eFIT: enabling Future Innovations through Transit wire

Dan Massey, Lan Wang, Beichuan Zhang, Lixia Zhang

CSU, U.Memphis, U.Arizona, UCLA

Routing Research Group Meeting

March 17, 2007

This set of slides has a few graphs at the end showing IP address allocation and our measurement results on the BGP talbel growth & prefix fragmentation (extracted from my IAB tech chat in September'05)

A High Level View

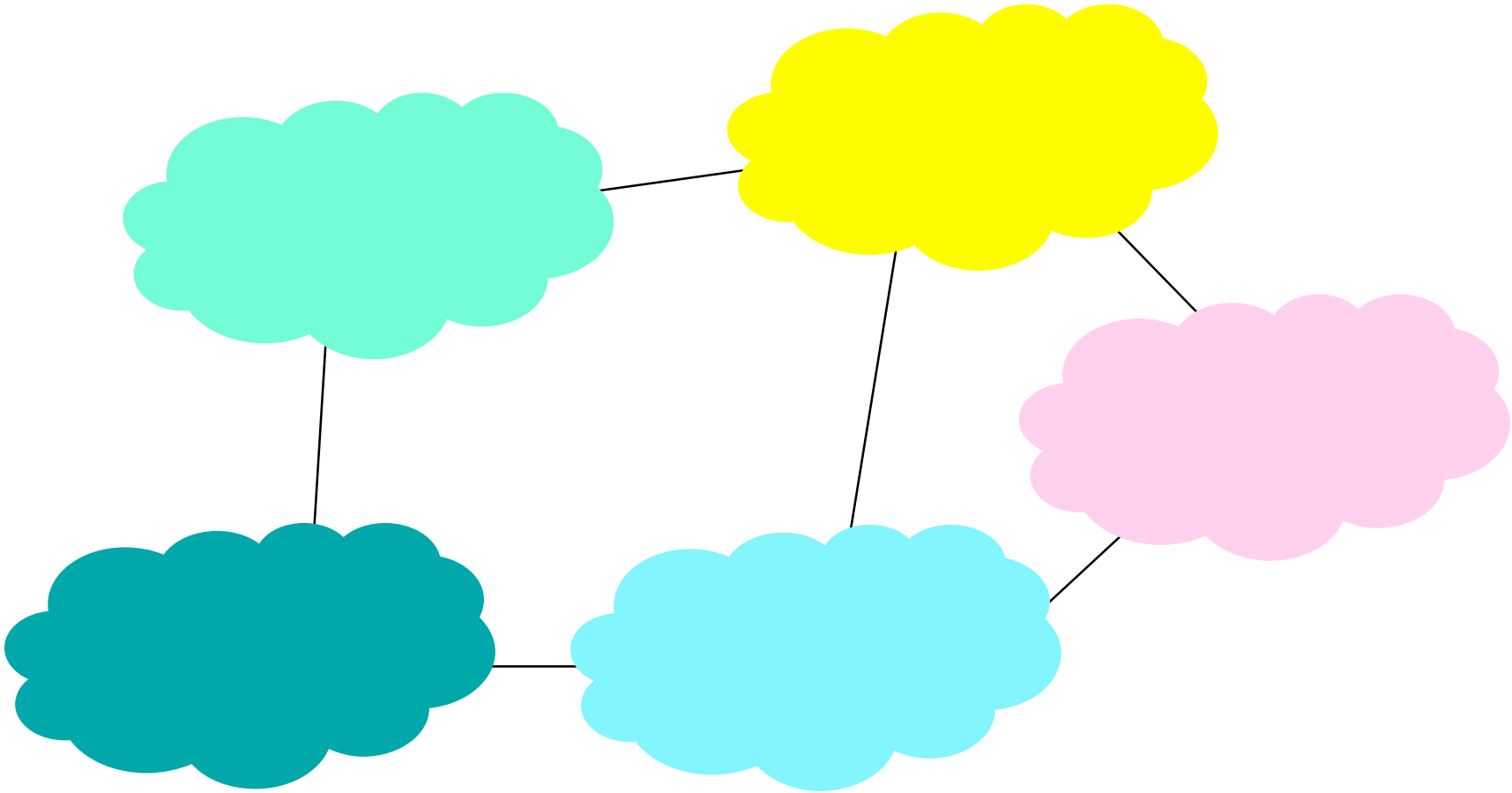
- Take a long term view of the solution space
  - Focus first on key ideas, then turn to incremental deployment challenges
- Key Idea: put ISPs and users in *separate* IP address space
  - A number of people independently came to this solution direction towards scalable routing
  - Identify synergy and join effort in solution development

This talk: focus on 2 points

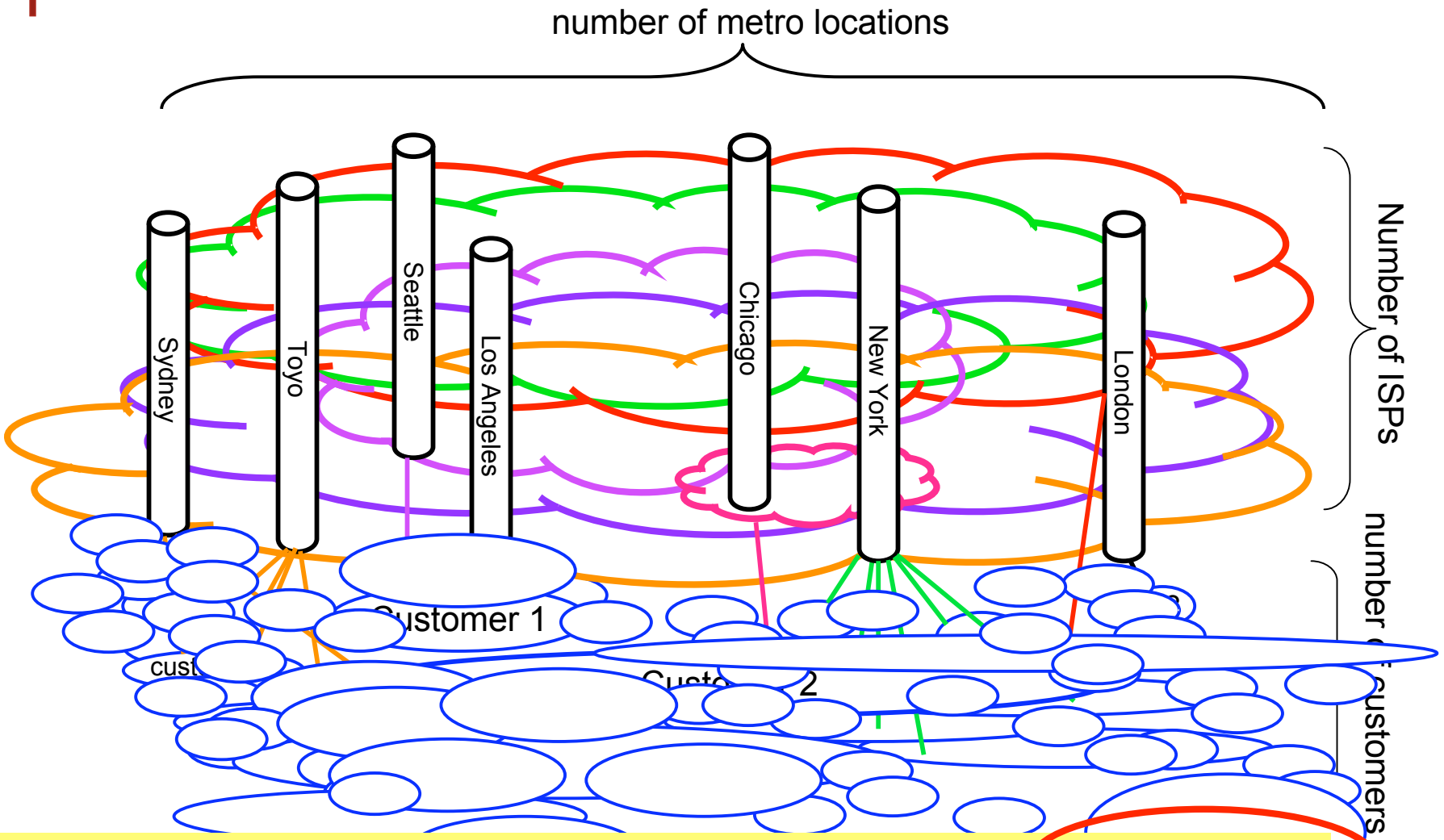
1. Terminology clarification
   - Locators, identifiers, addresses
   - Exactly what are we separating from what?
2. Proposed design of provider address structure

