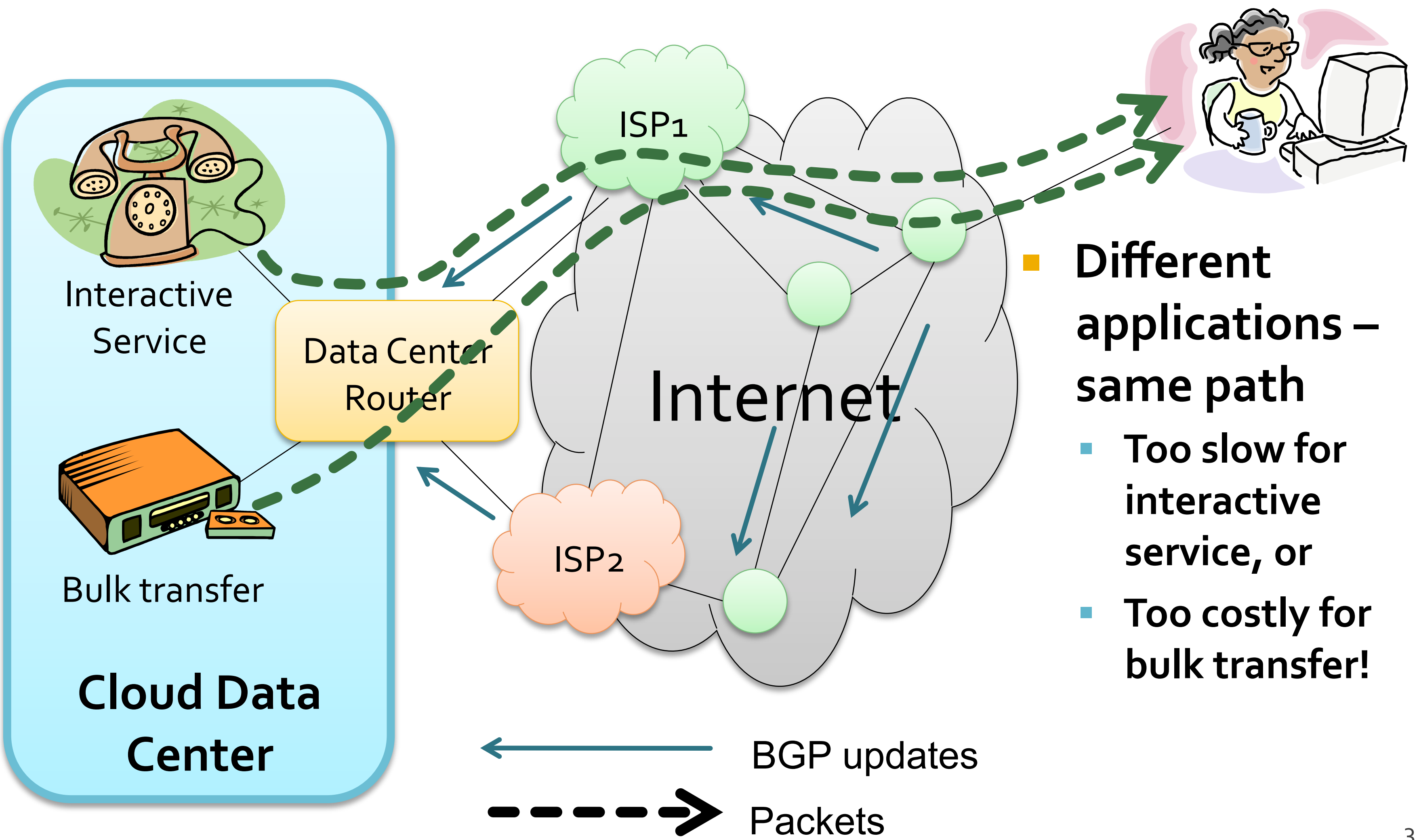
Vytautas Valancius, Nick Feamster, Akihiro Nakao, and Jennifer Rexford

Wide-Area Route Control for Distributed Services

Context

- Hosting and Cloud computing is on the rise
  - Collocation hosting
  - Cloud and data center hosting
- Different hosted applications have different requirements for routing
  - Interactive services vs bulk transfer
- Cloud and hosting does not expose routing to applications

Accessing Internet Services



- Different applications – same path
  - Too slow for interactive service, or
  - Too costly for bulk transfer!

