Dator- & telekommunikation

*Föreläsning 5*

# Hierarchical routing

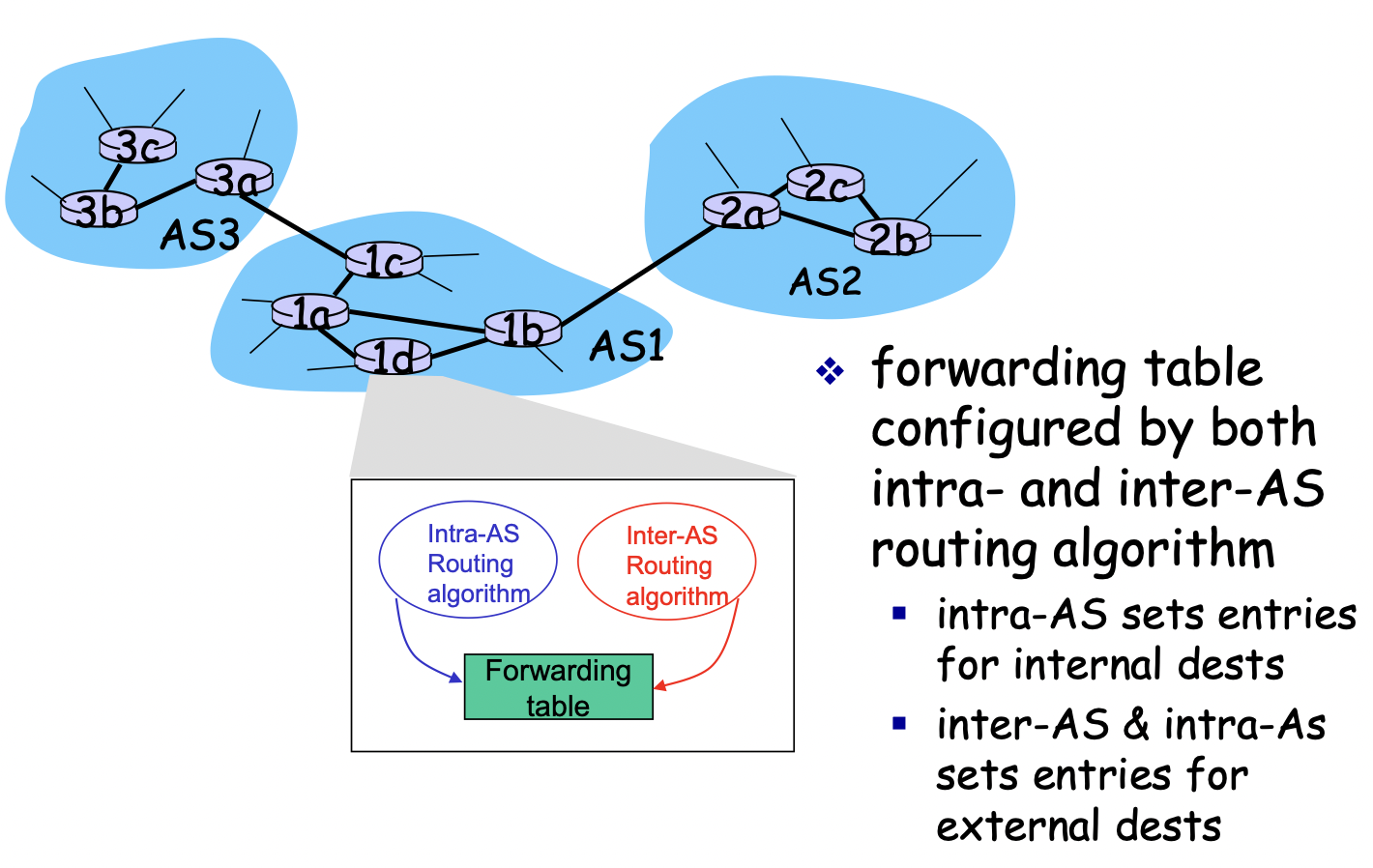
### Hierarchical Routing

| **Begrepp** | |
| --- | --- |
| **Hierarchical Routing** | * Smånätverk där en vägvalsalgoritm körs i varje delnätverk * Internt vägval, hur görs vägval mellan näten? (=inter-AS) |
| **Autonomous system (AS)** | * Aggregerade routrar i en region * Alla routrar i samma **AS** MÅSTE köra samma protokoll |
| **Intra-AS routing** | Protokoll inom ett AS (intern vägval) |
| **Inter-AS routing** | Protokoll utanför nätet/AS (externt vägval). Alla routrar på internet kör faktiskt samma inter-AS routing protokoll, nämligen **Border** **Gateway** **Protocol** (BGP). |
| **Gateway router** | Router som befinner sig i **kanten** (längst ut) på sitt egna AS  + har länk till en **annan** **router** i ett **annat** **AS** |
|  | |

