6.1800 Spring 2025

Lecture #10: Routing at scale, and with policy

Katrina's favorite protocol to teach

Katrina LaCurts | lacurts@mit.edu | 6.1800 2025

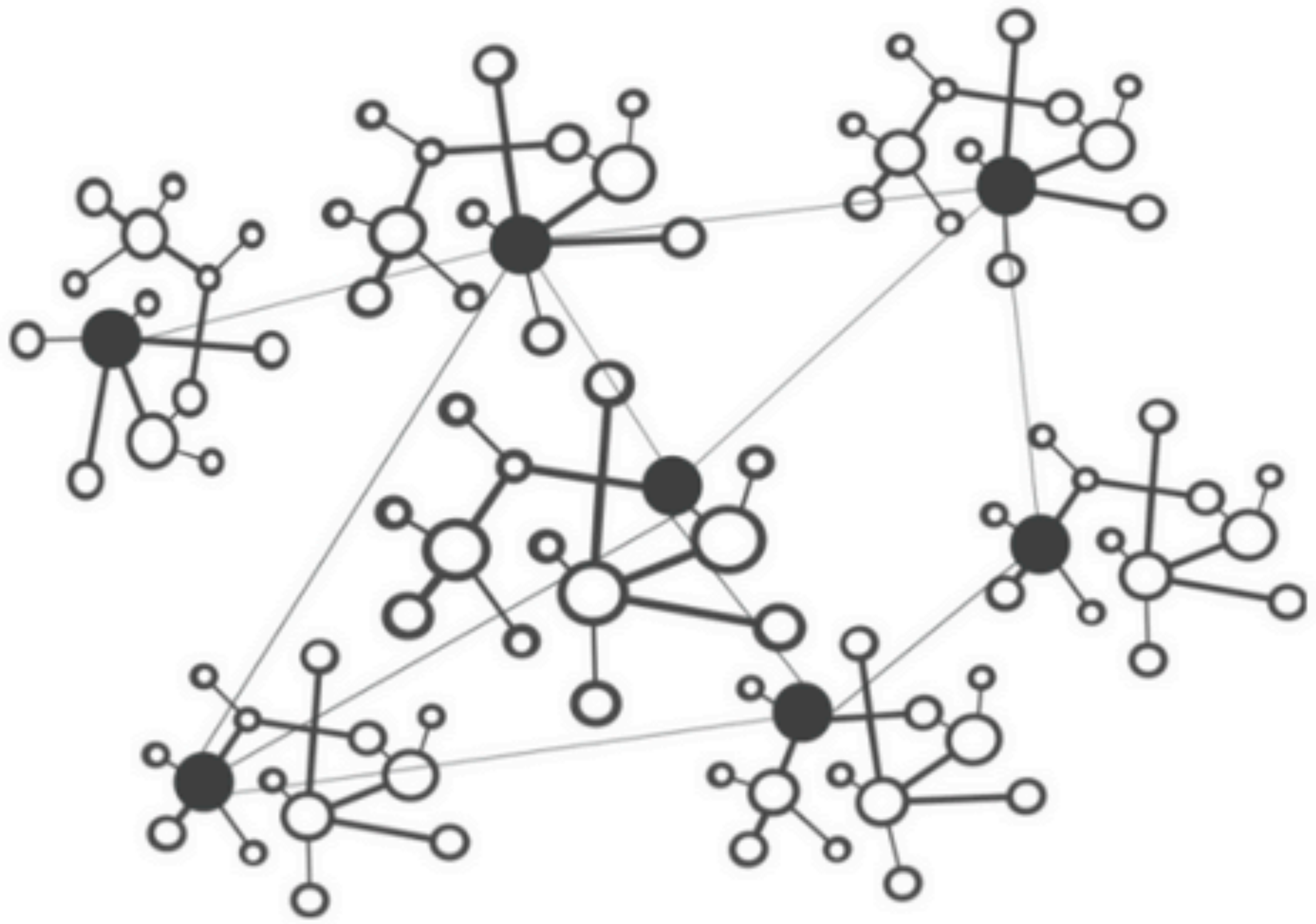
6.1800 in the news

Understanding How Facebook Disappeared from the Internet

10/04/2021

Celso Martinho Tom Strickx

This post is also available in 简体中文, 繁體中文, 日本語, 한국어, Deutsch, Français, Español, Português, Русский, and Italiano.

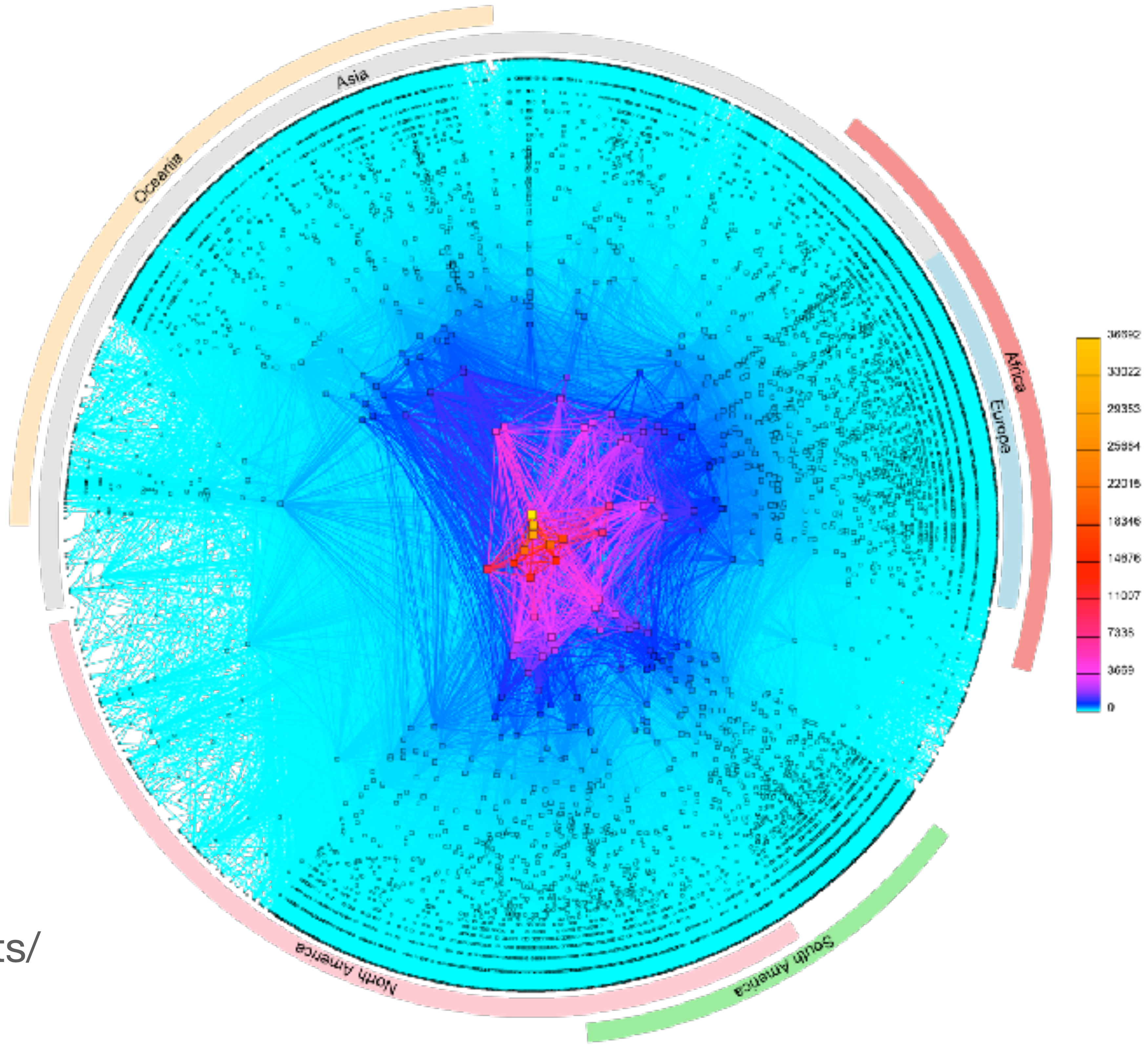


The Internet - A Network of Networks

“Facebook can't be down, can it?”, we thought, for a second.

1970s: ARPAnet — 1978: flexibility and layering — early 80s: growth → change — late 80s: growth → problems — 1993: commercialization

hosts.txt — **distance-vector routing** — TCP, UDP — **OSPF**, EGP, DNS (a link-state routing protocol) — congestion collapse — policy routing — CIDR



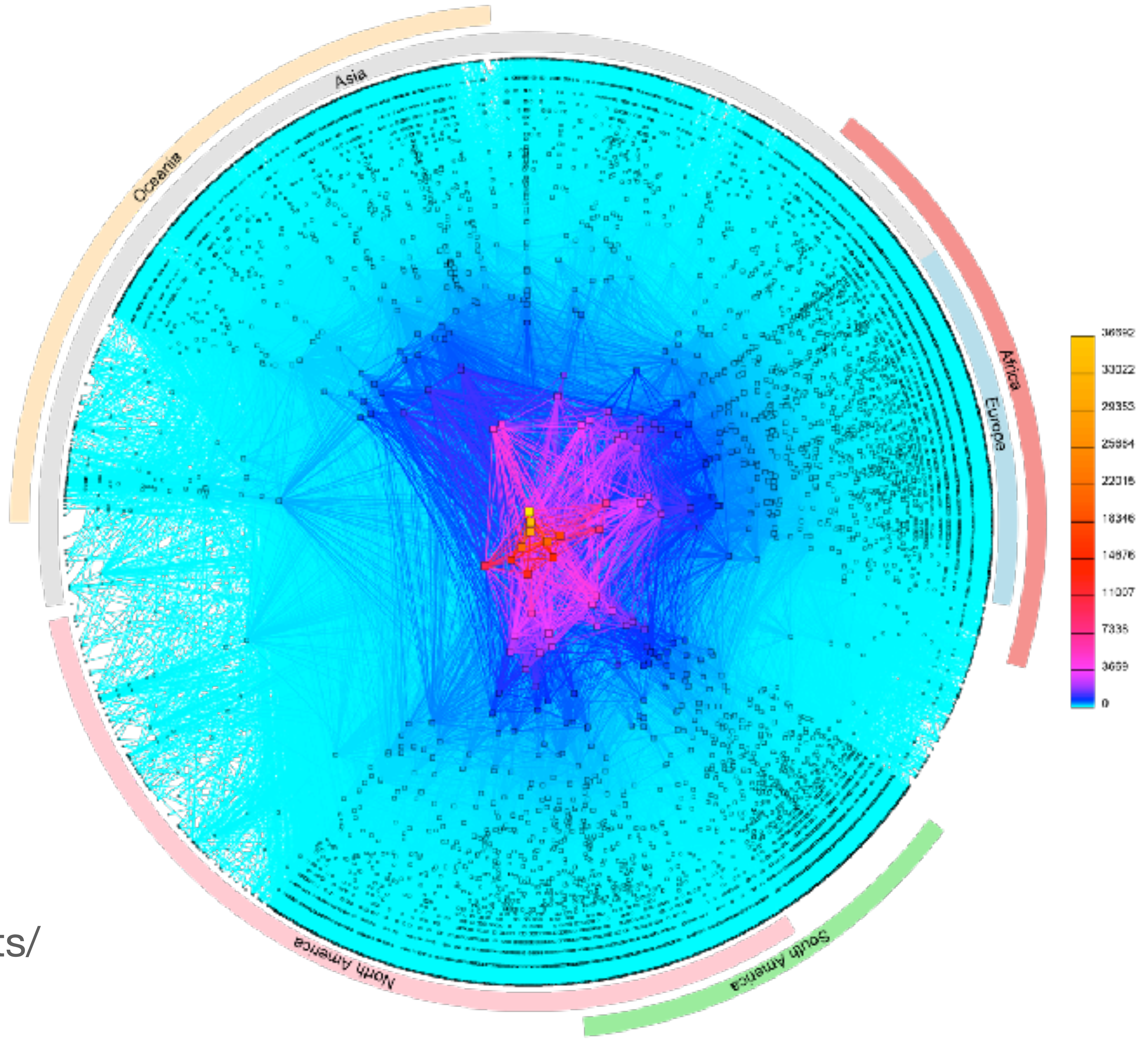
CAIDA's IPv4 AS Core, January 2020 (https://www.caida.org/projects/cartography/as-core/2020/)

last time: neither distance-vector nor link-state routing will scale to the size of the Internet, nor do either let us address policy routing

application	the things that actually generate traffic
transport	sharing the network, reliability (or not) *examples: TCP, UDP*
network	naming, addressing, routing *examples: IP*
link	communication between two directly-connected nodes *examples: ethernet, bluetooth, 802.11 (wifi)*