Routing From Data Center Today

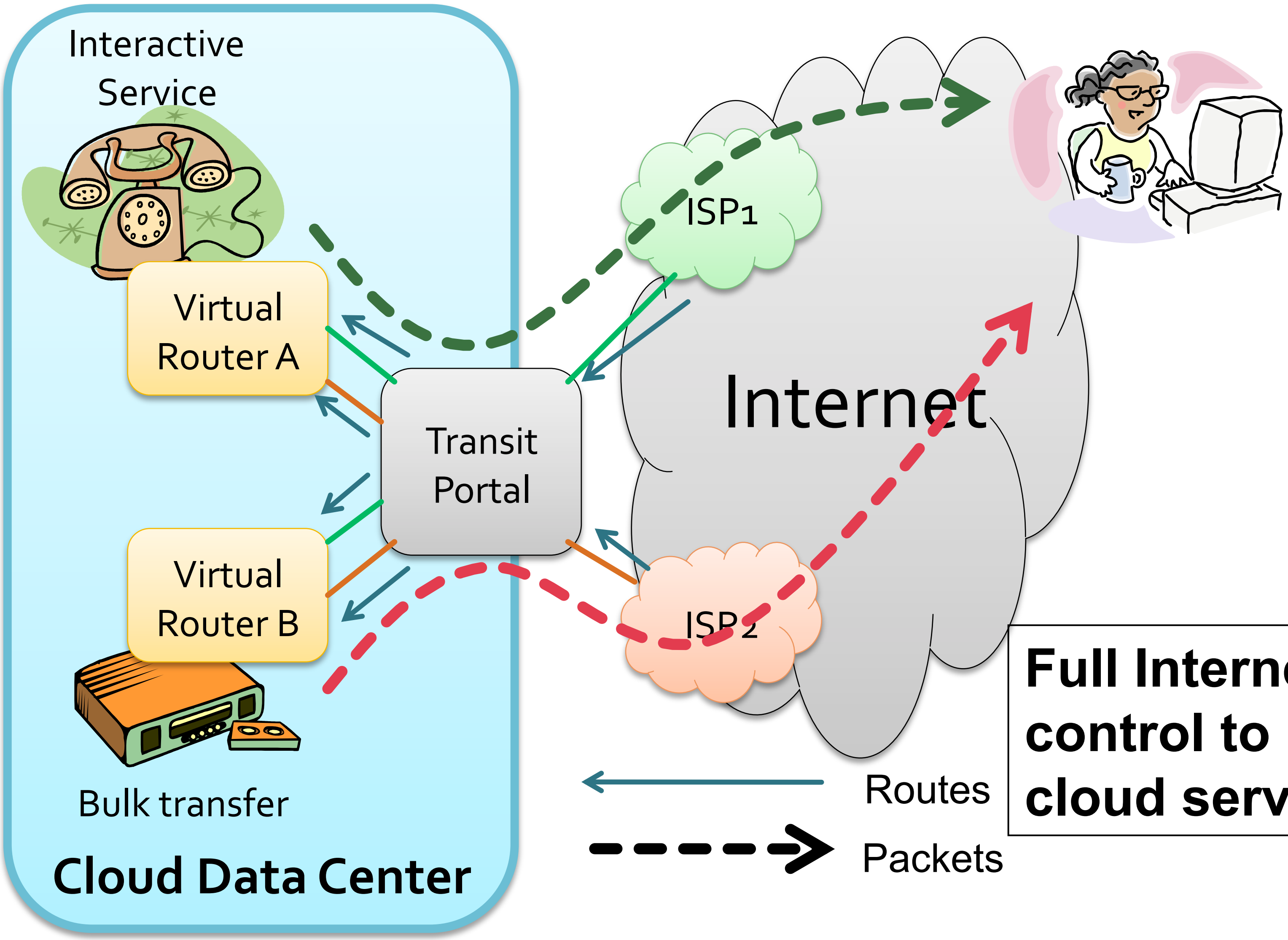
- Multiple upstream ISPs
  - Amazon EC2 has at least 58 external routing sessions in Virginia data center

- Data center router picks **one route** to a destination for all hosted services
  - Packets from all hosted applications use the same path to a destination prefix

Custom Route Control Today

- Obtain connectivity to upstream ISPs
  - Physical connectivity
  - Contracts and routing sessions
- Obtain the Internet numbered resources from authorities
- **Expensive and time-consuming!**

Routing with Transit Portal (TP)



Full Internet route control to hosted cloud services!