### Interconnected ASes

* Inom varje AS, måste samma teknik/protokoll köras

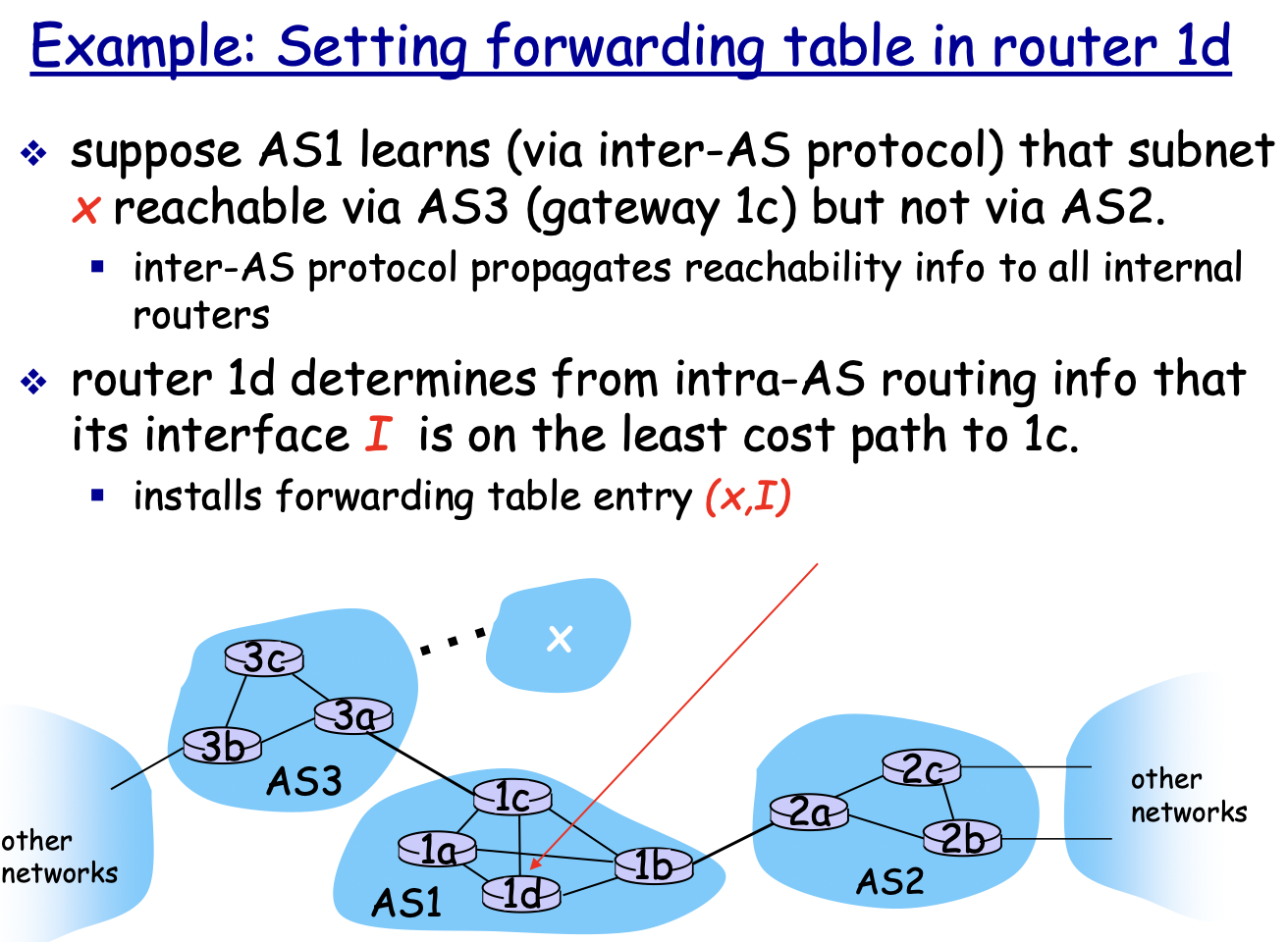
| Forwarding tabell byggs både upp av | |
| --- | --- |
| 1. Intra-AS routing | Har hand om det **interna** vägvalet & **externa** vägvalet |
| 2. Inter-AS routing | Har hand om det **externa** vägvalet |

****

# Inter-AS tasks (BGP=Border Gateway Protocol)

| **Problembeskrivning** | |
| --- | --- |
| * Anta att vi utgår **från** **AS1** och den tar emot ett datagram **utanför** **AS1**   + Hur ska routern veta till vilken **gateway** **router** den ska skicka till? | |
| **Lösning** | |
| 1. AS1 måste **lära** **sig** vilka **destinationer** som den kan nå **genom** **AS2** och **AS3** 2. Sprida vidare denna informationen (nåbarheten) till **ALLA** **routrar** i **AS1**   ⇒ Detta är **jobbet** som **inter-AS routing** har | |
|  | |