Why we have a routing scalability problem

When we draw network graphs, it tends to look like this



But in reality, it is more like this



DFZ Routing table size = Function(# of ISPs X # of PoPs X # of user sites)

Tensions between user sites and providers

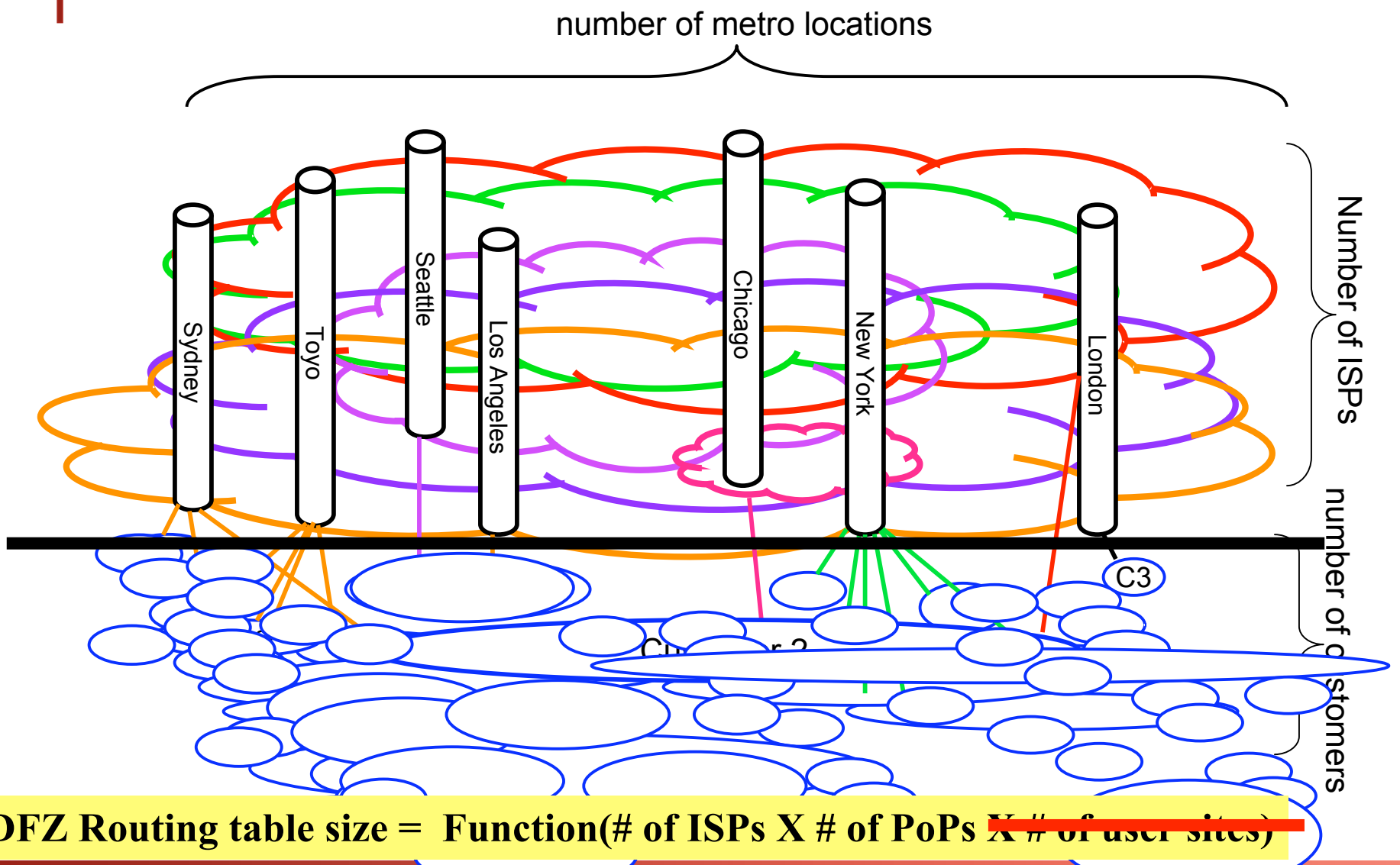
- User sites want Provider Independent (PI) prefixes
  - Nearly all sites want multihoming
  - no site desires renumbering
- Providers want provider-based addressing to scale

⇒ Head-on conflict

  - an address can't simultaneously be both PI and not-PI

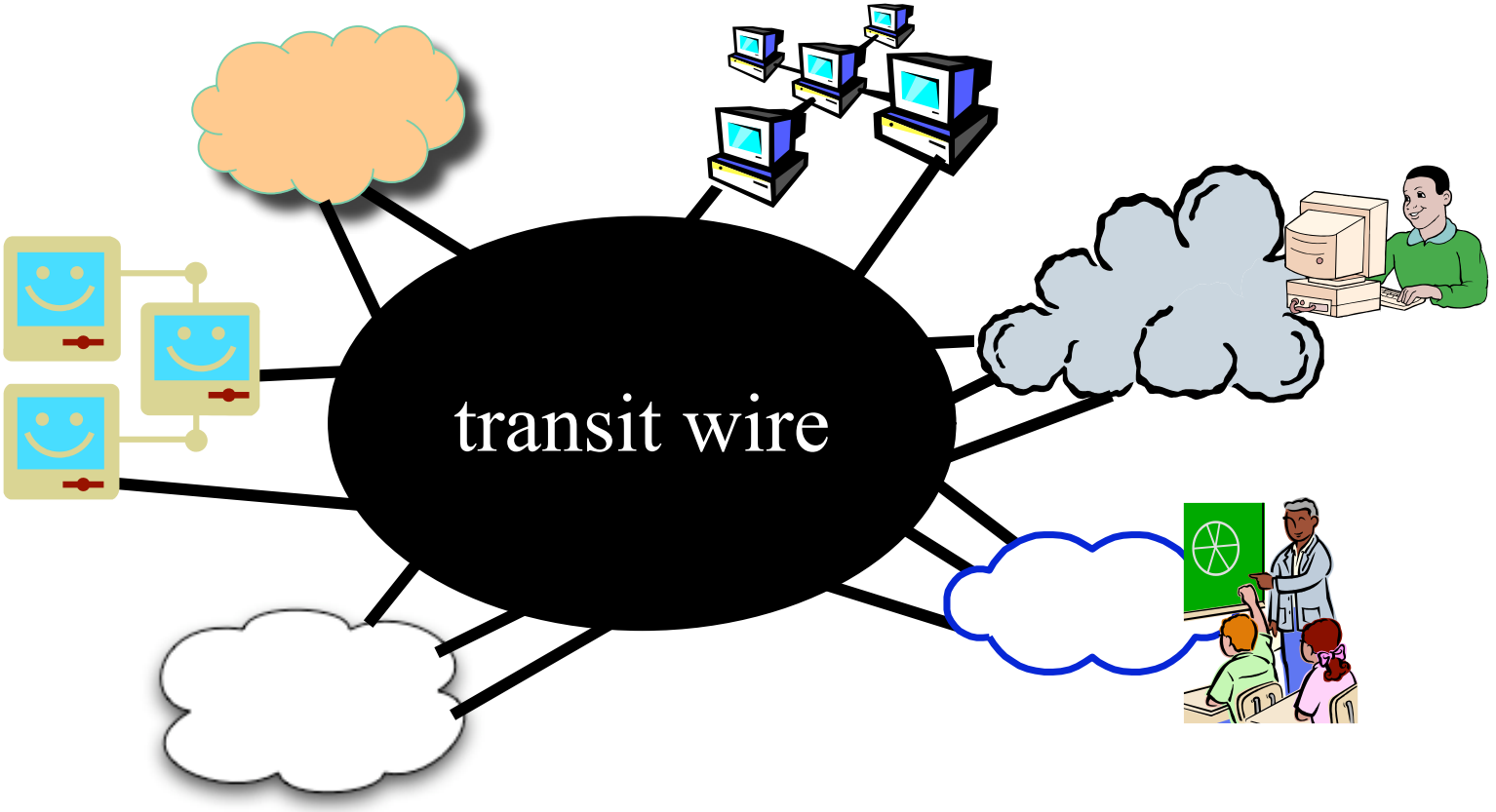
⇒ ***ISPs are losing the battle*** over topologically aggregatable prefixes

