The Edge, The Core and a bunch of Layers

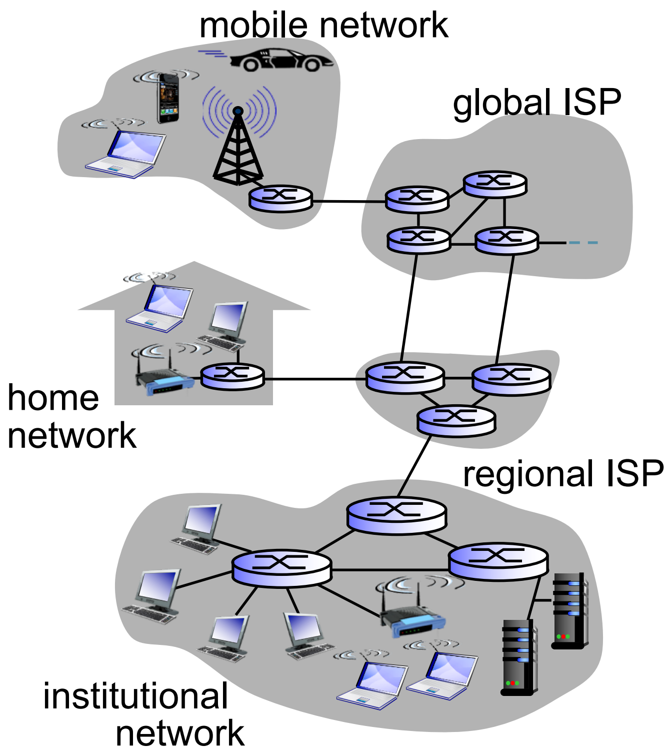
To do …

- The network edge
- The network core
- Protocol layers and service models

Northwestern

The Internet – In practice

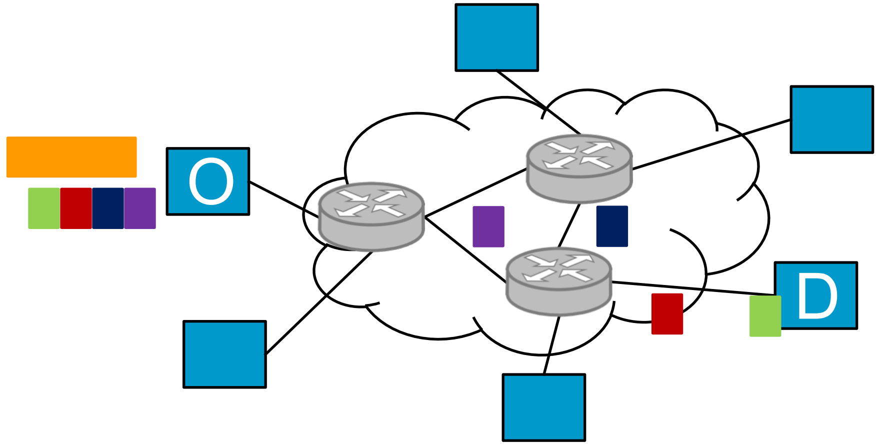
- To go deeper, let's look at the pieces
- Network edge
  - Hosts or end systems – Clients and servers (typically in data centers)
- Access networks, physical media
  - Wired and wireless communication links
  - Connecting end systems to edge (first) routers
- Network core
  - Mesh of routers and links connecting various Internet parts
  - Like hosts with multiple links, only there to relay others' packets



First, a bit on performance

- Most computer networks are packet-switched networks
  - Messages are sent as sets of self-contained packets, with an address
  - Each routed independently to its destination (may arrive out of order)

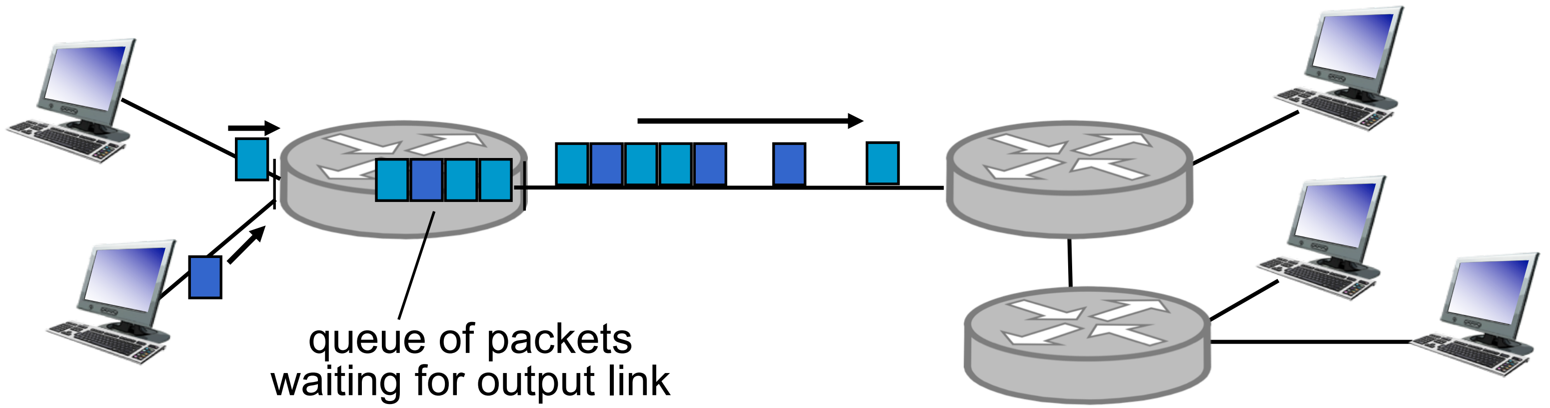
- Ideally, a network that can move as many packets as we want, instantaneously, without loss …



- *Real* networks constrain throughput between end systems, introduce delay and can even loose packets

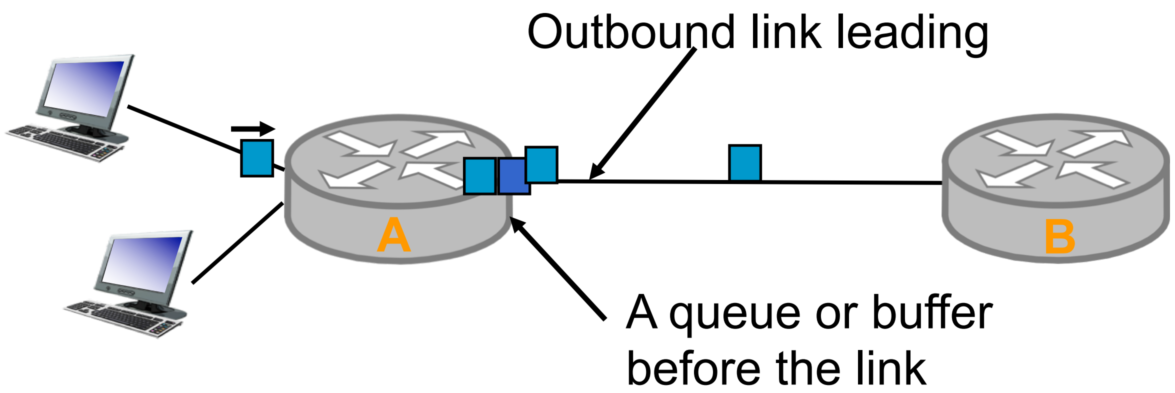
A quick look at performance

- Most routers use store-and-forward – The entire packet must arrive at router before it is forwarded on an outgoing link
- Each outgoing link has an output buffer or queue
  - Waiting in the queue introduces delays
  - Queues are finite-sized, if a queue is full, new packets are dropped – *packet loss*



Node delays

- As a packet travels from one node (host or router) to the next, it suffers from multiple delays at each node