### Example: Setting forwarding table

****

### Example: Choosing among multiple ASes

| **Slide 1.** | **Slide 2.** |
| --- | --- |
|  |  |

# Intra-AS Routing (IGB = Interior Gateway protocol)

| **3 most common Intra-AS routing protocols/algoritmer** | |
| --- | --- |
| 1. [RIP](#_ijpi14gbydt6) | Routing Information Protocol |
| 2. [OSPF](#_x3kezs4qv0b3) | Open Shortest Path First |
| (3. IGRP) | Interior Gateway Routing Protocol (Används enbart av Cisco) |

# (1) Intra-AS routing: RIP

### RIP (= Routing Information Protocol)

* **Distance vector algorithm** (håller reda på vad det kostar att ta sig till grannen)
  + Distance metric: **max 15 hops**
    - Längre än 15 hopp ⇒ anses hamna i en loop, etc., dvs kastas
  + Varje länk kostar 1
  + Var 30 sekund skickar grannen sin tabell
  + Tar sändning av ett paket >**180 sekunder** uppdateras tabellen med att det tar **oändligheten** att skicka genom den specifika vägen

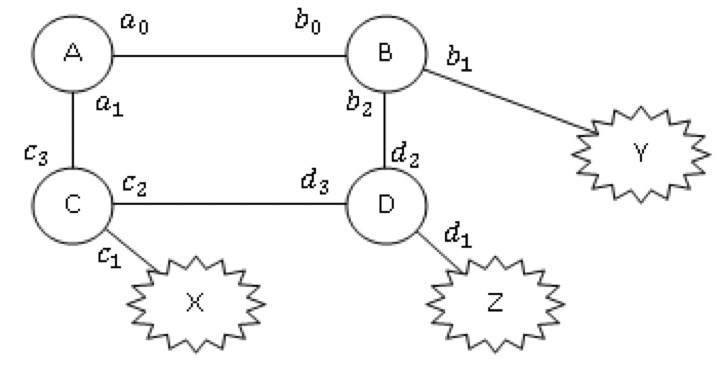
### RIP Table processing

* En **applikation** som ligger på **OSI-nivå 7**
* Skickas genom **UDP-paket**
* **Forwarding** **tabellen** ligger på **OSI-nivå 3** dock

### Skapa tabell till RIP

* Börja ALLTID med utgångspunkt i de routrar som är **närmast/har kontakt** med **subnäten**.
* Därefter jobbar du dig vidare tills **ALLA** routrar fått kontakt med subnäten.

### RIP Algoritm - Hur ska man tänka?

****

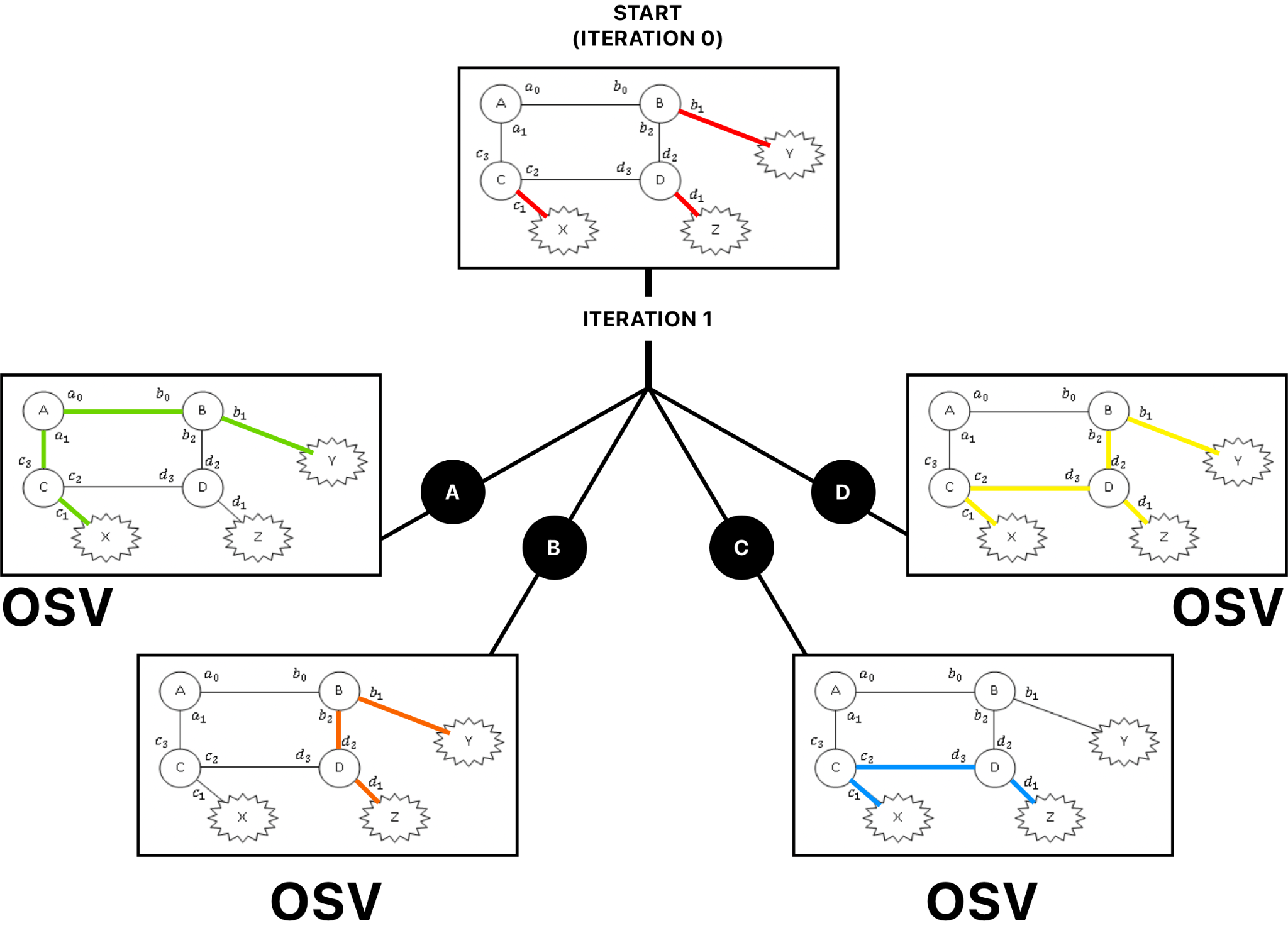
**Steg 1** - Hitta dem routrar som har direkt kontakt med de olika subnäten. Skriv ner tabell, med routrar som kolumner och rader som iterationer (ny rad var 30:e sekund).

