College of Information Studies

University of Maryland Hornbake Library Building College Park, MD 20742-4345

DNS

Session 24

INST 346

Technologies, Infrastructure and Architecture

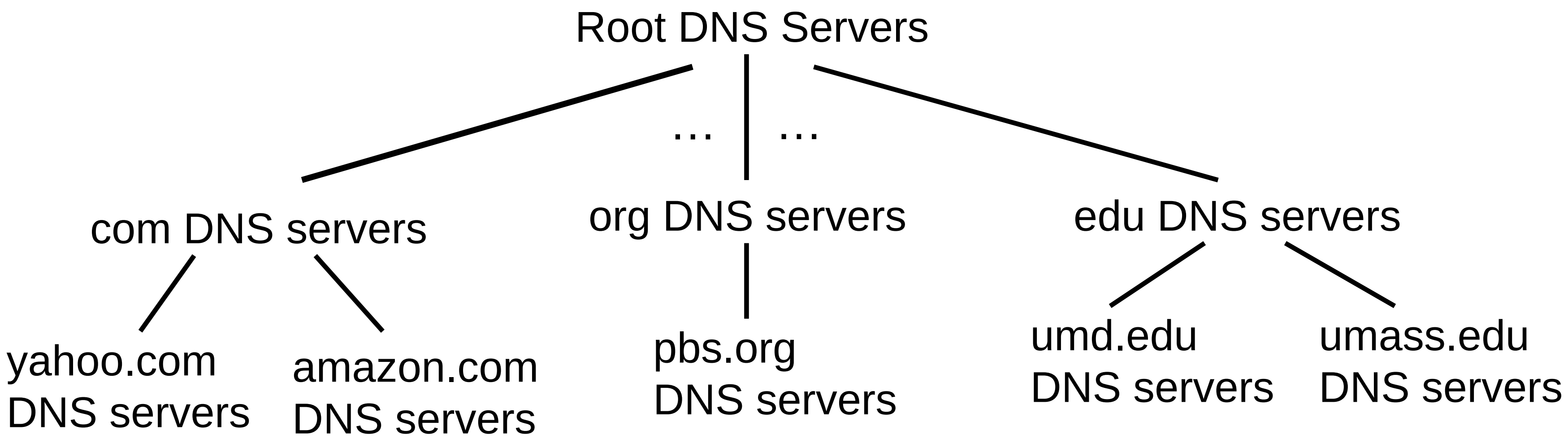
DNS: Domain Name System

- IP address (32 bit) - used for addressing datagrams
- Domain name used by humans (e.g., www.yahoo.com)

Domain Name System:

- *Maps between domain names and IP addresses*
- *Distributed database* implemented in hierarchy of *name servers*
- *Application-layer protocol:* hosts and name servers communicate to *resolve* domain names

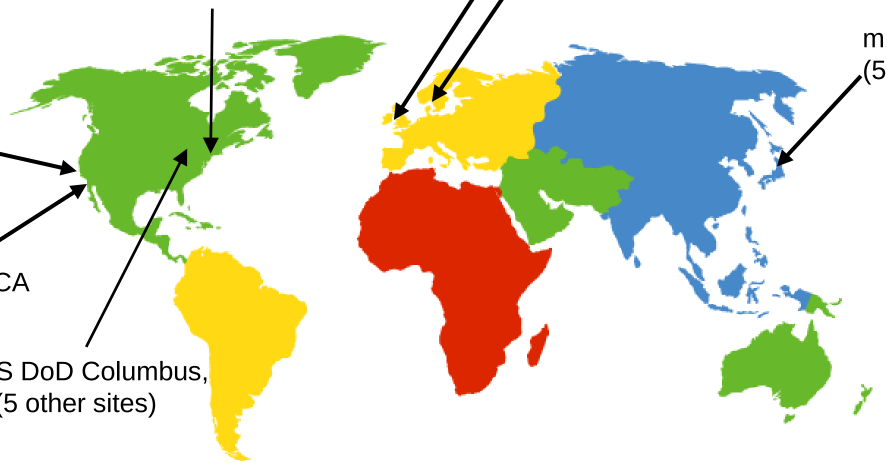
DNS: a distributed, hierarchical database



client wants IP for www.amazon.com; 1st approximation:

- client queries root server to find .com DNS server
- client queries .com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

DNS: root name servers



c. Cogent, Herndon, VA (5 other sites)
d. U Maryland College Park, MD
h. ARL Aberdeen, MD
j. Verisign, Dulles VA (69 other sites)

k. RIPE London (17 other sites)

i. Netnod, Stockholm (37 other sites)

e. NASA Mt View, CA
f. Internet Software C.
Palo Alto, CA (and 48 other sites)

m. WIDE Tokyo
(5 other sites)

a. Verisign, Los Angeles CA
(5 other sites)
b. USC-ISI Marina del Rey, CA
l. ICANN Los Angeles, CA
(41 other sites)

g. US DoD Columbus, OH (5 other sites)

13 logical root name “servers” worldwide

- *each “server” replicated many times*

Local DNS name server

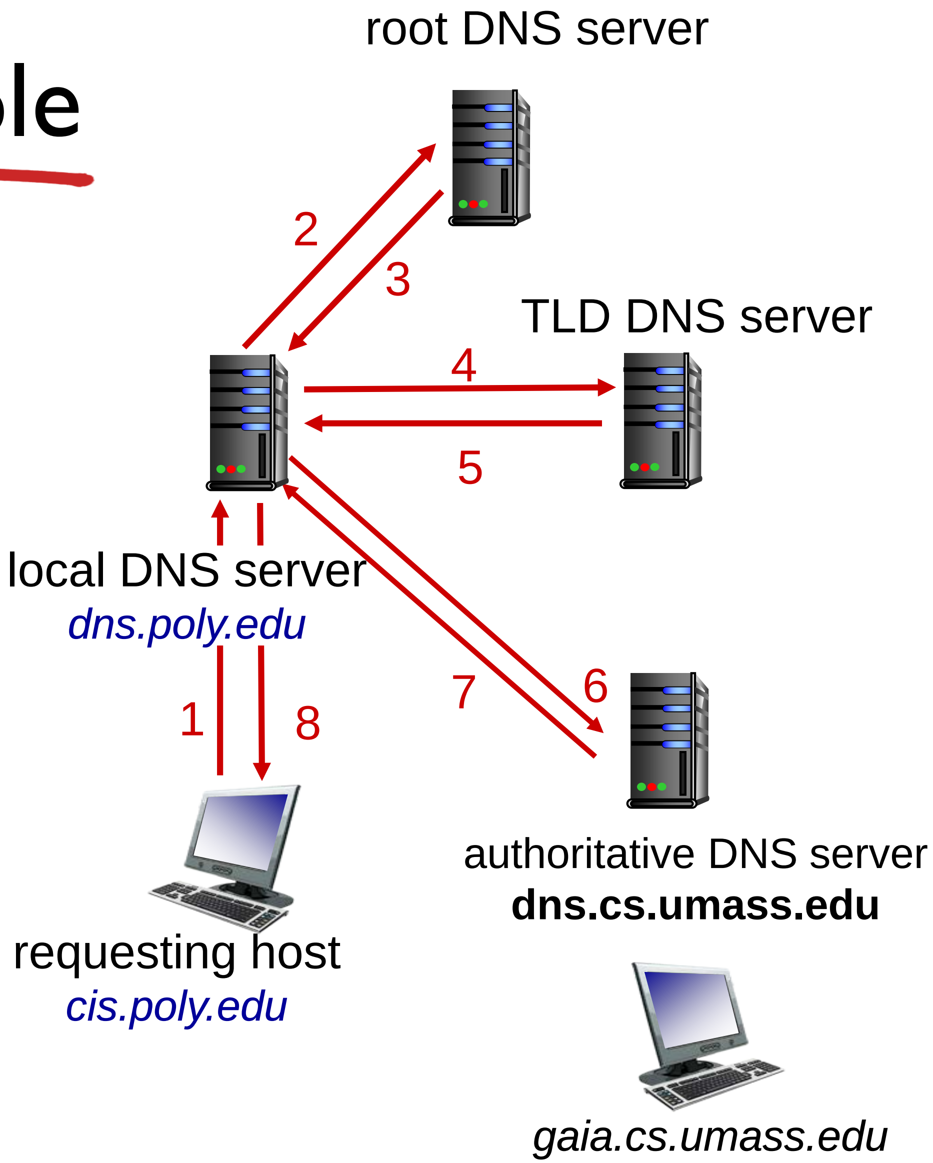
- Each Internet Service Provider (ISP) has one
  - also called “default name server”
- Hosts send DNS queries to their local DNS server
  - Answered from local cache of recent name-to-address translation pairs if possible
  - If not cached, obtains the translation from the DNS hierarchy