1970s: ARPAnet — 1978: flexibility and layering — early 80s: growth → change — late 80s: growth → problems — 1993: commercialization

hosts.txt — distance-vector routing — TCP, UDP — OSPF, **EGP**, DNS — congestion collapse — **policy routing** — CIDR



CAIDA's IPv4 AS Core, January 2020 (https://www.caida.org/projects/cartography/as-core/2020/)

this time: scale and policy!
(so we're thinking about the Internet specifically today, not just any network)

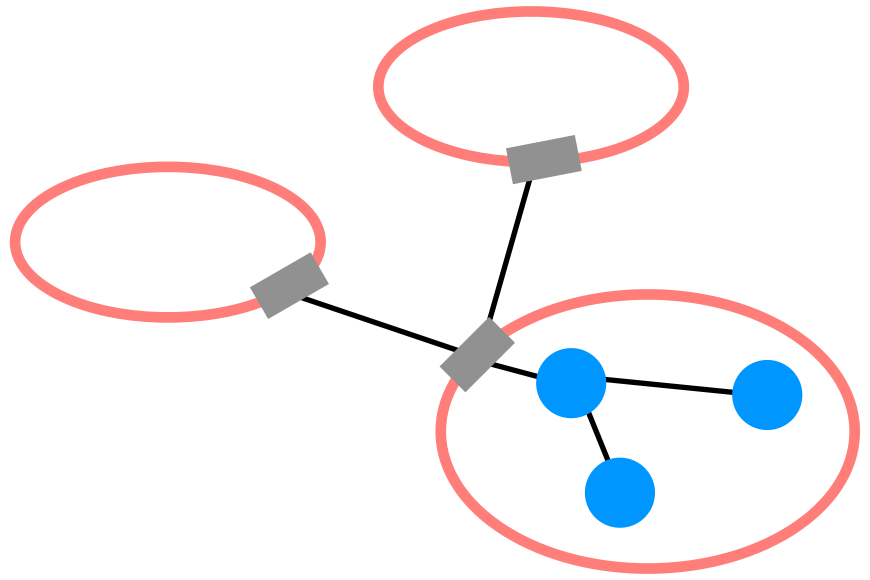
application	the things that actually generate traffic
transport	sharing the network, reliability (or not) *examples: TCP, UDP*
network	naming, addressing, routing *examples: IP*
link	communication between two directly-connected nodes *examples: ethernet, bluetooth, 802.11 (wifi)*

neither one of these algorithms will scale to the size of the internet, nor do either of them allow for *policy routing*

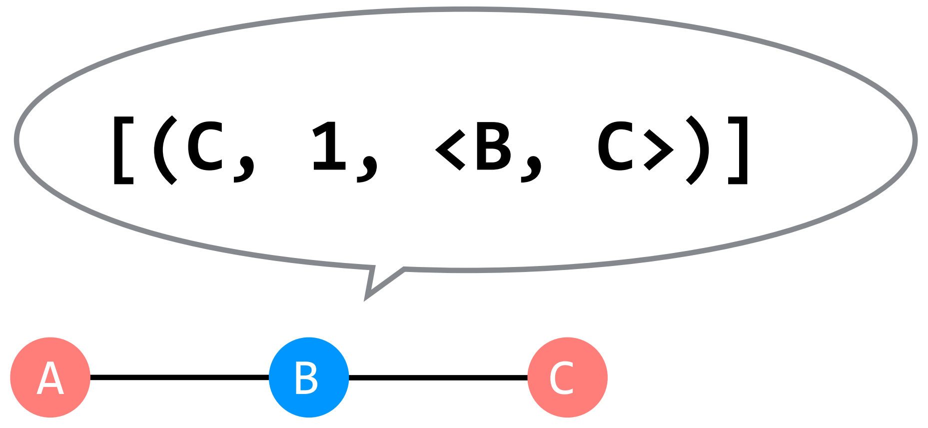
	link state	**distance vector**
what's in an advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	failures can be complicated because of timing
what limits scale?	the **overhead** of flooding	failure handling

scalable routing: a few different things allow us to route across the Internet

1. **hierarchy of routing:** route between ASes, and then within an AS



2. **path-vector routing:** similar to distance-vector, but advertisements include the path, to allow nodes to detect (and avoid) routing loops



3. **topological addressing:** assign addresses in contiguous blocks to make advertisements smaller

`18.0.0.0, … ,18.0.0.255`

↓

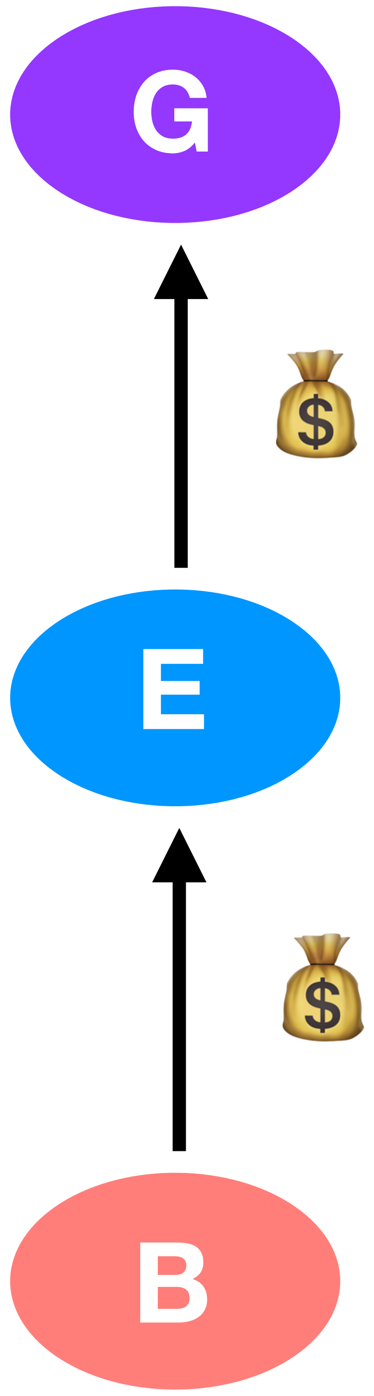
`18.0.0.*`

now that we have **scale**, we want a means to implement **policy**

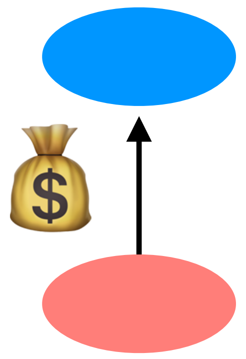
common AS relationships

arrows describe the flow of money; traffic may flow in both directions

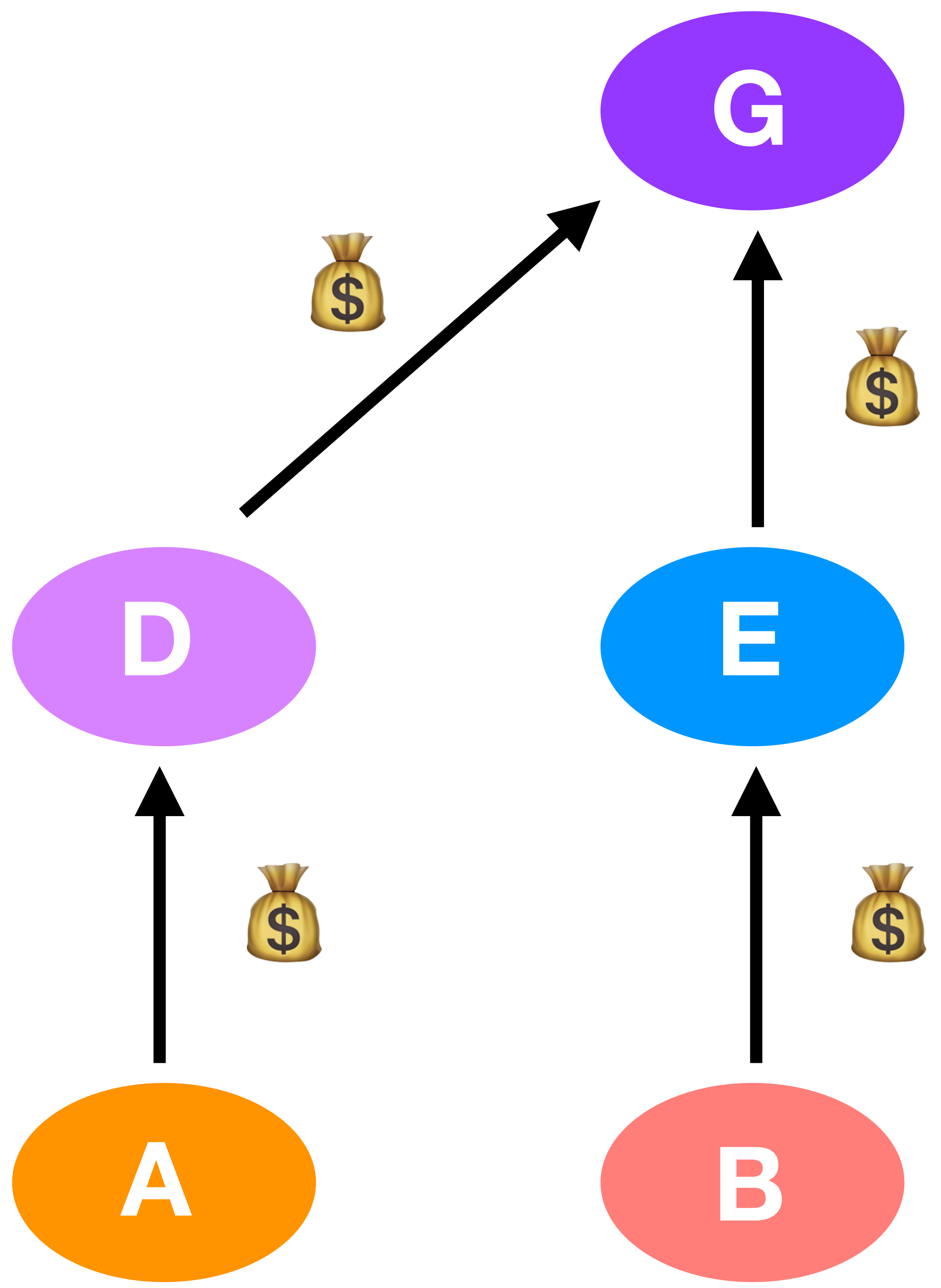
typically a provider will charge more money to its customers than it pays its own provider, so E makes a profit when traffic flows between B and G



