

ITKP104 Tietoverkot - Teoria 2

Ari Viinikainen

Jyväskylän yliopisto

5.6.2014

Teoria 2 - Sovelluskerrokselta kuljetuskerrokselle

- ▶ Rekursiivinen ja Iteratiivinen DNS
- ▶ DNS tarkemmin - kokeile itse
- ▶ Verkkotunnuksen hakeminen
- ▶ P2P vs. client server
- ▶ Sovelluskerroksen protokolla määrittelee
- ▶ Sovelluskerros haluaa kuljetuspalvelulta...
- ▶ Internetin kuljetuspalvelut

DNS Iteratiivinen vs. rekursiivinen

DNS Iteratiivinen vs. rekursiivinen - kokeile itse

DNS: kokeile itse (nslookup)

► Kokeile esim

- `www.google.com`
- `www.google.fi`
- `www.jyu.fi`
- `users.jyu.fi`
- `set type=MX`
- `jyu.fi`
- `student.jyu.fi`
- `set type=all`
- `jyu.fi`
- `nslookup - 8.8.8.8`
- `set type=all`
- `jyu.fi`

DNS: lisää

- ▶ Hosts tiedosto
- ▶ Palvelinten roolit (www.jyu.fi.[tyhjä])
 - ▶ Juurinimipalvelin [”piste tyhjä”]
 - ▶ Ylätason nimipalvelin [.fi] (TLD-nimipalvelin) (käytetään myös termiä ”toisen tason nimipalvelin”)
 - ▶ Autoritääriinen nimipalvelin [juu.fi]
 - ▶ Ratkaisija (Resolveri) nimipalvelin [se joka joka alkaa pyynnöstä selvittää IP osoitetta]
- ▶ DNS:n rooli Internetin sensuurissa

Verkkotunnuksen hakeminen: www.itkp104.fi

- ▶ Tarkista onko verkkotunnus varattu: domain.fi
- ▶ Hanki verkkotunnukselle nimipalvelimet tai
- ▶ Osta palvelu palveluntarjoajalta

P2P vs. client server

animaatio p2p:stä - kokeile itse

Sovelluserroksen protokolla määrittelee

- ▶ Mitä viestejä käytetään
- ▶ Viestien syntaksin: mitä kenttiä viestissä ja kuinka ne erotetaan toisistaan
- ▶ Viestien semantiikka (merkitys)
- ▶ Säännöt sille, milloin lähetetään viestejä ja miten reagoidaan viesteihin
- ▶ Avoimia protokollia (RFCt) ja yksityisiä protokollia

Sovelluskerros haluaa kuljetuspalvelulta...

- ▶ Data perille
 - ▶ Oikealle sovellukselle
- ▶ Mahdollisuus käyttää useita verkkosovelluksia samaan aikaan
 - ▶ Yhdellä kuljetuskerroksen palvelulla?
- ▶ Datan eheys?
 - ▶ Ei virheitä datassa eikä kadonnutta dataa
- ▶ Ajoitus?
 - ▶ Viive ei voi olla sekunttiluokkaa esim. Internet-puheluissa
- ▶ Kaistanleveys?
 - ▶ Videopuhelussa kuvan siirto vie paljon enemmän kaistaa kuin äänen siirto
- ▶ Tietoturva?
 - ▶ Palvelin tunnistaa asiakkaan? Asikas tunnistaa palvelimen?
 - ▶ Selkokielineen datansiirto kryptataan?

Internetin kuljetuspalvelut

- ▶ Data viedään oikealle sovellukselle
- ▶ Kuljetuskerroksen osoitteet (portit) laitteessa → useita samanaikaisia verkkosovelluksia
- ▶ Luotettava kuljetuspalvelu
- ▶ Yhteydellinen kuljetuspalvelu
- ▶ Ruuhkanhallinta ja vuonvalvonta
- ▶ Ajoitus?
 - ▶ Ei tällä hetkellä, ehkäpä jonain päivänä
- ▶ kaistanleveys?
 - ▶ Ei tällä hetkellä, ehkäpä jonain päivänä
- ▶ tietoturva?
 - ▶ Ei, toteutetaan tarvittaessa kuljetuskerroksen ja sovelluskerroksen välissä
- ▶ Käytännön implementointi?
 - ▶ Soketti (rajapinta käyttöjärjestelmän protokollapinon toteutukseen)

Internetin kuljetuspalveluiden nimet

- ▶ Transmission Control Protocol (TCP)
- ▶ User Datagram Protocol (UDP)
- ▶ Datagram Congestion Control Protocol (DCCP)
- ▶ Stream Control Transmission Protocol (SCTP)
- ▶ Resource Reservation Protocol (RSVP)
 - ▶ Yksi ehdotus ajoituksen ja kaistanleveyden takaamiseksi
 - ▶ IntServ (Integrated Services) arkkitehtuurin resurssienvaraus protokolla
 - ▶ DiffServ (Differentiated services) ei käytä RSVP:tä mutta mainitaan se IntServ:n 'kilpailijana' (kevyempi, ei takaa absoluuttista ajoitusta tai kaistanleveyttä)