Proposed solution: separation



DFZ Routing table size = Function(# of ISPs X # of PoPs ~~X # of user sites~~)

user's view: universal connectivity through transit wire



- Restore E2E connectivity model (if/when edges get global addresses)
- Enable core to evolve independently from edges

Draft minutes
6th discussion on IP addressing architecture
Thu 6/15/95

Participants: Clark, Deering, Postel, Yakov, Zhang (absent: Ford)

Clark: "There are clearly *two classes of network entities*, subscribers and providers; there may be a gray area but that is not important.

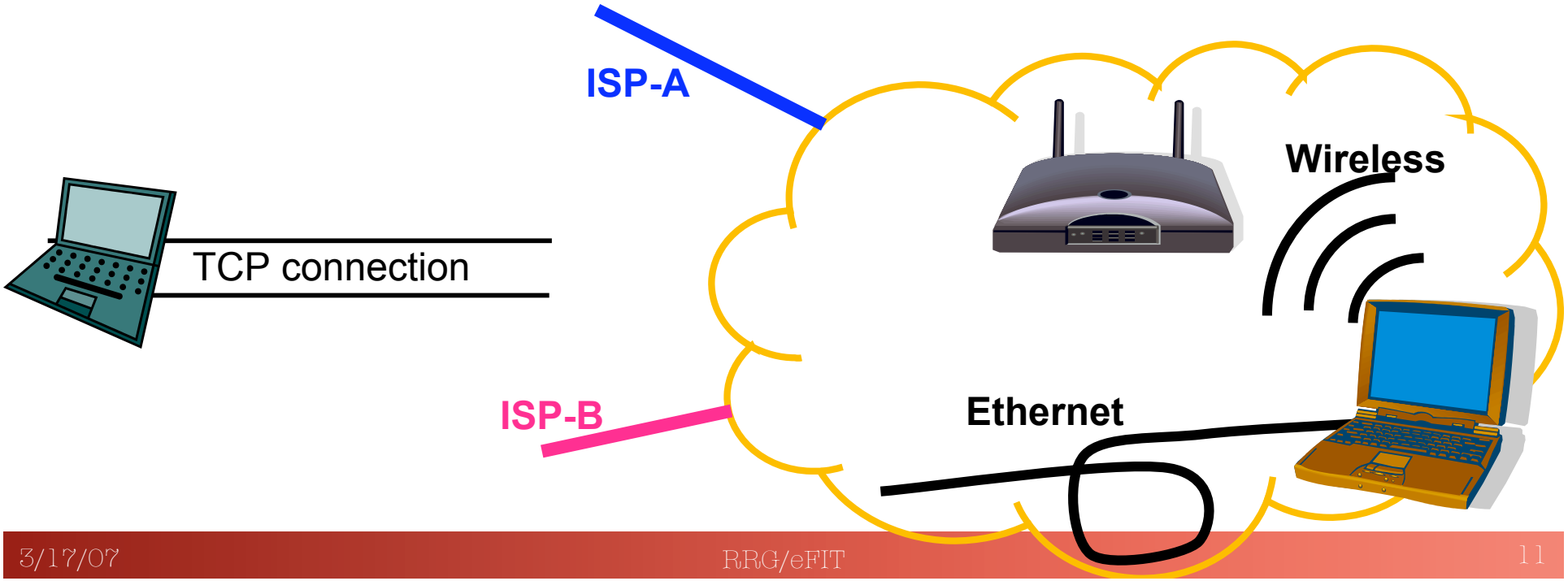
- "As the Internet gets bigger and bigger, *we can no longer make the assumption that subscriber addresses are globally routable,* therefore they cannot escape without having the provider part attached to it.
- "The idea is to let those people who are in the business of being internet providers do flat routing among themselves."

Terminology clarification

- What we've shown: need for separating providers and users into separate address space
- Is this “loc/ID split” ?
- Exactly
  - How many “things” out there, and
  - what needs to be separated from what?

Need for a different sepration:

- TCP user IP address as part of connection identifier
- Changing paths ⇒ breaking TCP connection
  - Either provider path (if PA address), or host interface



Terminology clarification

- Providers: want topologically aggregatible prefixes
- Users: want provider-independent address blocks
- TCP: want unique end point identifiers

To scale DFZ routing: separate these two

To make TCP conn. survive change of delivery path: separate IP addr and end idnetifiers

Towards scalable inter-domain routing

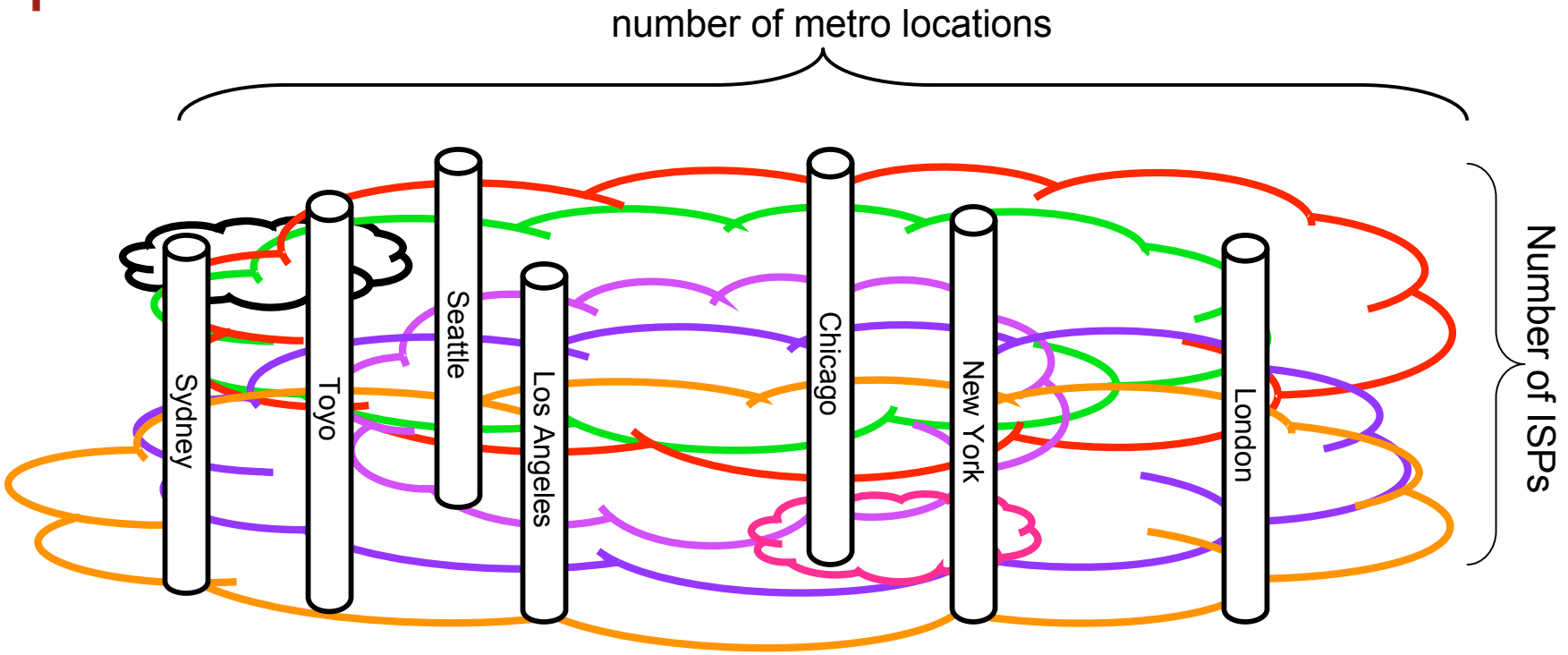
Idea 1: Divide up address space into 2 parts