E makes money whenever *any* traffic flows on this link

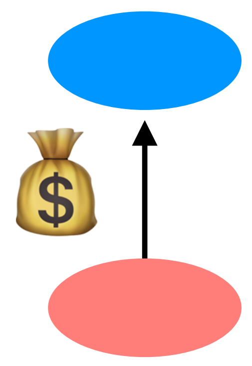


customer pays **provider** for transit

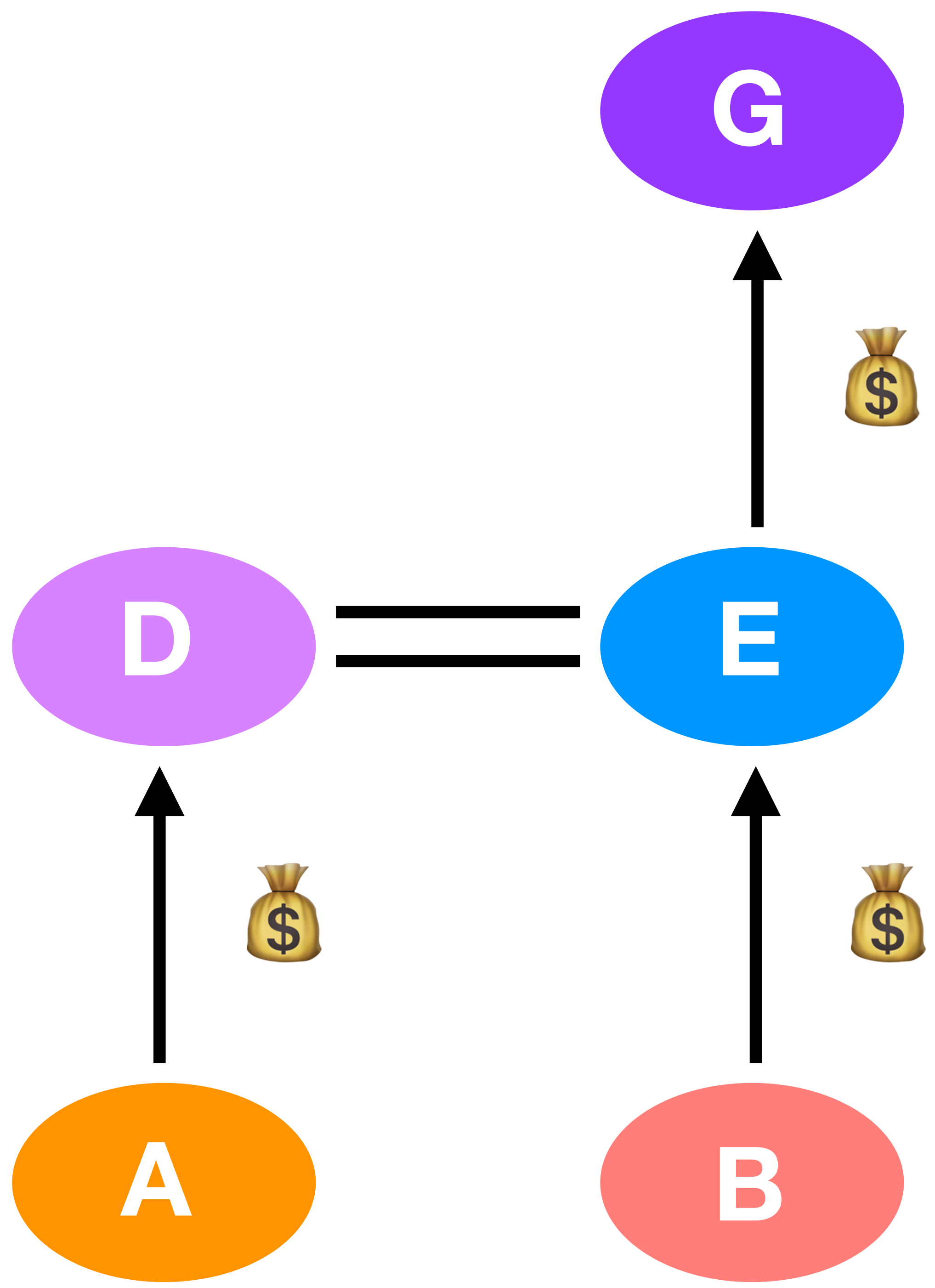


common AS relationships

arrows describe the flow of money; traffic may flow in both directions

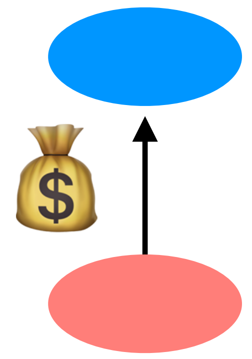


customer pays **provider** for transit

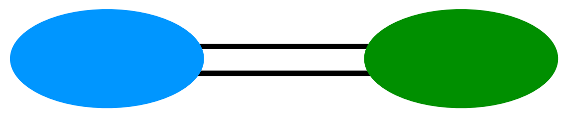


common AS relationships

arrows describe the flow of money; traffic may flow in both directions



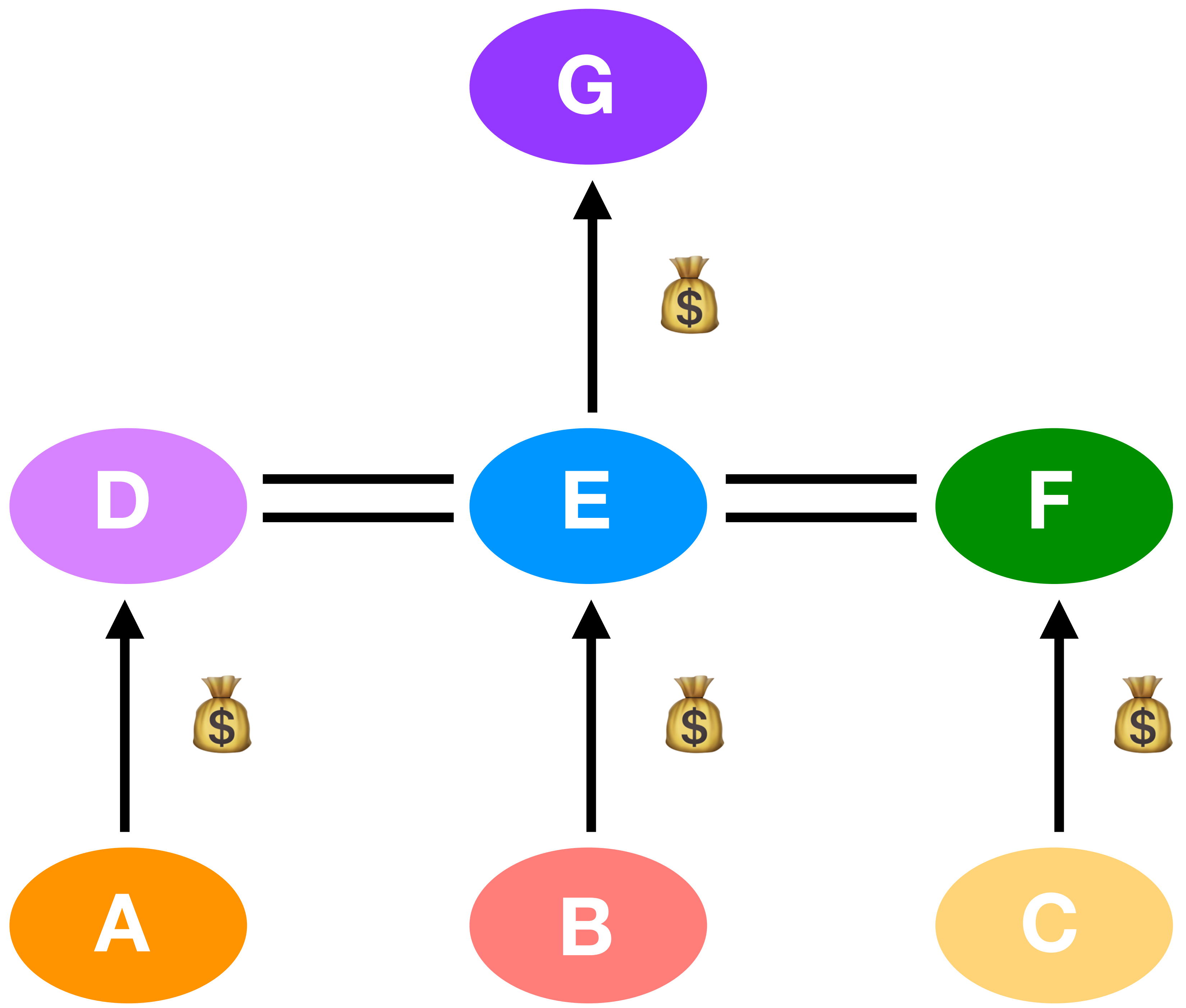
customer pays **provider** for transit



peers allow (free*) mutual access to each other's customers

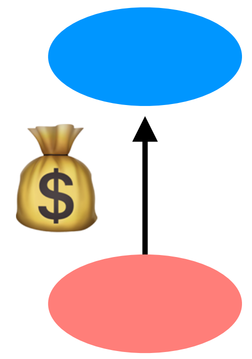
*as long as the amount of traffic in each direction is roughly equal

question: suppose traffic travels the path A-D-E-F-C. which of those ASes make money as a result?