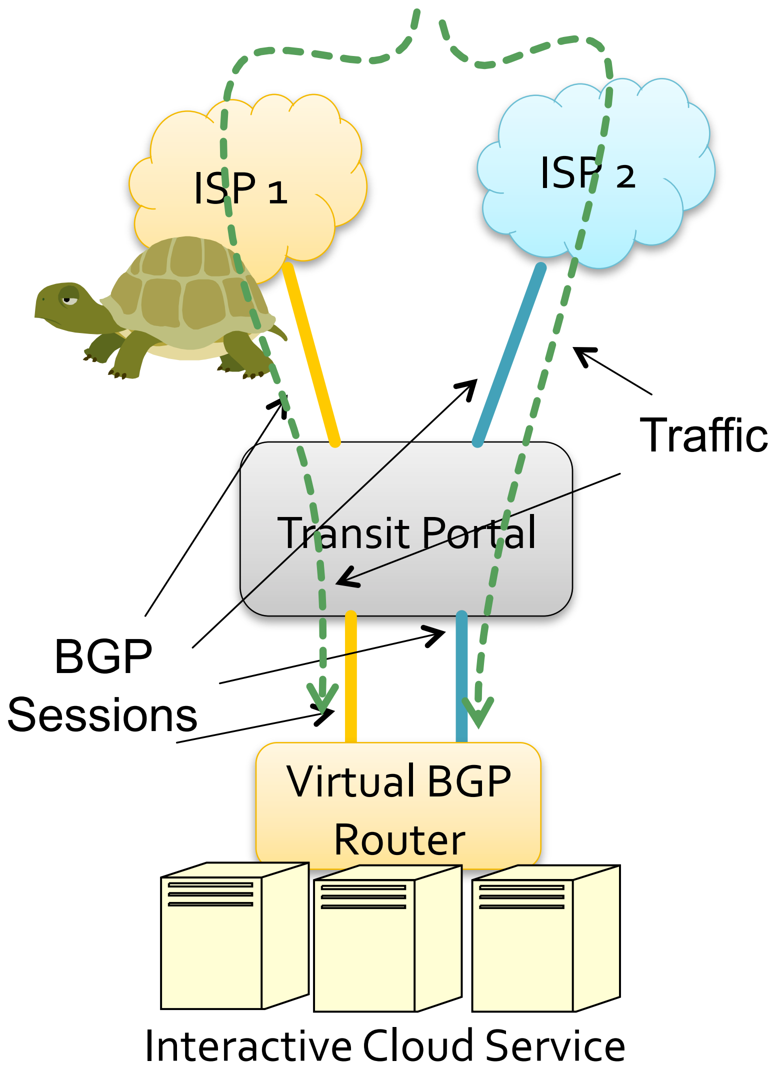
Outline

- Motivation and Overview
- Connecting to the Transit Portal
- Scaling the Transit Portal
- Current Transit Portal Deployment
- Future Work & Summary

Connecting to the TP

- Each hosted service runs its own “router”
  - This router might be physical or virtual
- Links between the service’s router and TP
  - Each link corresponds to (and emulates) a dedicated connection to upstream ISP
- Routing sessions to upstream ISPs
  - BGP for route control
  - The service’s router connects to the TP via BGP

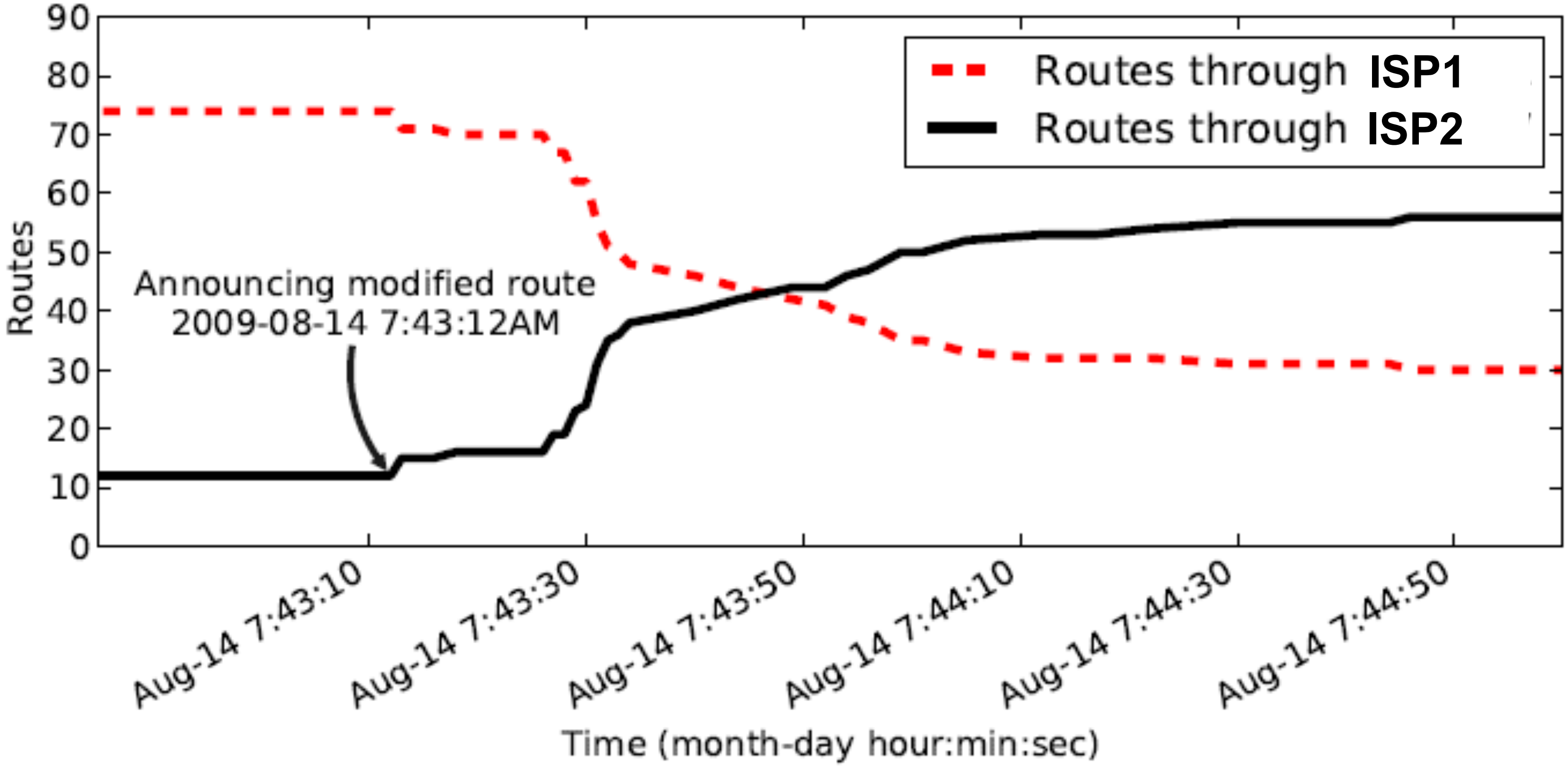
Basic Internet Routing with TP



- Initially, the service prefers ISP 1
- When ISP 1 has high jitter, the service can reroute through ISP 2 (not possible in today's cloud model)

