

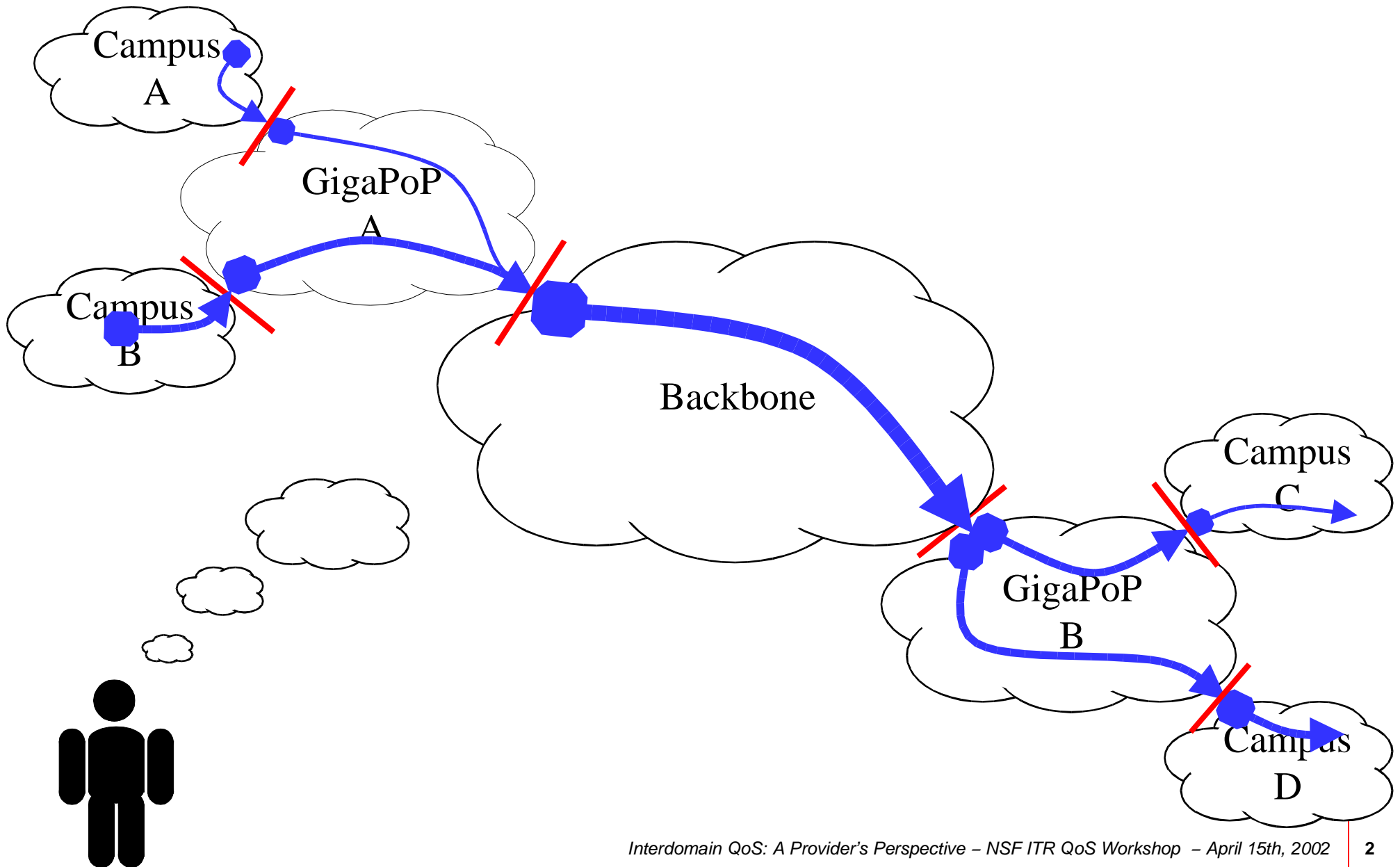


Interdomain QoS: A Provider's Perspective

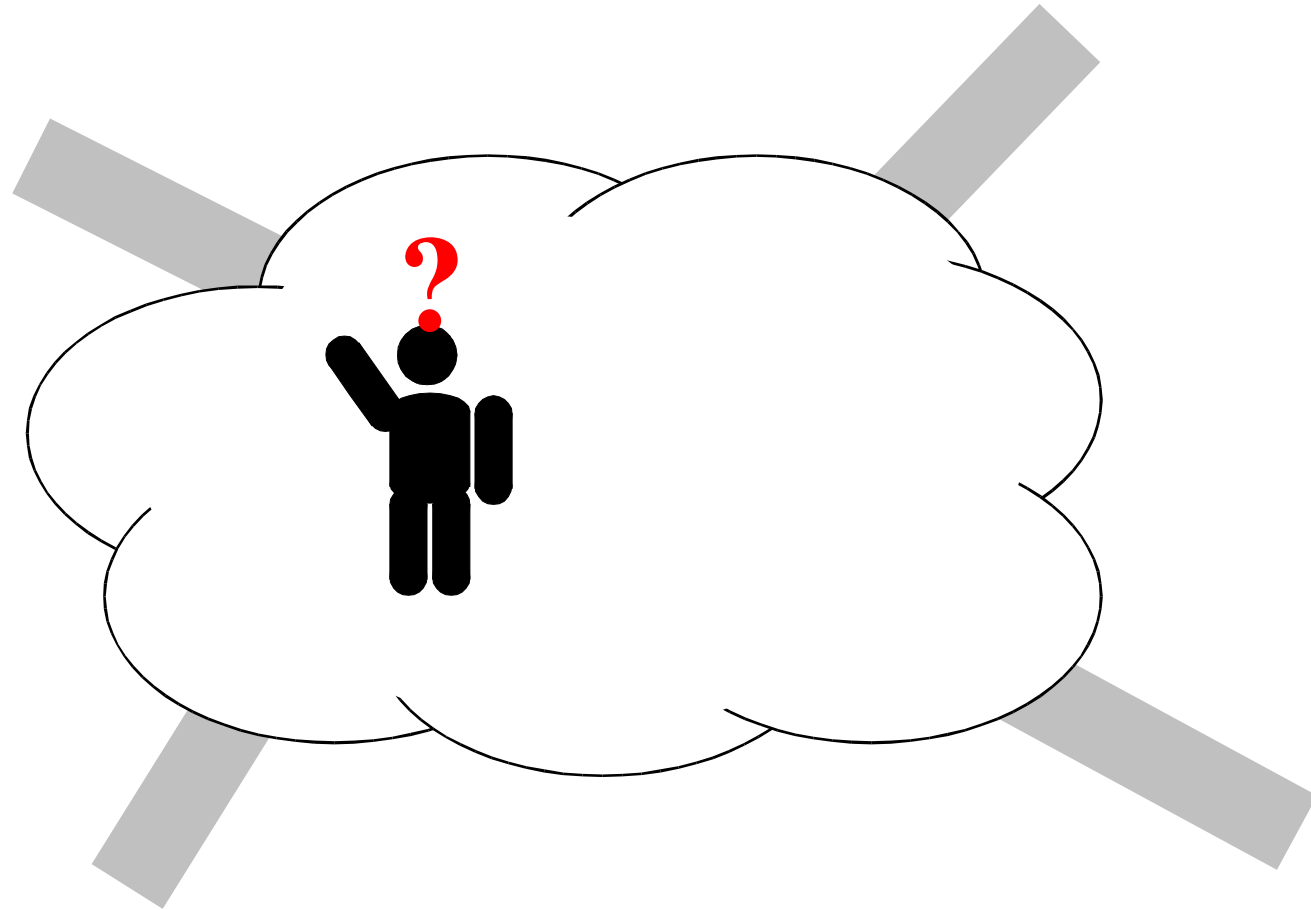
NSF ITR QoS Workshop
Annapolis, MD