Kuljetuskerros

Tarjoaa kuljetuspalvelun sovelluskerrokselle käyttäen verkkokerroksen palveluita apunaan

- ▶ Minkälainen on verkkokerros Internetissä?
 - ▶ Pakettikytkentäinen
 - ▶ Kun verkko ruuhkautuu, paketteja saattaa hävitä
 - ▶ IP protokolla ei tarkista onko sen kuljettamassa datassa virheitä
- ▶ Verkkokerroksen palvelu Kuljetuskerrokselle?
 - ▶ Vie datan virheettömästi oikeaan osoitteeseen, jos vain pystyy!
 - ▶ Best Effort -palvelu
 - ▶ IP protokollaa ei kiinnosta
 - ▶ jos reitittimet hukkaa paketteja
 - ▶ jos fyysisellä tai linkkikerroksella tulee virheitä

Sovelluskerros haluaa kuljetuspalvelulta...

- ▶ Data perille (saa tulla virheitä)
 - ▶ Verkkokerros: Ok, verkkokerros vie oikealle laitteelle
 - ▶ Kuljetuskerrokselle jää: Data oikealle sovellukselle/prosessille
 - ▶ Toteutus: määritellään vastaanottajan kuljetuskerroksen osoite
- ▶ Mahdollisuus käyttää useita verkkosovelluksia samaan aikaan
 - ▶ Verkkokerros: sama se mitä dataa paketissa on, paketit jonoon vaan ja laitetaan ne verkkoon
 - ▶ Kuljetuskerros:
 - ▶ Lähetys: erotellaan lähtevät paketit lähettäjän ja/tai vastaanottajan (kuljetuskerroksen) osoitteen perusteella
 - ▶ Vastaanotto: kun verkkokerrokselta saadaan paketteja, niin (kuljetuskerroksen) osoitteen perusteella ohjataan ne oikealle sovellukselle/prosessille
 - ▶ TCP: myös verkkokerroksen osoitteita käytetään lähettäjä/vastaanottajaprosessin määrittämiseen
- ▶ Kurose-Ross 3. luku, Transport layer, kalvot 8-14

Esimerkki kuljetuspalvelun toiminnasta

Piirros kuljetuspalvelun toiminnasta

Kuljetuskerroksen palvelut sovelluskerrokselle

- ▶ Datan eheys (virheettömyys)?
 - ▶ Verkkokerros: ei kiinnosta
 - ▶ Kuljetuskerros:
 - ▶ UDP: tarkastetaan otsikkokenttien sekä datan mahdolliset virheet (IPv4:n kanssa ei pakollinen), hylätään virheelliset paketit
 - ▶ TCP: tarkastetaan otsikkokenttien sekä datan mahdolliset virheet, hylätään virheelliset paketit
 - ▶ TCP: lisäksi uudelleenlähetetään paketit jotka eivät tule perille
 - ▶ Sovelluskerros (verkko-ohjelman tekijä) voi valita käyttääkö TCP vai UDP palvelua
- ▶ UDP ja TCP protokollien yhteiset palvelut sovelluskerrokselle
 - ▶ Data perille oikealle sovellukselle/prosessille
 - ▶ Palvellaan samanaikaisesti useita prosesseja samalla palvelulla
 - ▶ Tarkastetaan otsikkokenttien sekä datan virheettömyys

Sovelluskerros haluaa kuljetuspalvelulta...

- ▶ Ajoitus?
 - ▶ Verkkokerros - I am doing my best, but all those other users...
 - ▶ pakettikytkentäinen, viive voi vaihdella paljon riippuen muun liikenteen määrästä
 - ▶ Kuljetuskerros
 - ▶ UDP: Lähetetään niin nopeasti kuin sovellus/käyttöjärjestelmä/tietokone/tiedonsiirtolinkki kykenee ja toivotaan että viive (ja varsinkin sen vaihtelu) pysyy aisoissa
 - ▶ TCP: Luotettavuus pääasia, uudelleenlähetykset aiheuttavat viivettä
 - ▶ TCP: On muuten myös Vuonvalvonta ja Ruuhkanhallinta palvelut! Mutta lisäävät viivettä...

Sovelluskerros haluaa kuljetuspalvelulta...

► Ajoitus?