- Customers, transit service providers
  - Customers generate and receive data
  - Providers delivery data to destination networks

Idea 2: Design a new provider address format

- To facilitate routing policies (routing of $$$)
- To support traffic engineering
- To scale with growing, multihomed user sites

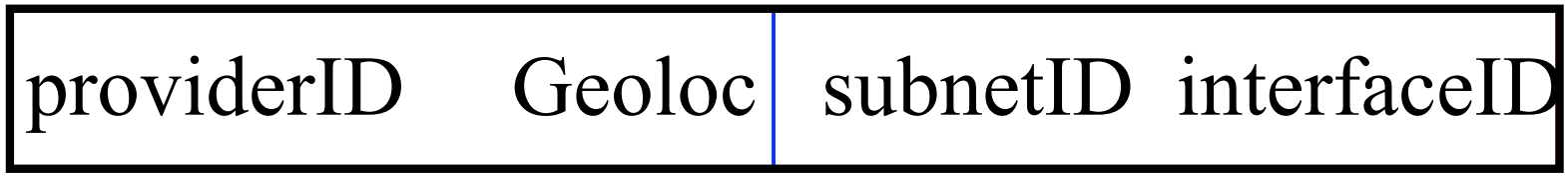
What To Carry in Provider Addresses



After moving users out of the picture, what values pinpoint a location in this mesh?

- Which provider
- Which location

eFIT provider address format



What's in address structure today

- ProviderID
  - Necessary information to make "*route money*" easier
  - Help reduce false routing announcements
- GeLoc
  - Useful info for traffic-engineering and multipath routing
- Support routing *aggregation at any desired granularity*

Traffic engineering

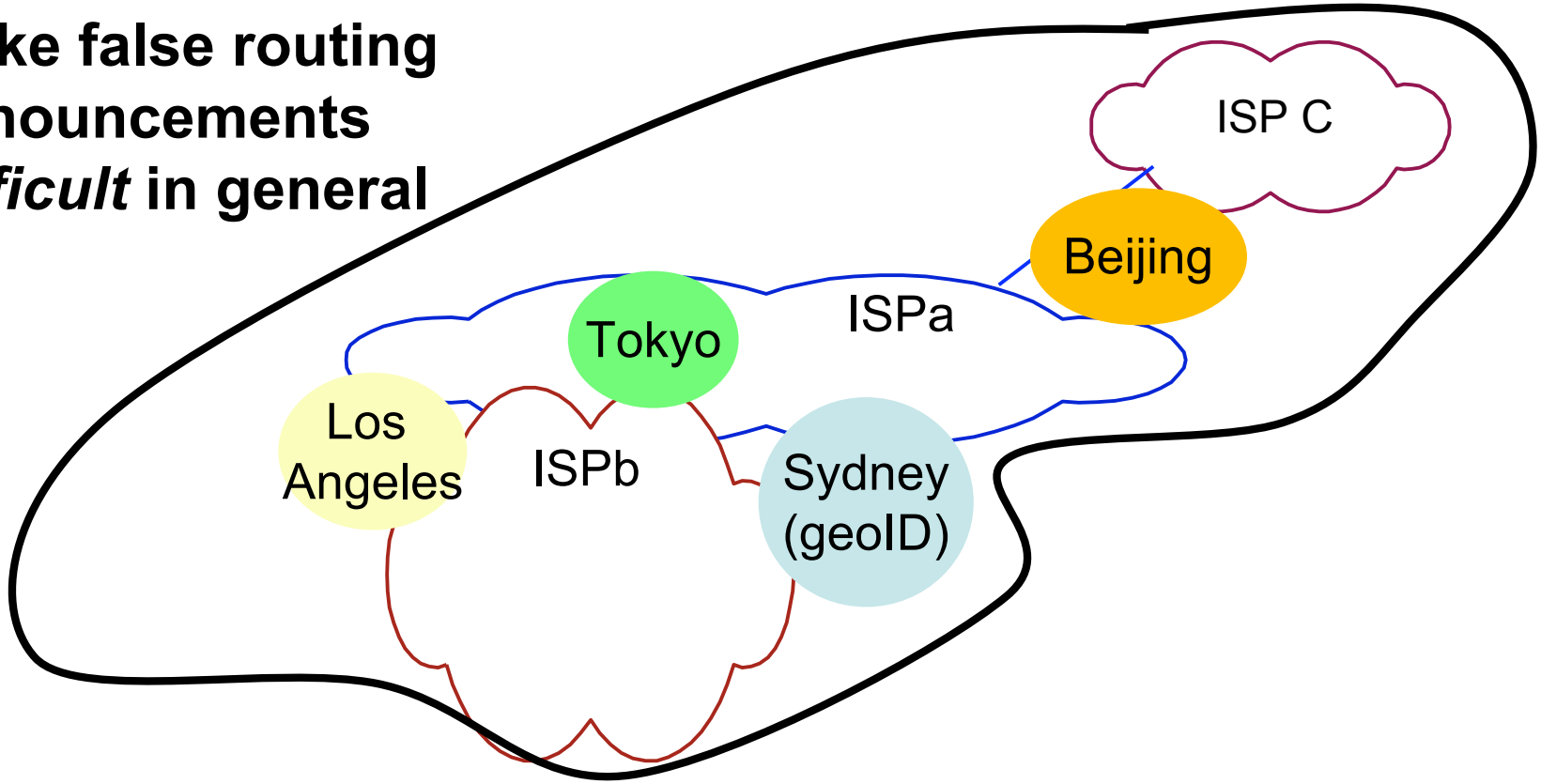
- Current practice: steering traffic by splitting prefixes
  - Whoever doing the split: a simple, effective approach
  - Whoever not benefitting from the split: bearing the cost of increased RIB/FIB size
- Scalable TE support: being able to re-aggregate effectively

How this new address structure helps

- Currently aggregation is risky at best
  - No information about whether prefix shares a common provider or common location
- We propose the new address structure to have fixed boundaries between subfields, to enable aggregation at any desired level

