

# IP routing research issues

Brian Carpenter  
Department of Computer Science



**THE UNIVERSITY  
OF AUCKLAND**

**NEW ZEALAND**

Te Whare Wānanga o Tāmaki Makaurau

*July 2008*

# Disclaimer and acknowledgements

- This talk represents my views only. I have no mandate to speak for any of the bodies mentioned.
- Thanks for valuable comments to Thomas Narten, Joel Halpern and Scott Brim in particular.
- Thanks to potaroo.net for the graphs.

# Why is this a research topic?

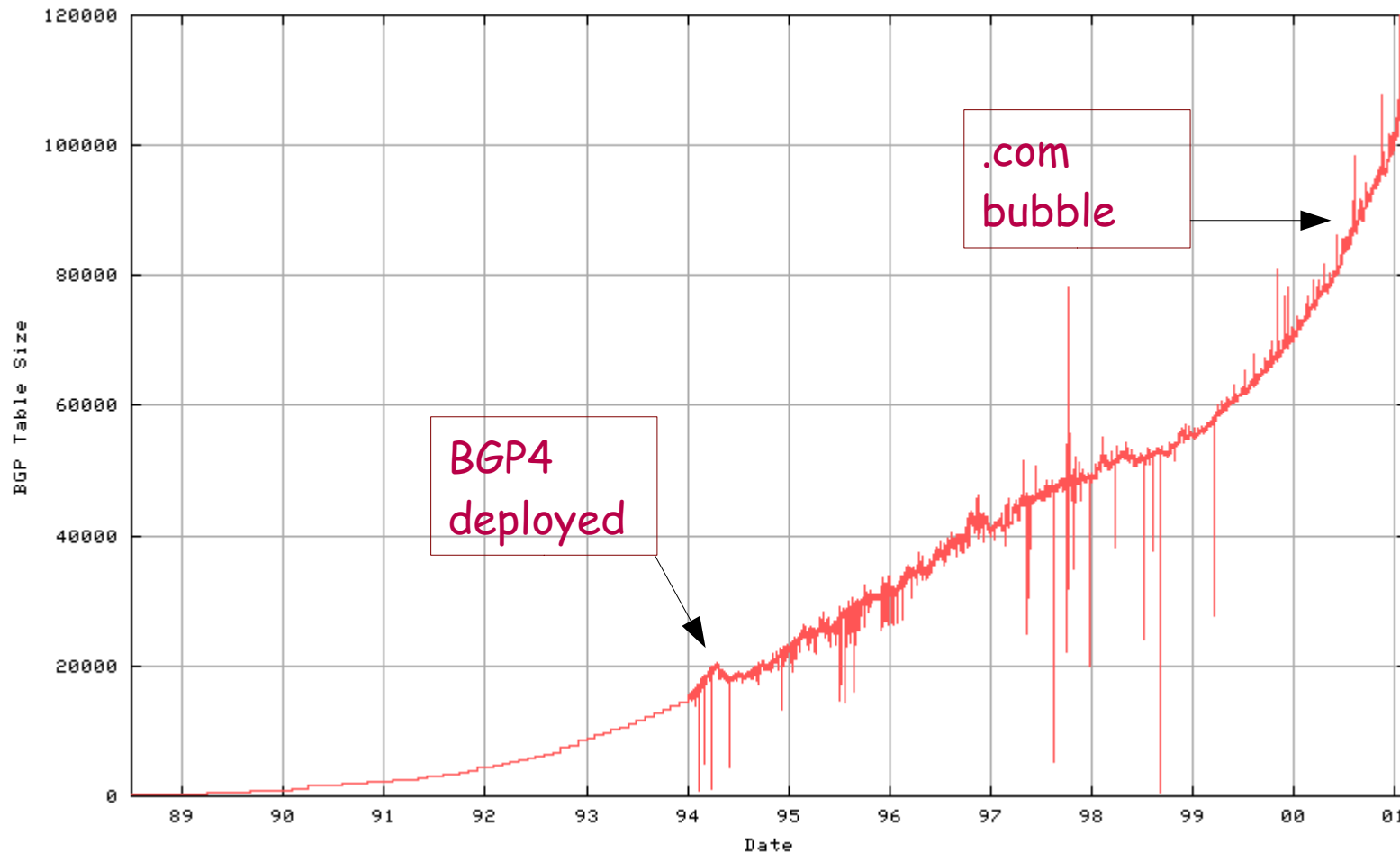
## Didn't Dijkstra, Bellman & Ford solve it all?