Ben Teitelbaum <ben@internet2.edu>
April 15th, 2002

Visions of Circuits from 30 Kilofeet

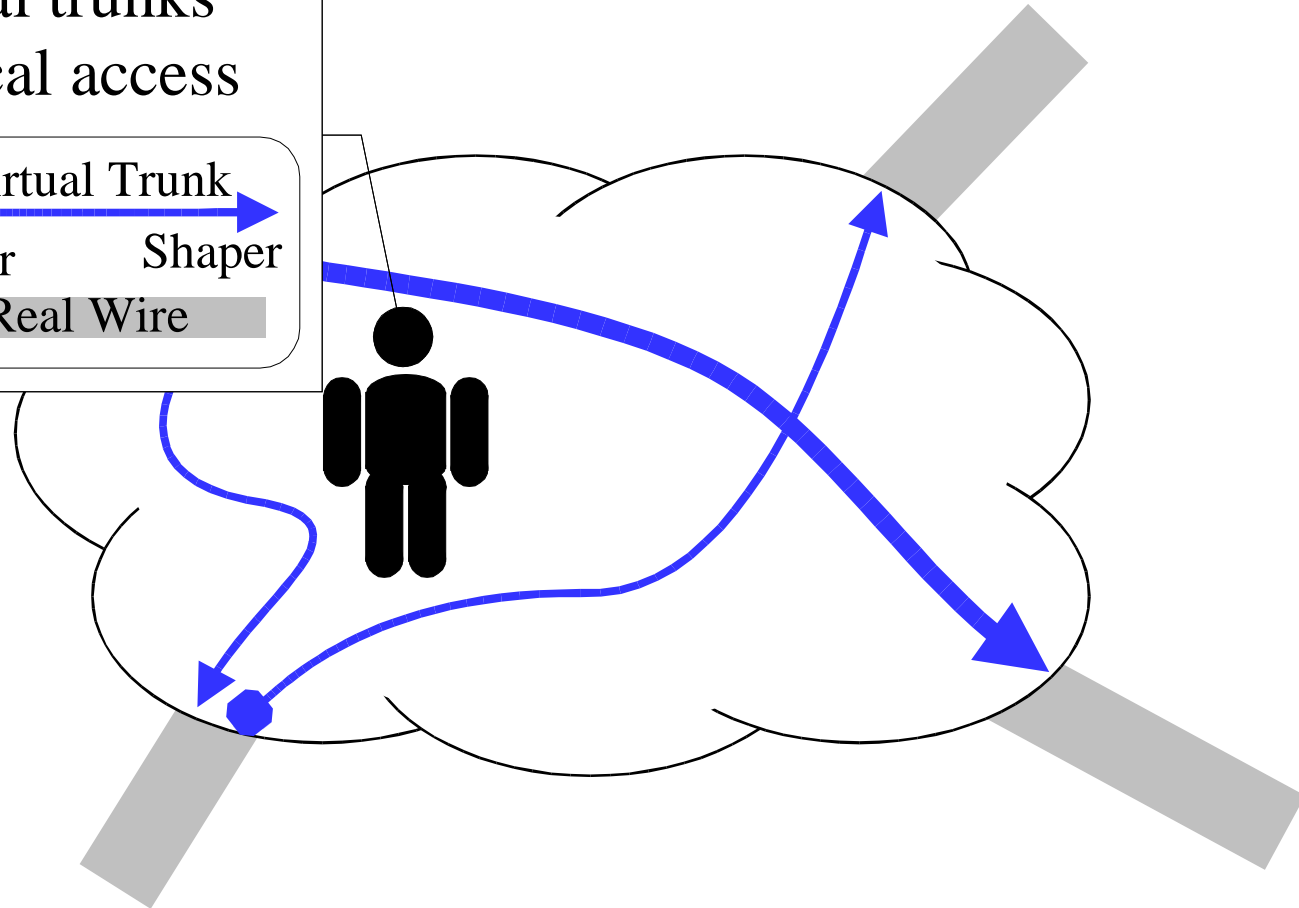
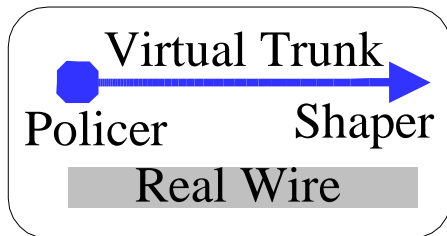