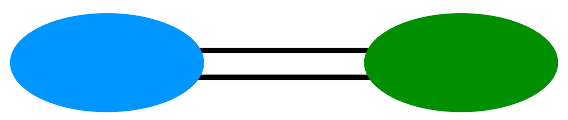


common AS relationships

arrows describe the flow of money; traffic may flow in both directions



customer pays **provider** for transit

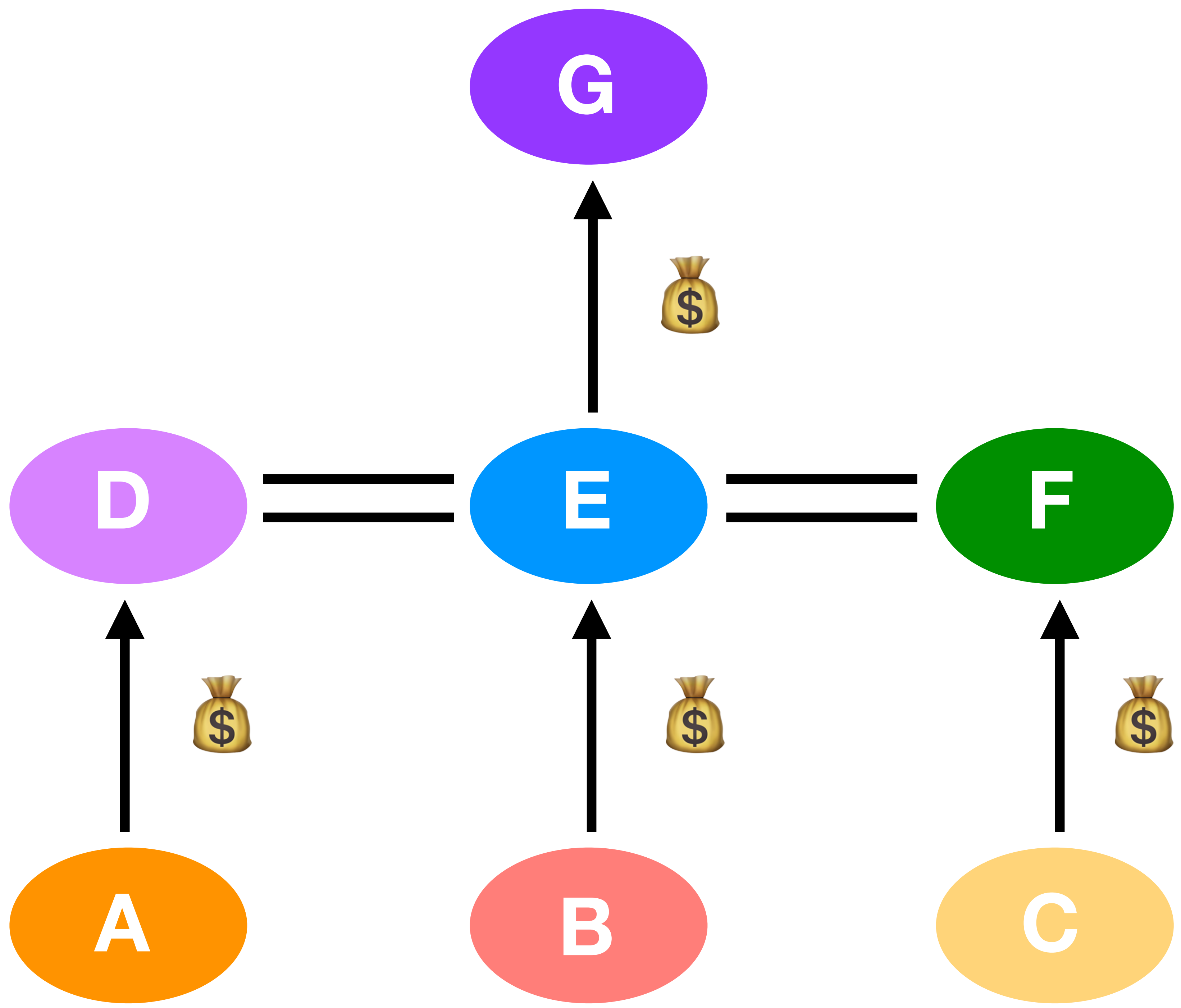


peers allow (free*) mutual access to each other's customers

*as long as the amount of traffic in each direction is roughly equal

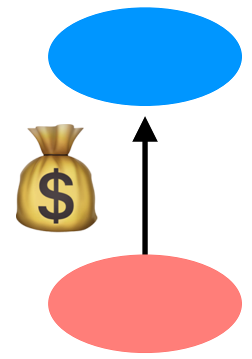
question: suppose traffic travels the path A-D-E-F-C. which of those ASes make money as a result?

if E allows its two peers to send traffic through it to their respective customers, it makes no money

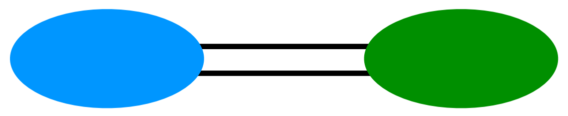


common AS relationships

arrows describe the flow of money; traffic may flow in both directions



customer pays **provider** for transit



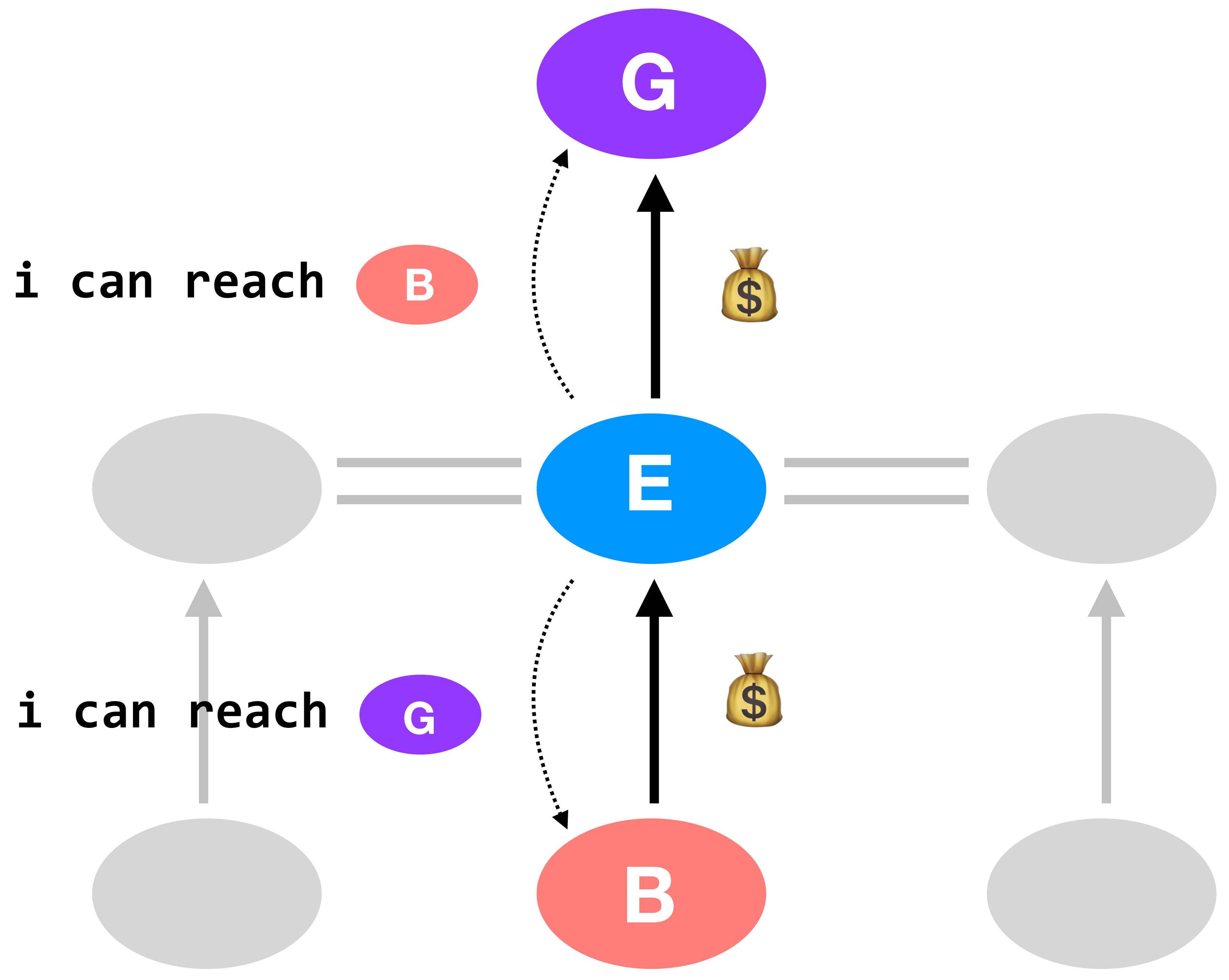
peers allow (free*) mutual access to each other's customers

*as long as the amount of traffic in each direction is roughly equal

these relationships are reflected in **export policies**

which routes to advertise, and to whom

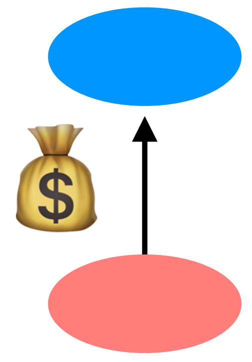
a provider wants its customers to send and receive *as much traffic through the provider as possible*



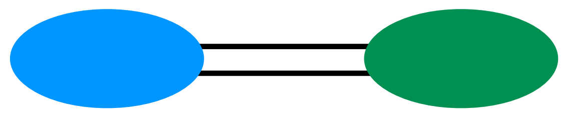
we're focusing on the middle node (E) right now; ignore the gray nodes

common AS relationships

arrows describe the flow of money; traffic may flow in both directions



customer pays **provider** for transit



peers allow (free*) mutual access to each other's customers

*as long as the amount of traffic in each direction is roughly equal

these relationships are reflected in **export policies**

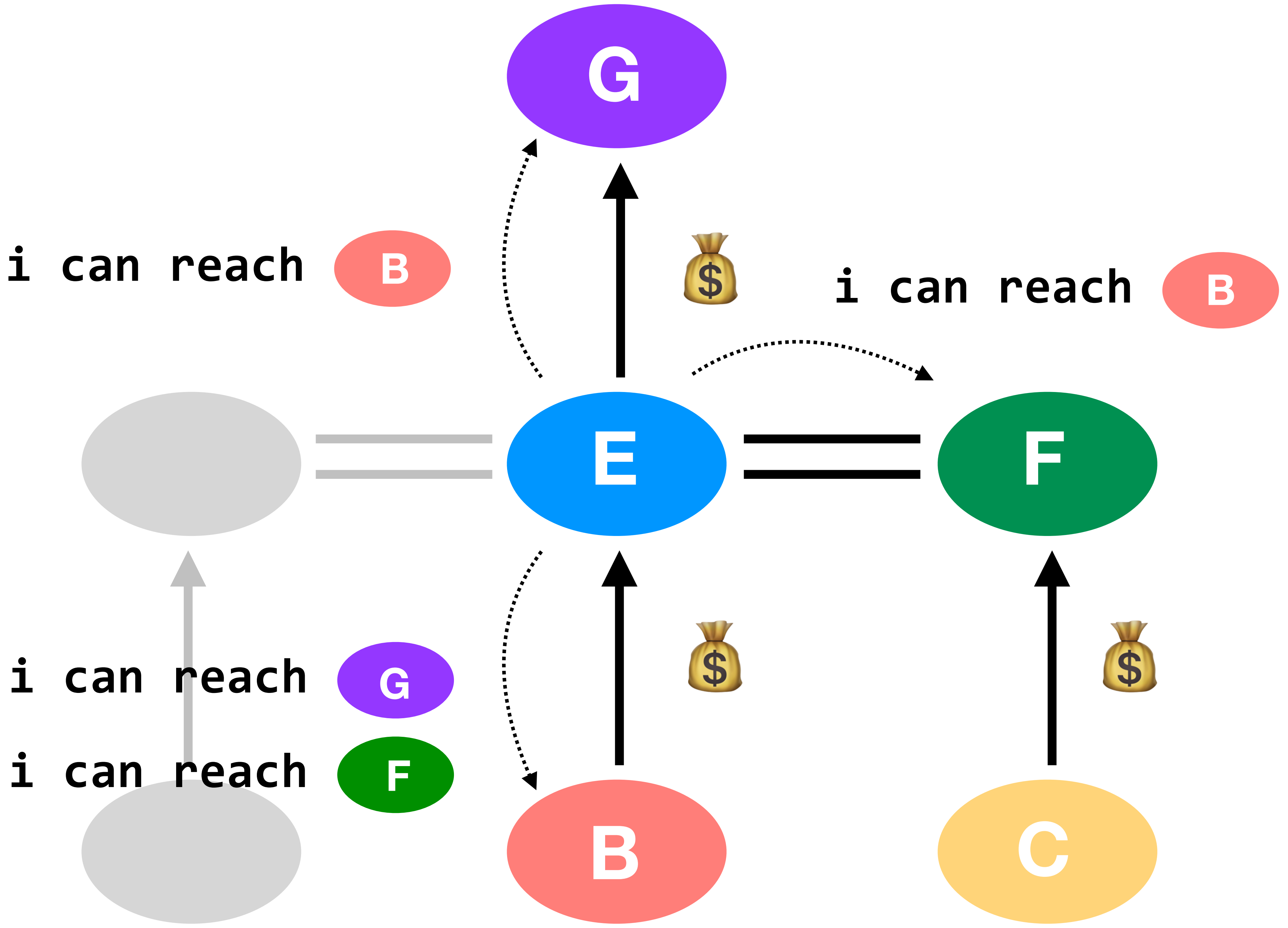
which routes to advertise, and to whom

providers tell all neighbors about their customers, and tell their customers about all neighbors*

* they'll also tell all neighbors about themselves; for example, E lets G know that it can reach all machines within E

question: after all advertisements have been sent, does C know about a route to G?