Route Convergence Behavior

- ~60 seconds for Internet to converge to a new upstream



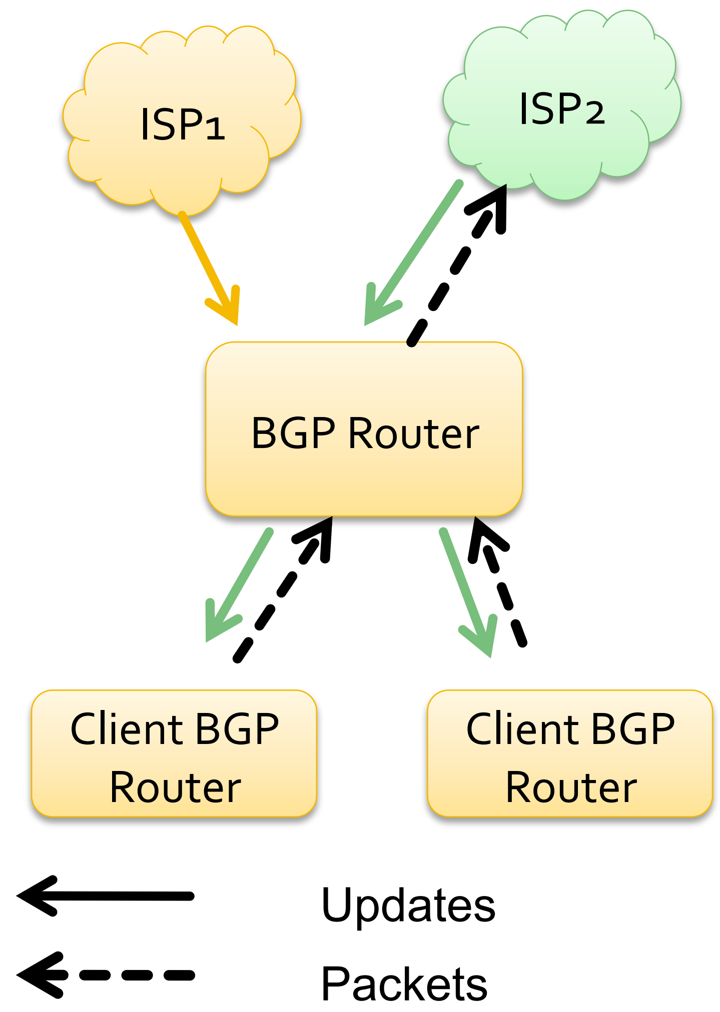
Convergence speed is the same as BGP today.

Scaling the Transit Portal: Goals

- Support dozens of sessions to upstream ISPs, hundreds of sessions to hosted services
- Provide the appearance of direct connectivity to each upstream ISP ("transparency")
- Prevent clients from introducing excessive routing instability, "leaking" routes, etc.