- Forget Dijkstra, the interesting part is wide area routing using BGP4, which is kind-of distance vector.
  - In any case, the actual route computation is the easy part.
- Graph theory doesn't really scale well when implemented as a distributed real-time algorithm
  - Especially when the graph keeps changing spontaneously
  - And some of the vertices misbehave
  - And economics is part of the problem

# The players

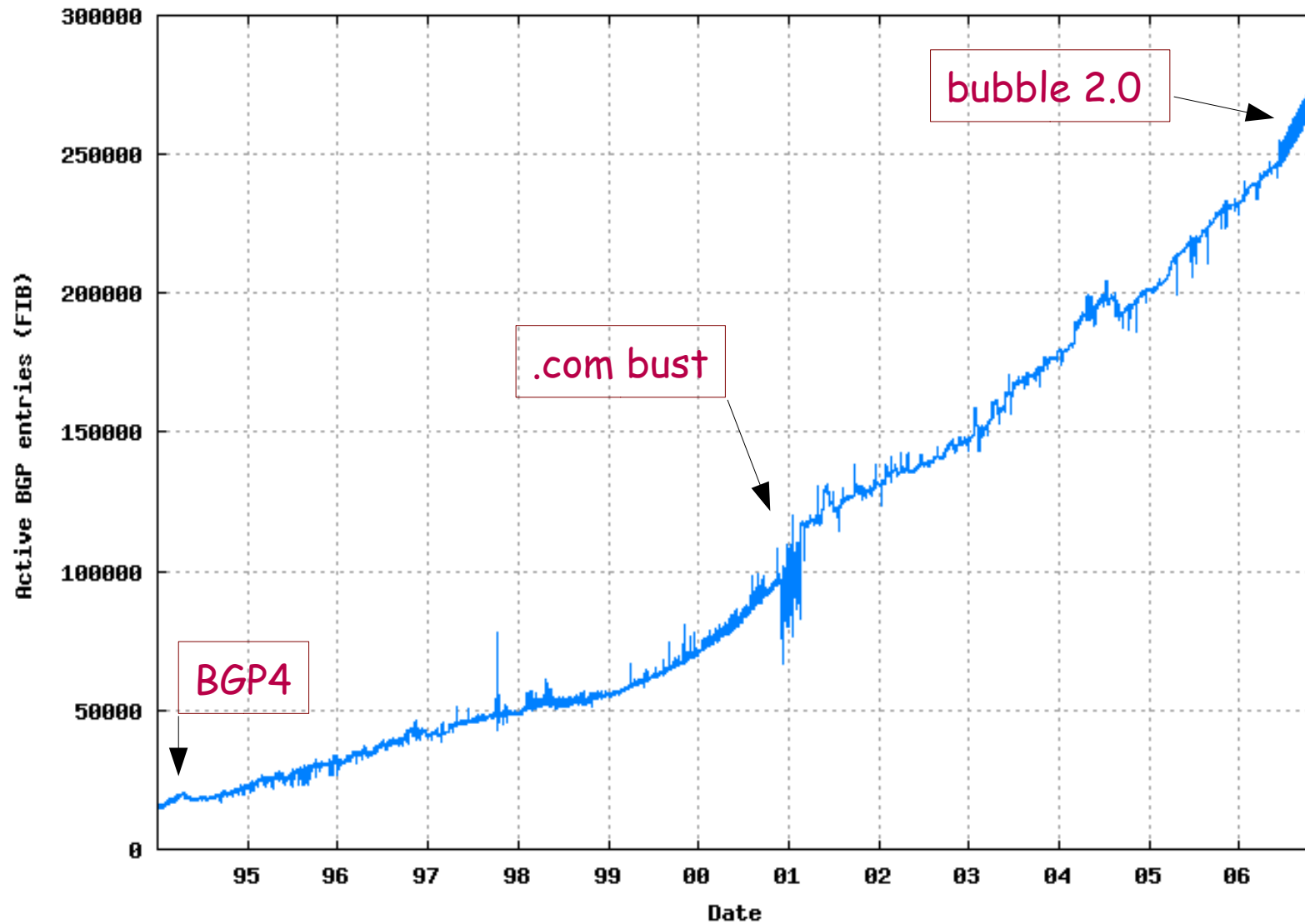
- The core (inter-continental) ISPs, offering transit at a price to...
  - The peripheral (local) ISPs
  - Major customers with their own BGP4 speakers
- Internet Exchange Points
- Core router vendors
- Address registries and other operational bodies
- IETF
- Academic researchers
- **Conflicting technical and economic interests**