Provider's View at Ground Level₁



Provider's View at Ground Level₂

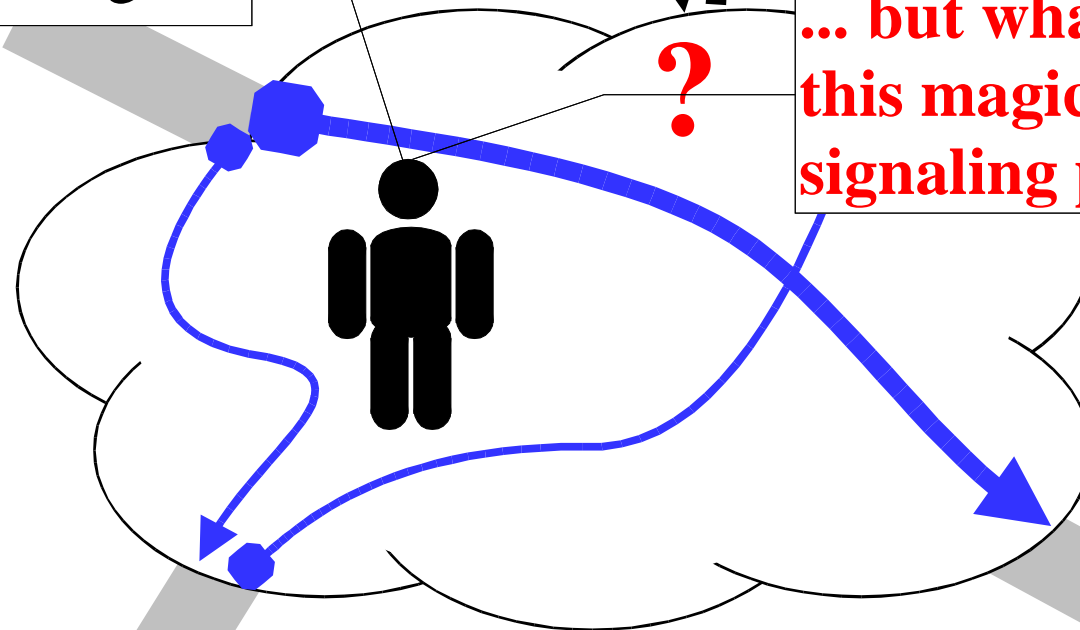
Makes sense to think in terms of "virtual trunks" between physical access interfaces