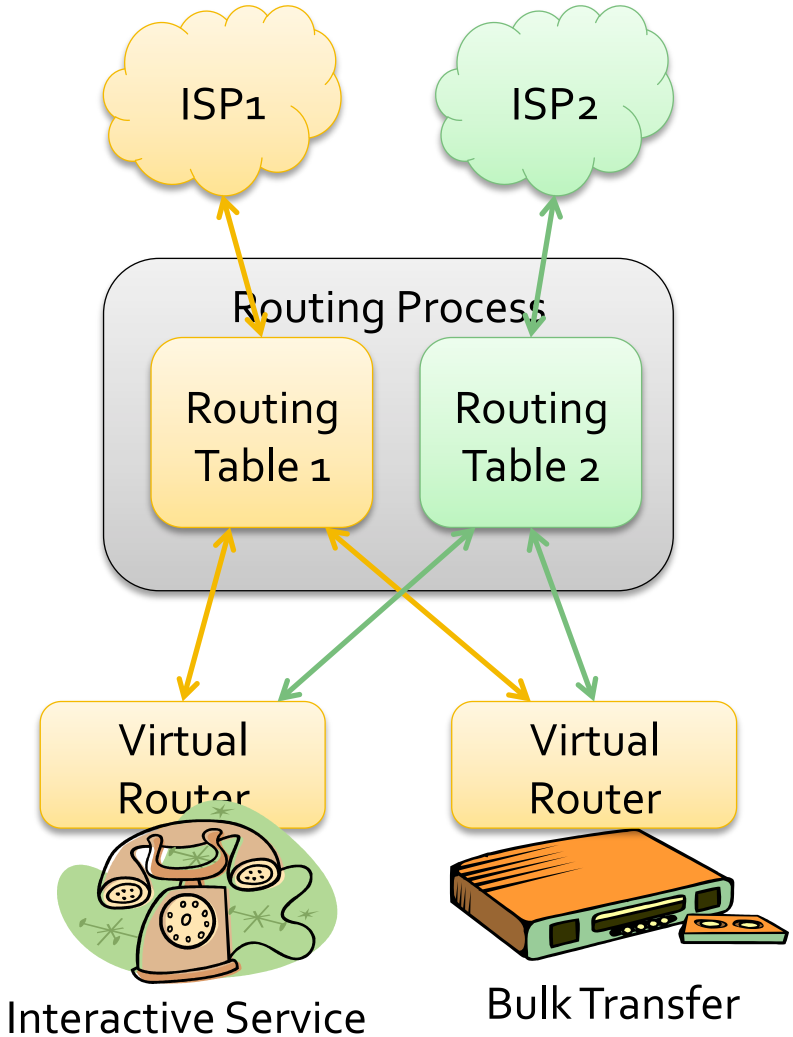
Conventional BGP Routing

- Conventional BGP router:
  - Receives routing updates from peers
  - Propagates routing update about one path only
  - Selects one path to forward packets
- **Scalable but not transparent or flexible**



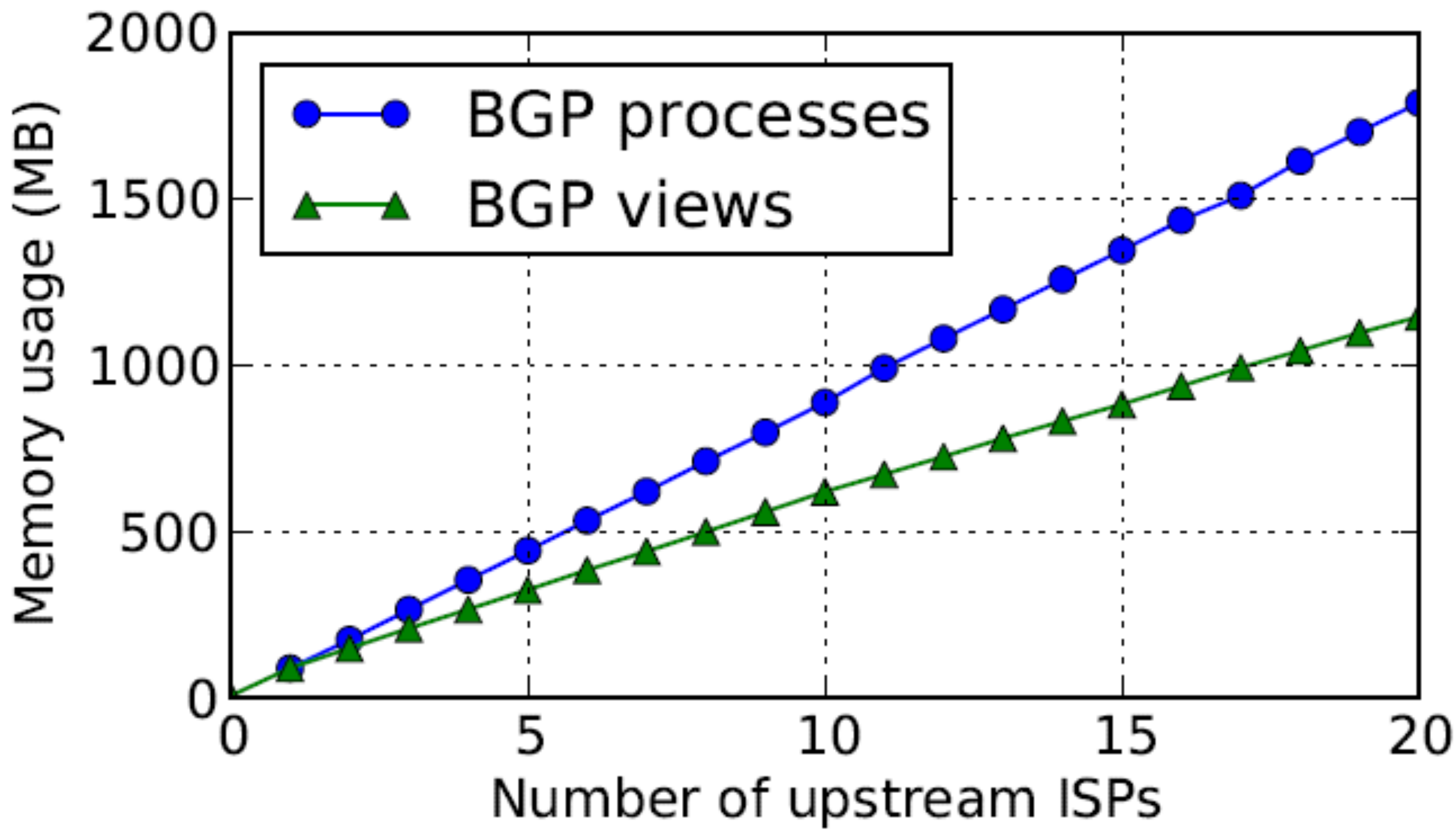
Scaling BGP Memory Use

- Quagga routing suite
- Store and propagate all BGP routes from ISPs
  - Isolated routing tables
  - Explosion of state
- Reduce memory consumption
  - Single routing process - shared data structures