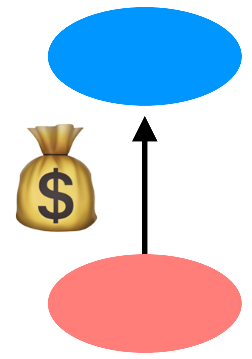
notice that peers *do not* tell each other about their own providers; they would lose money providing that transit



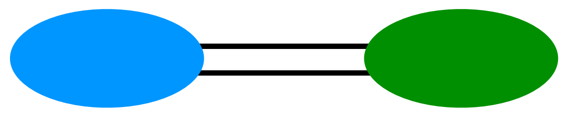
this slide represents one “round” of advertisements from node E; other routes will be discovered in subsequent rounds (see next slide)

common AS relationships

arrows describe the flow of money; traffic may flow in both directions



customer pays **provider** for transit



peers allow (free*) mutual access to each other’s customers

*as long as the amount of traffic in each direction is roughly equal

these relationships are reflected in **export policies**

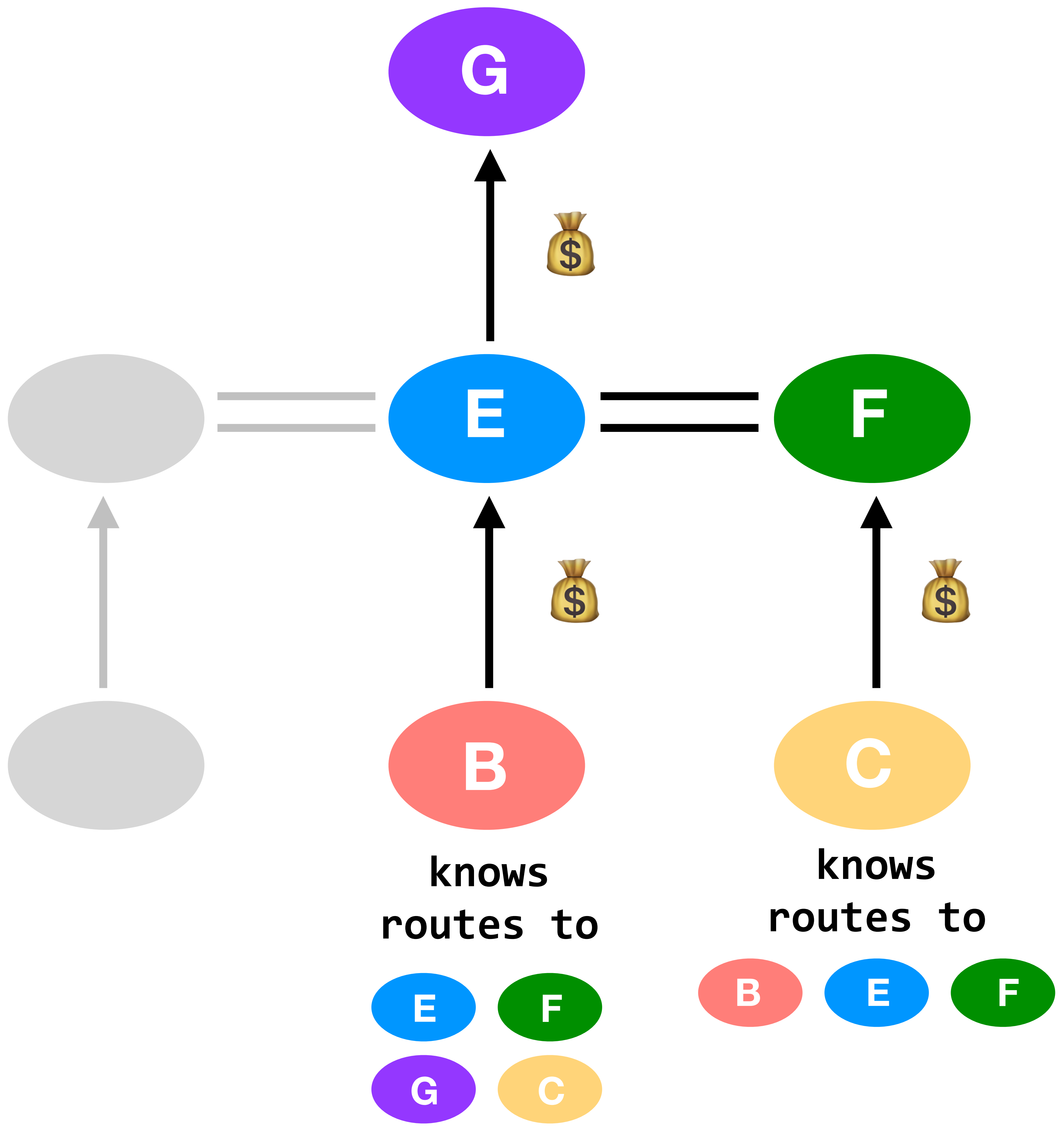
which routes to advertise, and to whom

providers tell all neighbors about their customers, and tell their customers about all neighbors*

peers tell each other about their customers

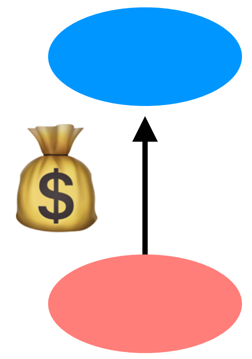
* they’ll also tell all neighbors about themselves; for example, E lets G know that it can reach all machines within E

in this example, some of our ASes are **unable** to send traffic to G ; they do not know about any routes to it

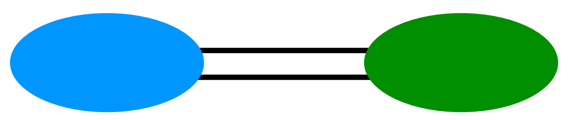


common AS relationships

arrows describe the flow of money; traffic may flow in both directions



customer pays **provider** for transit



peers allow (free*) mutual access to each other's customers

*as long as the amount of traffic in each direction is roughly equal

these relationships are reflected in **export policies**

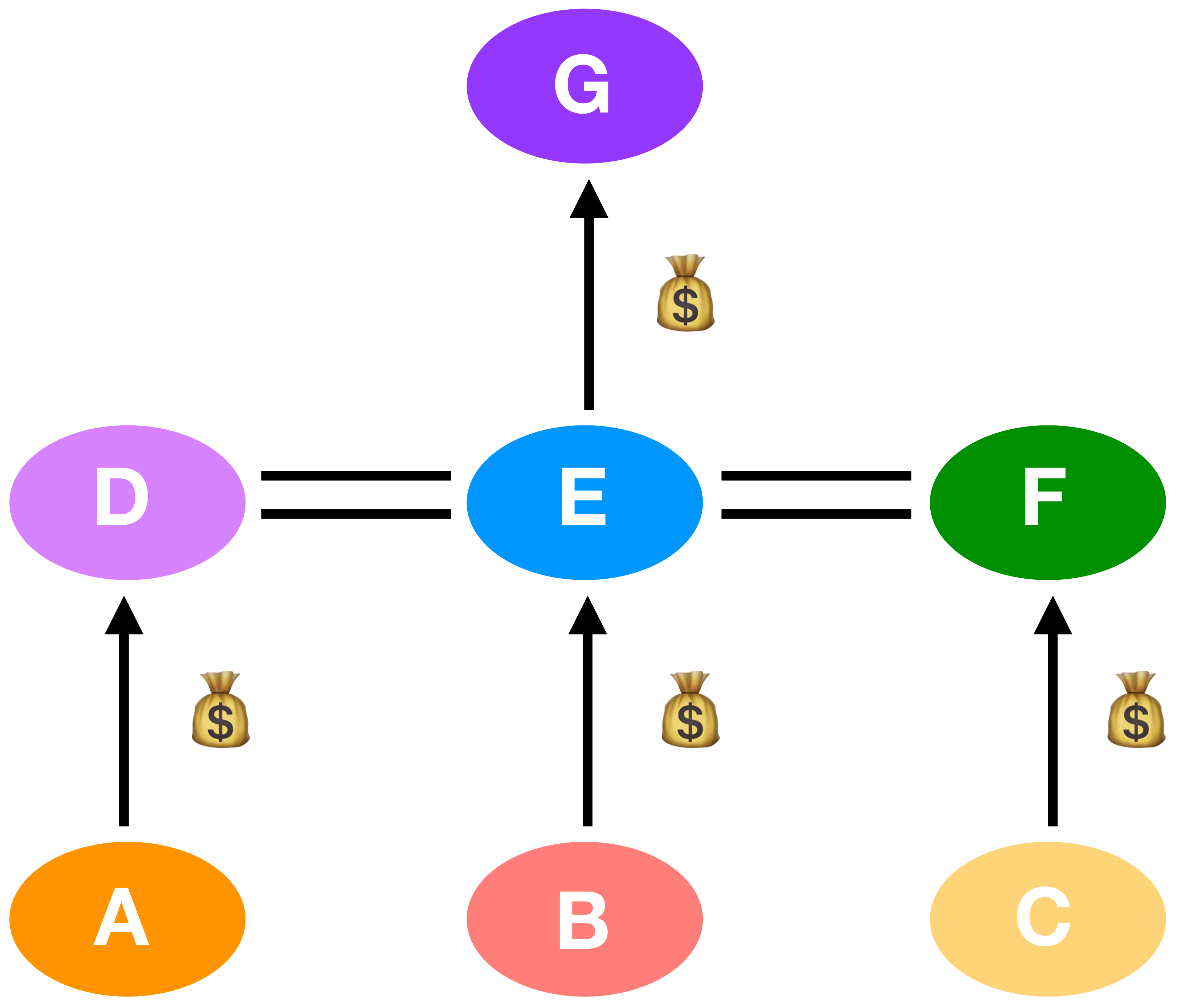
which routes to advertise, and to whom

providers tell all neighbors about their customers, and tell their customers about all neighbors*

peers tell each other about their customers

* they'll also tell all neighbors about themselves; for example, E lets G know that it can reach all machines within E