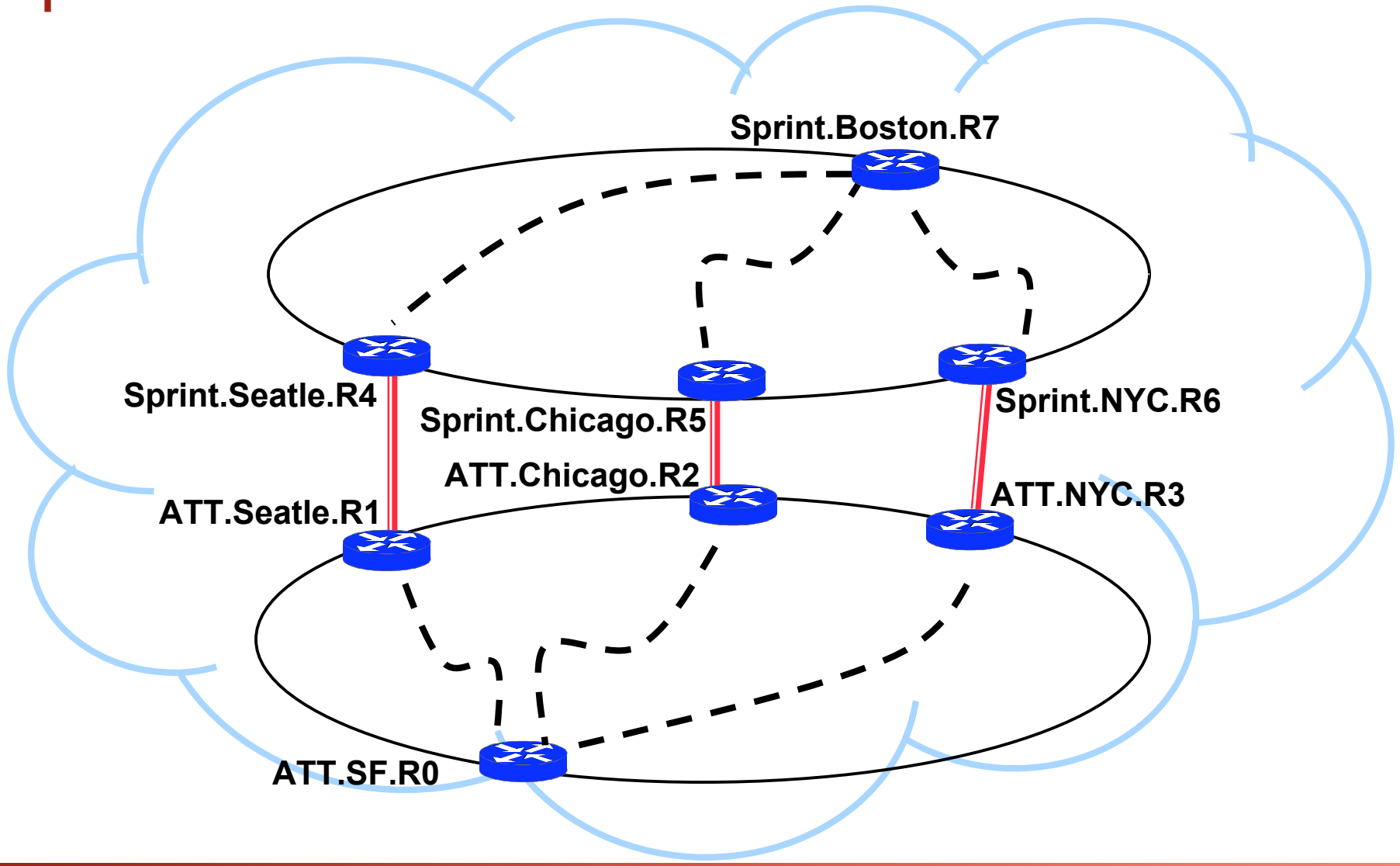
How this new address structure helps

Make false routing announcements *difficult* in general



providerID	GeLoc	subnetID	interfaceID

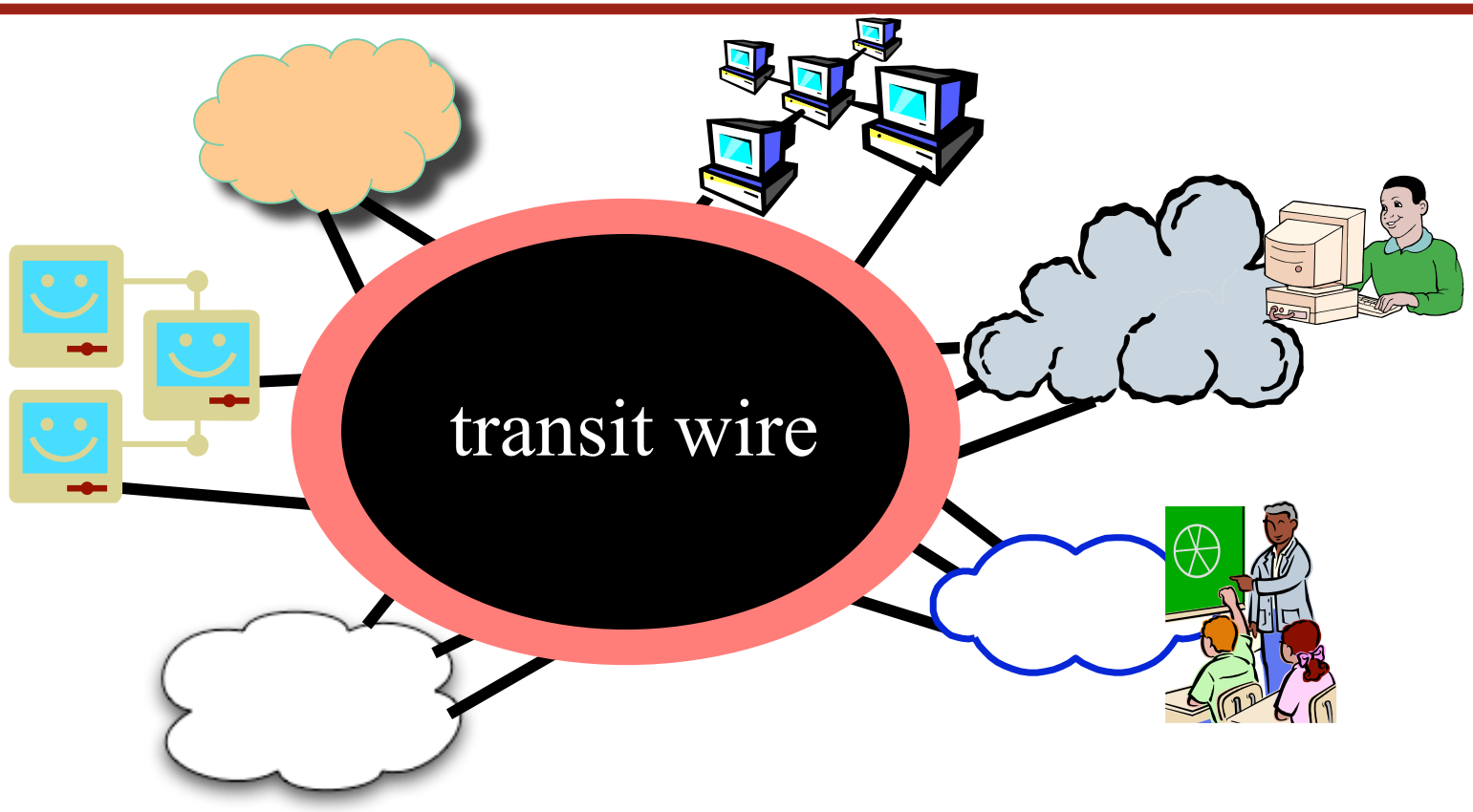
How this new address structure helps



Mapping Customers to Providers

- Customers appear to be directly connected to each other
- Reality is each customer connects to providers
  - Destination customer address mapped into provider address
  - Tunnel packet across core to provider address
  - Unpack the packet and deliver to customer network
- An essential part of any 2 address space is the mapping that links the two spaces together.
  - Mapping service design may vary, but some mapping needed to connect customer space to provider space.
  - We see other ***advantages*** in the mapping service….

A critical component: a mapping layer



insulate edges and core

- a cushion to hide core's inability of adopting edge changes instantly (or ever)
- A layer to add necessary functions that edges unable to do themselves

Some Broad Design Challenges

- Given new addresses space, design most effective routing inside the transit core
- Address heterogeneity and resiliency
- *Build the mapping service*
  - Both a challenge and a blessing: *One level of indirection can solve all the problems*
  - Currently sketching out initial designs, evaluating tradeoffs of different approaches
- Pop up a level: why adding this mapping compoment makes a worthwhile tradeoff

First, Why Change Anything?

- Why is it *necessary* to change the existing architecture?