DNS name resolution example

- host at cis.poly.edu wants IP address for gaia.cs.umass.edu

iterated query:

- contacted server replies with name of next server to contact



DNS: caching, updating records

- once (any) name server learns mapping, it *caches* mapping
  - Cache entries timeout (disappear) after some time (TTL)
  - TLD servers are typically cached in local name servers
    - Thus root name servers are not often visited
- Cached entries may be *out-of-date*
  - if a DNS host changes IP address, that may not be known Internet-wide until all TTLs expire

Resource Records

RR format: `(name, value, type, ttl)`

type=A

- **`name`** is hostname
- **`value`** is IP address

type=CNAME

- **`name`** is alias name for some "canonical" (the real) name
- **`value`** is canonical name
- **`Example: www.ibm.com`** is really **`servereast.backup2.ibm.com`**

type=NS

- **`name`** is domain (e.g., foo.com)
- **`value`** is hostname of authoritative name server for this domain

type=MX

- **`value`** is name of mailserver associated with **`name`**

Inserting records into DNS

- example: new startup "Network Utopia"
- register name networkuptopia.com at *DNS registrar* (e.g., Network Solutions)
  - provide names, IP addresses of authoritative name server (primary and secondary)
  - registrar inserts two RRs into .com TLD server:

    ```
    (networkutopia.com, dns1.networkutopia.com, NS)
    (dns1.networkutopia.com, 212.212.212.1, A)
    ```
- create authoritative server type A record for www.networkuptopia.com; type MX record for networkutopia.com