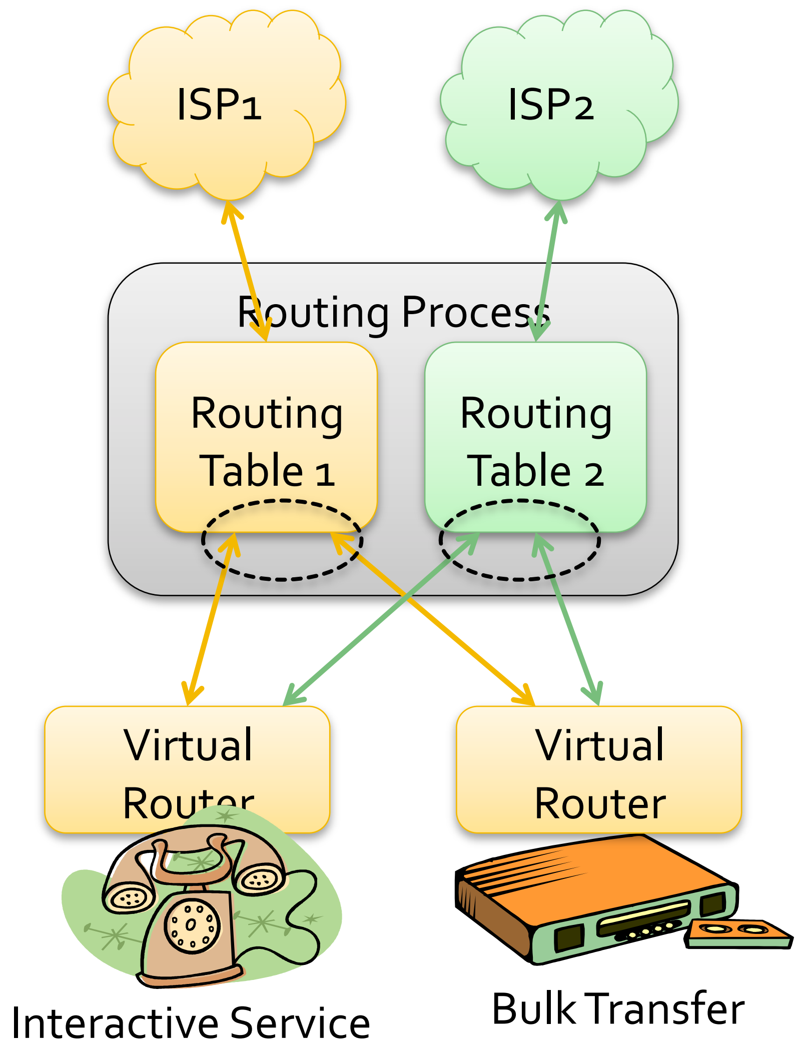
Memory Savings with Virtual Tables

Reduce memory use from 90 MB/ISP to 60 MB/ISP



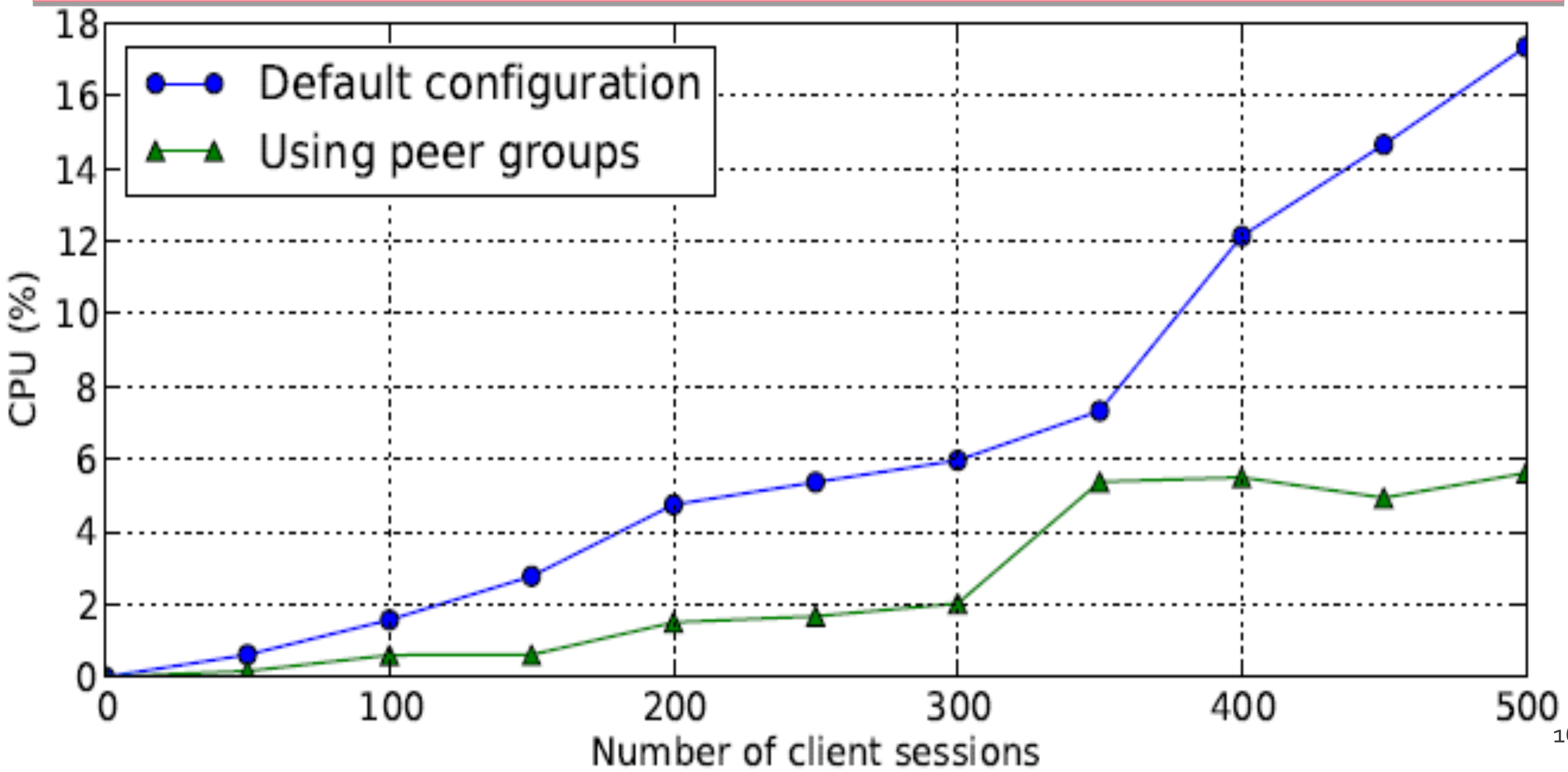
Scaling BGP CPU Use

- Hundreds of routing sessions to clients