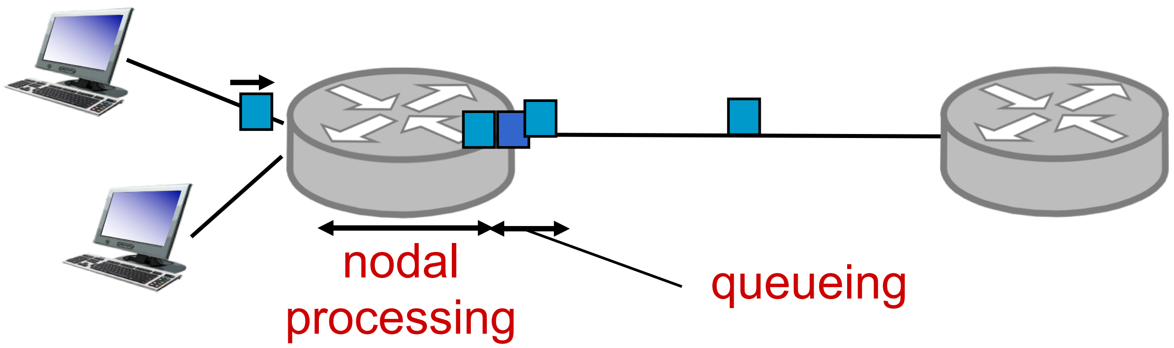


- When a packet arrives at router *A*, the router examines the header to determine outgoing link
- If there's nothing on the link or waiting in queue, transmit it

Sources of packet delay

d_{proc}: nodal processing delay

- check bit errors, choose output link
- typically < μsec



If there's something in the queue, the packet will have to wait …

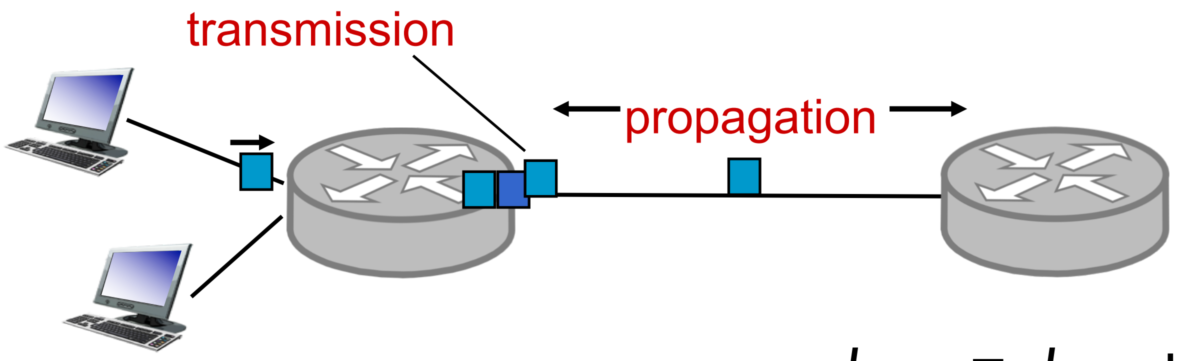
d_{queue}: queueing delay

- time waiting at output link
- depends on congestion level

Sources of packet delay

Time to push out pkt, like cars off a toll booth → transmission

d_{trans}: transmission delay
= pkt size (bits) / link bw (bps)



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

Time to propagate from the beginning of link to B

d_{prop}: propagation delay
≅ length of link / SoL (~2×10^8 m/s)

Queueing delay and Paquet loss

- d_{queue} – The most interesting part
  - Its importance depends on rate of pkt arrival, transmission rate and nature of the traffic (i.e., steady or in bursts)
- To get a sense – Traffic intensity, the ratio $L*a/R$
  - ***L*** bits/pkt | ***a*** rate of pkts arrival at queue | ***R*** transmission rate
  - If $L*a/R > 1$ arrive faster than they leave → *queueing*
  - If $L*a/R \lesssim 1$, depends on how bursty is the traffic
- As traffic intensity approaches 1
  - Average queueing delay grows fast
  - If queue can't hold the packet, drops it → *higher intensity, higher loss*

Latency and Bandwidth

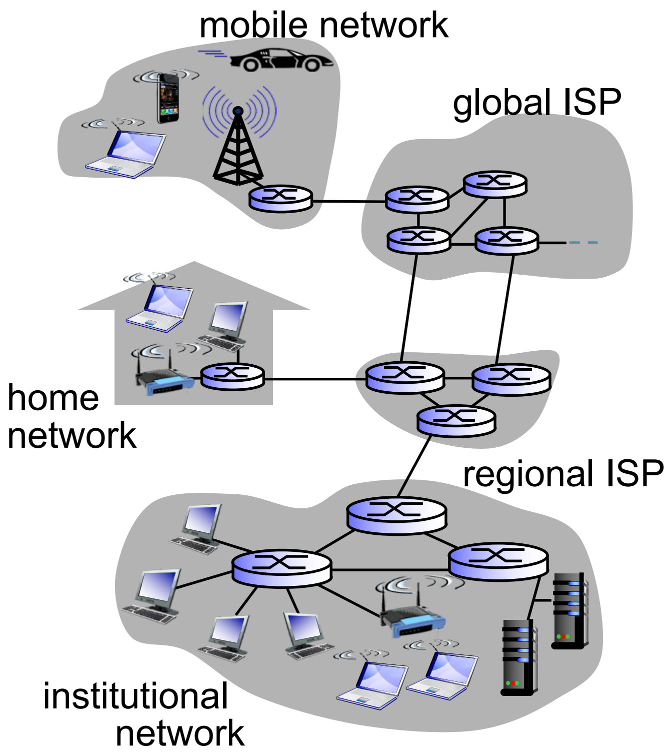
- Latency (aka delay) – how long it takes a message to travel from one end to the other
  - With *N - 1* routers between both ends and not queueing
  - E.g., 24ms transcontinental
  - Sometime you want the round-trip-time (RTT)

$$d_{end\text{-}end} = N(d_{proc} + d_{trans} + d_{prop})$$

- Bandwidth (aka throughput) – number of bits that can be transmitted over a network over a certain period of time
  - 10 Mbps ~ 10 million bits per second
  - Throughput more precisely refers to 'measured performance'
  - End-to-end throughput – the throughput of the bottleneck link

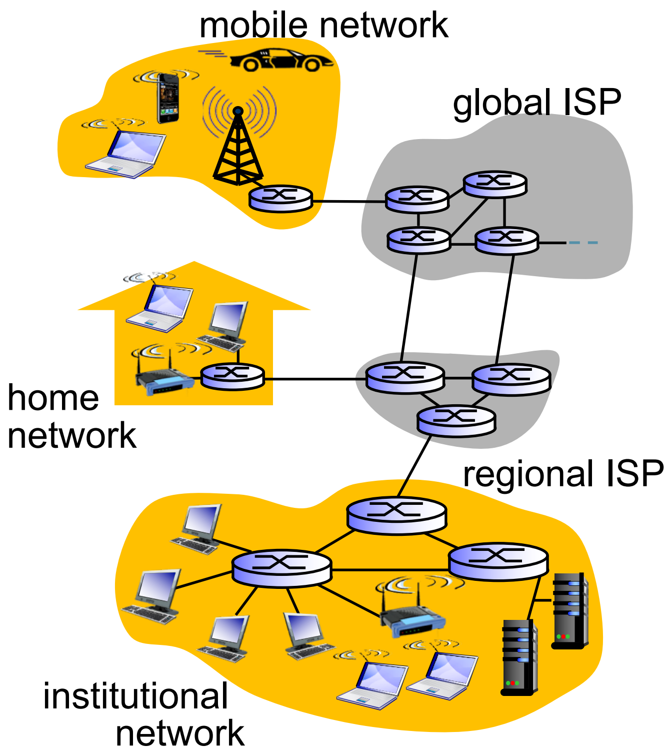
The Internet – In practice

- To go deeper, let's look at the pieces
- Network edge
  - Hosts or end systems – Clients and servers (typically in data centers)
- Access networks, physical media
  - Wired and wireless communication links
  - Connecting end systems to edge (first) routers
- Network core
  - Mesh of routers and links connecting various Internet parts
  - Like hosts with multiple links, only there to relay others' packets