  "Internet achieved unprecedented success without making the distinction between users and providers, don't change it"

"being the right size" by J. B. S. Haldane, 1928

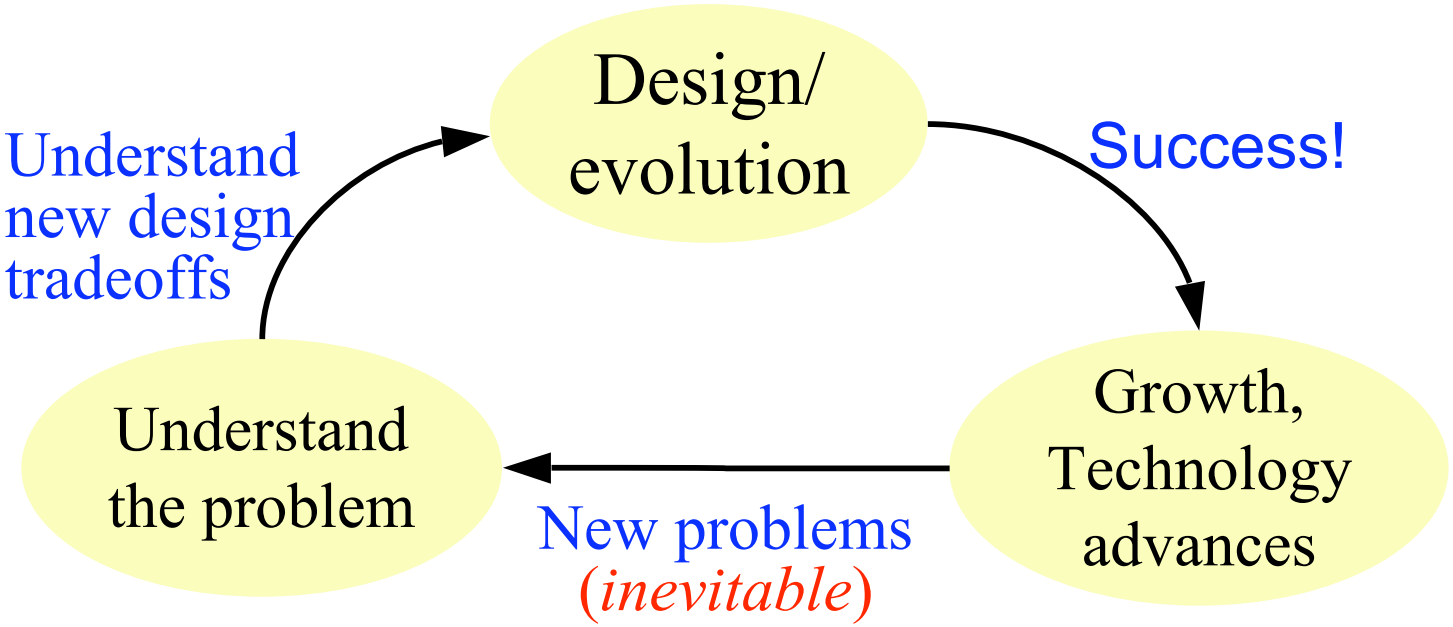
- "A typical small animal, say a microscopic worm or rotifer, has a smooth skin through which all the oxygen it requires can soak in.
- "Increase its dimensions tenfold in every direction, and its weight is increased a thousand times, so that if it is to use its muscles as efficiently as its miniature counterpart, it will need a thousand times as much food and oxygen per day

Change in size ⇒ change in form

- "Now if its shape is unaltered its surface will be increased only a hundredfold, and ten times as much oxygen must enter per minute through each square millimetre of skin...
- "*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*."
- It does not make sense for small insects to have lungs--impossible, on the other hand it is impossible for big animals to live without a lung
- Same story for Internet: probably did not make sense to have the complexity of mapping 2 spaces, but now the user base is big enough so that it become infeasible to have everyone live on the same address space

Necessary System Evolution

- All new systems start small
- Success ⇒ growing large ⇒ changed requirements
- Go through the evolution cycle (or otherwise)



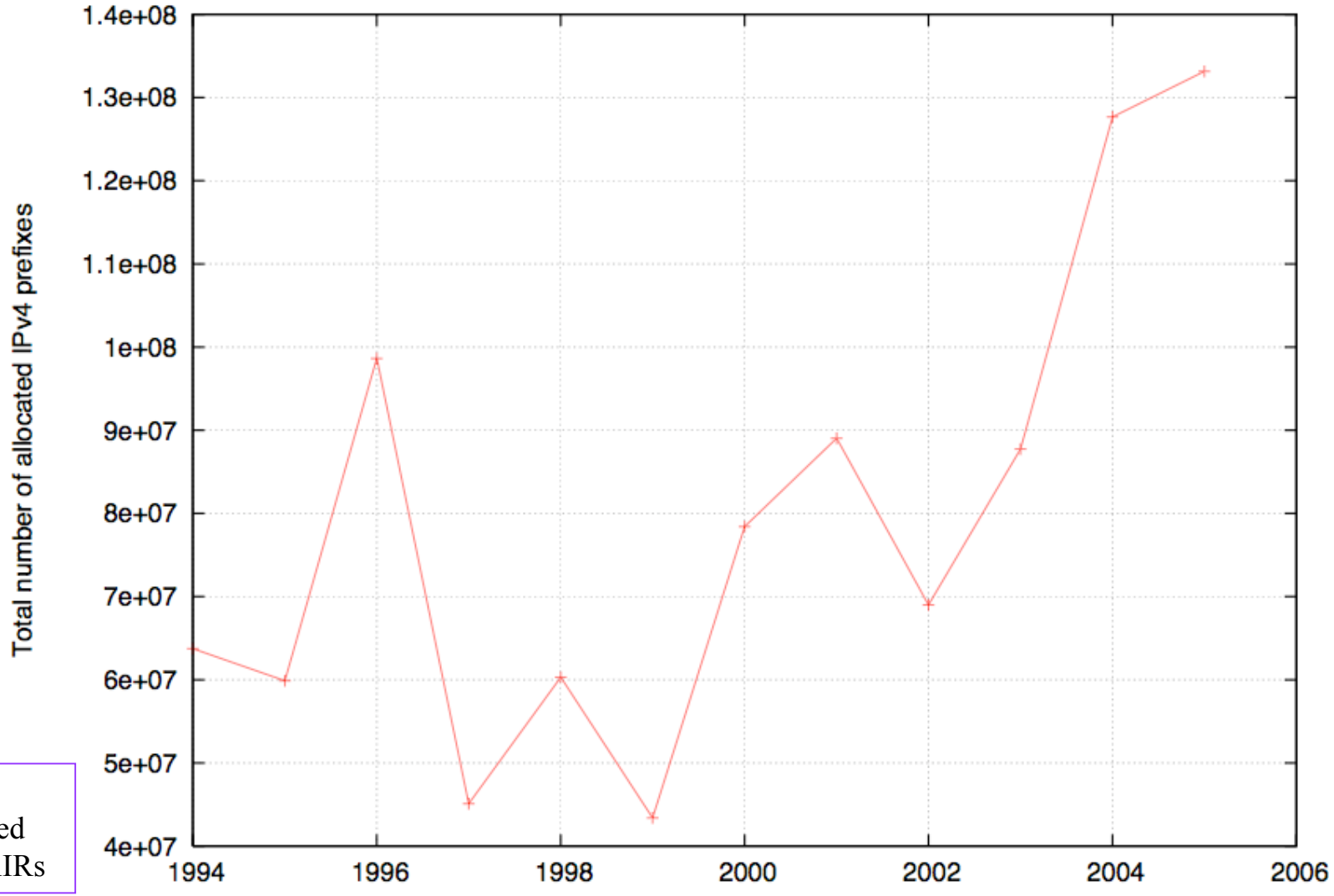
A few departing words

- A large change in size *necessarily* leads to a change of form
- Important *not* to change existing practice
  - Separation allows user sites to continue existing practices and no major change required
  - Separation allows providers to introduce new address structre and address challenges that came from change in Internet size
- A center piece in routing system design: the *address architecture*
  - *Get the address right and the rest can follow*

References to some of our measurement results about BGP table growth and update dynamics