  - High CPU load
- Schedule and send routing updates in bundles using **BGP Peer Groups**



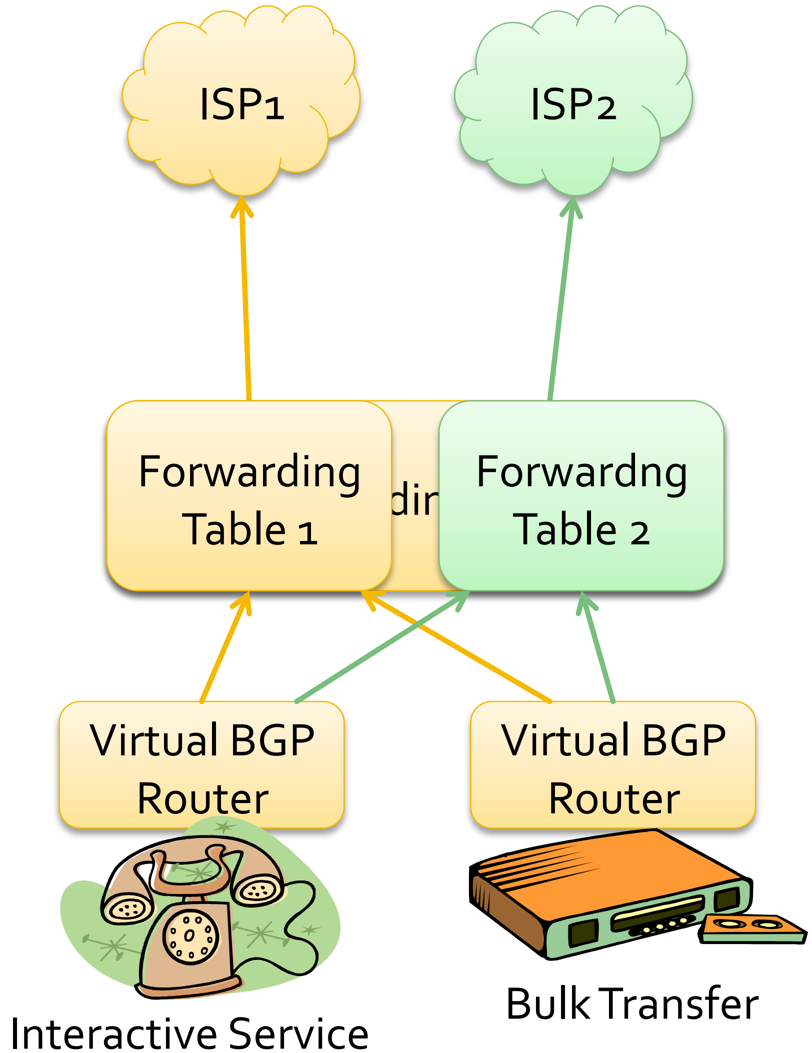
CPU Savings with Peer Groups

Using Peer Groups reduces CPU from 18% to 6% for 500 client sessions.