Example DNS lookup

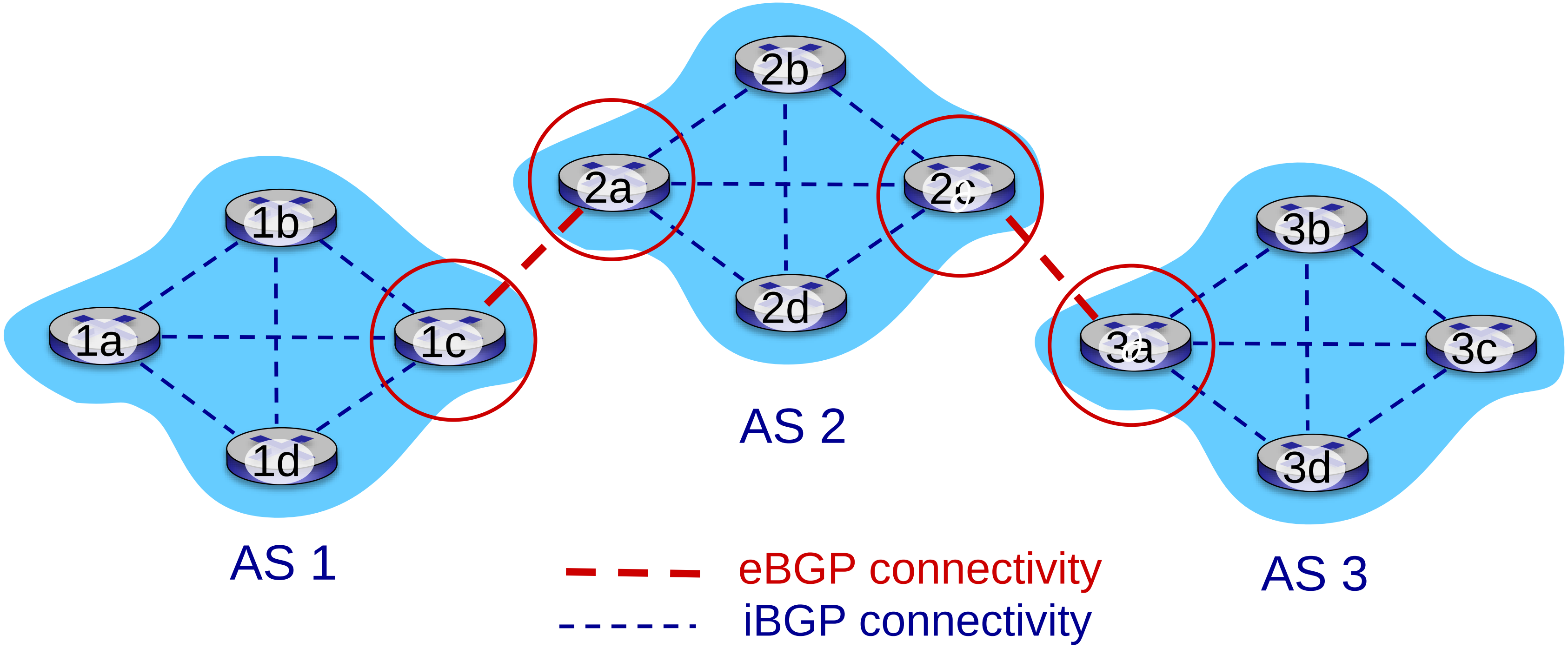
Try on your command line or terminal:

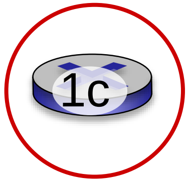
```
nslookup www.umd.edu
```

Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): *the* de facto inter-domain routing protocol
  - "glue that holds the Internet together"
- BGP provides each AS a means to:
  - eBGP: obtain subnet reachability information from neighboring ASes
  - iBGP: propagate reachability information to all AS-internal routers.
  - determine "good" routes to other networks based on reachability information and *policy*
- allows subnet to advertise its existence to rest of Internet: *"I am here"*