- Sovelluskerros - ei saa tätä palvelua kuljetuskerrokselta, täytyy itse keksiä jotain

► Videostreamit:

- valitaan yleensä TCP, lähetetään **datapaketteja** niin nopeasti kuin TCP kykenee (pitää odottaa kuittauksia, yms.)
- puskuroidaan saapuvaa dataa, toivotaan että puskurissa oleva data riittää viiveen noustessa
- jos ei riitä niin käyttäjä joutuu odottamaan

► Reaaliaikainen video/puhe:

- valitaan yleensä UDP, lähetetään **datapaketteja** niin nopeasti kuin UDP kykenee (tai sovelluksen tarve)
- mahdollisesti korjataan virheitä vastaanottajalla, mutta ei uudelleenlähetäksiä
- mahdollisesti puskuroidaan paketteja, jotta selvittää pienistä viiveen vaihteluista
- myöhästyneet paketit käytännössä sama kuin kadonnut paketti, kuvan/puheen laatu kärsii

Sovelluskerros haluaa kuljetuspalvelulta...

► Kaistanleveys?

- Kuljetuskerros: Lähetän niin nopeasti (bittiä sekunnissa) kuin verkkokerros kykenee palvelemaan
- Verkkokerros: Lähetän niin nopeasti (bittiä sekunnissa) kuin linkkikerros kykenee palvelemaan
- Linkkikerros: Lähetän niin nopeasti (bittiä sekunnissa) kuin fyysinen siirtotie vain kykenee lähettämään linkille
- Linkkikerros: linkeillä voi olla eri tekniikka käytössä
- Best Effort - ja riippuu fyysisestä sekä linkkikerroksesta
- Sovelluskerros: alemmat kerrokset eivät voi taata tiedonsiirtonopeutta, täytyy itse keksiä jotain
- Sovelluskerros: Videostream, puskuroidaan dataa ennenkuin käyttäjä painaa 'play'
- Sovelluskerros: Reaaliaikainen video/puhe, vaihdetaan/valitaan heikompi laatuinen kuvan/äänen laatu

Sovelluskerros haluaa kuljetuspalvelulta...

- ▶ Tietoturva?
 - ▶ Kuljetuskerros: Ei tunnistautumista tai salausta
 - ▶ Verkkokerros: IPsec (Internet Protocol Security), tunnistautuminen ja koko verkko liikenteen salausta kahden laitteen välillä
 - ▶ Verkkokerros: IPsec on yksi vaihtoehto VPN (Virtual Private Network) toteutukseksi
 - ▶ Linkkikerros: Riippuu linkistä, esim.
 - ▶ Ethernet: Ei tunnistautumista tai salausta
 - ▶ WLAN: Ei tunnistautumista, salausta optiona
 - ▶ 2G/3G/4G: tunnistautuminen ja salausta aina
 - ▶ Sovelluskerros: alempien kerrosten ei tarvitse taata tietoturvaa, täytyy itse keksiä jotain
 - ▶ Sovelluskerros: Transport Layer Security (TLS) tai Secure Sockets Layer (SSL) tai vastaava sovelluskerroksen ja kuljetuskerroksen väliin
 - ▶ Sovelluskerros: Jokin muu salausta- ja/tai tunnistautumismenetelmä liitettynä sovelluskerroksen protokollaan

Verkkokerroksen palvelut

- ▶ Data oikealle laitteelle
 - ▶ Pitää selvittää reitti: Reititysprotokollat
 - ▶ Pitää kuljettaa data: IP protokolla
 - ▶ Virhetilanteet ja verkon diagnostiikka: ICMP protokolla
- ▶ Laite: Reititin
 - ▶ Reititys (routing): selvitetään reitit IP osoitteisiin
 - ▶ Välitys (forwarding): lähetetään dataa eteenpäin reititysinformaation perusteella
 - ▶ Reititysalgoritmi (protokolla): määrittää reitit ja ylläpitää reititystaulua
 - ▶ Kun pakettia välitetään, katsotaan reititystaulusta seuraava laite jolle paketti lähetetään

IP osoite kertoo laitteen sijainnin?

- ▶ IP osoite ei kerro sijaintia maapallolla! Vain sen kuinka paketit reititetään kohteeseen
- ▶ IP osoitteiden sijainnista voi olla tietoa: Sijaintitieto joudutaan keräämään esim. seuraamalla reittejä
- ▶ Sijainnin voi piilottaa proxy palvelimella
 - ▶ Etsi ensin jokin proxy palvelin [google haku](#)
 - ▶ Katso optioista että scriptien suoritus sallitaan
 - ▶ Mene sitten kurssin kotisivulle kopiomalla ao. osoite proxy sivustolle
 - ▶ <http://users.jyu.fi/~arjuvi/opetus/itkp104/2014/>
 - ▶ Toivottavasti edes toinen vasemmassa reunassa olevista skripteistä näyttää "sijaintisi"
 - ▶ Sen jälkeen mene ao. sivustolle, joka kerää tietoa siitä mistä päin maailmaa vierailijat tulevat
 - ▶ http://users.jyu.fi/~arjuvi/opetus/itkp104/2014/ITKP104_proxy_hits_2014.html
 - ▶ Jos intoa riittää niin kokeile eri maissa sijaitsevia proxy palvelimia