* Rutorna fylls i med **subnät, länknamn, kostnad (längd)**
* Finns INGET subnät att nå, lämnas rutan tom

|  | A | B | C | D |
| --- | --- | --- | --- | --- |
| START | - | Y b1 1 | X c1 1 | Z d1 1 |

**Steg 2** - Fortsätt enligt de nya förutsättningarna och fyll i tabellen

* **Router A**: I iteration 2, kan nu **router A** nå **subnät X** respektive **Y** via **router B** respektive **C** (dvs router B och C kan nå subnät X och Y enligt första iterationen)
* **Router B**: I iteration 2 kan nu **router B** nå **subnät Z** genom **router D**
* Osv… med **router** **C** och **D**



|  | A | B | C | D |
| --- | --- | --- | --- | --- |
| START | - | Y b1 1 | X c1 1 | Z d1 1 |
| Iteration 1 | **X a1 5**  **Y a0 2** | Y b1 1  **Z b2 2** | X c1 1  **Z c2 2** | Z d1 1  **X d3 2**  **Y d2 2** |
| Iteration 2 | X a1 5  Y a0 2  **Z a0 3** | Y b1 1  Z b2 2  **X b2 3** | X c1 1  Z c2 2  **Y c2 3** | Z d1 1  X d3 2  Y d2 2 |

\*\*\* **FETSTILT** innebär nya tillagda vägar för den specifika iterationen \*\*\*

# (2) Intra-AS routing: OSPF

### OSPF (=Open Shortest Path First)

* Open = Publicly available
* Uses **link-state-algorithm**
  + Måste ha hela topologin i varje nod
  + **Dijkstra’s algorithm** for route computation

### Pros with OSPF compared to RIP

* **Säkerhet**: Lösenord krävs när en förändring i tabellerna ska göras
* Kan ha **flera** **vägar** som kostar **lika** mycket (i RIP finns endast en väg, alla kostar lika mycket)
* För **varje** **länk** kan det **kosta** **olika** mycket
  + Beror på ToS (Type of service)
  + Dvs beror på vilken “typ av paket” som skickas mellan routrarna
  + I IPv-paketen finns ett utrymme som bestämmer ToS
    - Ex. om det är ett “realtids-paket” (streaming) eller “fildelnings-paket”
    - “Realtids-paketet” kommer då att prioriteras och därmed vara billigare att skicka
* Integrated **uni**- and **multicast** support
  + **Unicast**
    - Skicka paket till en router
  + **Multicast**
    - Skicka paket till de som tillhör en viss grupp (AS)
  + **Broadcast**
    - Skicka paket till alla på nätet

