Datakommunikation och nätverksteknik - del 2

För en komplett lista av aktuella kommandon, användning och förklaring, se då till `cheat cheat.pdf`.

Vecka 1

Kapitel 1 - Routing

Routing handlar om att sammankoppla flera nätverk. Switchar är vanligtvis bara förgreningsdosor. Att hitta bästa vägen från en källa till en destination är nyckeln till bra routing.

I *RAM*-minnet finns det nu körande IOS, running-config, IP- och ARP-tabeller och paketbuffern.

I *ROM*-minnet finns boot-instruktioner, grundläggande diagnostik och en grundläggande version av IOS.

I *NVRAM*-minnet finns startup-config.

I *flash*-minnet finns permanent lagring så som IOS och andra system-relaterade filer.

Default gateway är den första routern som frågas om t.ex. datorn inte vet vart ett paket ska skickas. Exempelvis används en router i ett hemnätverk för att ta trafik från det lokala nätverket till ett *remote*-nätverk. DHCP används för att lösa konfigurationen automatiskt - IP-tilldelning och val av default gateway.

Lastbalansering. Om en router hittar flera bästa vägar till en destination används de lika.

Routing-tabellen används för att hitta nästa *hop* till en router på nätverket.

I IOS visas typen av anslutning med följande flaggor:

- **L**: Local route interface
- **C**: Directly connected interface
- **S**: Static route
- **D**: Learned dynamically from another router using the EIGRP routing protocol
- **O**: Learned dynamically from another router using the OSPF routing protocol

Vecka 3

Kapitel 4 - Switchade nätverk

Intro om konvergerade nätverk. Finns i anteckningar för del 1.

Ett *borderless switched network* är hierarkiskt (varje del har en roll), modulärt (sömlöst utökning av nätverket), *always on*, och flexibelt (intelligent belastnings-delning).

Generellt om en switch - MAC / Port- table o.s.v. Finns i anteckningar för del 1.

Cut-Through Switching gör det snabbare ….

Kapitel 5 - Switch

Om prompten i IOS är `switch:` så saknas OS.

För att helt starta om switchen:

1. Anslut med konsoll-kabel
2. Konfigurera terminalen på PC:n
3. Dra ut strömkabeln
4. Anslut strömkabeln, tryck in och håll ner Mode-knappen till dess att System LED lyser orange kort innan den blir grön

Det går att felsöka systemet genom att kontrollera LED-lamporna vid portarna:

- Av - ingen länk
- Grönt - länk ansluten
- Blinkande grönt - aktiv data-överförning
- Blinkande grönt och oranget - länk-fel
- Oranget - porten skickar inte data. Vanligt de första 30 sekunderna efter anslutning
- Blinkande oranget - porten är blockad för att stoppa en *switch-loop*

Säkerhet till switchar kan fås bland annat genom att avaktivera telnet och aktivera SSH.

Man kan even stänga av portar som inte används

Port-säkerhet:

Protect – data från en okänd MAC-adress droppas, ingen notifikation visas på switchen

Restrict - data från en okänd MAC-adress droppas, en notifikation visas av switchen och en räknare (violation counter) räknas upp

Shutdown – (standard) interfacet stängs ned och portens LED stängs av. En räknare (violation counter) räknas upp. För att sätta igång interfacet igen måste `shutdown` och sen `no shutdown` köras.

Kapitel 6 - VLAN

VLAN kan segmentera LAN-enheter utan att bry sig om den fysiska platsen för en användare eller enhet. Man skapar LAN-nätverk och VLAN:en är isolerade. Paket kan enbart färdas mellan olika VLAN om en router är konfigurerad.

Fördelar med VLAN är att det ökar säkerheten, reducerar kostnaden, förbättrar prestandan, medför mindre broadcast-domäner, ökat effektiviteten för IT-avdelningen (förbättrar hanteringsmöjligheter) etc.

Man kan vara anslutna till samma switch men ändå vara del av olika VLAN (Gäst-del, student-del, personal-del etc.). Man tänker helt enkelt mer logiskt än fysiskt.

- Default VLAN - VLAN 1, alla portar på switchen är del i VLAN 1 per default

- Data VLAN, skapas för en specifik grupp av användare eller enheter. De bär användar-genererad trafik
- Native VLAN, bär all otaggad trafik. Kommer inte från en VLAN-port. Använder VLAN 1 per automatik
- Management VLAN, skapas för att hålla hanteringstrafik så som SSH, SNMP, Syslog etc. Använder VLAN 1 per automatik

Visas genom *show vlan brief*.