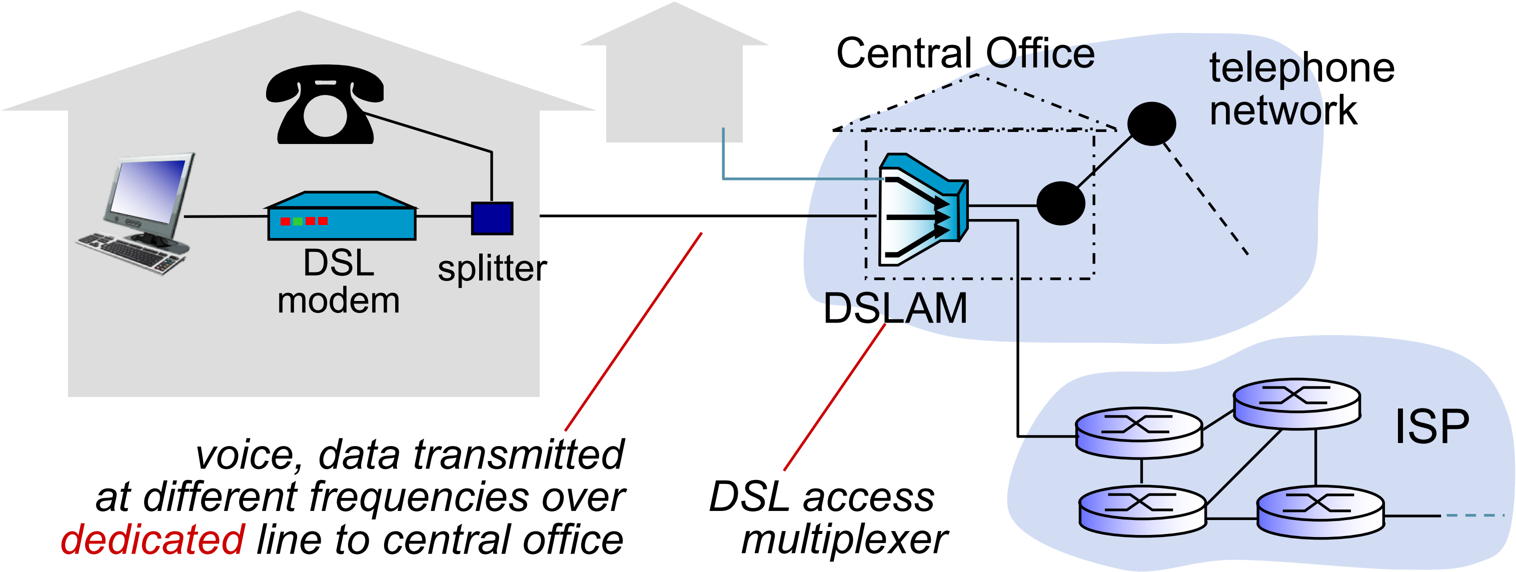
Access networks and local-area networks

- Different technologies are used to connect devices at the edge
  - WiFi, Ethernet, Cellular
  - Speed varies depending on the link technology
  - Links may be shared or dedicated
- Some access networks assign public IP addresses to hosts, and others use private IP addresses and network address translation (NAT)
- Most common type of broadband residential access – DSL and Cable



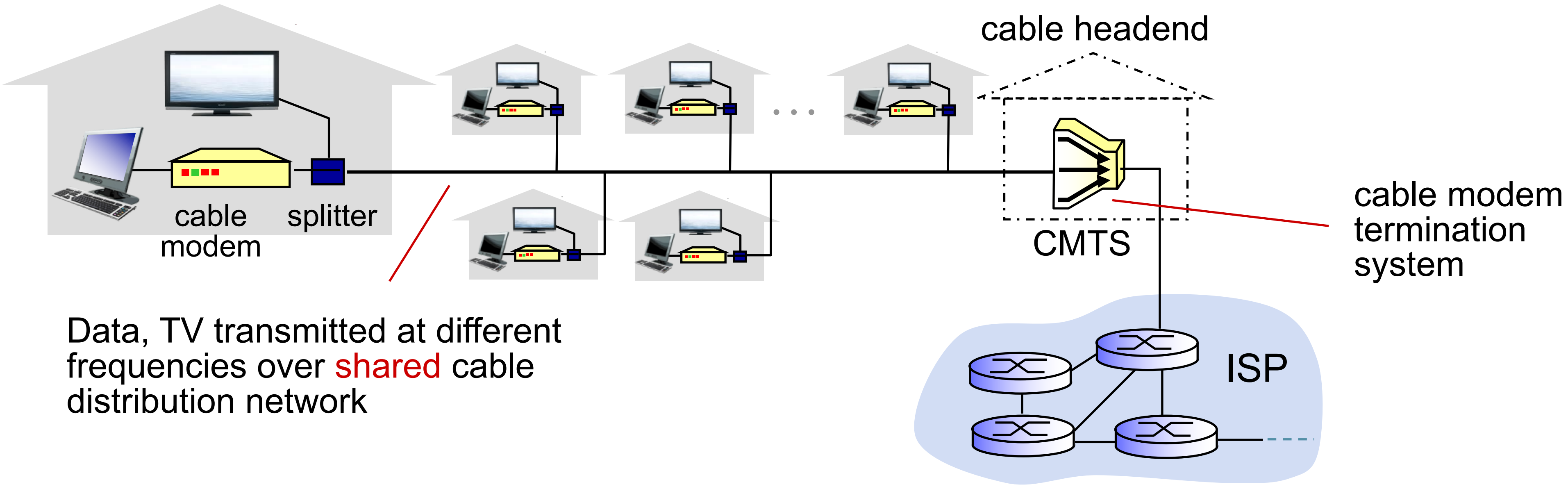
Digital Subscriber Line (DSL)

- Typically from your phone provider
  - One line for 2-way phone service, downstream/upstream data channels, each at different frequencies
- At home – Splitter to separate phone/data at home + modem
- On the telco side, the CO, DSLAM for 100-1000 of households
- Different standards, like 55/15 MBps in 2006 (asymmetric)
  - Also limited by noise in "last mile", so, distance to CO



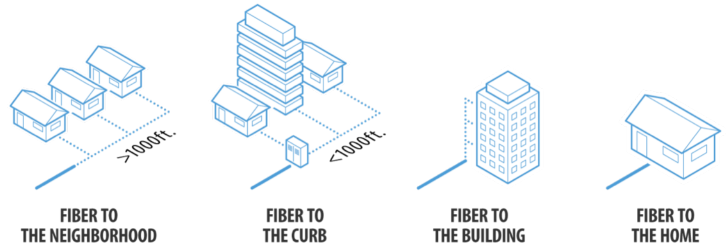
Cable Networks

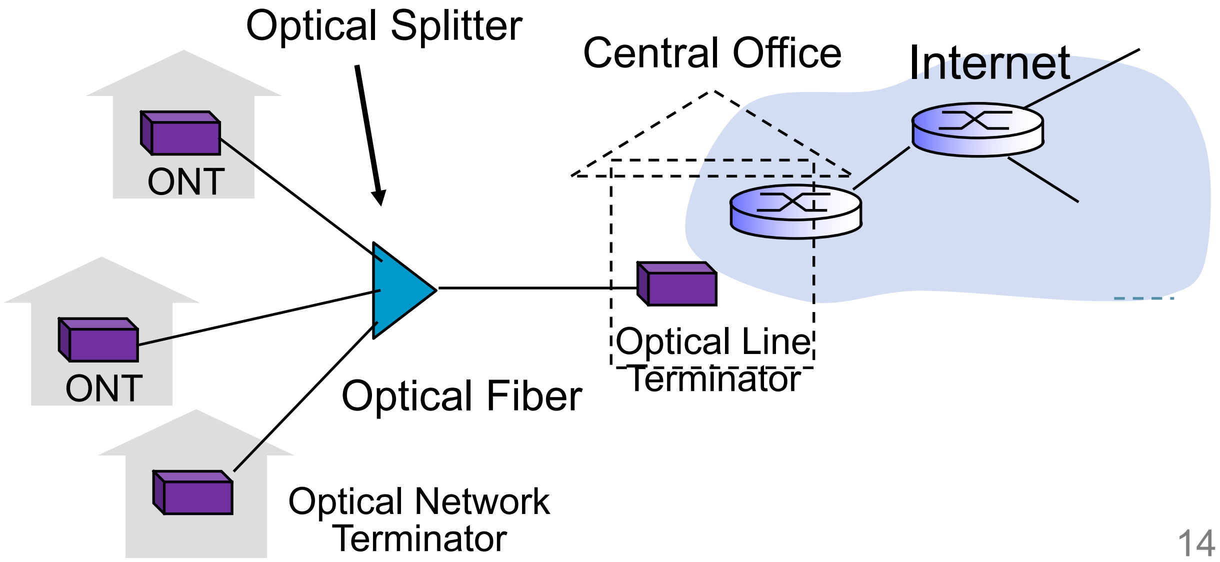
- Leveraging cable TV service
  - Entire neighborhood shares the same cable/signal
- Fiber optic connects to neighborhood level junctions (supporting 500-5000 homes); coaxial from there
- DOCSIS (Data-Over-Cable Service) 2.0 42.8/30.7Mbps, 100Mbps national average