# Inter-AS Routing (BGP = Border Gateway Protocol)

### 

### BGP algorithm: Basics + generell beskrivning

| **BGP ⇒ “The glue that holds the internet together”** | |
| --- | --- |
| * **BGP** hanterar destinationer som är utanför AS:et * I BGP så routar man INTE paket till specifika destinationsadresser, utan snarare till s.k. **prefix.**   + **Prefix** = representerar **ett** **subnät** eller en **grupp** av **subnät** * Routerns **forwarding-tabell** kommer bestå av datapunkter på formen **(x, I)**    + **x** = prefix   + **I** = interface nummer för en av routerns interface, dvs länk mellan sig själv och nästa router/etc. | |
| **BGP har 2 uppgifter / BGP förser varje router med 2 saker** | |
| 1. Nåbarhet till **prefix** genom närliggande AS | * BGP tillåter varje subnät (prefix) att berätta om sin existens till resten av internet * Subätet “skriker: Jag är här” och BGP låter all routrar på internet få veta om detta subnät * Utan BGP skulle ett subnät vara en isolerad ö. |
| 2. Fastställning av **“bästa”** vägen till **prefix** | * En router kan lära sig om **2 eller flera** olika vägar till ett **prefix** * För att bestämma den bästa vägen kommer routern köra BGP lokalt |
| **Generell (icke-detaljerad) beskrivning av BGP** | |
| * Anta att vi vill annonsera till **ALLA** **AS** i nedanstående system om **subnät (prefix) X** * Då kommer det att fungera enligt **följande** **steg**:  1. AS3 skickar BGP meddelande till AS2.    1. **Meddelande**: Subnät X existerar och finns i AS3 2. AS2 skickar BGP meddelande till AS1.    1. **Meddelande**: Subnät X existerar och kan nås genom att först passera AS2 och sedan gå till AS3.  * **Slutresultat**: Varje autonomt system får reda på X **OCH** lär sig **ÄVEN** om **vägen** till X | |

### BGP algorithm: Detaljerad beskrivning

| **Detaljerad beskrivning av BGP** | | | |
| --- | --- | --- | --- |
| * Vi utgår från samma nät som ovan!      * (I BGP) kommer **ett par** av routrar genomföra ett utbyte av **routing** **information** genom **TCP uppkoppling** (semi-permanent TCP connection) * TCP uppkoppling + BGP meddelanden (som sänds i TCP) = **BGP connection** | | | |
| **2 typer av BGP connections** | | | |
| 1. | | **eBGP** som är för den **externa** delen (används mellan gateway routrar) | |
| 2. | | **iBGP** som är för den **interna** delen (används inom ett AS) | |
|  | | | |
| **Beskrivning** (från slides: “BGP basics: distributing path information” ) | | | |
| **1.** | **eBGP** mellan **gateway-routrar** **2c (AS2)** och **3a (AS3)**  ⇒ **AS3** skickar alltså **information** **om** **“hur nås prefix X”** till **AS2** | | |
| **2.** | **Gateway-router 2c** kan nu överföra denna information till sina grann-routrar genom **iBGP** | | |
| **3.** | När **gateway-router 2a** tagit emotinformationen kan den m.h.a. **eBGP** föra över informationen till **gateway-router 1c** i **AS1. osv…**  **OBS! VARJE** router skapar **ny datapunkt (x, I)** i **routingtabell** när den lär sig om **nytt** **prefix**. | | |