IP osoite

- ▶ "A name indicates what we seek. An **address** indicates where it is. A route indicates how to get there." (RFC 791)
- ▶ Kertoo laitteen sijainnin verkkohierarkiassa (ei fyysistä sijaintipaikkaa)
 - ▶ Osoite on hierarkkinen - prefix-osa ja laiteosa (aiemmin oli verkko-osa, aliverkko-osa, ja laiteosa)
 - ▶ Prefix-osan perusteella paketti välitetään oikeaan verkkoon
 - ▶ IP osoitteen koko
 - ▶ IPv4: 32 bittiä (binääriluku - 4 294 967 296 (2^{32}) osoitetta), joka koostuu neljästä tavusta
 - ▶ Ihmisystävällinen merkintä tapa, muutetaan tavut positiivisiksi kokonaisluvuiksi ja erotetaan ne pisteillä
 - ▶ esim. 173.194.32.63
(10101101.11000010.00100000.00111111)
 - ▶ IPv6: 128 bittiä (binääriluku - noin 3.4×10^{38} (2^{128}) osoitetta)
 - ▶ Ihmisystävällinen merkintä tapa, jaetaan kahdeksaan neljän heksadesimaaliluvun ryhmään
 - ▶ esim. 2a00:1450:400f:801::1017, joka on lyhennys osoitteesta 2a00:1450:400f:8010:0000:0000:0000:1017 eli
0010101000000000 0001010001010000 0100000000001111
1000000000010000 0000000000000000 0000000000000000
0000000000000000 0001000000010111

IP osoitetyypit

- ▶ Unicast - täsmälähetys (esim. Yle Areena, Nettiradiot, yms.)
 - ▶ Yleisin osoitetyyppi, "yhden" laitteen osoite, voidaan käyttää lähettämiseen sekä vastaanottamiseen
- ▶ Broadcast - yleislähetys (TV ja radio -ohjelmat radio/kaapeliverkosta)
 - ▶ Yleislähetysosoite, voidaan lähettää (ali)verkon kaikille laitteille sama paketti
 - ▶ 255.255.255.255 kaikille lähiverkon laitteille (ei mene reitittimen läpi)
 - ▶ tai esim. aliverkkoon 192.0.2.0/24 yleislähetysosoite on 192.0.2.255
 - ▶ IPv6:ssa ei mahdollista, hoidetaan Multicast:llä

IP osoitetyypit - jatkuu

- ▶ Multicast - ryhmälähetys (näin 'pitäisi' Yle Areena yms. toimia!!!)
 - ▶ Monilähetysosoite, laitteet voivat liittyä Multicast ryhmään (IGMP protokolla), jossa paketit välitetään kaikille jäsenille
 - ▶ Varattu osoitteet väliltä 224.0.0.0 - 239.255.255.255 (IPv4) ja ff00::/8 (IPv6)
 - ▶ Reitittimet hoitavat paketin kopiomisen ja välittämisen kaikille ryhmän jäsenille
- ▶ Anycast - 'Jokulähetys' (...)
 - ▶ Paketti voidaan välittää monelle vastaanottajalle, mutta valitaan se joka on fyysisesti lähimpänä
 - ▶ Käytetään kuorman tasaukseen palvelimilla sekä esim. DSN juurinimipalvelinten hajauttamiseen
 - ▶ IPv4:ssa ei suoraa tukea, hoidetaan yleensä AS:ien välisellä reititysprotokollalla (BGP)
 - ▶ IPv6:ssa osa spesifikaatiota

IP aliverkotus

- ▶ Aluksi ensimmäiset 8 bittiä oli verkko-osa (ja loput 'muuta' osaa), sitten kehitettiin luokallinen osoitteistus

Class	Leading bit values	Network bits	Host bits	Number of networks	Addresses per network	Start address	End address
A	0	8	24	128 (2^7)	16,777,216 (2^{24})	0.0.0.0	127.255.255.255
B	10	16	16	16,384 (2^{14})	65,536 (2^{16})	128.0.0.0	191.255.255.255
C	110	24	8	2,097,152 (2^{21})	256 (2^8)	192.0.0.0	223.255.255.255

- ▶ D = Multicast, E = Varattu
- ▶ Lopulta: Classless Inter-Domain Routing (CIDR)
- ▶ Luokaton verkkotunnusten (domain) välinen reititys