Fiber-To-The-Home or Premise (FTTH or FTTP)

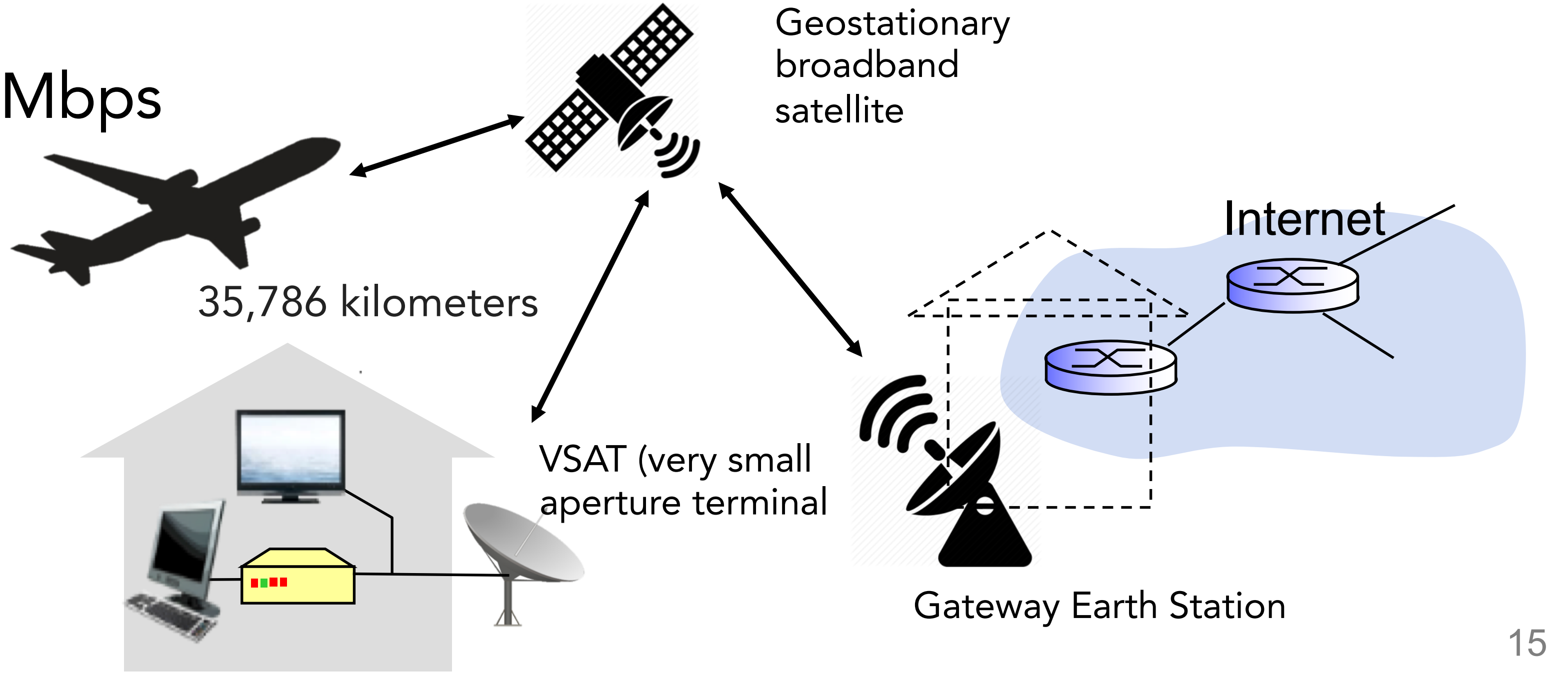
- Optical fiber path from CO to the home
  - UAE, South Korea, Lithuania, …. penetration rates >40%
  - ~25% in the US
- A number of options
  - Direct – fiber all the way from CO to the home
  - Passive Optical Networks (PON) – One fiber to a splitter (Verizon's FiOS)
- 1Gbps in Latvia in 2013 …





Satellite broadband

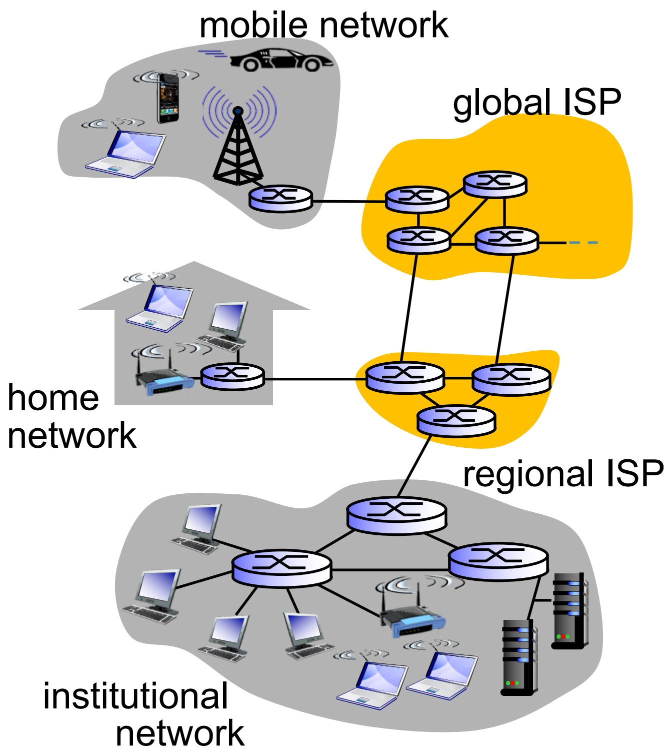
- Rural areas, planes, boat without DSL, cable or FTTH … a satellite link
- A geostationary satellite ~ 36,000 km
  - In perfect condition, 550ms round trip to gateway
- Affected by rain/snow (rain fade) and reflection
  - *Up to* 25/5Mbps
  - On a plane, more like 2Mbps