# Ancient history



**Things looked very worrying by early 1994,  
and very worrying again by late 2001.**

# Recent history



# What's really going on?

- At times of rapid growth in the network (1994, 2000, 2006) we see accelerating growth in the BGP4 routing table.
- Most likely cause is growth in multihoming of larger customers
  - with present technology, every multihomed customer adds one prefix to the BGP4 table
    - (i.e. has its own route instead of being buried inside a single ISP's address space, RFC 4116)
  - the more the Internet succeeds, the more this becomes a problem

# Why does it matter?

- Since 1990, core router memory size and forwarding speed have managed to keep up with the growth
  - In fact, core ISPs are carrying about a million routes internally (public BGP routes plus three times more customer VPN routes)
  - However, this costs money; routers that can handle a million routes and forward at many Gbit/s are not cheap
  - Maybe one day we will hit a hardware limit
- Route advertisements from ten or a hundred million autonomous systems are not a welcome prospect
  - Customer sites and customer "last kilometres" are much more subject to outages than ISPs
  - Thus, if each (multihomed) customer has its own route, there is concern that BGP4 UPDATE messages, and the consequent route re-computations, will maybe become overwhelming

# Did you say "maybe"?

- Yes. Twice. There may be a hardware limit somewhere in the future. There may be a dynamic limit to BGP4 updates somewhere in the future.
- I'm not aware of any convincing science behind those two *maybes*.
  - The hardware vendors are unlikely to reveal their technical projections to competitors
  - I haven't seen any convincing models of massive scale BGP4 dynamics (there are observational studies)

# Why didn't anybody tell me?

- They did. For example, RFC 1380 (November 1992) discussed "the routing table explosion"
  - The short term fix was classless addressing and BGP4
  - We're still looking for the long term fix
- The IETF NIMROD effort worked on a potential solution (1994-1998)
- The IAB routing workshops in 1998 and 2006 (RFC 2902 and RFC 4984) and the IAB network layer workshop in 1999 (RFC 2956) all considered these issues.
- Current efforts are focussed in the IRTF Routing Research Group (RRG)

# Why didn't the Internet explode?

- If the increasing growth rates that we saw in 1994 and 2000 had continued indefinitely, no doubt there would have been a meltdown.
  - BGP4/CIDR saved us once
  - The .com bust saved us once
  - Sad to say, NAT has saved us many times
  - We need to understand the renumbering bogeyman and the PI heresy

# The renumbering bogeyman



This ship has sailed for IPv4,  
but we should really try to  
do better for IPv6.