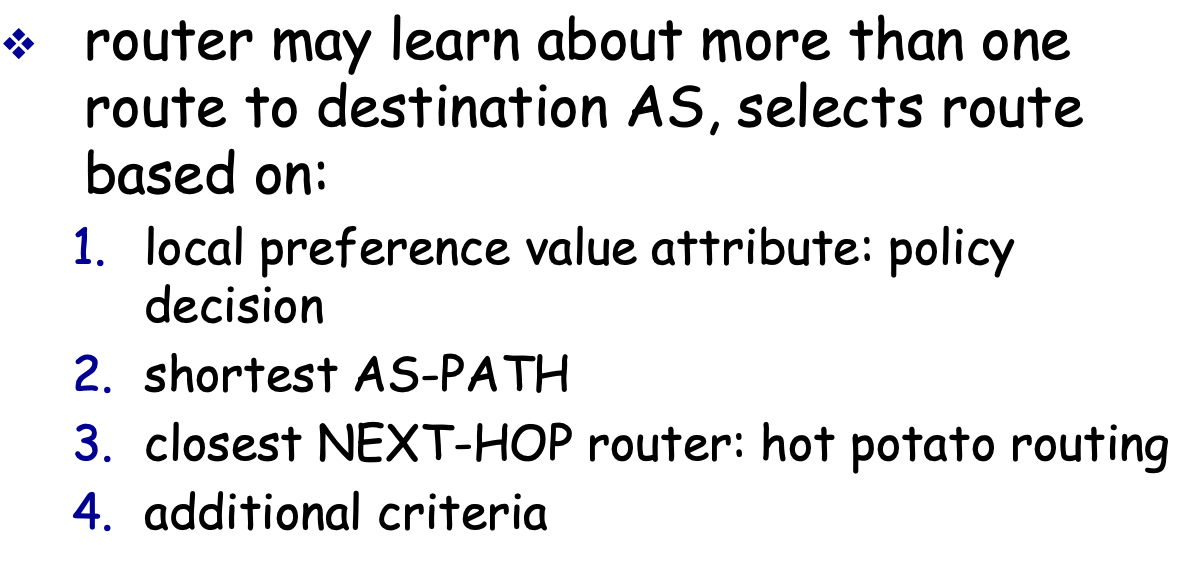
### Path attributes and BGP routes

* When router advertises a prefix across a BGP connection it includes 2 things:
  + **prefix** (subnet/group of subnets)
  + **attributes**
  + prefix + attributes = **route**

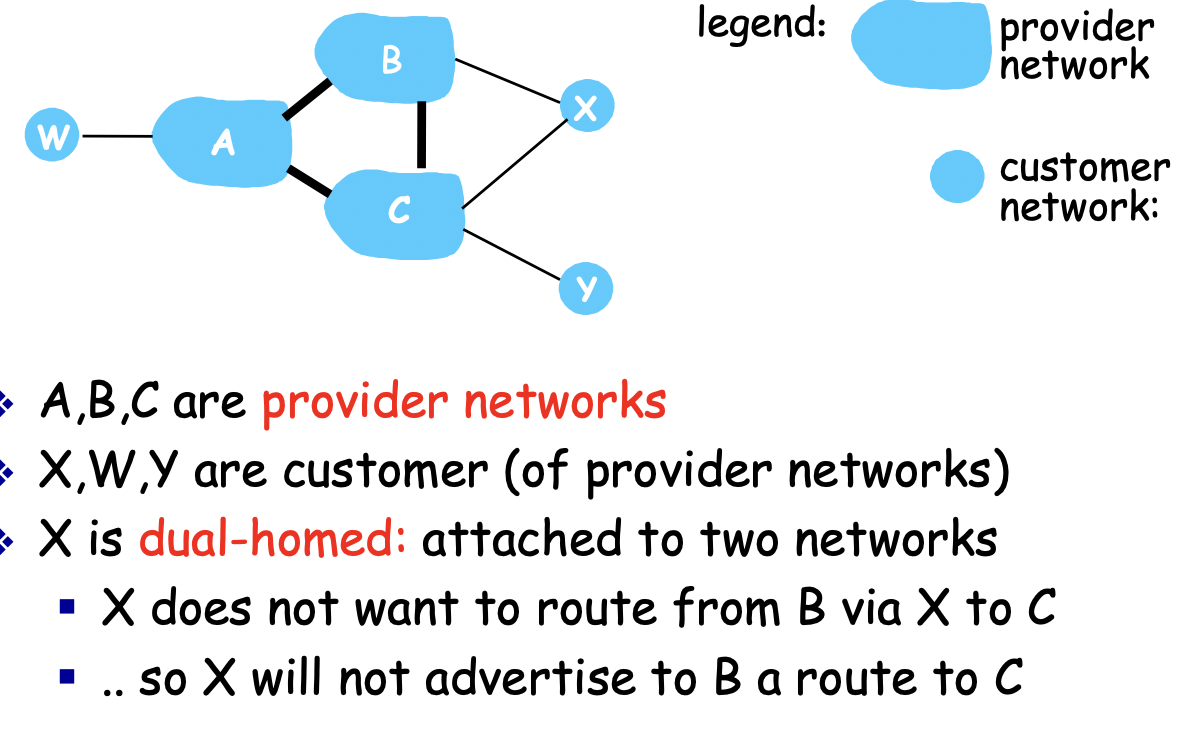
| **2 important attributes** | | |
| --- | --- | --- |
| **1.** | **AS-path** | Attribut som innehåller de autonoma system som “advertisement:et” passerat, dvs genom vilka **prefix** (AS/subnät) den har passerat  Ex. **AS-path** från AS1 till prefix X:  **⇒ “AS2 AS3”** |
| **2.** | **Next-hop** | IP-adressen av router-interface:et som börjar AS-path.  Ex. **next-hop** för rutten **“AS2 AS3 X”**, (dvs från AS1 till X, genom AS2).  **⇒ IP-adressen av** router-interfacepå vänstra sidan av **router 2a** |
|  | | |