Cisco använder vanligtvis VLAN 150 för VoIP där data prioriteras i form av en tillförlitlig bandbredd, låg latens (<150ms).

En *VLAN trunk* är en point-to-point-länk som bär mer än ett VLAN. Konfigureras vanligtvis mellan switchar för att tillåta kommunikation mellan VLAN. En VLAN-trunk eller *trunk ports* är inte associerade med något specifikt VLAN.

Cisco IOS stöder *IEEE 802.1q*, ett vanligt VLAN-trunk-protokoll.

Cisco använder vanligtvis VLAN 99 för Management och Native.

Broadcast-domäner kan hanteras av VLAN. Exempelvis kan en VLAN-trunk skicka vidare broadcast-paket till andra VLAN.

Innan en frame är vidarebefodrad till enn *VLAN-trunk-länk* måste den taggas med VLAN-information. Man taggar helt enkelt framen i ett fält *Tag*. Denna tag består av 2 bytes för typ, 3 bitar för prioritet, 1 bit för CFI (tillhör *token ring*-protokollet) och 12 bitar för VLAN ID (i den ordningen). VLAN-trunken tar bort informationen från framen och skickar vidare den.

Trafik som skickas på Native VLAN taggas inte. Kommer det in otaggade frames, placeras de i Native VLAN och skickas vidare. Finns det inga portar som är associerade med Native VLAN eller en VLAN-trunk så droppas de. Det ska inte finnas trafik på Native VLAN som inte härstammar från någon typ av VLAN-trunk - vanliga datorer ska alltså inte jobba mot Native VLAN.

Implementation av VLAN:

Normal range VLANs: 1-1005 (primära, prioriterade - sparas i vlan.dat i flashminnet)

Extended range VLANs: 1006-4096 (sparas i NVRAM)

Vecka 4

Kapitel 7 - Access Control Lists (ACL)

Funktionalitet för brandväggar. Dessa ACL:er kan göra flera olika saker så som att begräna nätverkstrafiken så att prestandan i nätverket ökar, blockera videotrafik, styra flödet, filtrera olika typer av trafik så som SSH, blockera nätverk eller hosts. Rent konkret är i CISCOs IOS ACL en serie kommandon som kontrollerar huruvida ett paket ska droppas eller ej. Det normala är att ACL är avaktiverat.

En ACL är en sekventiell lista som har "tillåt"-rader och "blockera"-rader. Dessa rader refereras som *ACE - Access Control Entries* eller *ACL Statements*. När trafik passerar så kontrolleras paketen gentemot listan. Detta kallas *packet filtering*. Man kan med denna teknik analysera inkommande och/eller utgående data. Detta sker antingen på lager 3 eller 4. Det sista ACEn är alltid en *implicit deny* (blockera allt som inte tillåts). Därför borde listan innehålla minst en tillåtelse.

ACL ingående filtreras innan de skickas vidare på nätverket beroende på interface. Utgående ACL filtrerar efter att paketet routas, oberoende av vilket interface det kom ifrån.

Man kan använda sig av masker för att definiera ACL, så som wildcard-masker (0.0.255.255), 32-bitars binärt tal där de sista 16 bitarna är 1. En 1 är en låsning, 0 släpper igenom. Exempelvis så är 0.0.0.0 *match all*, 255.255.255.255 *ignore all*. Att blockera 192.168.0.0 ger masken 11000000 10101000 00000000 00000000.

Att beräkna en wildcard-mask går att göra med hjälp av subnet-masken.

$$\begin{array}{r} 255.255.255.255 \\ -\ 255.255.255.000 \\ \hline 255 \end{array}$$

Det vill säga att wildcard-masken för att tillåta 192.168.3.0-nätverket med subnetmasken 255.255.255.0 är 0.0.0.255. Man subtraherar alltså wildcard-masken med subnet-masken.

Ett annat sätt är att använda IP-adresser (`host` och `any`).

`host` blir 0.0.0.0

`any` blir 255.255.255.255

```
# Ekvivalenta uttryck
R1(config)# access-list 1 permit 0.0.0.0 255.255.255.255
R1(config)# access-list 1 permit any
```

```
# Ekvivalenta uttryck
R1(config)# access-list 1 permit 192.168.10.10 0.0.0.0
R1(config)# access-list 1 permit host 192.168.10.10
```

Det är lätt att göra bort sig när det kommer till konfiguration av ACL.