TAKE 5

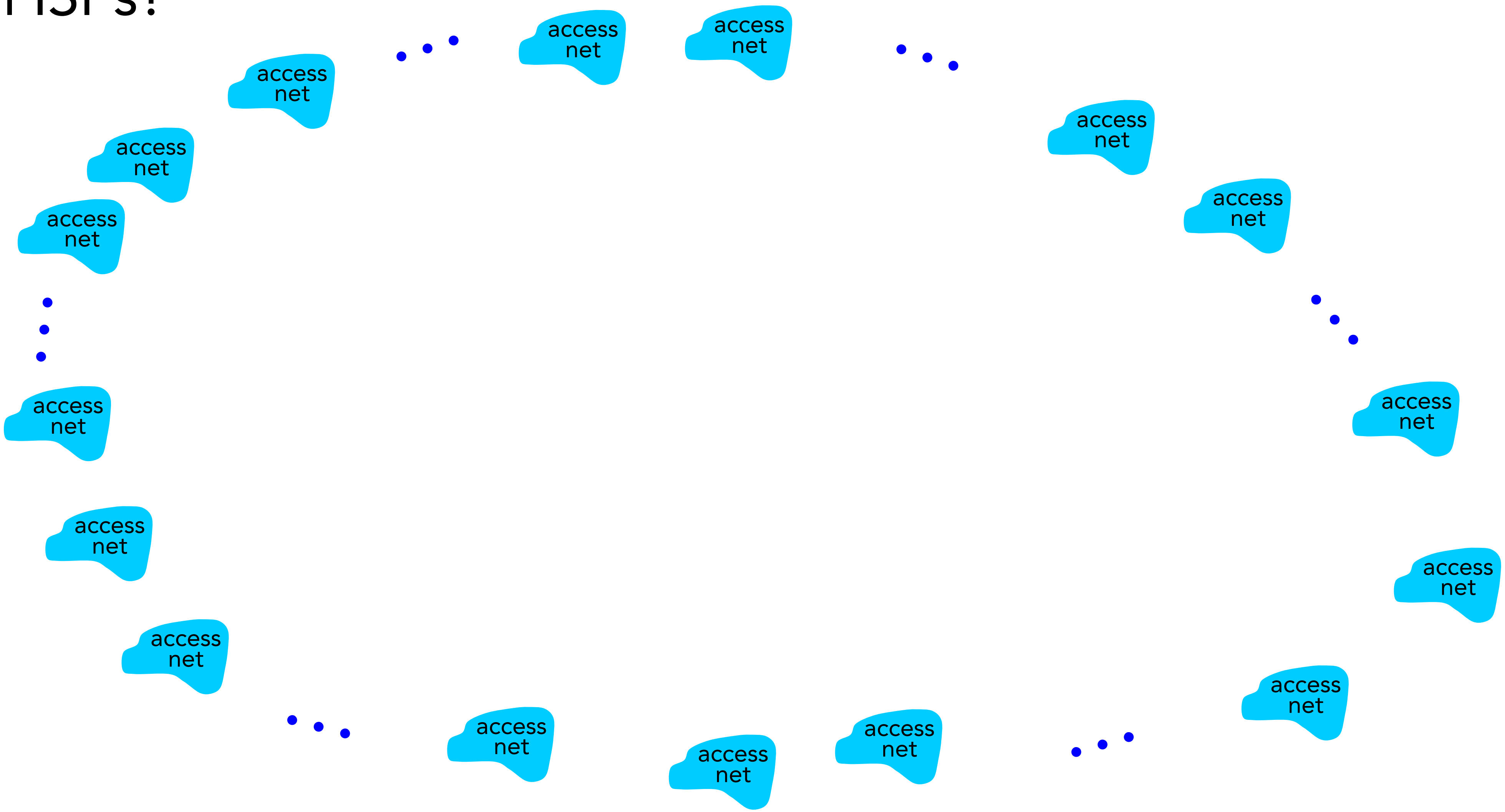
The network core

- End systems connect to the Internet via an access ISP
- But the access ISPs themselves have to be interconnected
- What does that network of networks look like?
  - *Let's build it piece by piece*



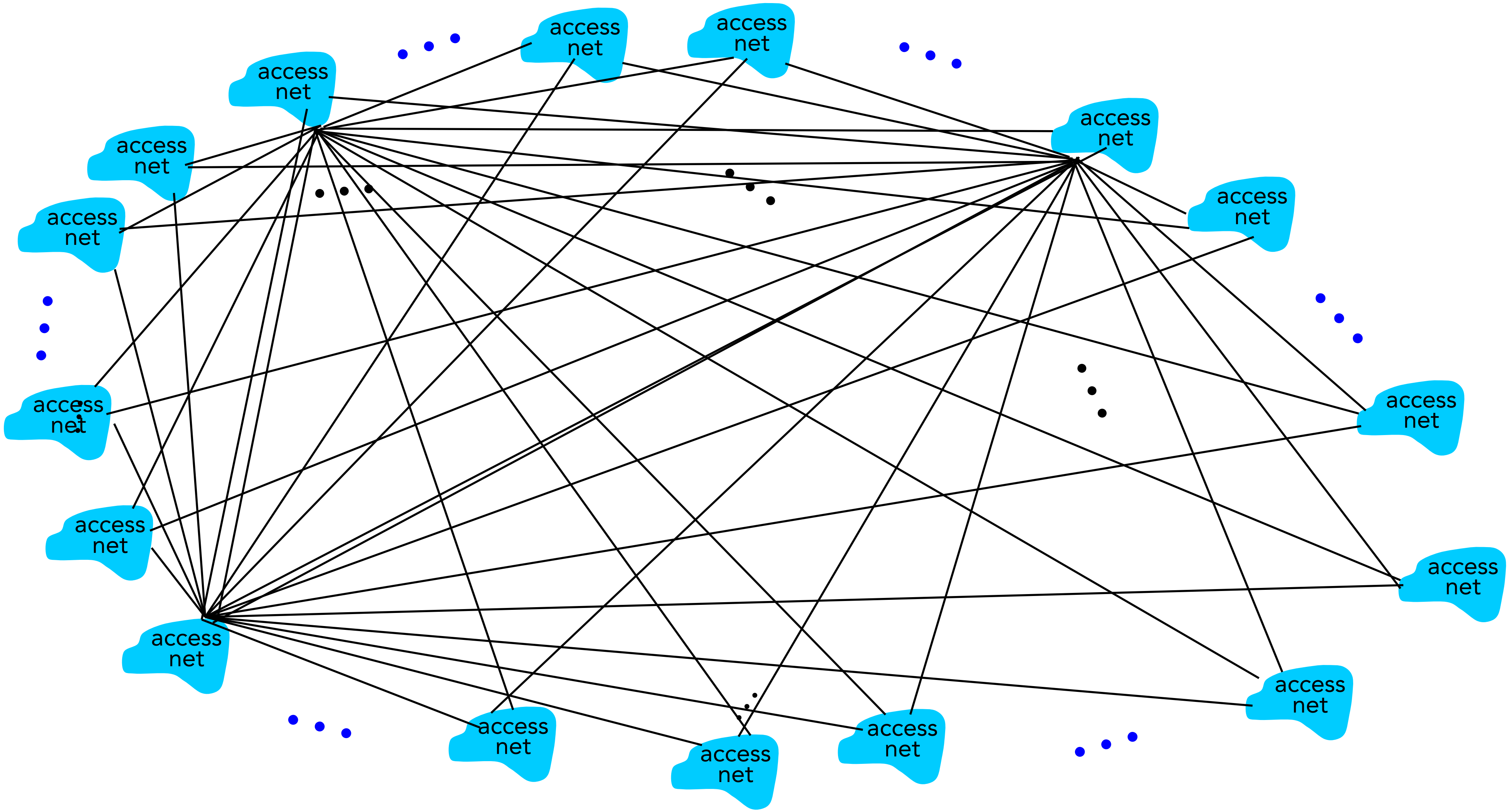
The Internet is a network of networks

- Hundredths of access ISPs – How should we connect all these local ISPs?



Idea 1: Connect them pairwise?