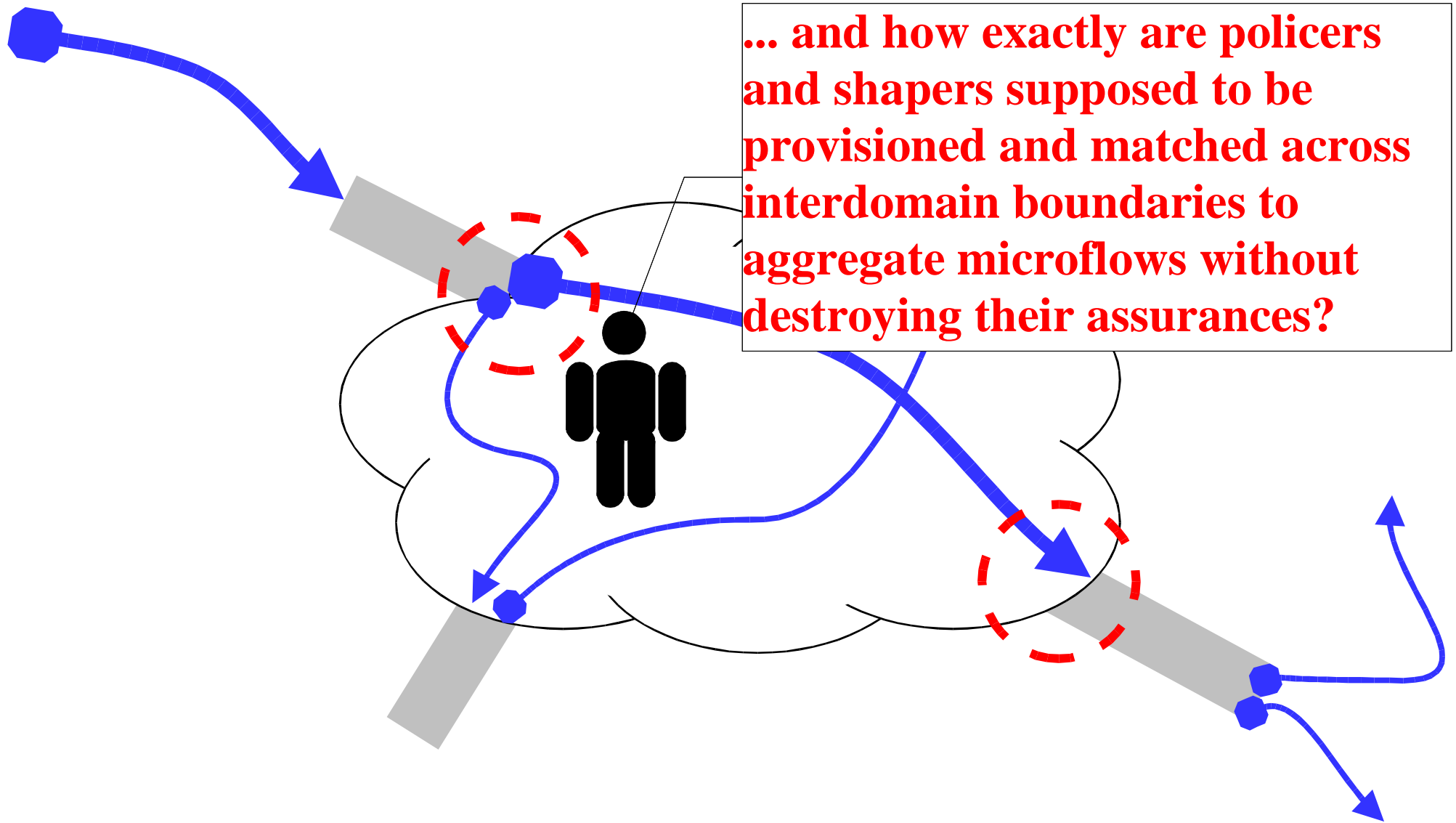
Provider's View at Ground Level₃

... and to participate in e2e signaling

... but what exactly is this magical interdomain signaling protocol?



Provider's View at Ground Level₄



Non-Architectural Concerns₁

DiffServ functionality still missing in modern routers or not available at line rate

- Route-based edge classification, anyone?
- How about multiple shaped aggregates within a PQ?

Per-net deployment granularity

- Must police EF traffic at every ingress interface