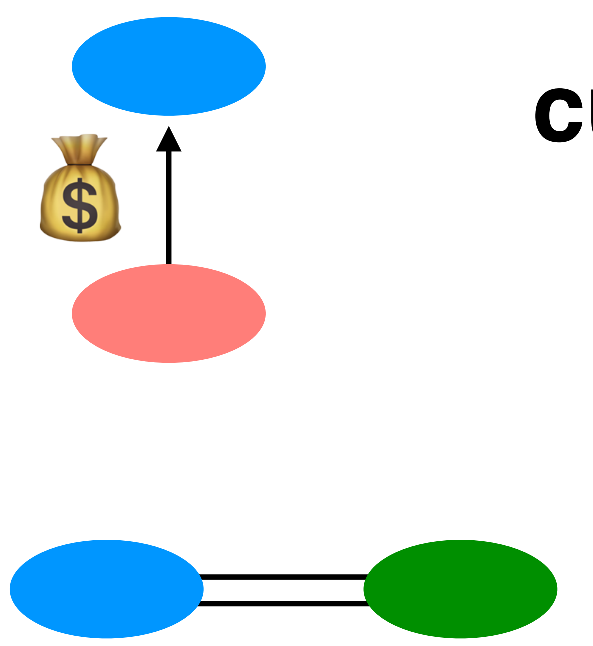
in this example, some of our ASes are **unable** to send traffic to G ; they do not know about any routes to it



in fact, there are quite a few ASes here that are disconnected from one another

common AS relationships

arrows describe the flow of money; traffic may flow in both directions



customer pays **provider** for transit

peers allow (free*) mutual access to each other's customers

*as long as the amount of traffic in each direction is roughly equal

these relationships are reflected in **export policies**

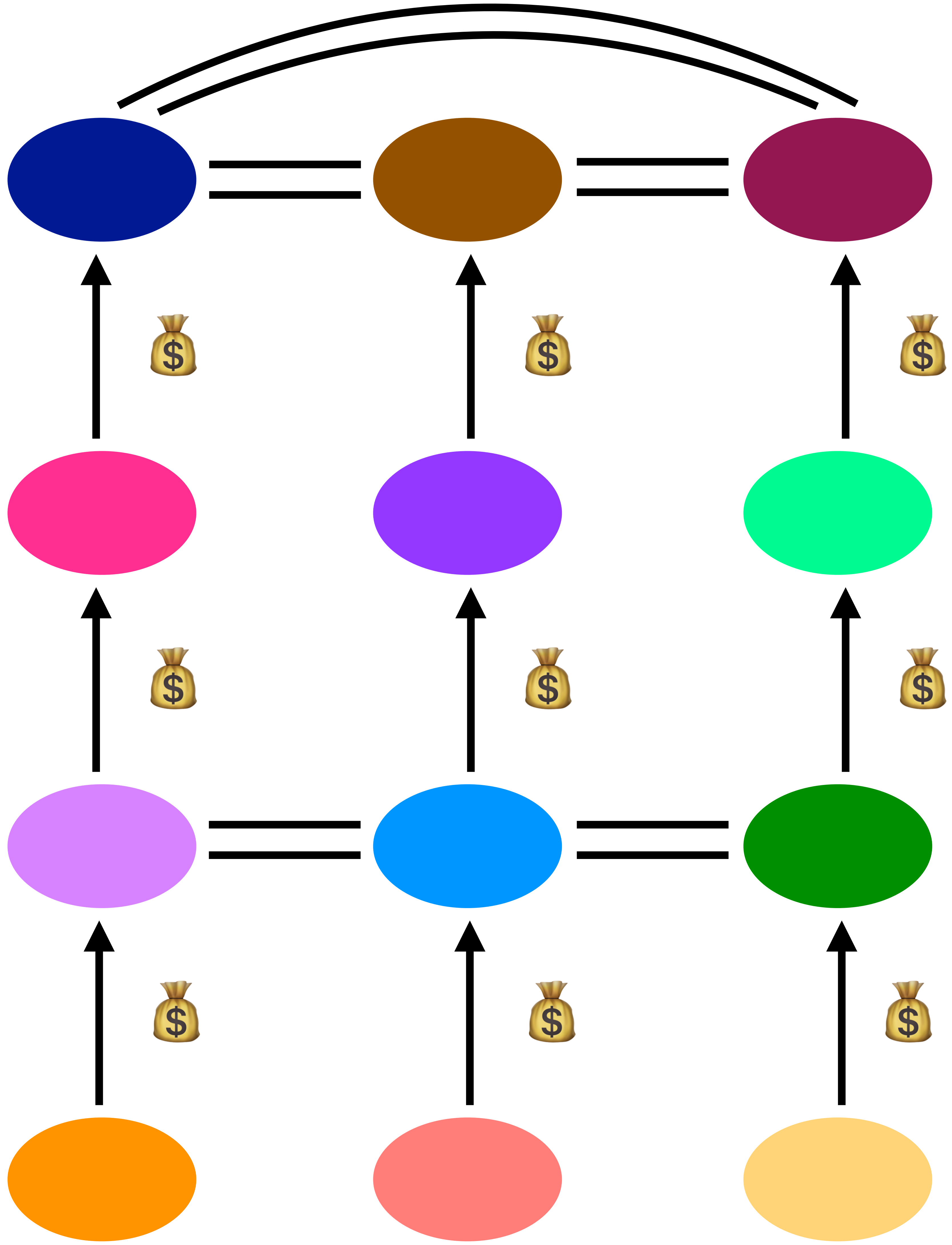
which routes to advertise, and to whom

providers tell all neighbors about their customers, and tell their customers about all neighbors*

peers tell each other about their customers

* they'll also tell all neighbors about themselves; for example, E lets G know that it can reach all machines within E

Katrina LaCurts | lacurts@mit.edu | 6.1800 2025

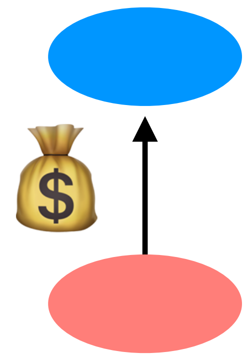


on the Internet, all of the top tier ("tier-1") ISPs peer, to provide global connectivity

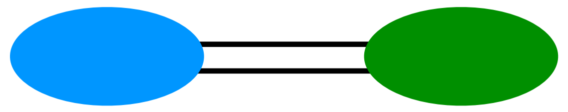
this is an extremely simplified diagram. you'd expect to see other sorts of peering agreements in this graph, and in fact other sorts of AS relationships

common AS relationships

arrows describe the flow of money; traffic may flow in both directions



customer pays **provider** for transit



peers allow (free*) mutual access to each other's customers

*as long as the amount of traffic in each direction is roughly equal

these relationships are reflected in **export policies**

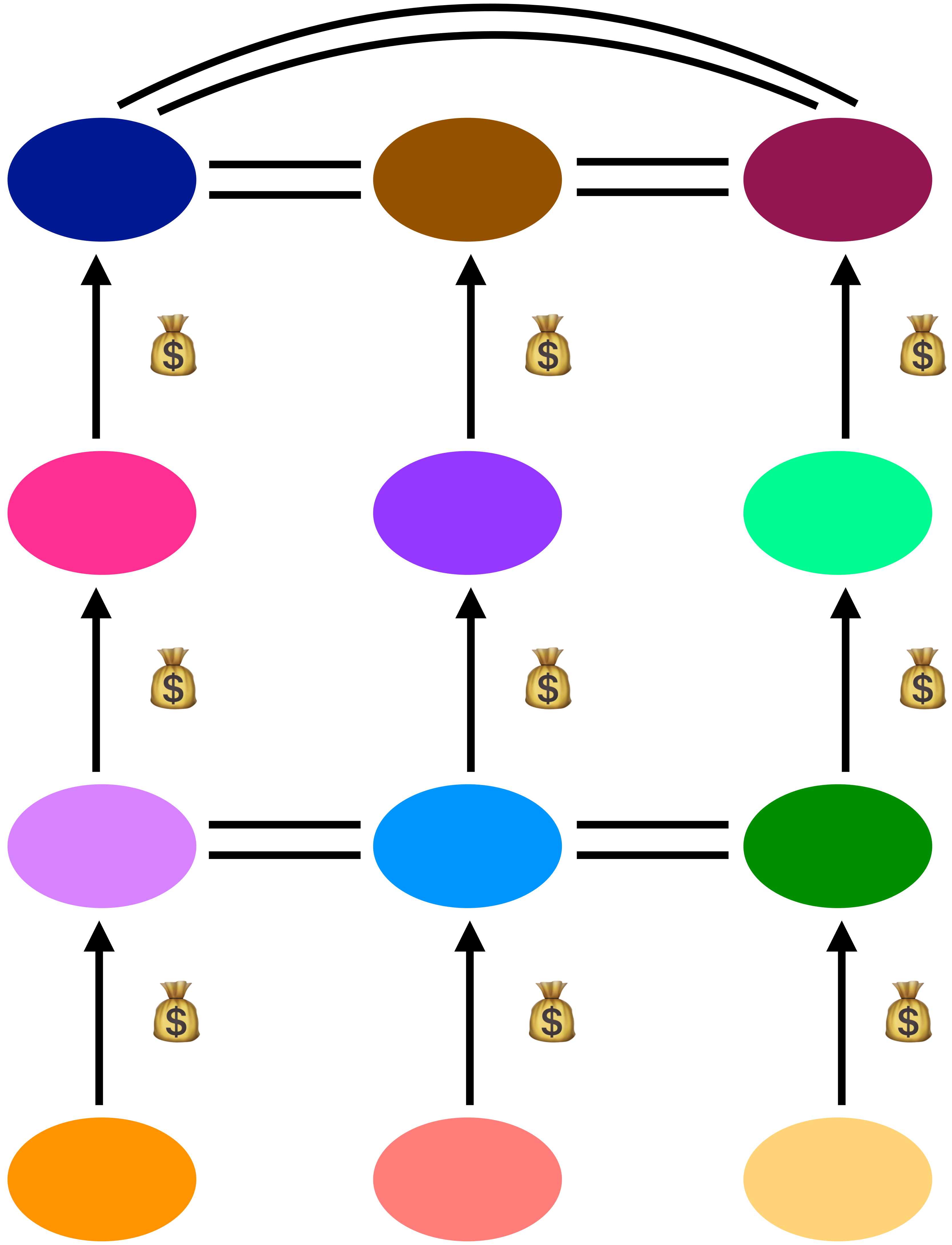
which routes to advertise, and to whom

providers tell all neighbors about their customers, and tell their customers about all neighbors*

peers tell each other about their customers

* they'll also tell all neighbors about themselves; for example, E lets G know that it can reach all machines within E

Katrina LaCurts | lacurts@mit.edu | 6.1800 2025

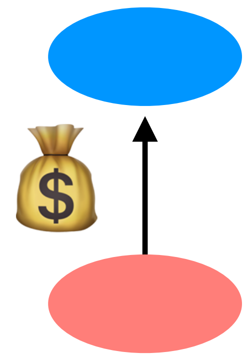


on the Internet, all of the top tier (“tier-1”) ISPs peer, to provide global connectivity

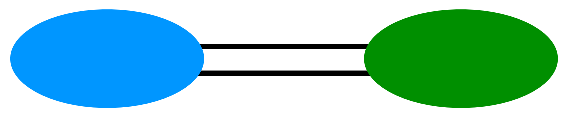
this is an extremely simplified diagram. you’d expect to see other sorts of peering agreements in this graph, and in fact other sorts of AS relationships

common AS relationships

arrows describe the flow of money; traffic may flow in both directions



customer pays **provider** for transit



peers allow (free*) mutual access to each other’s customers

*as long as the amount of traffic in each direction is roughly equal

these relationships are also reflected in **import policies**

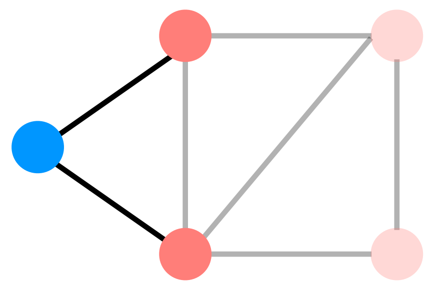
which routes to *use*

ASes set their own *import policies*. typically, if an AS hears about multiple routes to a destination, it will prefer to use its customers first, then peers, then providers

if that’s not enough, a variety of other attributes are provided

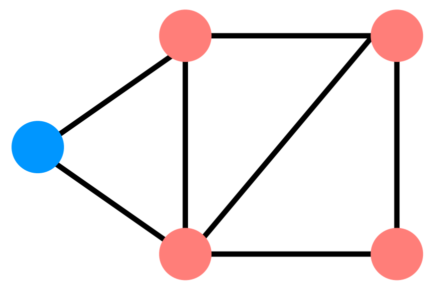
BGP as a distributed routing protocol

1. nodes learn about their neighbors via the `HELLO` protocol



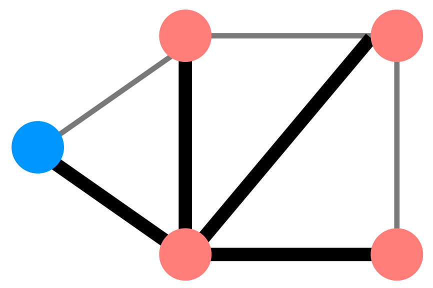
nodes send “KEEPALIVE” messages to their neighbors once every ~sixty seconds