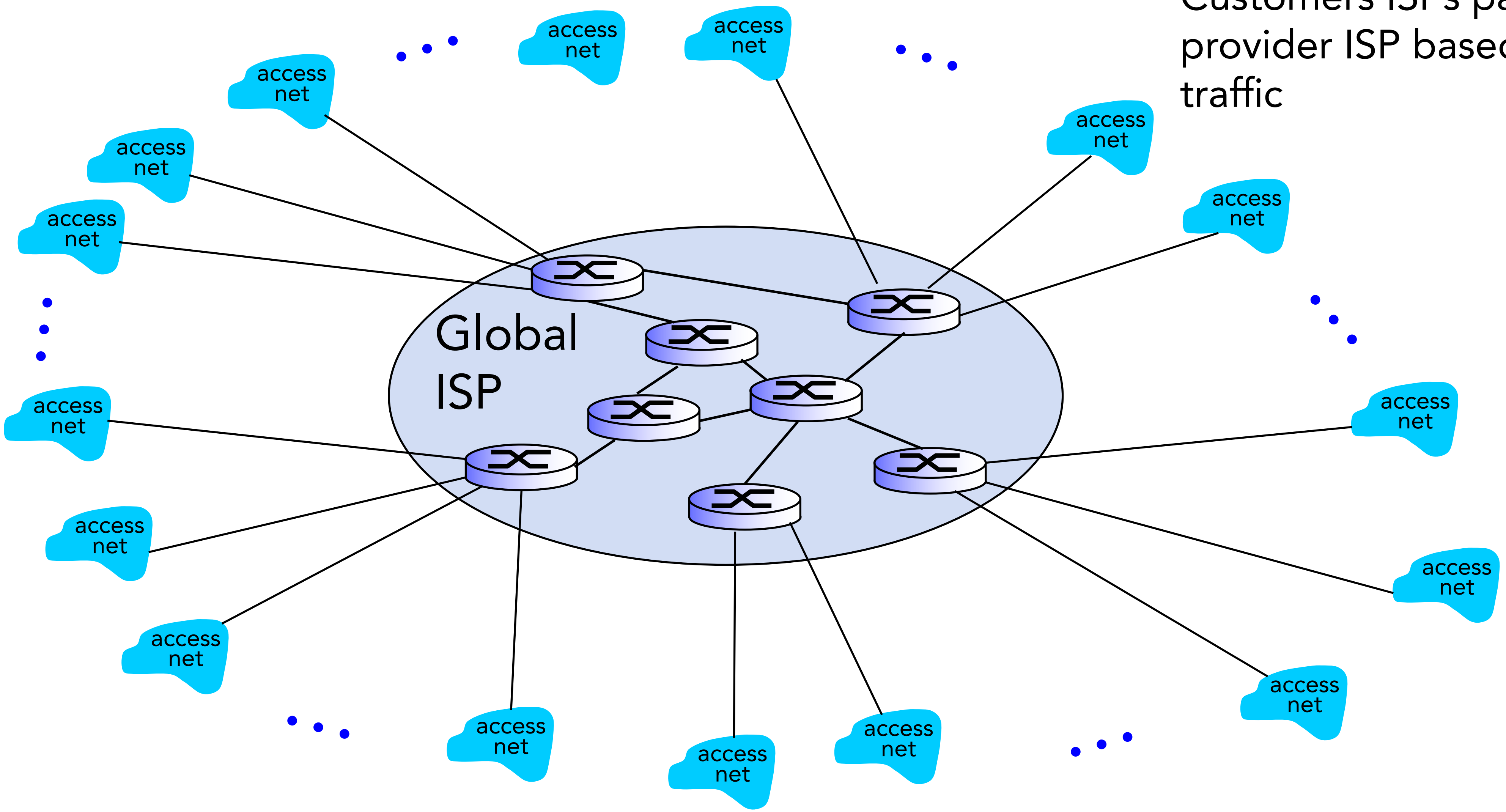
- N^2, very long, connections



Idea 2: A single transit ISP

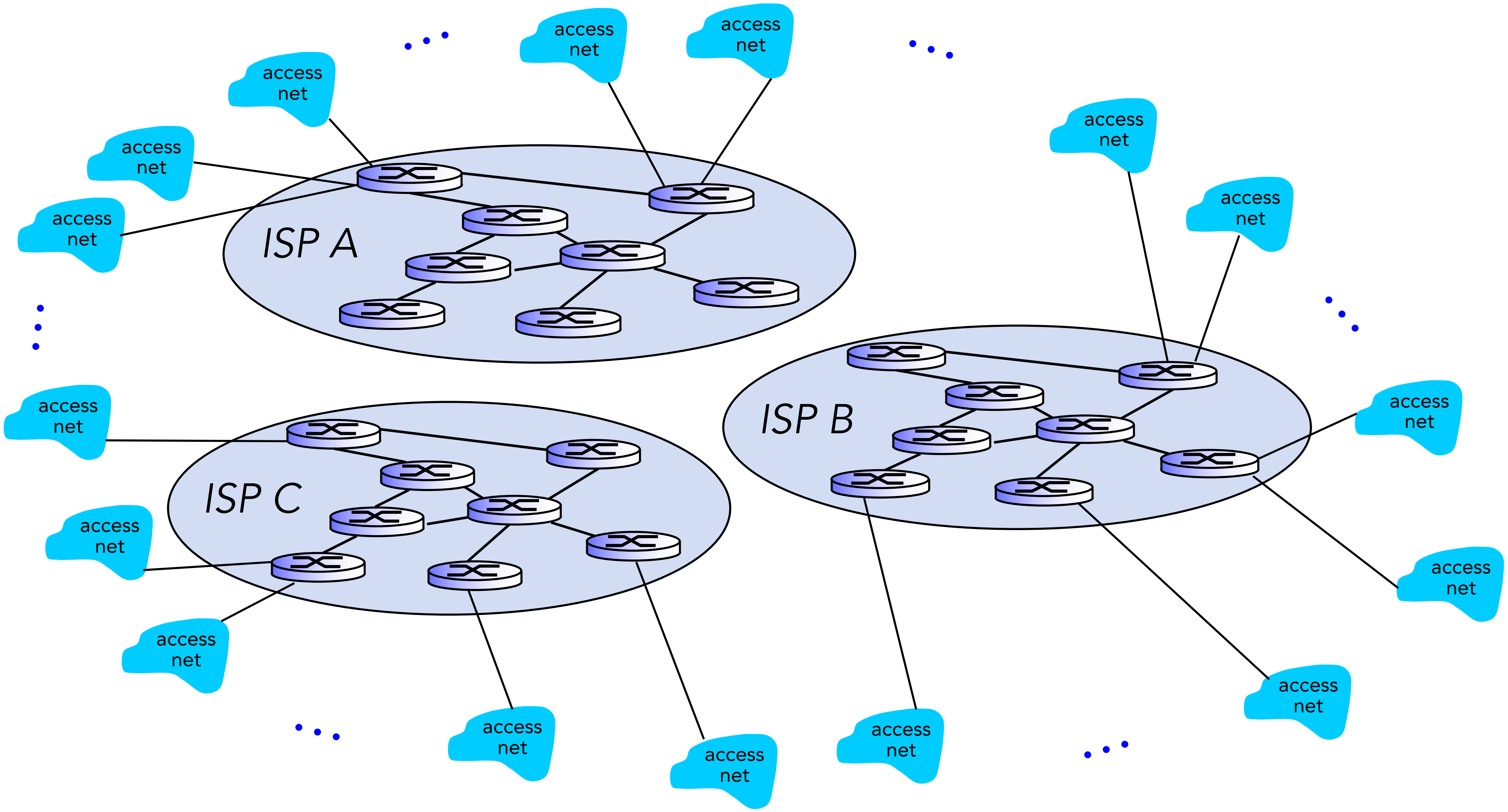
- An ISP for ISPs, with a few long and fast connections