IP aliverkotus - CIDR

▶ Aliverkkomaski

- ▶ Kertoo kuinka monta bittiä IP osoitteesta on prefix osaa (loput on sitten laitteille)
- ▶ Esim. /8 tarkoittaa että 8 ensimmäistä bittiä on prefix osaa
- ▶ Tällöin siis aliverkkomaskissa kahdeksalla ensimmäisellä bitillä on arvo 1 (ja lopuilla arvo 0)
- ▶ esim. 11111111 00000000 00000000 00000000
- ▶ yo. voidaan merkitä myös 255.0.0.0
- ▶ Tai esim. /23 jota vastaa maski
- ▶ 11111111 11111111 11111110 00000000
- ▶ eli 255.255.254.0

IPv4 Route Table

Windows tietokone: *netstat -rn* tai *route print*

Network	Destination	Netmask	Gateway	Interface	Metric
	0.0.0.0	0.0.0.0	130.234.169.1	130.234.169.82	276
	130.234.169.0	255.255.255.128	On-link	130.234.169.82	276
	130.234.169.82	255.255.255.255	On-link	130.234.169.82	276

- ▶ Koneen osoite: 130.234.169.82
- ▶ Aliverkon osoite: 130.234.169.0/25
- ▶ Aliverkkomaski: 255.255.255.128

IPv4 reititys - kohde samassa verkossa

Aliverkkomaskit				Verkko-osoite
11111111	11111111	11111111	11111111	130.234.169.82
11111111	11111111	11111111	10000000	130.234.169.0
00000000	00000000	00000000	00000000	0.0.0.0

- ▶ Lähetetään osoitteeseen: 130.234.169.47 (10000010 11101010 10101001 00101111)
- ▶ Käydään reititystaulun kaikki rivit läpi
- ▶ Tehdään binäärinen AND operaatio kohdeosoitteen ja tarkasteltavan rivin maskin kanssa
- ▶ Verrataan yllä saatua tulosta tarkasteltavan rivin verkko-osoitteeseen
- ▶ Reititys bittien vastaavuuden perusteella
- ▶ Jos usealla verkko-osoitteella sama määrä, valitaan se jonka maskissa on eniten ykkösiä

IPv4 reitin valinta - kohde samassa verkossa

```
=====
AND          11111111 11111111 11111111 11111111    255.255.255.255
             10000010 11101010 10101001 00101111    130.234.169.47
             10000010 11101010 10101001 00101111    130.234.169.47
verrataan    10000010 11101010 10101001 01010010    130.234.169.82
             ---> 26 bittiä samoja
=====
```

```
=====
AND          11111111 11111111 11111111 10000000    255.255.255.128
             10000010 11101010.10101001 00101111    130.234.169.47
             10000010 11101010.10101001 00000000    130.234.169.0
verrataan    10000010 11101010.10101001 00000000    130.234.169.0
             ---> 32 bittiä (kaikki) samoja
=====
```

```
=====
AND          00000000 00000000 00000000 00000000    0.0.0.0
             10000010 11101010 10101001 00101111    130.234.169.47
             00000000 00000000 00000000 00000000    0.0.0.0
verrataan    00000000 00000000 00000000 00000000    0.0.0.0
             ---> 32 bittiä (kaikki) samoja
=====
```

IPv4 reitin valinta - kohde samassa verkossa

=====					
Network	Destination	Netmask	Gateway	Interface	Metric
	0.0.0.0	0.0.0.0	130.234.169.1	130.234.169.82	276
130.234.169.0	255.255.255.128		On-link	130.234.169.82	276
130.234.169.82	255.255.255.255		On-link	130.234.169.82	276
=====					

- ▶ Paketti lähetetään 130.234.169.0 verkkoon (koska sen maskissa enemmän ykkösiä edellisellä kalvolla)
- ▶ Koska kyseisellä rivillä ei ole gateway sarakkeessa IP osoitetta, on kohde siis samassa verkossa
- ▶ Seuraavaksi selvitetään kohteen (130.234.169.47) linkkikerroksen osoite (MAC osoite)
- ▶ Linkkikerroksen osoite selvitetään ARP protokollalla
- ▶ Ja kapsuloidaan verkkokerroksen paketti linkkikerroksen kehykseen

IPv4 reititys - kohde eri verkossa

Aliverkkomaskit				Verkko-osoite
11111111	11111111	11111111	11111111	130.234.169.82
11111111	11111111	11111111	10000000	130.234.169.0
00000000	00000000	00000000	00000000	0.0.0.0

- ▶ Lähetetään osoitteeseen: 130.234.169.147 (10000010 11101010 10101001 10010011)
- ▶ Koska meillä on 130.234.169/25 verkko, niin 147 on eri verkossa kuin 47
- ▶ Laite 'tietää' ylläolevan aliverkkomaskin perusteella