Det är viktigt att använda ACL i routrarna som ska agera brandväggar - så som mellan det interna nätverket och det externa nätverket.

Man kan använda dem i routrar mellan två delar i ett internt nätverk för att kontrollera trafiken.

Man kan bara ha **en** ACL **per protokoll** (IPv4, IPv6), **per interface** (G0/0), **per riktning** (IN eller OUT). Man brukar inte blanda protokoll etc. i samma ACL.

En ACL bör ingå i säkerhetspolicyn för företaget. En beskrivning bör skapas för att förtydliga exakt vad ACLn ska göra. Skriv dina ACLer i en textredigerare så att de kan sparas och återanvändas. Testa ACLn i ett utvecklingsnätverk innan de används i produktion.

Man ska placera en ACL där de ger mest betydelse och inverkan. Generellt finns två typer. *Extended ACL* (baserad på destinationsadresser) och *Standard ACL* (baserad på sourceadresser). Extended ACL placeras så nära källan som möjligt. Standard ACL placeras så nära destinationen som möjligt. Närhet antyder både router och interface som ska användas.

Exempel:

```
# All trafik från alla hostar på 192.168.10.x släpps igenom
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255

# Välj interface som listan ska användas på
R1(config)# interface serial 0/0/0

#Tilldela listan
R1(config-if)# ip access-group 1 out
```

```
# Tillåt all trafik från ett subnät men tillåt en specifik host
# Tar bort nuvarande ACL 1
R1(config)# no access-list 1
# Deny 192.168.10.10
R1(config)# access-list 1 deny host 192.168.10.10
# Allow andra hosts på 192.168.10.0/24
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
# Notera implicit match i slutet på ACLn som blockerar alla andra
# Tilldela ACL
R1(config)# interface S0/0/0
R1(config)# ip access-group 1 out
```

```
# Ta bort nuvarande ACL 1
R1(config)# no access-list 1
# Deny 192.168.10.10
R1(config)# access-list 1 deny host 192.168.10.10
# Allow alla andra
R1(config)# access-list 1 permit any
# Tilldela listan G0/0
R1(config)# interface G0/0
R1(config-if)# ip access-group 1 in
```

```
# Skapa en ny lista med ett namn NO_ACCESS istället för 1
# Namn är alfanumeriska, case-sensitive och unika
R1(config)# ip access-list standard NO_ACCESS
# Blockera 192.168.11.10
R1(config-std-nacl)# deny host 192.168.11.10
# Tillåt alla andra
R1(config-std-nacl)# permit any
# Gå tillbaka till globalt konfigurationsläge
R1(config-std-nacl)# exit
# Tilldela interfacet G0/0 listan
R1(config)# interface G0/0
R1(config-if)# ip access-group NO_ACCESS out
```

Man bör dock använda en textredigerare för att konfigurera listorna. Man kan då använda `show running-config` för att visa ACLn, kopiera den till en textredigerare och ändra den där. Sedan kan man klistra in ACLn från textredigerare till konsollen.

För att redigera en ACL använder man `show access-lists 1`. Indexet visas då till vänster (10, 20, ...). För att ta bort en rad skriver man då `ip access-list standard 1` och sedan `no 10` och ändrar den med `10 deny host 192.168.10.10`. Man kan lägga till en rad så kan man skriva `ip access-list standard NO_ACCESS` följt av `15 deny host 192.168.11.11` där 15 är raden i listan som ska läggas till.

För att verifiera en ACL använder man som tidigare `show ip interface S0/0/0` där en outgoing och ingoing ACL visas. `show access-lists` går även bra att använda.

Man kan styra ACL för VTY-portar så som för SSH.

```
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# access-class 21 in
R1(config)# exit
R1(config)# access-list 21 permit 192.168.10.0 0.0.0.2555
R1(config)# access-list 21 deny any
```

Kapitel 8 - DHCP

Protokollet DHCP delar ut IP-adresser dynamiskt. En dedikerad DHCP-server är enkel att hantera och växer i takt med behov. En CISCO-router kan agera DHCP i ett medelstort nätverk.