Service verification

- To "jiggle door" service provider or customer must launch EF DoS attack

Non-Architectural Concerns₂

Scary new business model

- Accounting, billing, etc.
- Complex new peering agreements with QoS SLAs
- More attractive to separate provider/customer service models to allow for statistical provisioning?

Scary new operational responsibilities

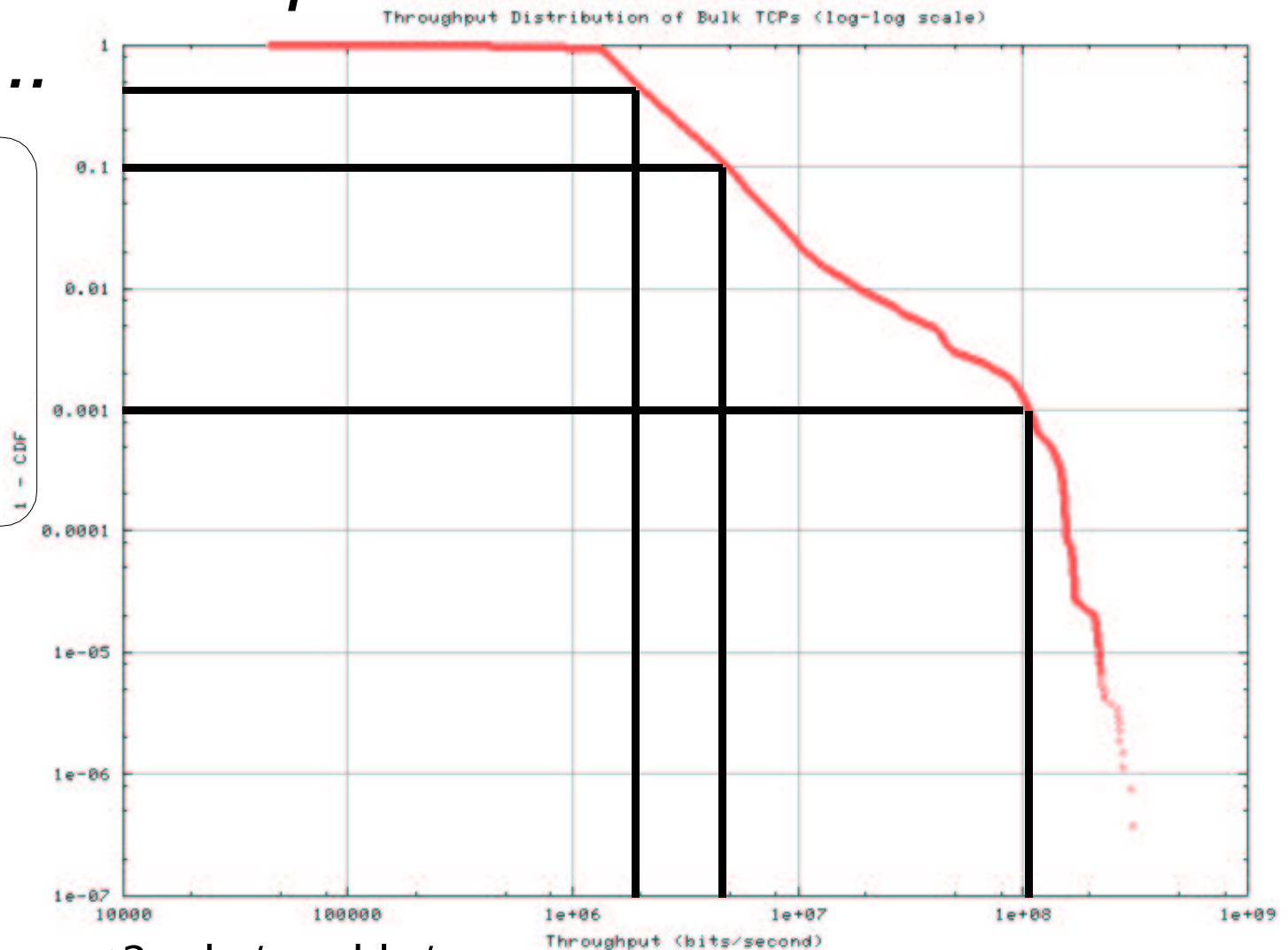
- Admissions control
- Increased vulnerability to DoS attacks