IPv4 reitin valinta - kohde eri verkossa

```
=====
AND          11111111 11111111 11111111 11111111 255.255.255.255
             10000010 11101010 10101001 10010011 130.234.169.147
             10000010 11101010 10101001 10010011 130.234.169.147
verrataan    10000010 11101010 10101001 01010010 130.234.169.82
             ---> 29 bittiä samoja
=====
```

```
=====
AND          11111111 11111111 11111111 10000000 255.255.255.128
             10000010 11101010.10101001 10010011 130.234.169.147
             10000010 11101010.10101001 10000000 130.234.169.128
verrataan    10000010 11101010.10101001 00000000 130.234.169.0
             ---> 31 bittiä samoja
=====
```

```
=====
AND          00000000 00000000 00000000 00000000 0.0.0.0
             10000010 11101010 10101001 10010011 130.234.169.147
             00000000 00000000 00000000 00000000 0.0.0.0
verrataan    00000000 00000000 00000000 00000000 0.0.0.0
             ---> 32 bittiä (kaikki) samoja
=====
```


IPv4 reitin valinta - kohde eri verkossa

=====					
Network	Destination	Netmask	Gateway	Interface	Metric
	0.0.0.0	0.0.0.0	130.234.169.1	130.234.169.82	276
	130.234.169.0	255.255.255.128	On-link	130.234.169.82	276
	130.234.169.82	255.255.255.255	On-link	130.234.169.82	276
=====					

- ▶ Paketti lähetetään 0.0.0.0 verkkoon, eli oletusreittiin (koska sen vertailussa eniten samoja bittejä edellisellä kalvolla)
- ▶ Koska kyseisellä rivillä on gateway sarakkeessa oletusreitittimen osoite, on kohde siis eri verkossa
- ▶ Seuraavaksi selvitetään reitittimen (130.234.169.1) linkkikerroksen osoite (MAC osoite)
- ▶ Linkkikerroksen osoite selvitetään ARP protokollalla
- ▶ Ja kapsuloidaan verkkokerroksen paketti linkkikerroksen kehykseen

Tutki mitä tietoja löydät

- ▶ <https://stat.ripe.net/>

Esimerkki hierarkisesta reitityksestä

- ▶ ao. kuva draw.io (SaaS) sovelluksessa

Hierarkinen reititys.svg

Hierarkinen reititys - jos sitä ei olisi...

- ▶ Skaalautuvuus

- ▶ Jokaisen reitittimen täytyisi ylläpitää tietoa koko Internetin rakenteesta, jos ei olisi hierarkiaa
- ▶ Internet ruuhkautuisi reitittimien päivitysviesteistä
- ▶ Reititysalgoritmitmien laskenta kestäisi kauan

- ▶ Autonomia

- ▶ Kaikki maailman verkot joutuisivat käyttämään samaa reititys protokollaa
- ▶ Yritykset/organisaatiot eivät pystyisi piilottamaan oman verkkonsa toteutusta ulkopuolisilta
- ▶ Oman verkon hallinta olisi maailmanlaajuisen reititysprotokollan vastuulla, eikä itsellä

Hierarkinen reititys - Toteutus

- ▶ IP osoite on hierarkinen (CIDR)
- ▶ Autonomous System (AS)
 - ▶ Reitittimet organisoidaan kuuluvaksi tiettyyn AS:ään
 - ▶ Yleensä yksi operaattori on yksi AS (isoilla operaattoreilla on useita AS tunnuksia)
 - ▶ AS:n sisällä
 - ▶ operaattori hallitsee verkkoaan ja käyttää haluamaansa reititysprotokollaa
 - ▶ reitittimet osaavat välittää sisäisen liikenteen
 - ▶ ulosmenevästä liikenteestä tiedetään reunareititin, joka vastaa AS:n ulkoisesta liikenteestä
 - ▶ AS:stä ulos
 - ▶ Mainostetaan mitkä osoiteavaruudet kuuluvat AS:n verkkoon, ei yksittäisiä osoitteita
 - ▶ Mainostetaan myös mitä muita AS:iä voidaan kyseisen verkon kautta saavuttaa
 - ▶ ulosmenevä liikenne lähetetään sellaiseen AS:ään joka mainostaa reittiä
 - ▶ ulosmenevän liikenteen kohde-AS:ään vaikuttaa myös tiedonsiirron hinta

Mistä niitä IPv4 osoitteita saa?