1. Dator: DHCP Discover (Broadcast) - "Jag skulle vilja ha en adress"
2. Server: DHCP Offer (Unicast) - "Jag är en DHCP-server, här är adressen jag kan ge dig"
3. Dator: DHCP Request (Broadcast) - "Jag accepterar adressen"
4. Server: DHCP Ack (Unicast) - "Jag förstår att du accepterar adressen"

DHCP-protokollet för datorn använder UDP på source-port 68 och destination-port 67. Om det skickas från servern används UDP source-port 67 och destination-port 68.

```
# Ta bort några adresser från poolen
R1(config)# ip dhcp excluded-address 192.168.10.1 92.168.10.9
R1(config)# ip dhcp excluded-address 192.168.10.254
# Starta igång poolen och konfigurera den LAN-POOL-1 är namnet som tilldelas poolen
R1(config)# ip dhcp pool LAN-POOL-1
# Konfigurera servern
R1(dhcp-config)# network 192.168.10.0 255.255.255.0
R1(dhcp-config)# default-router 192.168.10.1
R1(dhcp-config)# dns-server 192.168.11.5
R1(dhcp-config)# domain-name example.com
R1(dhcp-config)# end
# Verifiera konfiguration
R1# show running config | section dhcp
R1# show ip dhcp binding
R1# show ip dhcp server statistics
```

DHCP Discover skickas som en broadcast, men det normala är att routrar inte skickar vidare broadcasts. En CISCO IOS *helper address* konfigureras så att dessa meddelanden skickas vidare.

```
# Konfigurera interface
R1# interface G0/0
# Tilldela "helper"
R1(config-if)# ip helper-address 192.168.11.6
R1(config-if)# end
# Verifiera
R1# show ip interface G0/0
```

Man kan använda `debug` för att undersöka fel - `debug ip dhcp server events`.

DHCPv6 använder ett lite annat protokoll - *SLAAC - Stateless Address Autoconfiguration*. Det finns även *Stateful DHCPv6* som liknar DHCPv4.

1. Dator: Router solitication (RS) (IPv6 all-routers multicast)- "Jag behöver en Router Advertisement från routern"
2. Server: Router Advertisement (RA) (IPv6 all-nodes multicast) - "Här är ditt prefix, prefixlängd och lite annan information"
3. Dator: IPv6 global unicast address - "Jag genererar mitt ID, sedan används prefix och prefixlängd för att skapa min IPv6-adress"
4. Dator: Duplicate Address Detection (IPv6 solicited node multicast) - "Används den här adressen av någon annan?"

Med andra ord tilldelas ingen IP-adress utan annan information skickas till alla enheter på nätverket.

Stateless:

1. Dator: "Jag behöver en RA från servern"
2.

A. Server: (SLAAC) "Använd bara den här RAn"

B. Server: (SLAAC och DHCPv6) "Använd den här RAn och DHCPv6-servern"

C. Server: (DHCPv6) "Använd den här DHCPv6-servern"

Kapitel 9 - NAT & PAT

NAT: Network Adress Translation

PAT: Port Address Translation

I IPv4 är ett "problem" att det finns privata adress-områden som aldrig routas ut på nätverket. Detta är en åtgärd för att säkerställa att adresserna inte skulle ta slut. En publik adress kan med NAT hålla flera privata adresser;

- 10.0.0.0 - 10.255.255.255 - 10.0.0.0/8
- 172.16.0.0 - 172.31.255.255 - 172.16.0.0/12
- 192.168.0.0 - 192.168.255.255 - 192.168.0.0/16

Grundtanken i internet är egentligen att varje enhet ska ha en unik adress så att alla kan komminucera med varandra. IPv4 (NAT) bryter lite mot detta tänk.

Vanligtvis är det en *border router* som är konfigurerad att använda NAT. Det är routern som ansluts till det externa nätverket (så som internet).

I grund och botten byter routern ut den interna IP-adressen till en av routerns IP-adresser. En server på internet ser då routern som en klient, snarare än den interna enheten. Routern omvandlar tillbaka adresser hos mottagna paket till den interna adressen. Klient och server är omedvetna om att en NAT har skett. Det viktiga är att routern har koll på vilka paket som skickades från/till vilken enhet. Detta löses med en NAT-tabell.

Inside global	**Inside local**	**Outside local**	**Outside global**
209.165.200.226	192.168.10.10	209.165.201.1	209.165.201.1

Detta ger routern mer att göra och kan resultera i prestandaförluster.

Det finns statisk NAT och dynamisk NAT. Statisk NAT förser en koppling mellan en intern dators adress och en egen publik adress - användbart för servrar. Dynamisk NAT förser en intern dator med en publik adress i mån av plats. Först till kvarn gäller. Generellt gäller dessa två lägen enbart då man köpt en pool av IP-adresser så som för företag.

PAT använder en enda publik IP. Detta förekommer oftast i nätverk för hem och småföretag. PAT använder portar för att använda en publik IP till flera interna klienter. Tabellen liknar den ovan men använder även en port.

