Designs for Scale

How to deal with large numbers (millions) of entities in a system?
- IP devices in the internet (0.5 billion)
- Users in P2P network (millions)

More generally:
- Are there advantages to large scale?
- “For every type of animal there is a most convenient size, and a large change in size *inevitably* carries with it a change of form.”
  True for networks?
Dealing with scale: Hierarchical routing

Scale: > 500 million destinations:
- Can’t store all dest’s in routing tables!
- Routing table exchange would swamp links!

Administrative autonomy
- internet = network of networks
- Each network admin may want to control routing in its own network
Hierarchical routing

- Aggregate routers into regions, "autonomous systems" (AS)
- Routers in same AS run same routing protocol
  - "Intra-AS" routing protocol
  - Routers in different AS can run different intra-AS routing protocol

Gateway routers

- Special routers in AS
- Run intra-AS routing protocol with all other routers in AS
- Also responsible for routing to destinations outside AS
  - Run *inter-AS routing* protocol with other gateway routers
Intra-AS and inter-AS routing

Gateways:
- Perform inter-AS routing amongst themselves
- Perform intra-AS routers with other routers in their AS

Inter-AS, intra-AS routing in gateway A.c
Intra-AS and inter-AS routing (2)

Intra-AS routing within AS A

Intra-AS routing within AS B

Inter-AS routing between A and B
Dealing with scale: Addressing

Old-fashioned “class-full” addressing:

<table>
<thead>
<tr>
<th>Class</th>
<th>Prefix</th>
<th>0 network</th>
<th>Host</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0 network</td>
<td>host</td>
<td>1.0.0.0 to 127.255.255.255</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>network</td>
<td>host</td>
<td>128.0.0.0 to 191.255.255.255</td>
</tr>
<tr>
<td>C</td>
<td>110</td>
<td>network</td>
<td>host</td>
<td>192.0.0.0 to 223.255.255.255</td>
</tr>
<tr>
<td>D</td>
<td>1110</td>
<td>multicast address</td>
<td></td>
<td>224.0.0.0 to 239.255.255.255</td>
</tr>
</tbody>
</table>

32 bits
IP addressing: CIDR

- Classful addressing:
  - Inefficient use of address space, address space exhaustion
  - E.g., class B net allocated enough addresses for 65K hosts, even if only 2K hosts in that network

- CIDR: Classless InterDomain Routing
  - Network portion of address of arbitrary length
  - Address format: `a.b.c.d/x`, where `x` is # bits in network portion of address

```
11001000  00010111  00010000  00000000
```

```
200.23.16.0/23
```
IP addresses: How to get one?

Q: How does network get network part of IP addr?

A: Gets allocated portion of its provider ISP’s address space

<table>
<thead>
<tr>
<th>ISP’s block</th>
<th>11001000 00010111 00010000 00000000</th>
<th>200.23.16.0/20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization 0</td>
<td>11001000 00010111 00010000 00000000</td>
<td>200.23.16.0/23</td>
</tr>
<tr>
<td>Organization 1</td>
<td>11001000 00010111 00010010 00000000</td>
<td>200.23.18.0/23</td>
</tr>
<tr>
<td>Organization 2</td>
<td>11001000 00010111 00010100 00000000</td>
<td>200.23.20.0/23</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Organization 7</td>
<td>11001000 00010111 00011110 00000000</td>
<td>200.23.30.0/23</td>
</tr>
</tbody>
</table>
Hierarchical addressing allows efficient advertisement of routing information:

- Organization 0: 200.23.16.0/23
- Organization 1: 200.23.18.0/23
- Organization 2: 200.23.20.0/23
- Organization 7: 200.23.30.0/23

Fly-By-Night-ISP

"Send me anything with addresses beginning 200.23.16.0/20"

ISP-Rs-Us

"Send me anything with addresses beginning 199.31.0.0/16"

Internet
Hierarchical addr: Route aggregation

ISPs-R-Us has a more specific route to Organization 1

Organization 0
200.23.16.0/23

Organization 2
200.23.20.0/23

Organization 7
200.23.30.0/23

Organization 1
200.23.18.0/23

Fly-By-Night-ISP

"Send me anything with addresses beginning 200.23.16.0/20"

ISPs-R-Us

"Send me anything with addresses beginning 199.31.0.0/16 or 200.23.18.0/23"

Internet
Hierarchical addr: More specific routes

- Multiple advertised routes could hold destination
  - 200.23.16.0/20
  - 200.23.18.0/23
  both hold 200.23.18.7

- Always route to more specific destination (longest prefix match)
Dealing with scale

**Question:** what are the advantages of large scale?

- Take advantage of having to do similar things for others (caching)
- Fault tolerance:
  - Large number of servers
  - We have redundancy; multiple routes between sites
- Metcalfe’s law:
  - “Value” of a network is proportional to square of number of things connected (bigger is better)
- Law of large numbers
  - Allocation of resources based on average usage rather than peak
- Amortizing upgrade maintenance over a large population
  - Popular network and services likely to be upgraded/improved
- Denial of service:
  - Size/replication makes it harder to attack
  - More generally, a system with replicated components is more survivable
Dealing with Scale

*Discussion:* “For every type of animal there is a most convenient size, and a large change in size inevitably carries with it a change of form.”

*Question:* True for networks? Why? How so? Examples?

- Ethernet doesn’t scale up: Geographical distance, speed of light delays degrade performance of random access protocols. (geographic scaling). Maybe scale with # users in geographically narrow net if bandwidth scales with users

- As number of communicants scales, need to change/improve manner in which to access communication channel
  - Example: small number of students, versus 500-class lecture. Keeping bandwidth fixed as # users scales

- Email versus HTTP
  - Push systems work ok when small number of sender (email)
  - Pull is better with large number of senders (http)
Dealing with Scale

**Discussion:** “For every type of animal there is a most convenient size, and a large change in size inevitably carries with it a change of form.”

**Question:** True for networks? Why? How so? Examples?

- **Routing:**
  - Large number of users and optimal routes => requires lots of info to compute routes, etc...
  - Doesn’t scale

- **Certain services become necessary when you get big**
  - Name storage/translation: DNS, phone books

- **A single centralized site eventually breaks**
  - Need replication or other form of distribution

- **As network gets bigger flooding breaks**
  - Use limited flooding, caching (Gnutella)

- **Switched vs. routed networks**
  - Change from layer 2 switched networks to layer 3 routed networks as # users gets bigger