2. nodes learn about other reachable nodes via advertisements



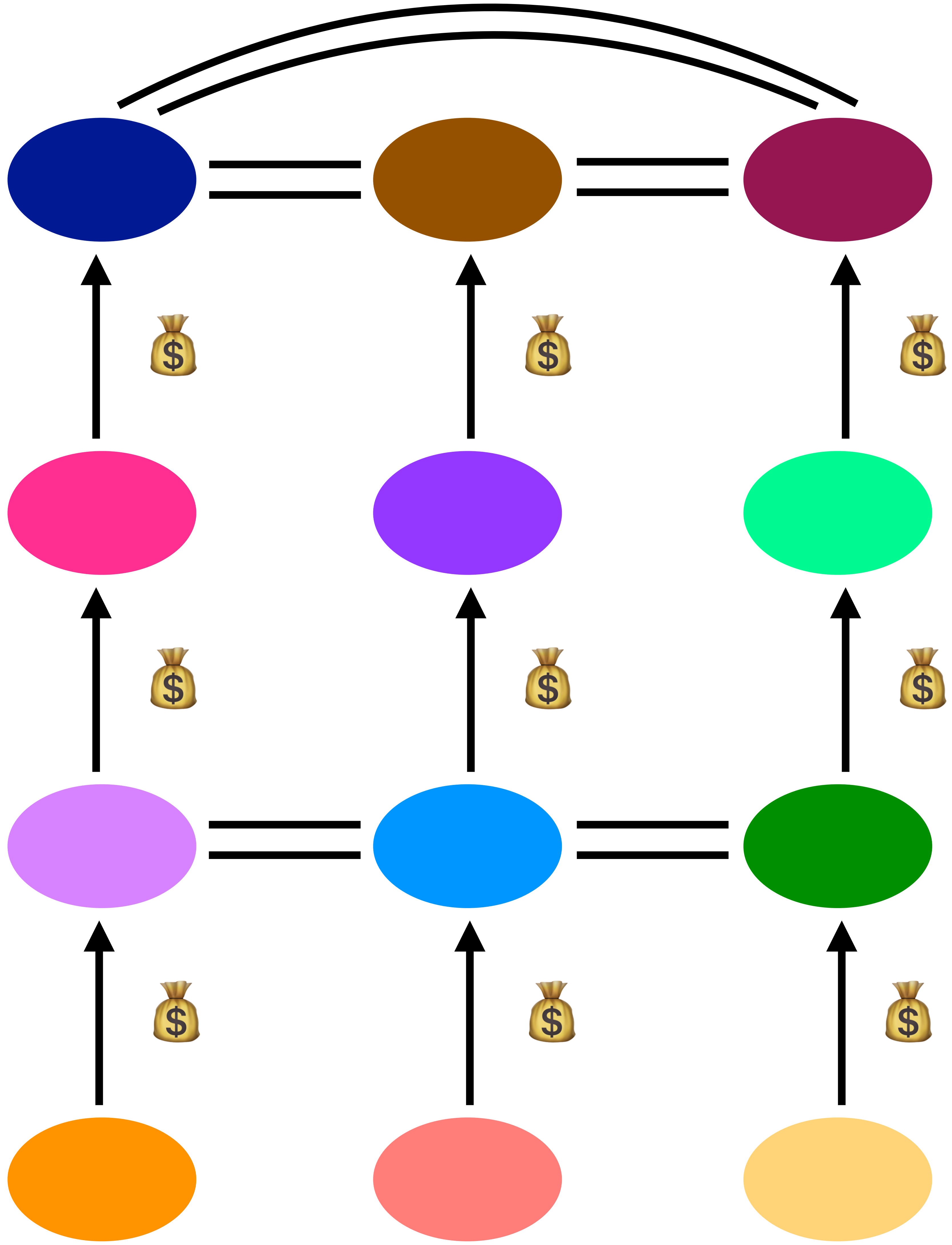
nodes send advertisements to their neighbors, but the content of each advertisement will differ depending on the AS relationships (e.g., customer/provider, peer). this is where we see the “export policies” play out

3. nodes determine the minimum-cost routes (of the routes they know about)



nodes choose which routes to use based on AS relationships and a number of other properties (e.g., path length) when needed. this is where we see the “import policies” play out

BGP is an application-layer protocol, even though it deals with routing. It runs on top of TCP, which provides reliable transport; doing this lets BGP handle failures differently than link-state and distance-vector routing



on the Internet, all of the top tier ("tier-1") ISPs peer, to provide global connectivity

this is an extremely simplified diagram. you'd expect to see other sorts of peering agreements in this graph, and in fact other sorts of AS relationships

does BGP scale?

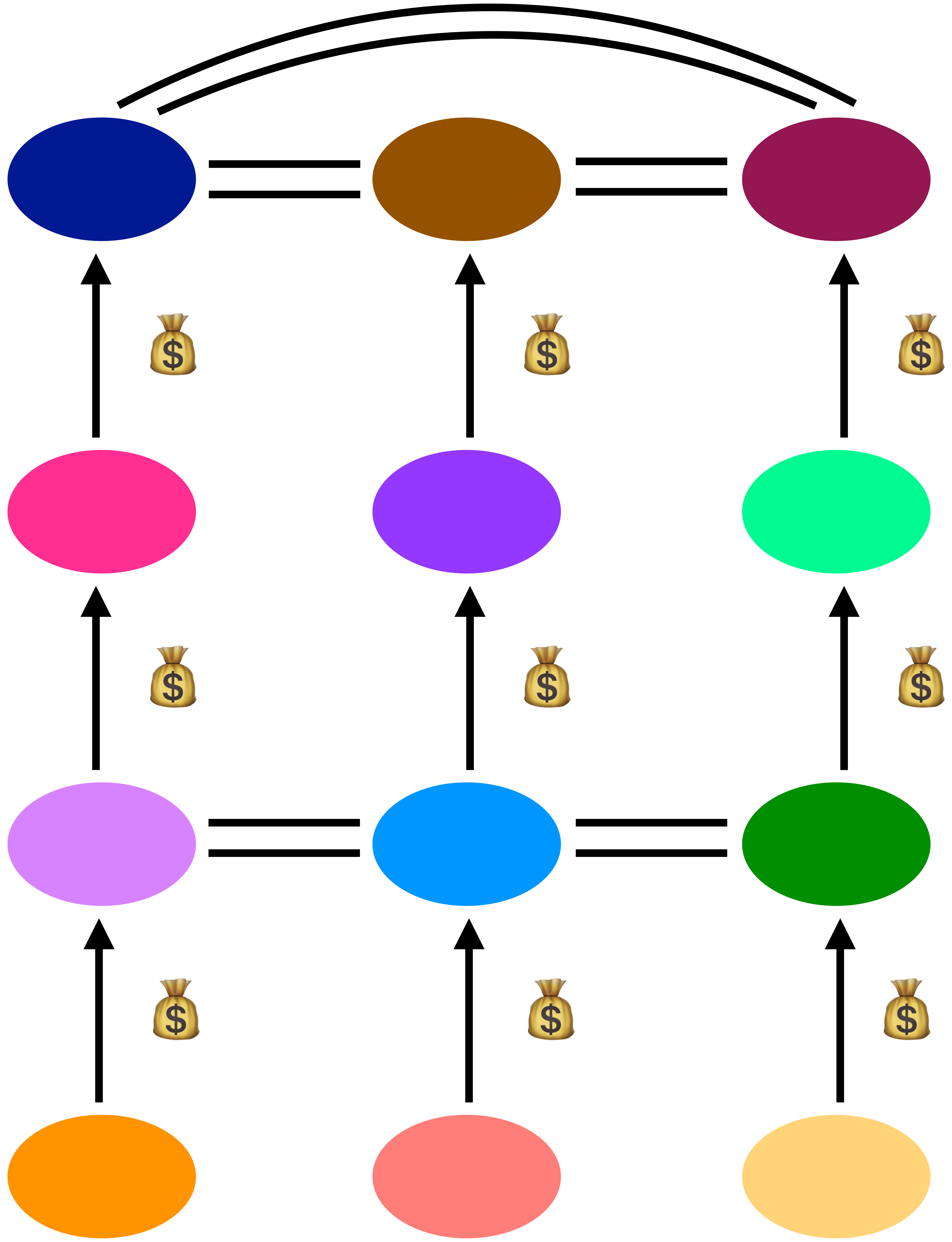
it works on the Internet (which is good), but the size of routing tables, route instability, multihoming, and iBGP all cause scaling issues

is BGP secure?

it is not!

does BGP matter?

absolutely — it is a huge part of the Internet's infrastructure



on the Internet, all of the top tier ("tier-1") ISPs peer, to provide global connectivity

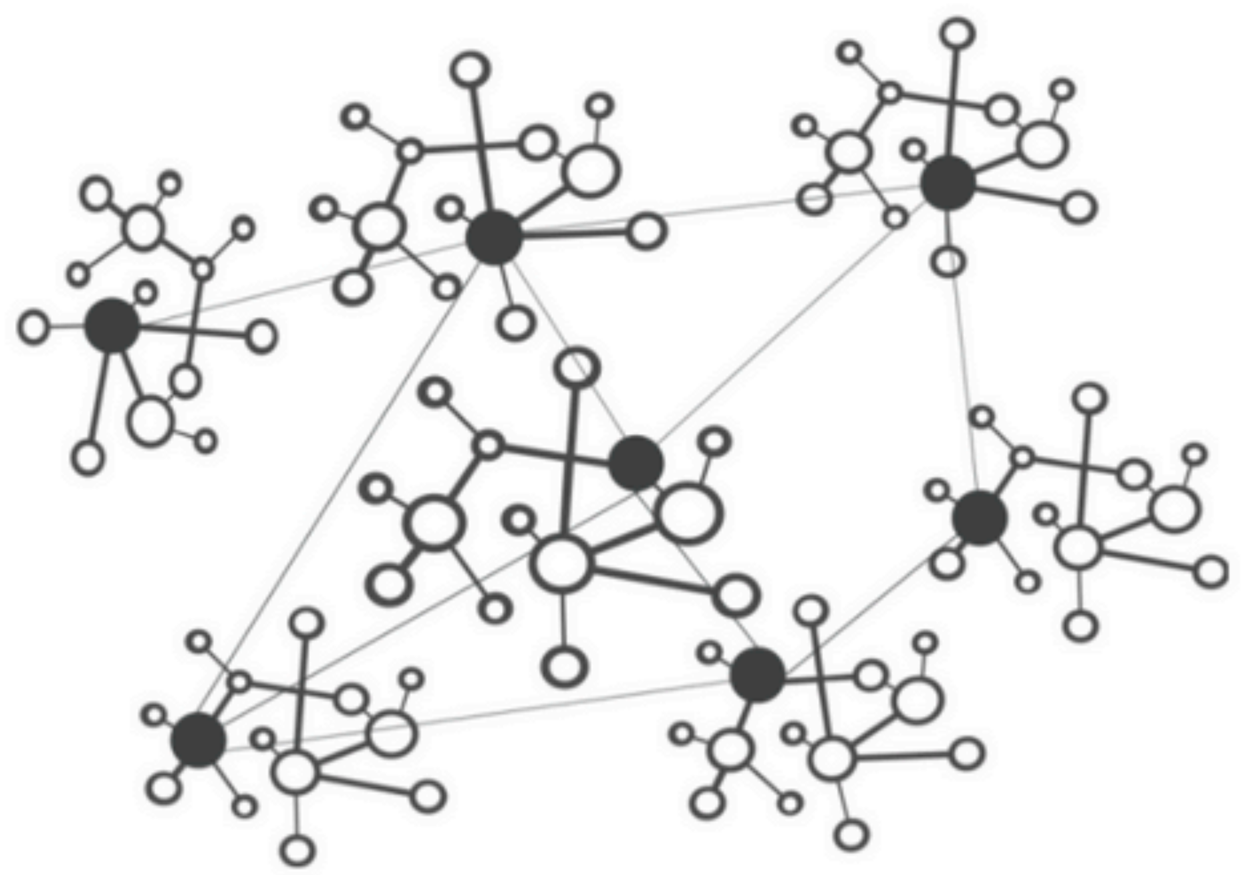
this is an extremely simplified diagram. you'd expect to see other sorts of peering agreements in this graph, and in fact other sorts of AS relationships