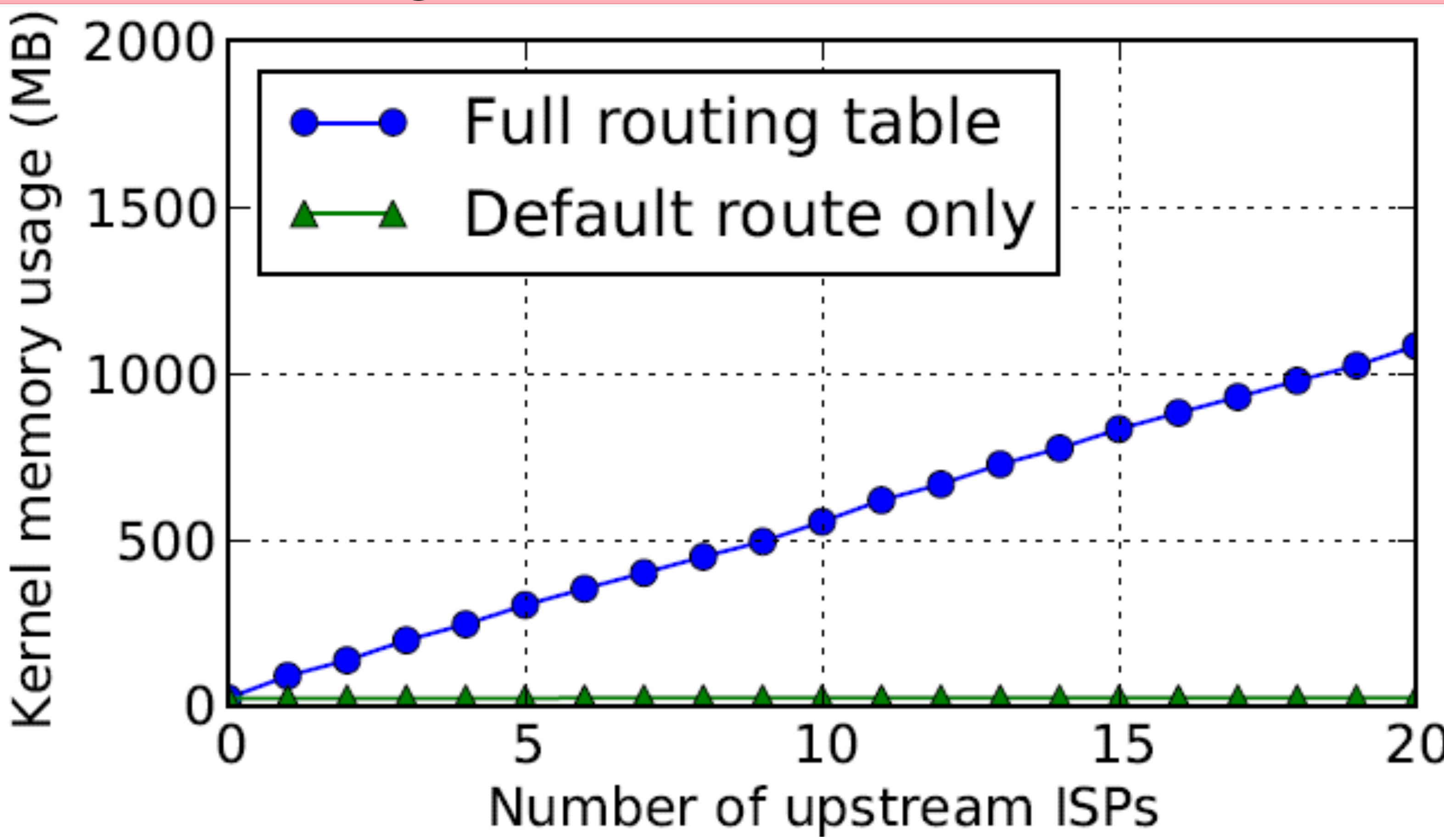
Scaling Forwarding Memory for TP

- Connecting clients
  - Tunneling and VLANs
- Curbing memory usage
  - Separate virtual routing tables with **default to upstream**



Memory Savings with Default Routing

Default routes reduce memory use from 50 MB/ISP to ~0.1 MB/ISP



Current TP Deployment

- Servers with customized routing software
  - 4GB RAM, 2x2.66GHz Xeon cores
- Three active sites with upstream ISPs
  - Atlanta, Madison, and Princeton
- Numbered resources:
  - AS 47065
  - IPv4 prefix: 168.62.16.0/21

Current/Future Experiments

- IP Anycast (Princeton University)
  - Performance measurements
- Next-Generation Networking class (Georgia Tech)
  - Hands-on BGP route control for undergrads
- BGP poisoning (University of Washington)
  - Topology inference

Future Work

- More deployment sites
- Making TP accessible for network research testbeds (e.g., GENI, CoreLab)
- Faster forwarding (NetFPGA, OpenFlow)
- Lightweight interface to route control

Conclusion

- **Problem:** Limited routing control for hosted services today

- **Our solution:** Transit Portal gives wide-area route control to services hosted in clouds
  - Scalable, open-source implementation based on Quagga
  - The deployment is real

http://www.gtnoise.net/tp

TP Applications: Education

- Used in a “Next-Generation Internet” Course at Georgia Tech in Spring 2010

- Students set up virtual networks and connect directly to TP via OpenVPN
  - Live feed of BGP routes
  - Routable IP addresses for in class topology inference and performance measurements

Current TP Deployment

- Three active sites
  - Princeton, NJ
  - Atlanta, GA
  - Madison, WI
- Internet numbered resources from ARIN
  - AS 47065
  - IPv4 prefix: 168.62.16.0/21