Customers ISPs pay the provider ISP based on traffic



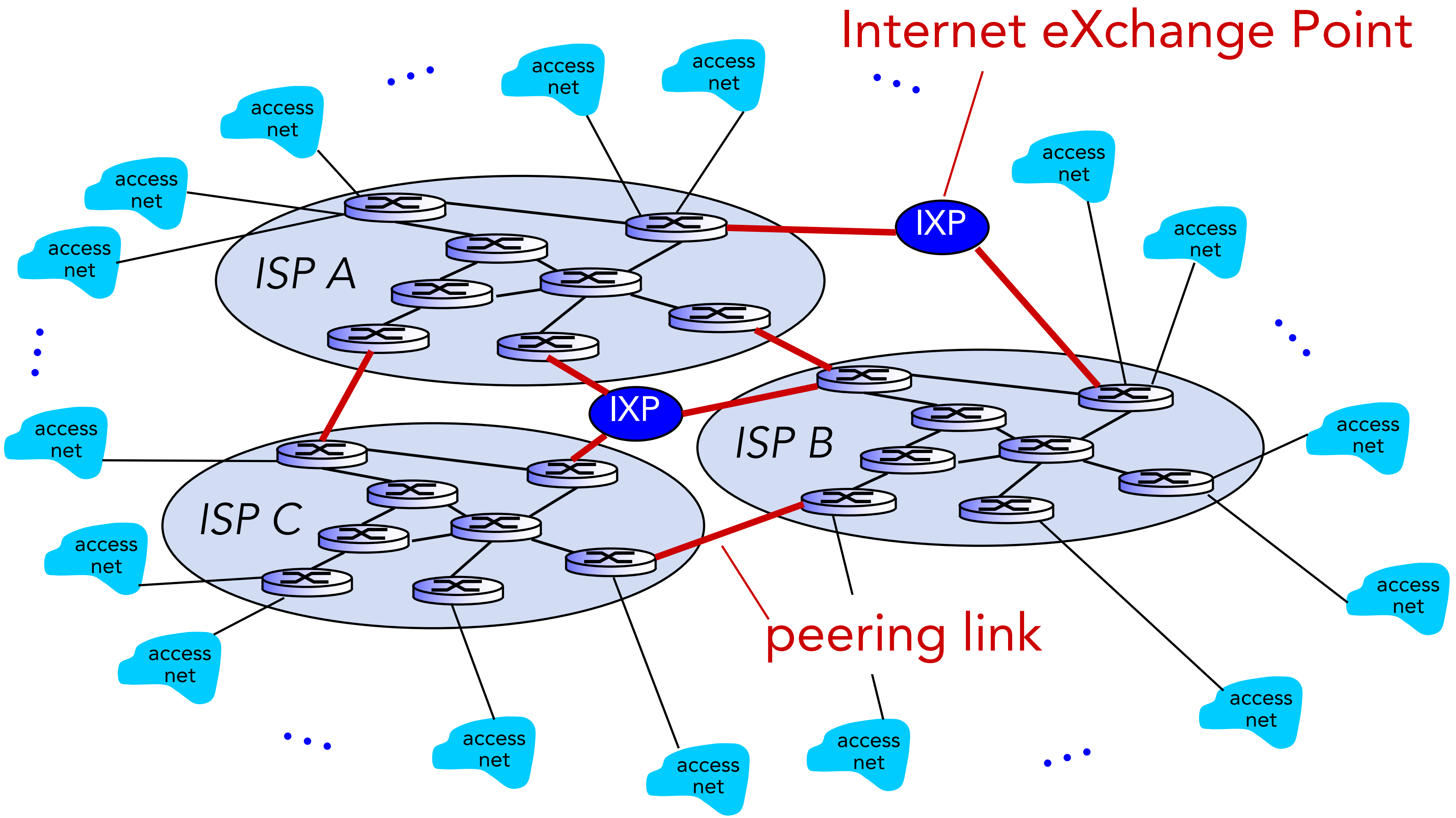
Idea 3: Multiple Global ISPs

- Other global ISPs to compete; these global ISPs need to connect as well …



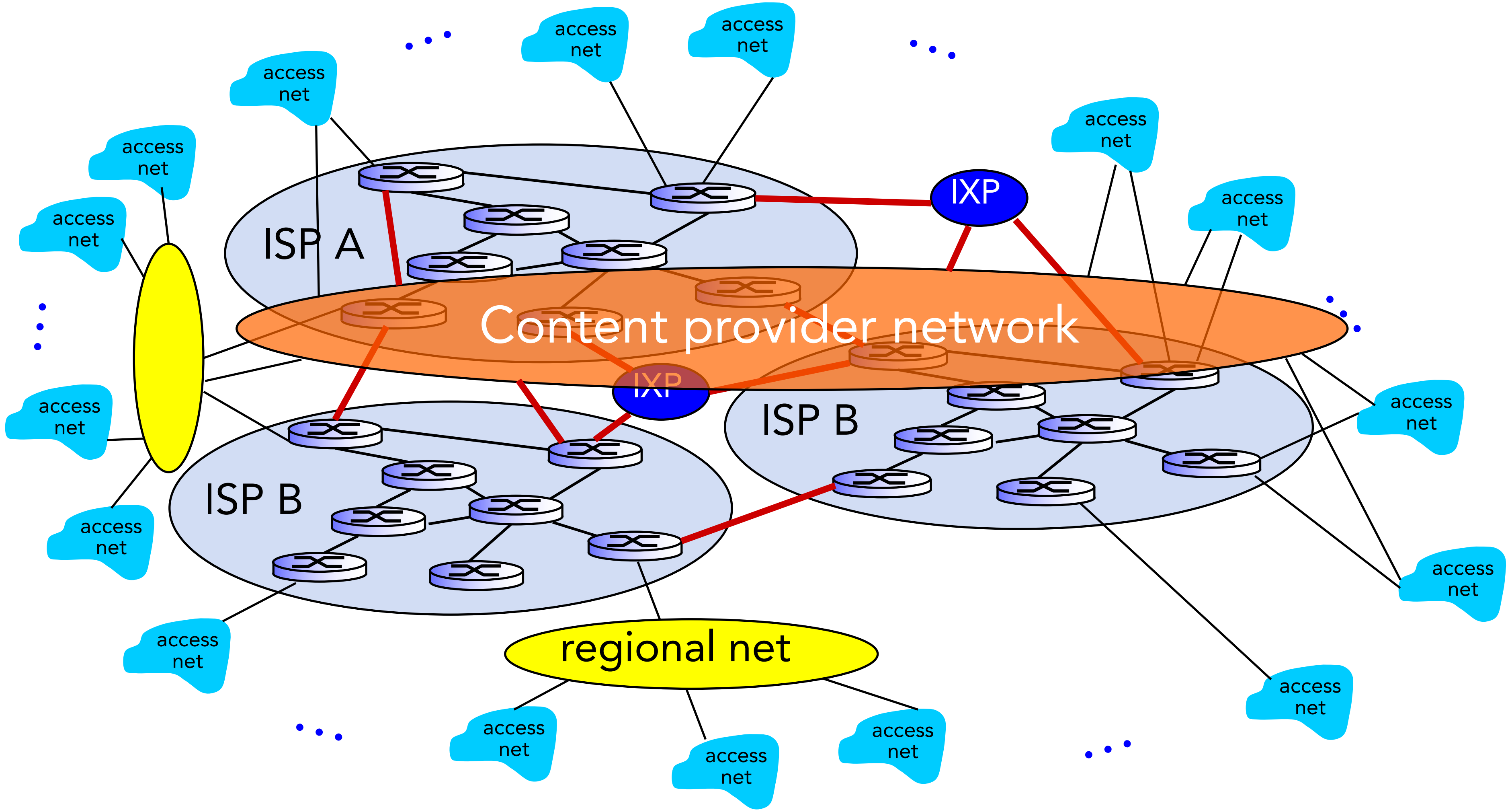
Global ISPs must be connected

- All connected at Points of Presence and IXPs, … in customer-provider or peer relationships



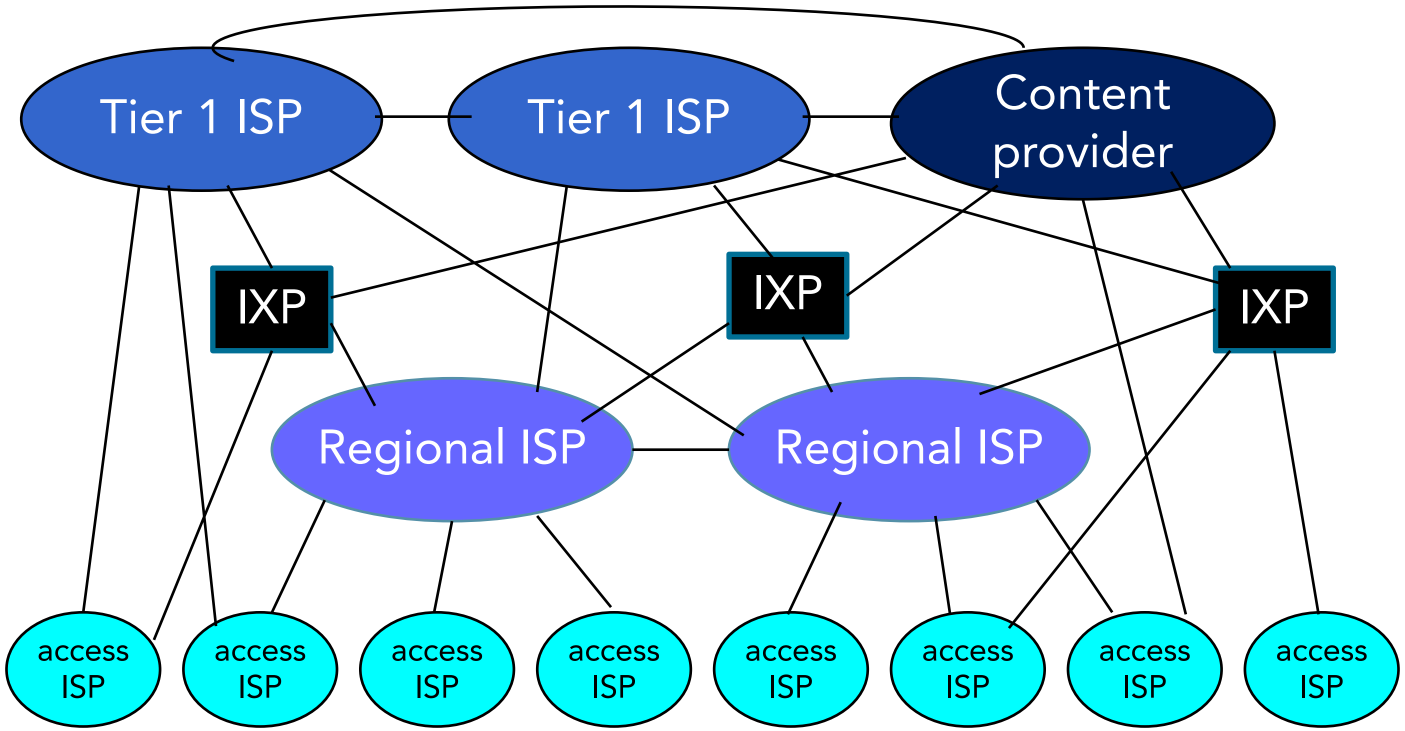
Final version: add regional ISPs and Content Providers

