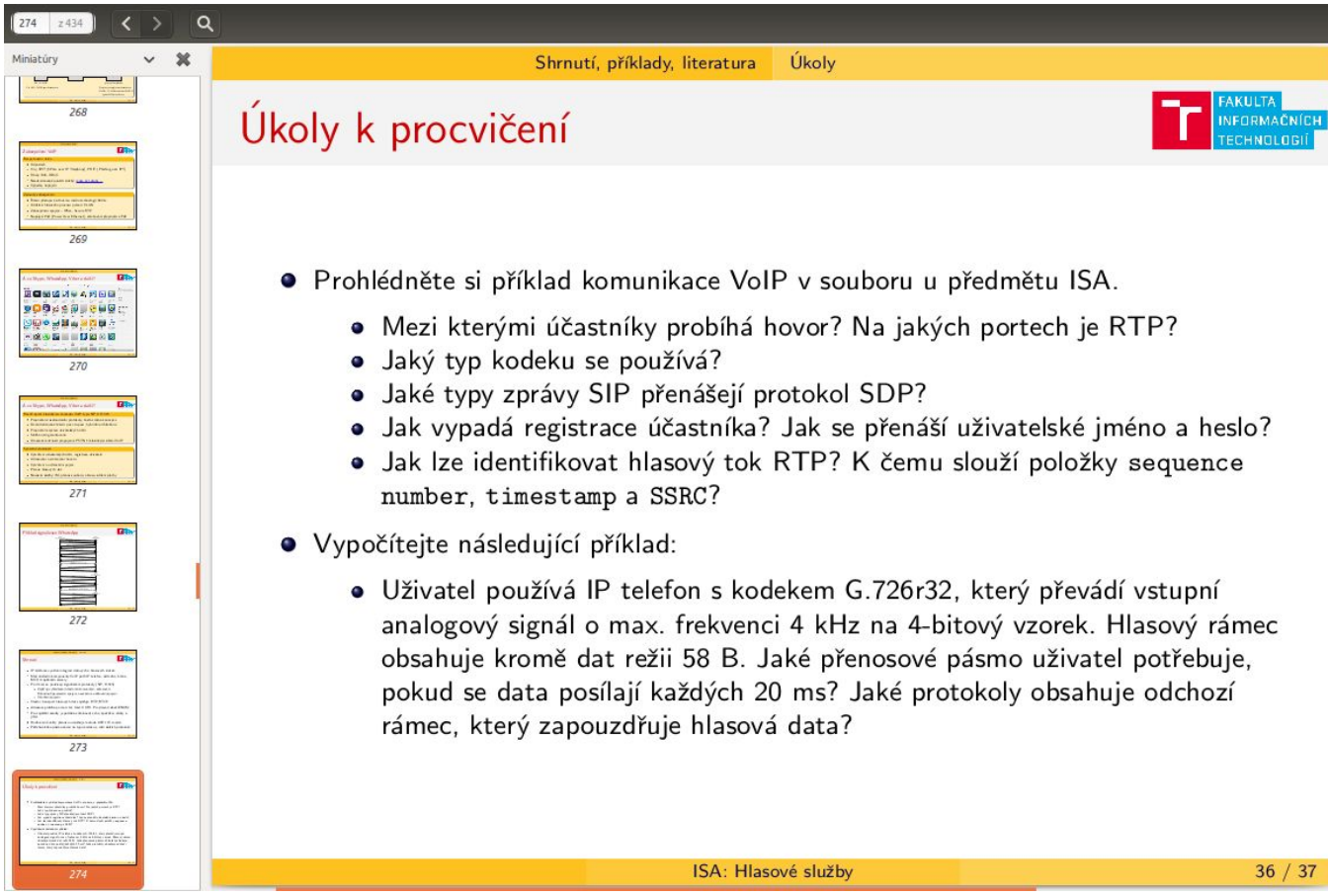


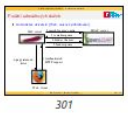
Úkoly k procvičení

- Projděte si kódy a vyzkoušejte příklady komunikace uložené v souboru `examples.tar` v IS.
- Vyzkoušejte si program tcpdump z různými typy filtrů:
  1. Vytvořte filtr pro filtrování HTTP provozu na konkrétní HTTP server. Vypište pouze pakety bez příznaků SYN a FIN.
  2. Vytvořte filtr pro zachytávání DNS komunikace.
  3. Vytvořte filtr pro zachytávání provozu IPv6 typu RA.
  4. Vytvořte filtr pro zachytávání multicastu IPv6.
- Vyzkoušejte si, jak ukládat data z programu tcpdump do souboru a jak zpracovávat data uložená v souboru ve formátu PCAP.
- Upravte program sniff.c pro zpracování a analýzu ICMP komunikace.
- V programu Wireshark si vyzkoušejte sledování sekvenčních čísel u paketů TCP. Co jsou relativní a absolutní čísla? Zjistěte, jak je lze zobrazit v programu Wireshark.