Inside global	**Inside local**	**Outside local**	**Outside global**
209.165.200.226:1555	192.168.10.10:1555	209.165.201.1:80	209.165.201.1:80
209.165.200.226:1331	192.168.10.11:1331	209.165.202.129:80	209.165.202.129:80

Med andra ord så översätts den interna IP-adressen till en publik adress och porten byts till en för routern tillgänglig port. Här släpper vi nätverkslagret och utnyttjar transportlagret. Generellt ska allt på internet använda IP-adresser, inte portar. Detta bryter också mot internets principer.

Vanligtvis försöker PAT använda den ursprunliga porten, men om den är upptagen används en annan port från samma grupp;

- 0-511
- 512-1023
- 1024-65535

Finns det ingen tillgänglig port försöker PAT gå vidare till nästa IP-adress ur den publika poolen.

Fördelarna med NAT är bland andra:

- Varje företag kan behålla privata IP-adresser
- Ökar flexibiliteten
    - Flera NAT-pooler
    - Backup-pooler
    - *load balancing*
- Man kan byta ISP utan att byta interna IP-adresser
- Gömmer interna IPv4-adresser

Nackdelarna är bland andra:

- Prestandan degraderas
- End-to-end-funktionalitet tappas
- End-to-end-spårning tappas
    - Man ser inte vilken dator i ett nätverk som användes. Adressen kanske inte ens ursprungar från samma land som enheten som ursprungligen användes
- Vissa applikationer som kräver end-to-end-adressering kan inte användas
- Felsökning kan bli mer problematisk
- *Tunneling* blir mer komplicerat
- Instansering av TCP-anslutningar kan störas

Det finns NAT för IPv6 (NAT64) även om det inte var tanken från början. Prefix: FC00::/7 (FC00 till FDFF). Detta nätverk routas inte ut på internet. Används mest för att koppla ihop IPv6- med IPv4-nätverk, inte för att omvandla privata till publika adresser.

Kapitel 10 - Device Discovery etc.

Device Discovery, Management och Maintenance. Lite av ett restkapitel.

För Device Discovery finns huvudsakligen två protokoll - *Cisco Discovery Protocol (CDP)*. Det bygger på lager 2 och innebär att enheter delar med sig av vilka enheter som upptäckts, namn av enheterna, typer av enheter, interface som används etc. Skickas var trettionde sekund. Man bör inte köra CDP om man inte behöver det - det avslöjar mycket om hur nätverket ser ut.

Ett annat protokoll, *Link Layer Discovery Protocol (LLDP)* är mer öppet. Fungerar även för andra fabrikat än CISCO. Delar med sig av sin identitet för lager 2. Skickas var trettionde sekund.

För management finns det flera saker man kan göra. En viktig sak för loggning är att bestämma klockan - att loggar förblir synkad. Bestäms genom att köra `R1# clock set 20:36:00 dec 11 2015` eller konfigurera *Network Time Protocol (NTP)*. NTP använder UDP port 123 och hämtar datum och tid från en viss källa (så som atomur).

Syslog är ett sätt att konfigurera ett logg-system. En astandard som använder UDP port 514. Skickar notifieringsmeddelanden över nätverket med information vad som händer switchar, servrar, brandväggar och routrar. Loggarna kan ses över konsol-lina, en terminal eller en dedikerad syslog-server. Default är att det går ut i terminalen.

Det finns flera *security levels*.

- Level 0 - Emergency
- Level 1 - Alert
- Level 2 - Critical
- Level 3 - Error
- Level 4 - Warning
- Level 5 - Notification
- Level 6 - Informational
- Level 7 - Debugging

Level 0-4 handlar om att mjukvara eller hårdvara har en felaktighet - funktionaliteten är påverkad. Level 5 handlar om normala event - notifieringar så som att ett interface har stängts av, systemet startats om osv. Level 6 är normal information som inte påverkar enhetens funktionalitet.

Att sköta router och switch handlar om att se att filsystemet, IOS:et är igång.

Att spara konfigurationsfilerna (running-config) till en textfil görs genom att använda Tera Term. Klicka på *File menu -> Log*, välj plats att spara filen på. Tera Term kommer nu att spara utmatning. Skriv in `show running-config` i terminalen och Tera Term sparar utmatningen.

Man kan även använda TFTP för att spara konfigurationen. Detta görs genom att köra `copy running-config tftp`. Vissa enheter har USB-portar som kan användas för lagring.