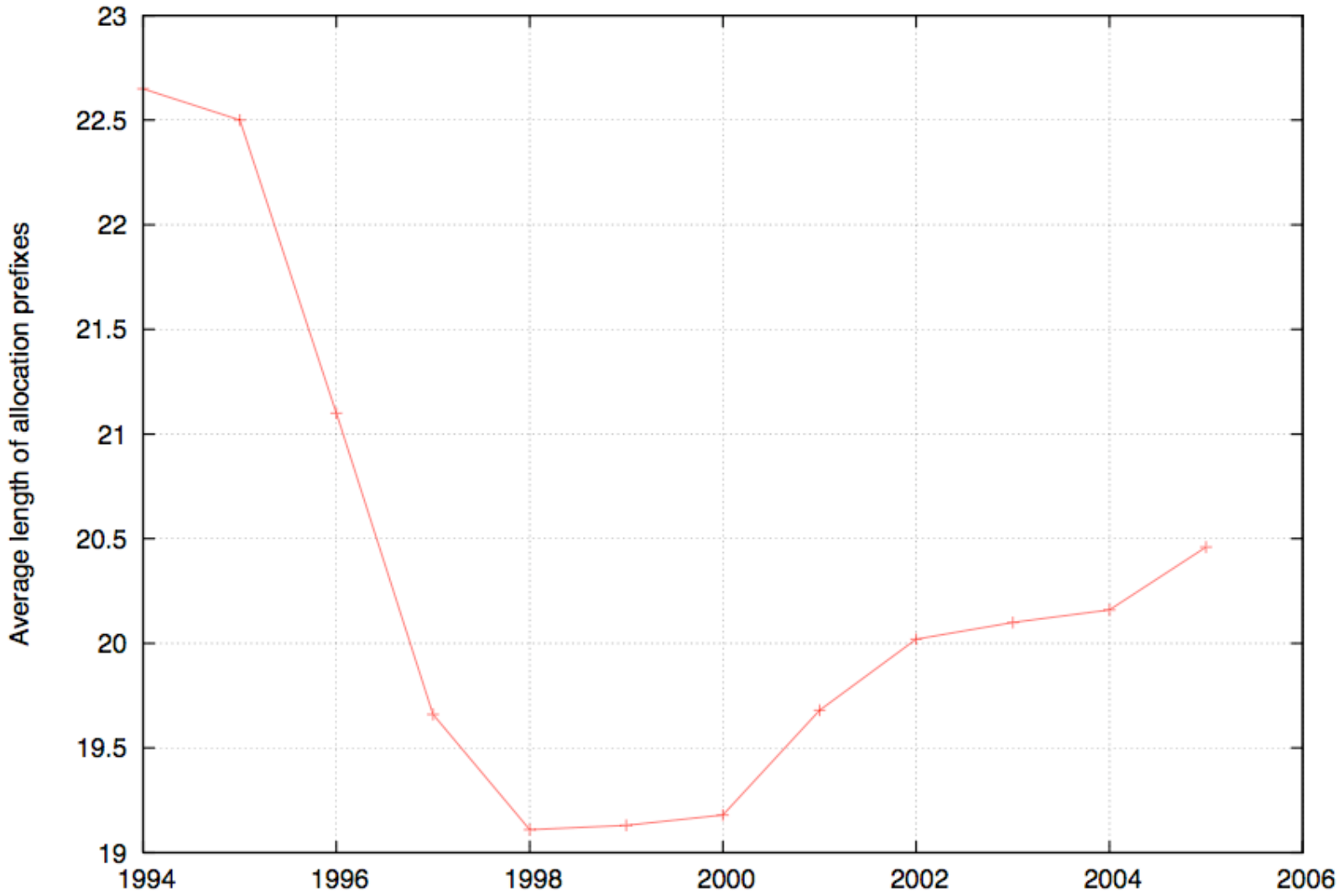
- *"IPv4 Address Allocation and the Evolution of the BGP Routing Table"*
  - ftp://ftp.cs.ucla.edu/tech-report/2003-reports/030009.pdf
- *"An Analysis of BGP Routing Table Evolution"*
  - ftp://ftp.cs.ucla.edu/tech-report/2003-reports/030046.pdf
- *"IPv4 Address Allocation and BGP Routing Table Evolution"*
  - ACM Computer Comm. Review, January 2005, www.cs.arizona.edu/~bzhang/paper/05-ccr-address.pdf
- *"Measurement of Highly Active Prefixes in BGP"*
  - GLOBCOM 2005, www.cs.ucla.edu/~rveloso/papers/activity.pdf