eBGP, iBGP connections

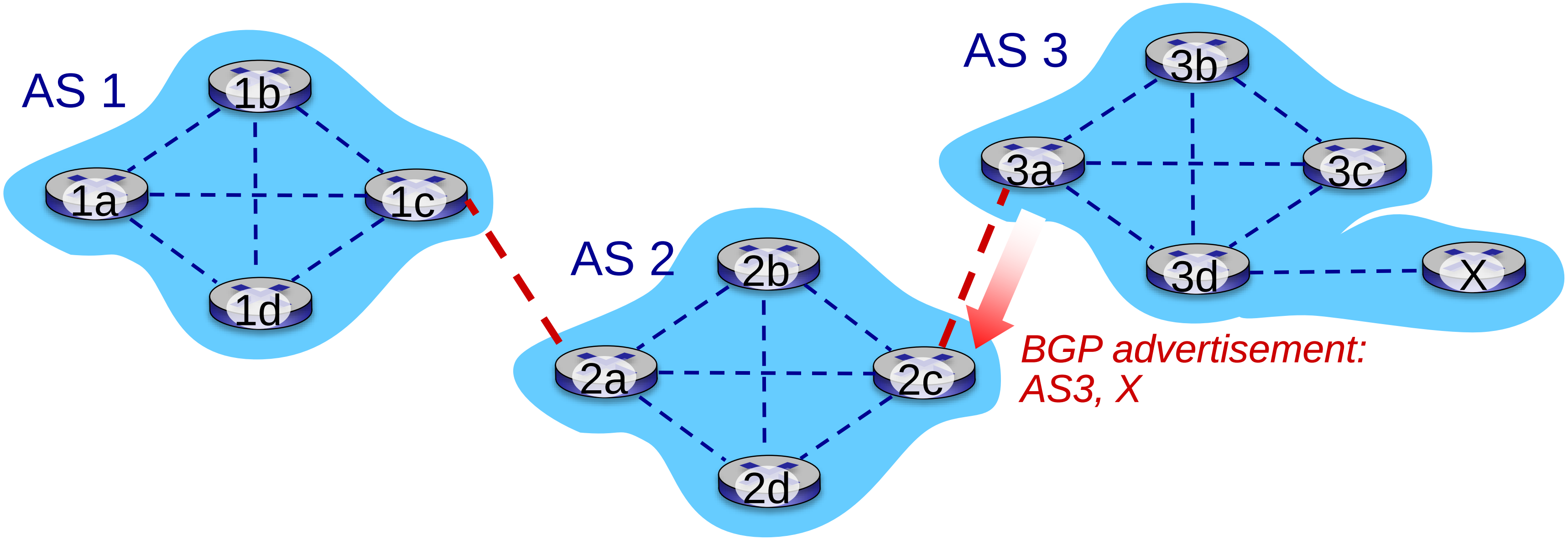




gateway routers run both eBGP and iBGP protools

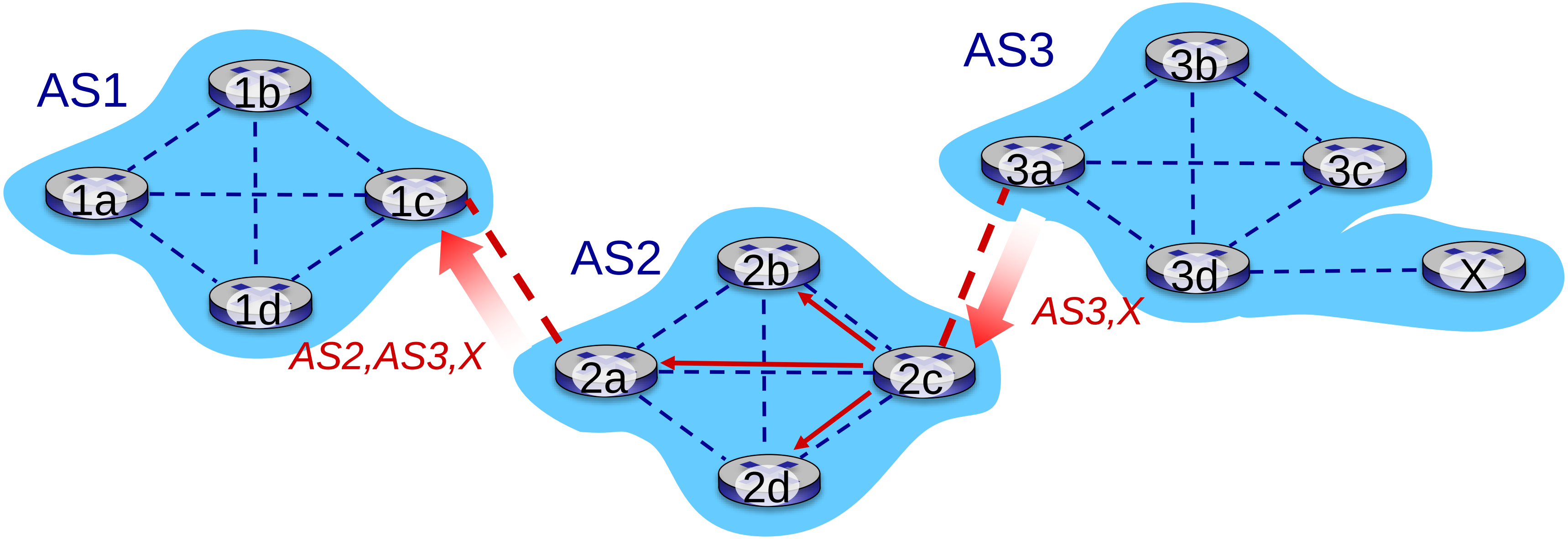
BGP basics

- BGP session: two BGP routers ("peers") exchange BGP messages over semi-permanent TCP connection:
  - advertising *paths* to different destination network prefixes (BGP is a "path vector" protocol)
- when AS3 gateway router 3a advertises path AS3,X to AS2 gateway router 2c:
  - AS3 *promises* to AS2 it will forward datagrams towards X



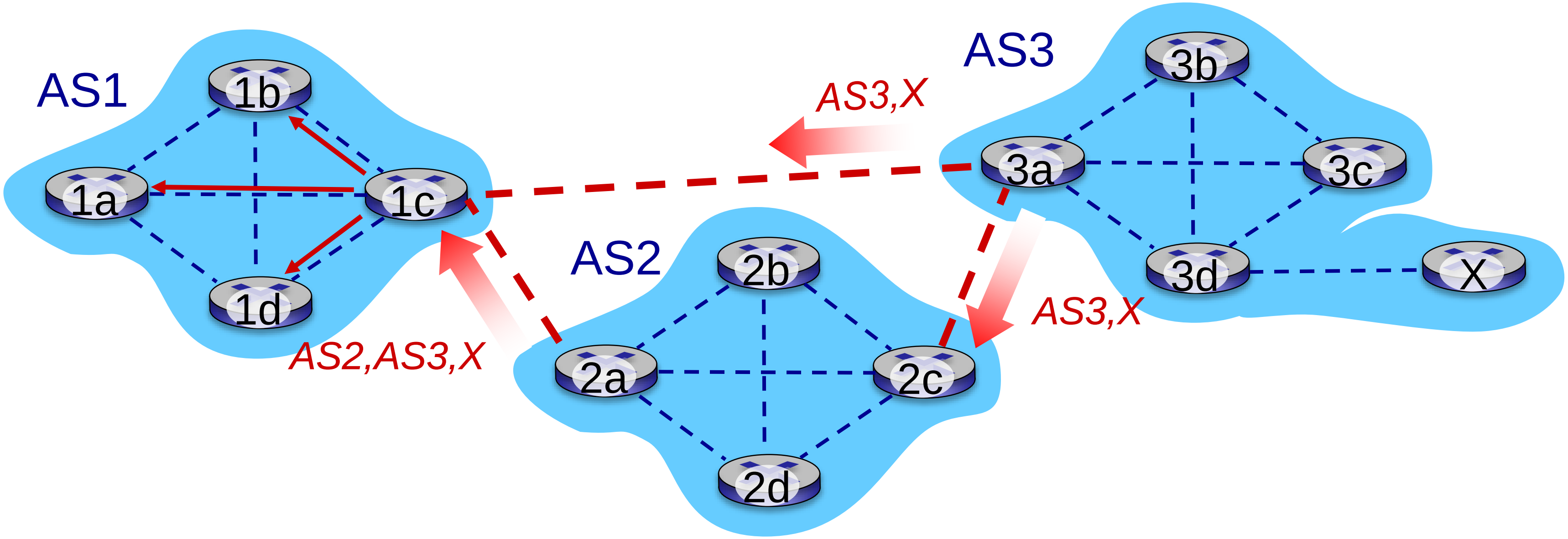
BGP advertisement: AS3, X

BGP path advertisement



- AS2 router 2c receives path advertisement AS3,X (via eBGP) from AS3 router 3a
- Based on AS2 policy, AS2 router 2c accepts path AS3,X, propagates (via iBGP) to all AS2 routers
- Based on AS2 policy, AS2 router 2a advertises (via eBGP) path AS2, AS3, X to AS1 router 1c