Analýza paketů Formát PCAP

Riesenie prikladu s kodekom G.726r32 (nizsie):
G.726r32 -> 32 kb/s -> 4b
hlasovy ramec = 58B
kazdych 20ms sa posle ramec -> za 1s sa posle 50 ramcov
50*58B = 2900B
2900B = 2900*4 = 11600 bitov //z bytu na bity se nasobi 8! - jo, je to chyba, počítal jsem to trochu jinak a vyšlo mi to 55,2 kb/s
este sa prida prenos kodeku -> 11600b + 32000b = 43600b
spravne? podle me ano, **IMHO nie**
Aplikovanie postupu z prednasok:
G.726r32 -> prenosove pasmo zistime ako **4kHz** (zadane) * 2 * 4 bit (vzorkovanie) **= 32kb/s**
// pripadne proste vieme ze je to 32kbps pretoze to toto kodovanie ma v specifikacii (aj nazve)
32kb * 0.02 (1s/50paketov je 20ms) **= 640b = 80B**
(58B (velkost rezie) **+ 80** (velkost jedneho paketu)**) * 8** (prevod na bity) *** 50** (pocet za sekundu)
= 55200 b/s
protokoly: **Ethernet** (18B), **IP**(20B), **UDP**(8B) a **RTP**(12B) // spolu 58B rezia
// ak by sa jednalo o 64B reziu tak obsahuje + FrameRelay (6B)

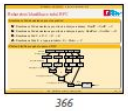
- 



Úkoly k procvičení

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

- Podívejte se na celou hlavičku vybrané emailové zprávy a zkuste zjistit, kdo a kdy ji odeslal, ze kterého počítače a kdo byl adresát.
- Z hlavičky emailu určete typ obsahu, způsob kódování obsahu. Dekódujte obsah typu `quoted-printable` a `base64`.
- Nastavte vyhledávání adres v adresáři LDAP **ldap.fit.vutbr.cz** pro strom **dc=fit,dc=vutbr,dc=cz**. Zkuste napsat email za použití adresářové služby.
- Pomocí řádkové klienta **ldapsearch** vyhledejte informace o vaší osobě v adresáři LDAP na FIT. Kolik je na FIT osob se stejným příjmením, jaké máte vy?
- Prohlédněte si certifikát Certifikační autority na VUT uložený v textové podobě na `http://ca.vutbr.cz`. Jaké jsou základní atributy certifikátu ve formátu X.509.
- Odchytněte si v programu Wireshark/tcpdump komunikaci LDAP. Podívejte se na formát a strukturu dat LDAP. Jaké příkazy a jaká data obsahují pakety s protokolem LDAP?



Úkoly k procvičení

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

1. Pro zadaná filtrovací pravidla vytvořte dvoudimenzionální binární strukturu trie (a) se zpětným vyhledáváním a bez duplicit, (b) se zpětným vyhledáváním a s ukazateli.

```
R1: deny ip from 194.0.0.0/5 to any
R2: deny ip from 0.0.0.0/4 to any
R3: deny ip from 200.0.0.0/5 to any
R4: deny ip from 160.0.0.0/5 to any
R5: permit ip from any to 48.0.0.0/4
R6: permit ip from 160.1.0.0/4 to 16.0.0.0/4
R7: deny ip from any to any
```

2. Implementujte níže uvedená pravidla firewallu (a) pomocí bitového vektoru, (b) pomocí kartézského součinu rekurzivní klasifikace RFC s třídami ekvivalence (srcIP-dstIP) a (SrcPort-DstPort).

```
R1: permit TCP from 147.229.0.0 to any dst-port 80
R2: permit UDP from 147.229.0.0 to any dst-port 53
R3: permit UDP from any to 147.229.0.0 src-port 53
R4: permit ICMP from 147.229.0.0 to any
R5: deny ICMP from 147.229.1.15 to any
R6: deny IP from any to any
```