### 

### BGP route selection (determine best route)



**BGP routing policy**



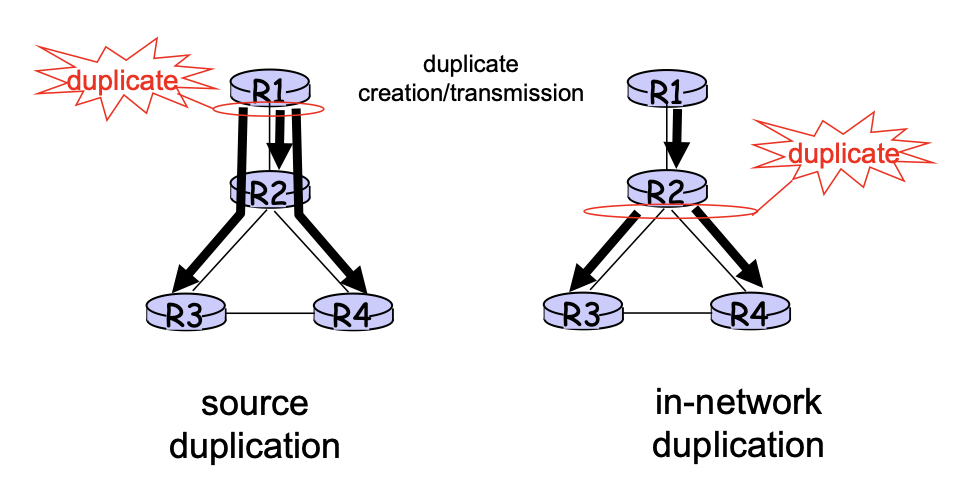
**Why different intra and inter-AS routing**

* Policy: Som tidigare nämnt
* Scale
* Performance

# Broadcast routing

### Broadcast routing

* Deliver packets from source to **ALL** other node
* 2 sätt
  + **1. Source** **duplication** = **INEFFICIENT!!**
  + **2. In-network duplication**

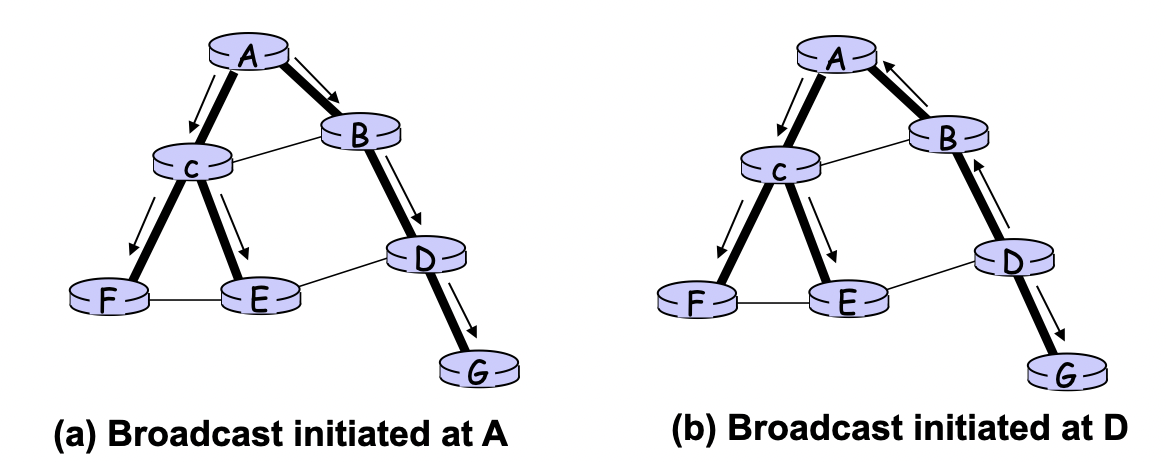


### In-network duplication

* Flooding
* Controlled flooding
* **Spanning tree** (aktuellt idag)

### Spanning tree

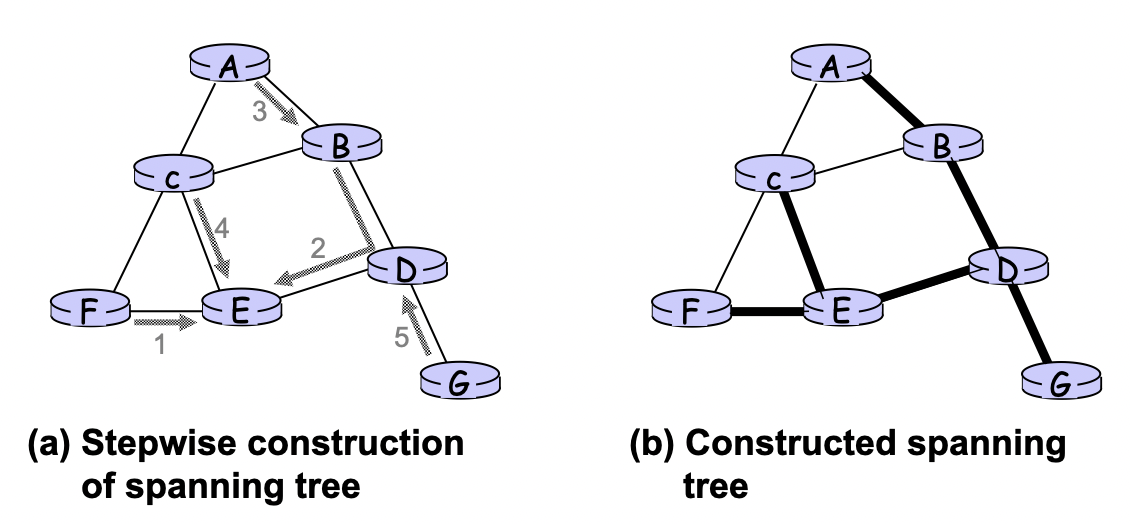
* Konstruera ett träd
* Noderna skickar paket i spanning tree (de tjocka noderna)