Finally, where's the demand that's going to make this all worth doing?!

“Low Demand: A Partial Explanation₁”

...typical Internet2 performance ain't what it should be...

2.6M bulk TCPs
week of 20020325
Observed x-puts:
50% 1.870M
90% 4.898M
99.9% 106.3M



- <http://netflow.internet2.edu/weekly/>

“Low Demand: A Partial Explanation₂

Evidence suggests that most performance problems are non-congestive and are in or near hosts

- Broken TCP stacks (e.g. inadequate socket buffering, no window scaling)
- Ethernet duplex mismatch
- Crummy cabling (e.g. CAT3, shared, or damaged)

But users are blissfully ignorant!

- In part because of low penetration of interactive multimedia applications

Where Does This Leave Us?

We have "suspended" our Premium efforts

Currently working to fix common e2e performance faults and raise user expectations

Also, pushing VoIP to the masses

As for QoS, we are going with the theory that less is more¹

1. More deployment anyway!

“Enter, Non–Elevated Services

“Worse”

- QBone Scavenger Service (QBSS)
- Bulk Handling PDB (B. Carpenter, K. Nichols)

“Different–but–equal”

- Alternative Best Effort (ABE)
- Best–effort Differentiated Services (BEDS)

Why do we like these wacky services?!

- Require no policing, admissions, settlement, etc.
- Deploy incrementally at the granularity of single interfaces
- Consistent with end–to–end principle

QBone Scavenger Service

Basic idea

- **Voluntary** marking hints to network that degraded service is OK (like Un*x `nice` for the network)
- Scavenger traffic **may** be degraded at congestion points
- **Think:** thin, bottom-feeding best-effort network that can expand to full capacity in absence of congestion
- Formal service definition:
`http://qbone.internet2.edu/qbss/qbss-definition.txt`

Goals

- A tool to preserve/extend uncongested BE experience for interactive applications

Alternative Best Effort (ABS)

Monolithic best-effort service class split into:

- **Blue** –lower loss / higher delay
- **Green** –higher loss / lower delay

Fairness relationship between classes

Each app knows its utility function and trades off loss for delay accordingly

Could we do an ABE-like low-delay class today (e.g. with WFQ and RED)?

<http://www.abeservice.com/>

But What to Tell Those Developers Who Still Think They Need Circuits?

First, we flame them...

- "Listen amigo, this is the Internet. It's fast, cheap, and global, but there are no guarantees. You should be grateful for what you can get and ask not what the network can do for you, but what you can do on the end-systems to make your application work better."

Then, we try to encourage them to think about how they might use QBSS & ABE

Finally, we try to disseminate best-practices for writing adaptive apps and for enabling those apps across a BE cloud



INTERNETTM