Addresses Allocated Per Year



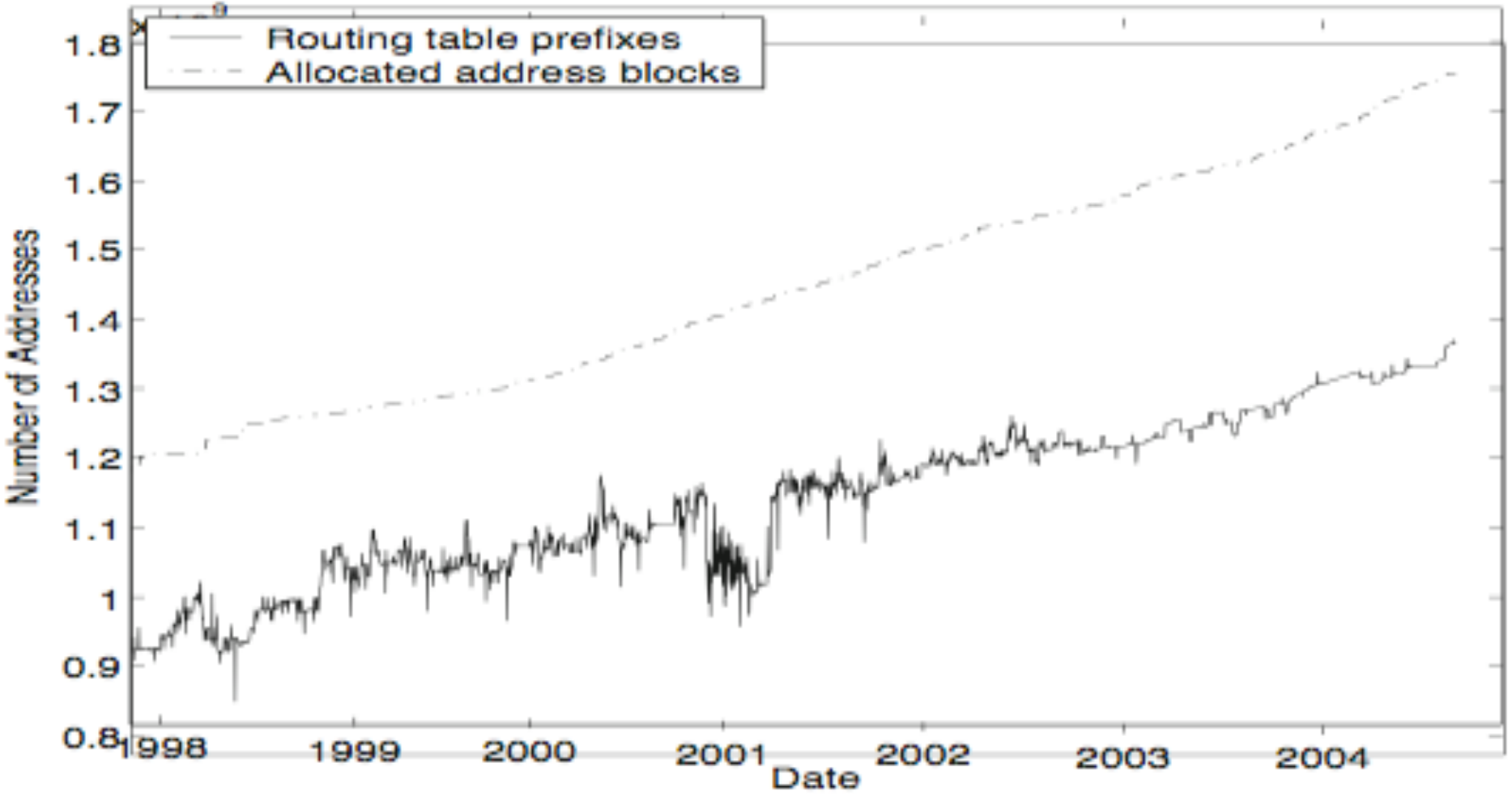
Data collected from RIRs

Average Allocation Length per year

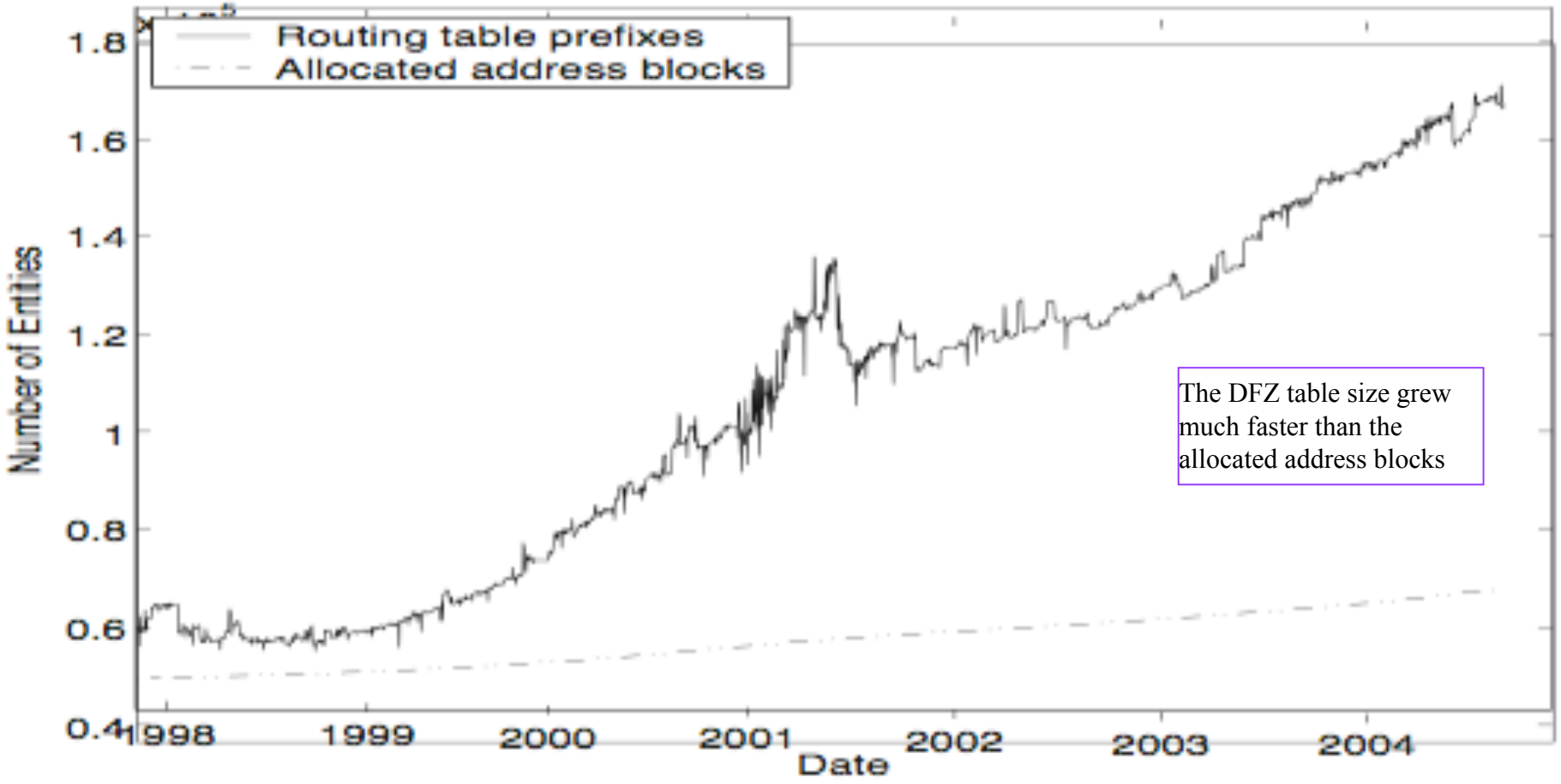


Allocated Addresses vs Routed Addresses

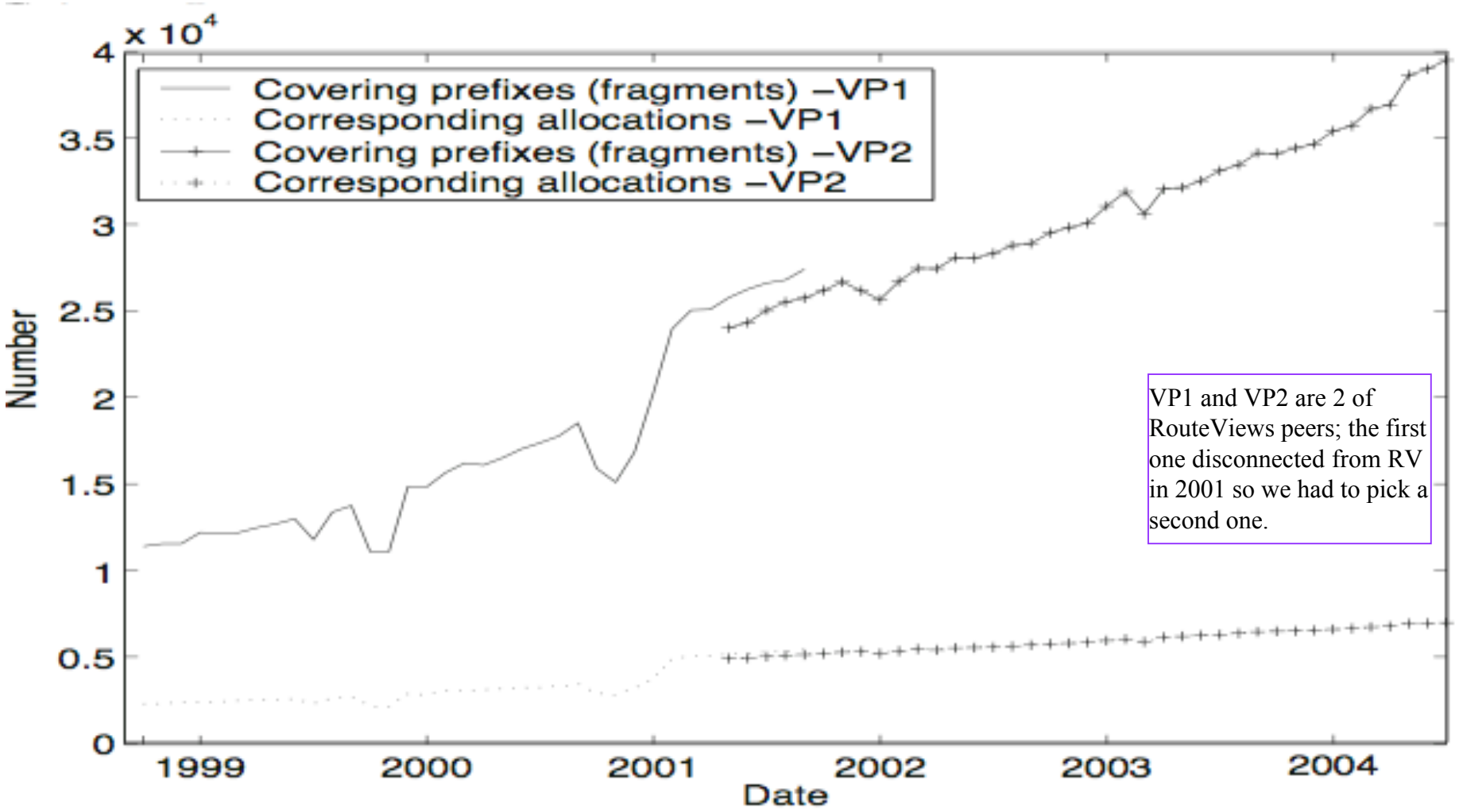
Note the gap between allocated and announced has been slightly increasing over time



Allocated Address Blocks vs BGP Routing Table Size



Covering Prefixes Fragmentation



The length of allocated prefixes is going up each year

The # fragmentations per allocation is also going up each year

The Percentage of Covering and Covered prefixes in routing table