- Hierarchies of ISPs – regional ISPs, tier-1 ISPs (~ our global)
- Big content providers build their own global nets, much like ISPs



Internet's structure

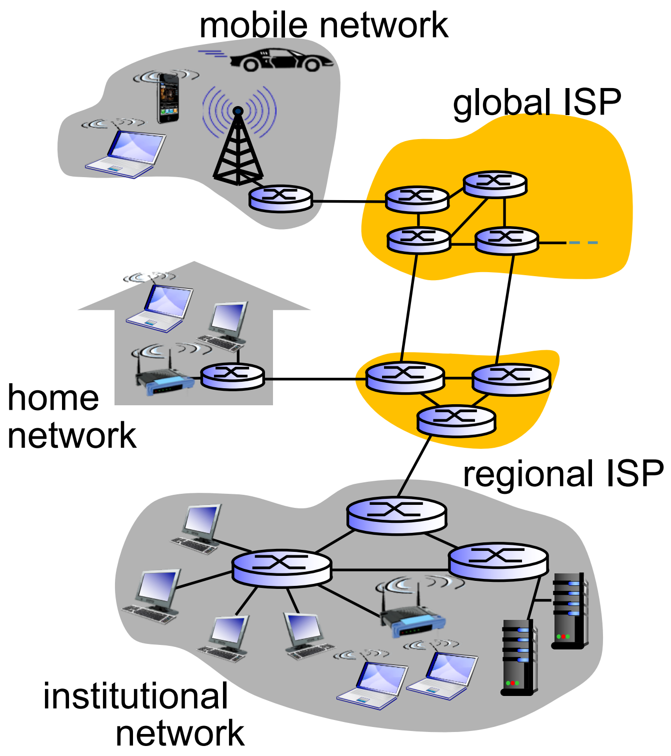
- At the center, a few well-connected, large networks
  - Tier-1 commercial ISPs (e.g., Century Link, AT&T, China Telecom), national and international coverage
  - Content provider networks (e.g, Google, FB): private network that connects it data centers to Internet, often bypassing tier-1, regional ISPs



Routing packets through this ...

Routers' main responsibilities

- Distributed routing algorithms to determine which address ranges are most quickly reachable on each of its outbound links
- Packet forwarding, to direct packets according to the decisions made above



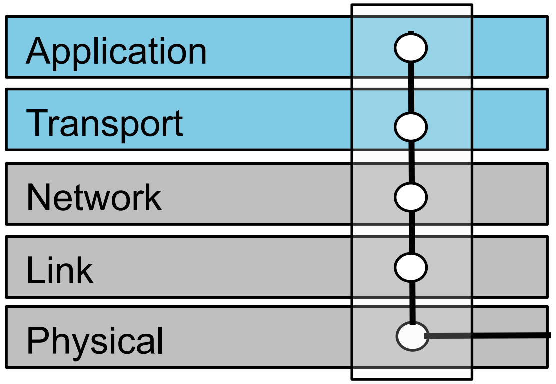
Protocol Layers and Service Models

- For communication, parties must agree on several things
  - Is this the last bit of a msg? How long are integers? …
- To give structure to the design of network protocols – layers
  - Each protocol belong to one of the layers
  - Each layer provides services to the layer above – the service model

- Layering pros and cons
  - A structured way to consider systems components
  - Modularity make it easier to update components
  - Layers may duplicate lower-layer functionality
  - A layer may need info from other layers, violating goal of separation

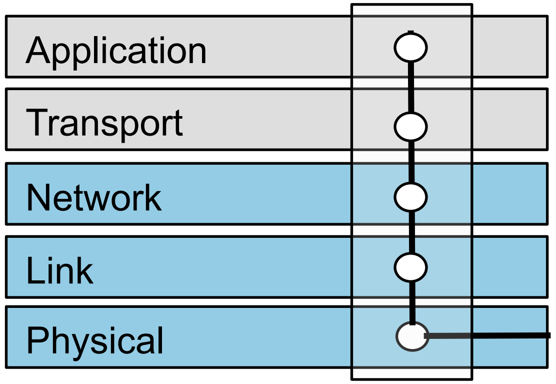
Internet layers and protocols

- Application layer – Where network applications and their protocols reside, e.g. HTTP, DNS, Bittorrent, …
  - End systems running the protocol exchange messages
- Transport layer provides inter-process communication
  - Transport address = net add + port
  - Two key protocols
    - TCP – connection oriented, reliable stream communication
    - UDP – connectionless, unreliable datagram communication



Internet layers and protocols

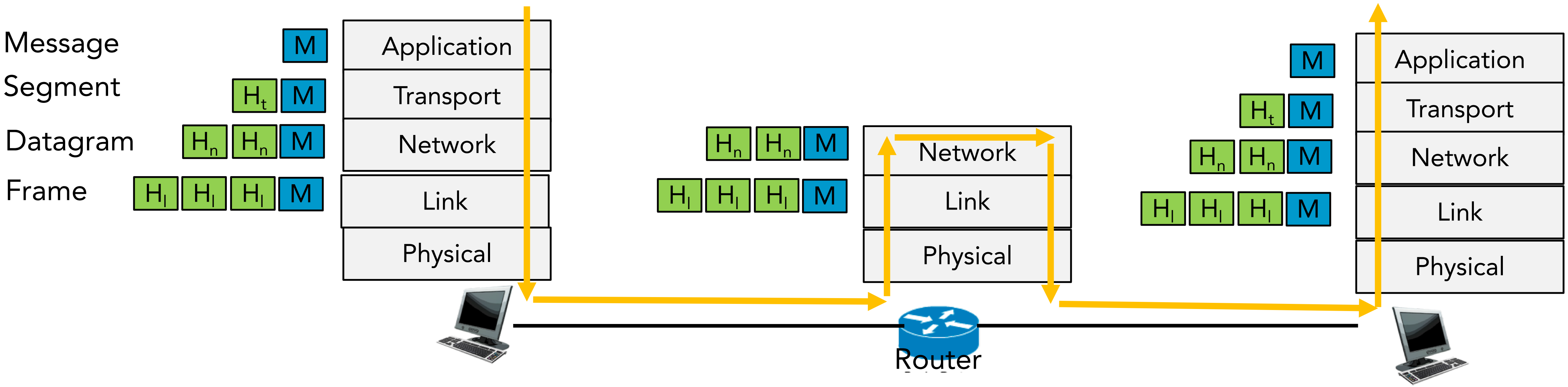
- Network layer provides communication host to host, over sequences of links
  - Moving datagrams between hosts
  - Provided by routers doing packet forwarding
  - Key protocol: IP
  - Other protocols for routing



- Link layer moves frames between locations
  - E.g., Ethernet, WiFi, DOCSIS …
- Physical layer moves bits within the frame between nodes
  - Protocols depend on the medium, Ethernet has many physical-layer protocols for twisted-pair copper wire, coaxial cable, …

Encapsulation

- Each protocol needs to transmit control data to do its job
- Up and down the stack
  - As a message is pass down, each layer adds a header (trailer)
  - On the receiver side the message is push upward with each layer stripping off and examining their own headers
  - Router and switches do not implement the whole stack



Summary

- A lot of content from a bird's eye view
  - From the edge to the core, Internet structure, protocols and layer, …
- Rest of the term, a more detailed view
- Starting with the application layer