- ▶ IANA jakaa osoitelohkot alueellisille rekistereille
 - ▶ jakoi viimeiset viisi IPv4 osoitelohkoa (Last /8) 3.2.2011
 - ▶ /8 lohko sisältää noin 16 miljoona osoitetta
- ▶ Alueelliset rekisterit - Regional Internet Registry (RIR)
 - ▶ RIPE Network Coordination Centre (RIPE NCC)
 - ▶ Käytti osoitelohkonsa toisena loppuun (14.9.2012)
 - ▶ Jakaa nyt (rajoitetusti) osoitteita viimeisestä /8 lohkostaan
 - ▶ American Registry for Internet Numbers (ARIN)
 - ▶ Lähestyy vaihetta jossa joudutaan jakamaan osoitteita viimeisestä /8 lohkoista
 - ▶ Asia-Pacific Network Information Centre (APNIC)
 - ▶ Käytti osoitelohkonsa ensimmäisenä loppuun (15.4.2011)
 - ▶ Jakaa nyt (rajoitetusti) osoitteita viimeisestä /8 lohkostaan
 - ▶ Latin American and Caribbean Internet Address Registry (LACNIC)
 - ▶ Jakaa osoitteita normaalisti kunnes n. 8 miljoonaa osoitetta jäljellä (last /9)
 - ▶ African Network Information Centre (AfriNIC)
 - ▶ vuonna 2014 vielä 50+ miljoona osoitetta jaettavissa

Saako euroopassa enää IPv4 osoitteita?

- ▶ RIPE Network Coordination Centre (RIPE NCC)
 - ▶ Käytti vapaat osoitteensa loppuun alkamalla allokoimaan osoitteita viimeisestä /8 lohkoista
 - ▶ Jakaa vain yhden /22 (1024 IPv4 osoitetta) kullekin LIR (Local Internet registry):lle
 - ▶ Myönnetään vain jos hakijalla on olemassa oleva IPv6 allokatio
 - ▶ Enää ei ole mahdollista (uuden) operaattorin olla yhteydessä Internettiin vain IPv4 tekniikalla
 - ▶ Yksi /16 lohko varataan IPv4 → IPv6 siirtymistä varten
 - ▶ Kaikkien operaattoreiden tulisi ottaa IPv6 käyttöön

IPv4 vs IPv6

- ▶ Erot
 - ▶ osoite
 - ▶ kehysrakenne yksinkertaistettu
 - ▶ poistettu harvoin käytettyjä kenttiä
 - ▶ Vähentää pakettien käsittelyaikaa reitittimillä
 - ▶ ei DHCP:tä
 - ▶ IPv6 kykenee automaattisesti generoimaan itselleen osoitteen
 - ▶ ei erillisiä yleislähetysosoitteita

Linkkikerros - Teoria 2

- ▶ Johdanto ja palvelut
- ▶ Monipääsyprotokollat
- ▶ Linkkikerroksen osoitteet

Linkkikerroksen palvelut

- ▶ ”Linkkikerros kuljettaa verkkokerroksen datan linkin yli seuraavalle (verkko)laitteelle, ja tekee tämän jokaisen linkin osalta vaikka linkkiteknologia vaihtuisi, jottei verkkokerroksen tarvitse siitä huolehtia”
- ▶ Yksittäisiä palveluita, joilla yo. toteutetaan
 - ▶ Kehystys ja linkille pääsy
 - ▶ Kapsuloidaan ylemmän kerroksen data (lisätään header/trailer)
 - ▶ Jos jaettu media, niin kanavanvaraus/kanavalle(linkille) pääsy
 - ▶ MAC osoite, linkkikerroksen osoite
 - ▶ Luotettavuus linkillä *kahden* laitteen välillä
 - ▶ Käytetään harvoin jos linkillä pieni virhetodennäköisyys
 - ▶ Käytetään esim. langattomilla linkeillä
 - ▶ Miksi luotettava tiedonsiirto sekä kuljetus että linkkikerroksella?
 - ▶ Ks. viestiseinä tai [täältä](#)
 - ▶ Vuonvalvonta
 - ▶ Virheiden havaitseminen
 - ▶ Virheiden korjaaminen
 - ▶ half-duplex vs. full-duplex

Monipääsylinkit

- ▶ Kaksi linkkityyppiä
 - ▶ point-to-point (Pisteestä-pisteeseen, kahden laitteen välinen)
 - ▶ ADSL
 - ▶ Kytketty Ethernet (yhdistetään kytkimeen)
 - ▶ Jaettu (yleislähetys) langallinen tai langaton
 - ▶ Vanha Paksu Ethernet
 - ▶ WLAN
 - ▶ Matkapuhelinyhteys