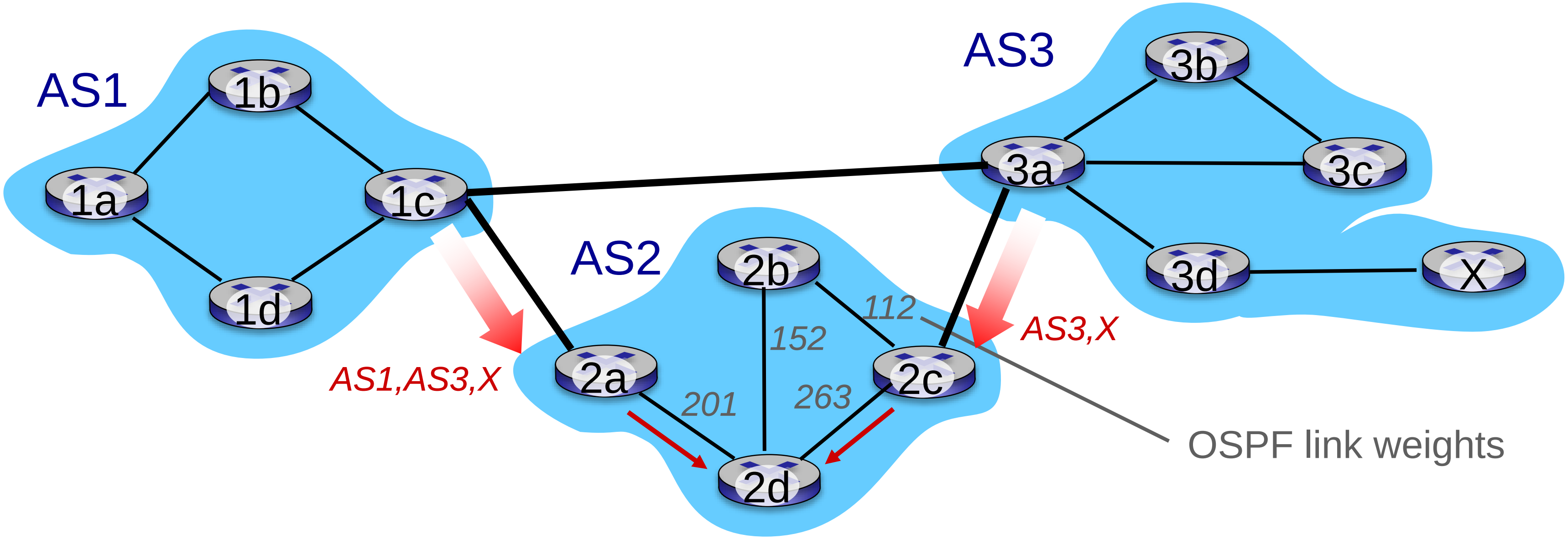
BGP path advertisement



gateway router may learn about multiple paths to destination:

- AS1 gateway router 1c learns path *AS2,AS3,X* from 2a
- AS1 gateway router 1c learns path *AS3,X* from 3a
- Based on policy, AS1 gateway router 1c chooses path *AS3,X, and advertises path within AS1 via iBGP*

Hot Potato Routing



- 2d learns (via iBGP) it can route to X via 2a or 2c
- *hot potato routing:* choose local gateway that has least intra-domain cost (e.g., 2d chooses 2a, even though more AS hops to *X*): don't worry about inter-domain cost!

BGP route selection

- router may learn about more than one route to destination AS, selects route based on:
    1. local preference value attribute (policy decision)
    2. shortest AS-PATH
    3. closest NEXT-HOP router (hot potato routing)