Understanding How Facebook Disappeared from the Internet

10/04/2021

Celso Martinho Tom Strickx

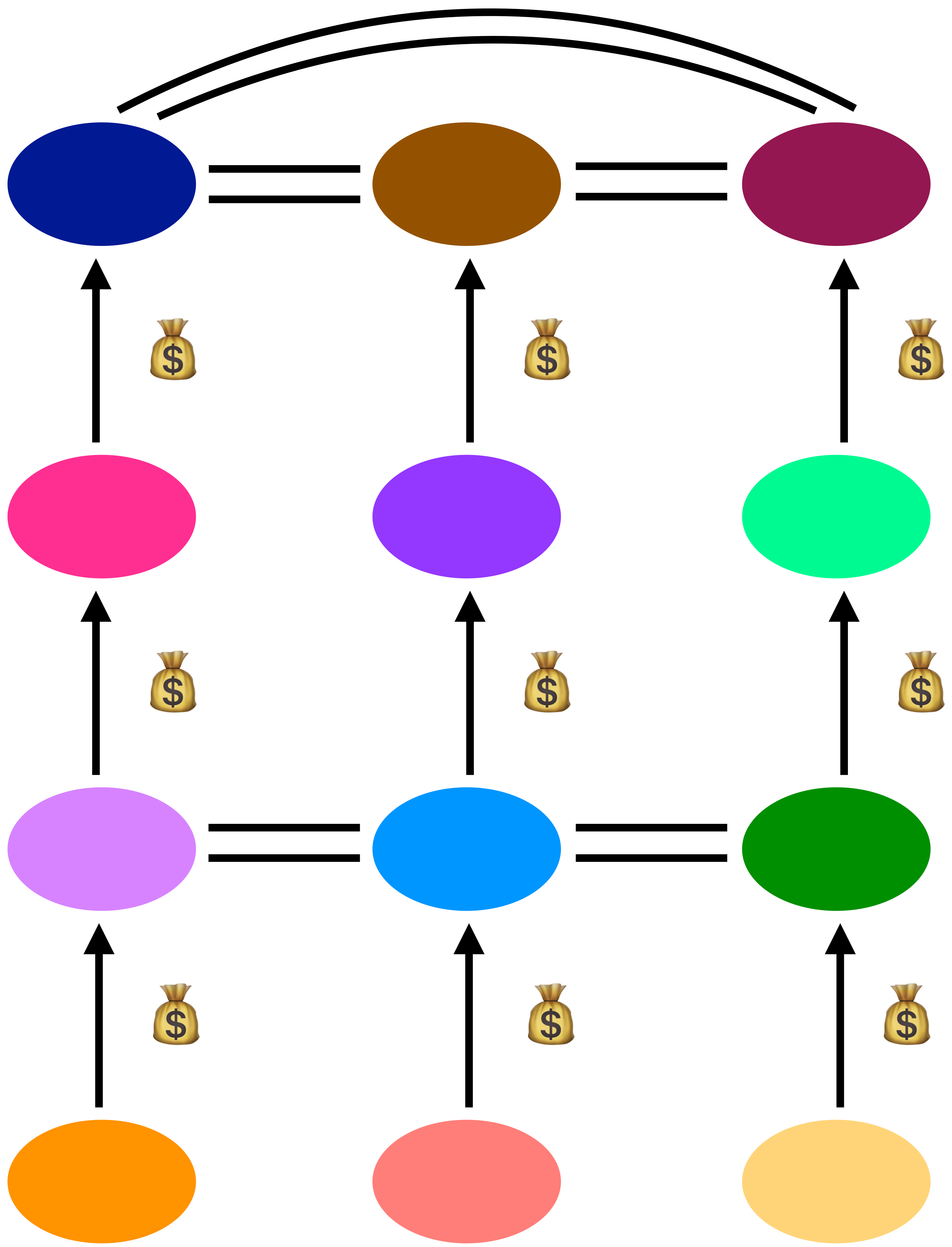
This post is also available in 简体中文, 繁體中文, 日本語, 한국어, Deutsch, Français, Español, Português, Русский, and Italiano.



The Internet - A Network of Networks

"Facebook can't be down, can it?", we thought, for a second.

https://blog.cloudflare.com/october-2021-facebook-outage/



on the Internet, all of the top tier (“tier-1”) ISPs peer, to provide global connectivity

this is an extremely simplified diagram. you’d expect to see other sorts of peering agreements in this graph, and in fact other sorts of AS relationships

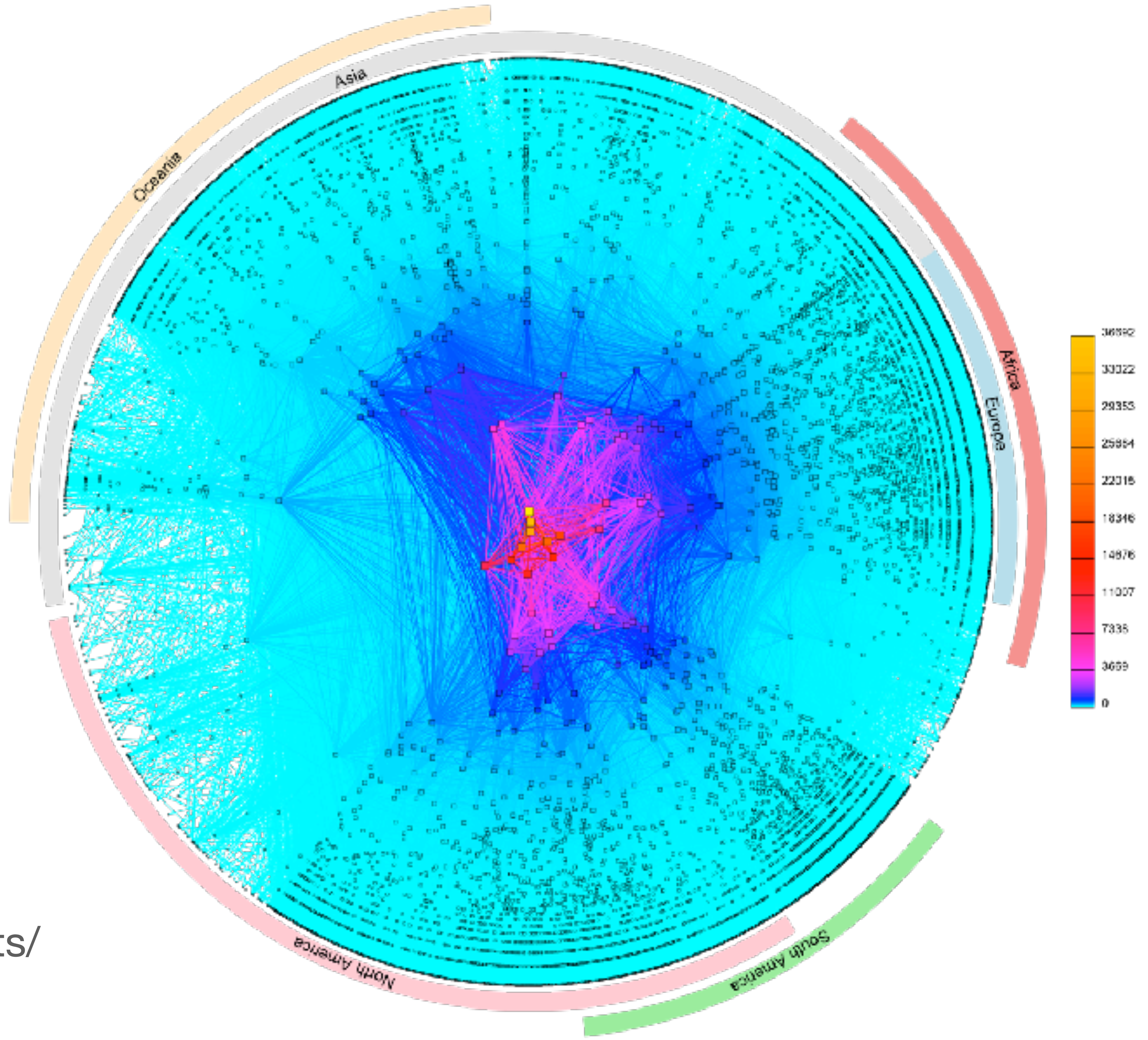
This was the source of yesterday's outage. During one of these routine maintenance jobs, a command was issued with the intention to assess the availability of global backbone capacity, which unintentionally took down all the connections in our backbone network, effectively disconnecting Facebook data centers globally. Our systems are designed to audit commands like these to prevent mistakes like this, but a bug in that audit tool prevented it from properly stopping the command.

All of this happened very fast. And as our engineers worked to figure out what was happening and why, they faced two large obstacles: first, it was not possible to access our data centers through our normal means because their networks were down, and second, the total loss of DNS broke many of the internal tools we'd normally use to investigate and resolve outages like this.

https://engineering.fb.com/2021/10/05/networking-traffic/outage-details/

1970s: ARPAnet — 1978: flexibility and layering — early 80s: growth → change — late 80s: growth → problems — 1993: commercialization

hosts.txt | **distance-vector routing** | TCP, UDP | **OSPF**, EGP, DNS (a link-state routing protocol) | congestion collapse | policy routing | CIDR



CAIDA's IPv4 AS Core, January 2020 (https://www.caida.org/projects/cartography/as-core/2020/)

on the Internet, we have to solve all of the "normal" networking problems (addressing, routing, transport) **at massive scale, while supporting a diverse group of applications and competing economic interests**

application	the things that actually generate traffic
transport	sharing the network, reliability (or not) *examples: TCP, UDP*
network	naming, addressing, routing *examples: IP*
link	communication between two directly-connected nodes *examples: ethernet, bluetooth, 802.11 (wifi)*