* **Tjocka** **noder**: tillhör nätet
* **Smala** **noder**: tillhör INTE nätet
* Paket skickas **BARA** genom de tjocka noderna

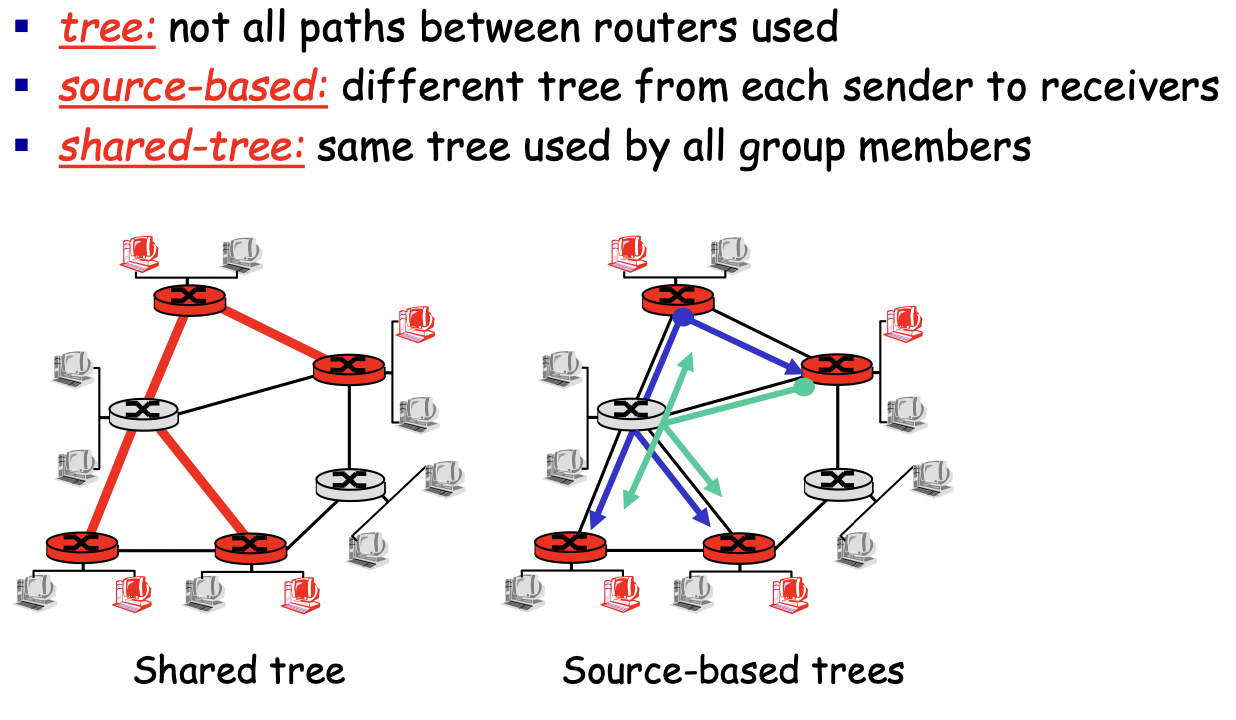
### Spanning-tree creation

* Center-nod (E)
* Varje nod skickar ett **unicast** till center-noden
  + “Jag vill vara med i detta träd”
  + Skickar paket tills den träffar en nod som redan är med i trädet
* **Router E** är center-nod i nedanstående bild



**Multicast Routing**

* Mål: Hitta ett/flera träd som kopplar samman routrar som har lokala multicast grupper



### 

### Approaches to Multicast tree

* **source-based tree**
  + 1. Shortest path tree
  + 2. Reverse path tree
* **group-shared tree**
  + 1. Minimal spanning
    - 2. Center-based trees
* **Shortest** **past** **tree** (source based)
* **Reverse** **path** **Forwarding** (source based)
  + Om jag fått ett paket från användare
    - Där vi erhåller information om vilken väg den har gått
  + Då använder vi det åt “andra hållet” också
  + Dvs den utnyttjar varje nods “unicast”