Monipääsyprotokollat

- ▶ Kolme kategoriaa
 - ▶ Kanavan jakaminen
 - ▶ Jaetaan tiedonsiirtokanava pienempiin paloihin (aika, taajuus, koodi)
 - ▶ Pala on ainoastaan yhden laitteen käytössä
 - ▶ Hajasaanti (Random Access)
 - ▶ Kaikki laitteet käyttävät (koko) kanavaa vapaasti
 - ▶ Hyväksytään törmäykset (useampi kuin yksi yrittää lähettää samaan aikaan)
 - ▶ Toivutaan törmäyksistä
 - ▶ Vuorottelu
 - ▶ Jokainen laite lähettää vuorollaan
 - ▶ Vuoro kiertää laitteelta toiselle
 - ▶ Vuoron pituus voi riippua lähetettävästä datan määrästä

Kanavan jakaminen

Kuva

Hajasaantiprotokollat

- ▶ Ei varata resursseja etukäteen
 - ▶ Jos on lähetettävää, niin lähetetään täydellä nopeudella
- ▶ Hajasaantiprotokolla määrittelee
 - ▶ miten havaitaan törmäykset
 - ▶ miten selvittää törmäyksistä
 - ▶ Esimerkkejä: (Slotted) Aloha, CSMA, CSMA/CD, CSMA/CA

Vuorotteluprotokollat

- ▶ Kaksi kategoriaa
 - ▶ Polling: Isäntälaitte kyselee orjalaitteilta "onko lähetettävää" ja jakaa lähetysvuoroja tarpeen mukaan
 - ▶ Token passing: "Lähetyslupa (token)-viestiä välitetään laitteelta toiselle"

Monipääsyprotokollien vertailua

- ▶ Kanavan jakaminen (FDMA ja TDMA)
 - ▶ Jakaa kanavan tehokkaasti ja reilusti suurella kuormalla
 - ▶ Pienellä kuormalla tehoton: Viive kanavalle pääsyssä, ja vain osa kanavasta käytössä
 - ▶ CDMA:ssa koko kanava käytössä (jaetaan aktiivisten kesken - kun tulee lisää aktiivisia käyttäjiä niin kaikkien tiedonsiirtonopeus pienenee)
- ▶ Hajasaantiprotokolla
 - ▶ Tehokas pienellä kuormalla: yksi laite voi käyttää koko kanavan kapasiteetin
 - ▶ Suuri kuorma: törmäyksistä aiheutuu uudelleenlähetyksiä ja viivettä
- ▶ Vuorotteluprotokolla
 - ▶ Ylläolevien parhaat ominaisuudet, mutta
 - ▶ viivettä vuoron jakamisen vuoksi
 - ▶ Kanavan kapasiteettia 'tuhlataan' vuoronjakamiseen, koko kapasiteetti ei datansiirron käytössä
 - ▶ Single point of failure: Isäntälaitte tai token viesti

Linkkikerroksen osoitteet

- ▶ MAC (tai LAN tai Fyysinen tai Ethernet) osoite
 - ▶ Tehtävä: Käytetään kehyksen viemiseen 'paikallisesti' liittynnältä toiselle (fyysisesti kytketylle) liittynnälle (saman IP-verkon sisällä)
 - ▶ 48 bittinen MAC osoite (useimmissa lähiverkoissa), poltettu verkkokorttiin ROM muistiin (yleensä)
 - ▶ Esim. 00-24-E8-42-9E-89, (heksadesimaalimuodossa), missä 8 bittiset tavut erotettu toisistaan '-' -merkillä
- ▶ MAC osoitteiden allokoinnista vastaa IEEE
- ▶ (Verkkokortti)valmistajat ostavat MAC osoitteita
- ▶ Analogia
 - ▶ MAC osoite: sosiaaliturvatunnus
 - ▶ IP osoite: postiosoite
- ▶ Toinen vertaus (nämä oli väärinpäin luentotaltioinissa tällä kalvolla)
 - ▶ MAC osoite: istumapaikka luentosalissa
 - ▶ IP osoite: Henkilön nimi
- ▶ MAC osoite ei hierarkinen: voidaan siirtää verkosta toiseen
- ▶ IP osoite on hierarkinen, ei voida siirtää verkosta toiseen

ARP (Address Resolution Protocol)

- ▶ Who has 130.234.169.1? Tell 130.234.169.82
- ▶ Posti tuo kirjeen:
 - ▶ Kysyy käytävällä: Missä on Ari Viinikaisen huone? (IP)
 - ▶ Minä huudan henkilötunnukseni (MAC) (henkkarit asetettu huoneiden oviin)
 - ▶ Posti kävelee käytävää ja verraten henkkareita jättää postin minun ovelle
- ▶ Luentosali:
 - ▶ Opettaja kysyy: missä istuu Oiva Opiskelija? (IP)
 - ▶ Opiskelija nostaa käden pystyyn, opettaja katsoo ja kirjaa rivin ja paikan ylös (MAC)
 - ▶ Kirjoittaa rivin ja paikan kirjekuoreen ja laittaa kirjeen kulkemaan luentosalissa.