- There is unimaginable resistance to IP address renumbering among site IT operations people.
- They, and many application developers, have broken Rule 1:

**NEVER embed an IP address in software or store it in a file;  
ALWAYS use DNS names.**

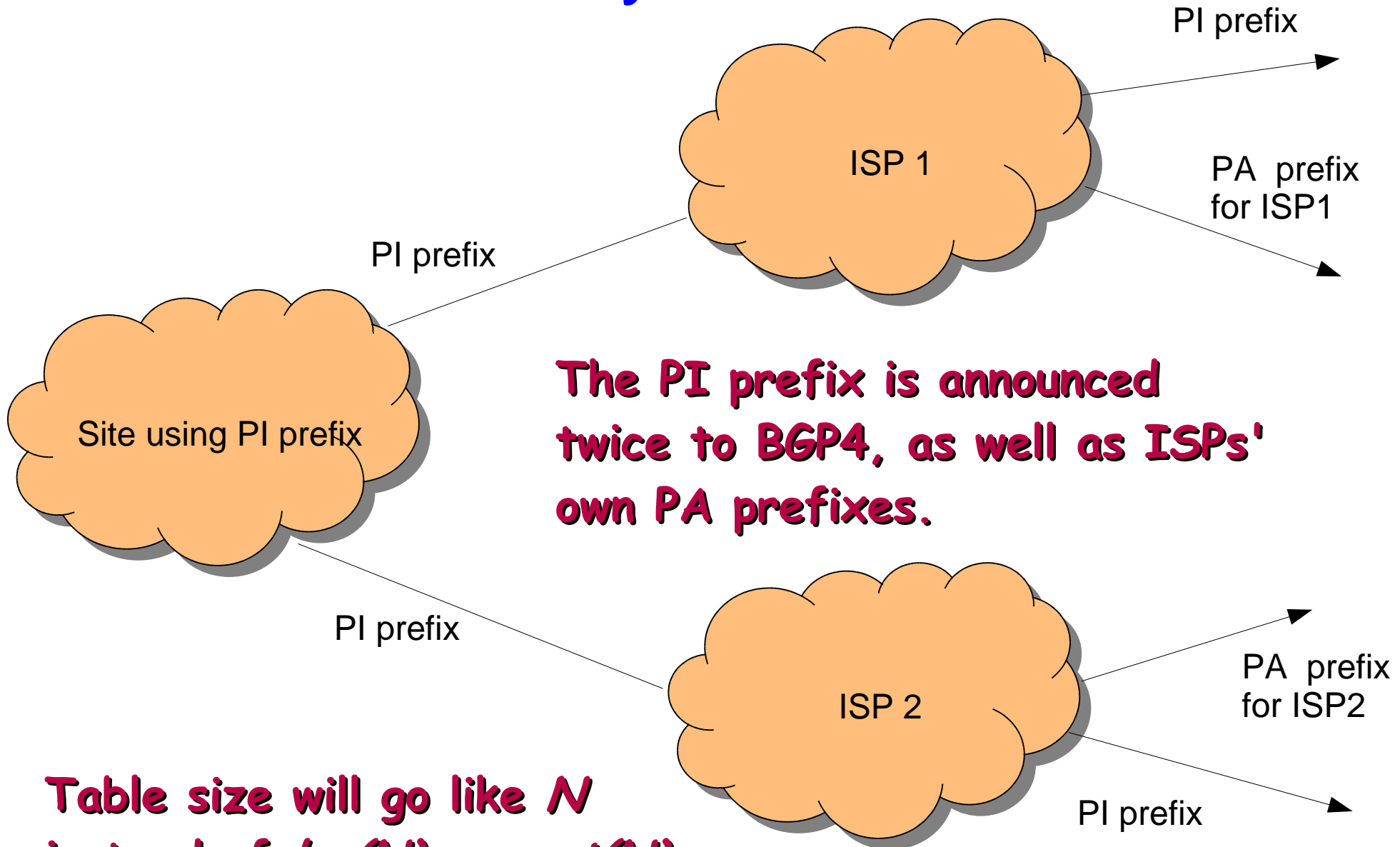
- There are many aspects of network management that are address-dependent.
- The consequence is a great attachment to having "my own address space."

# The PI heresy

- BGP4/CIDR contained growth in 1994 because of the move to ISP-based addresses (*provider aggregation* or PA addressing).
- Sadly, the address registries were persuaded to assign *provider independent* (PI) address prefixes to individual user sites, in direct contradiction to address aggregation.
  - The site avoids the renumbering bogeyman
  - If it multihomes, the PI prefix will lead to its own BGP4 table entry (and UPDATES whenever connectivity flips between ISPs)



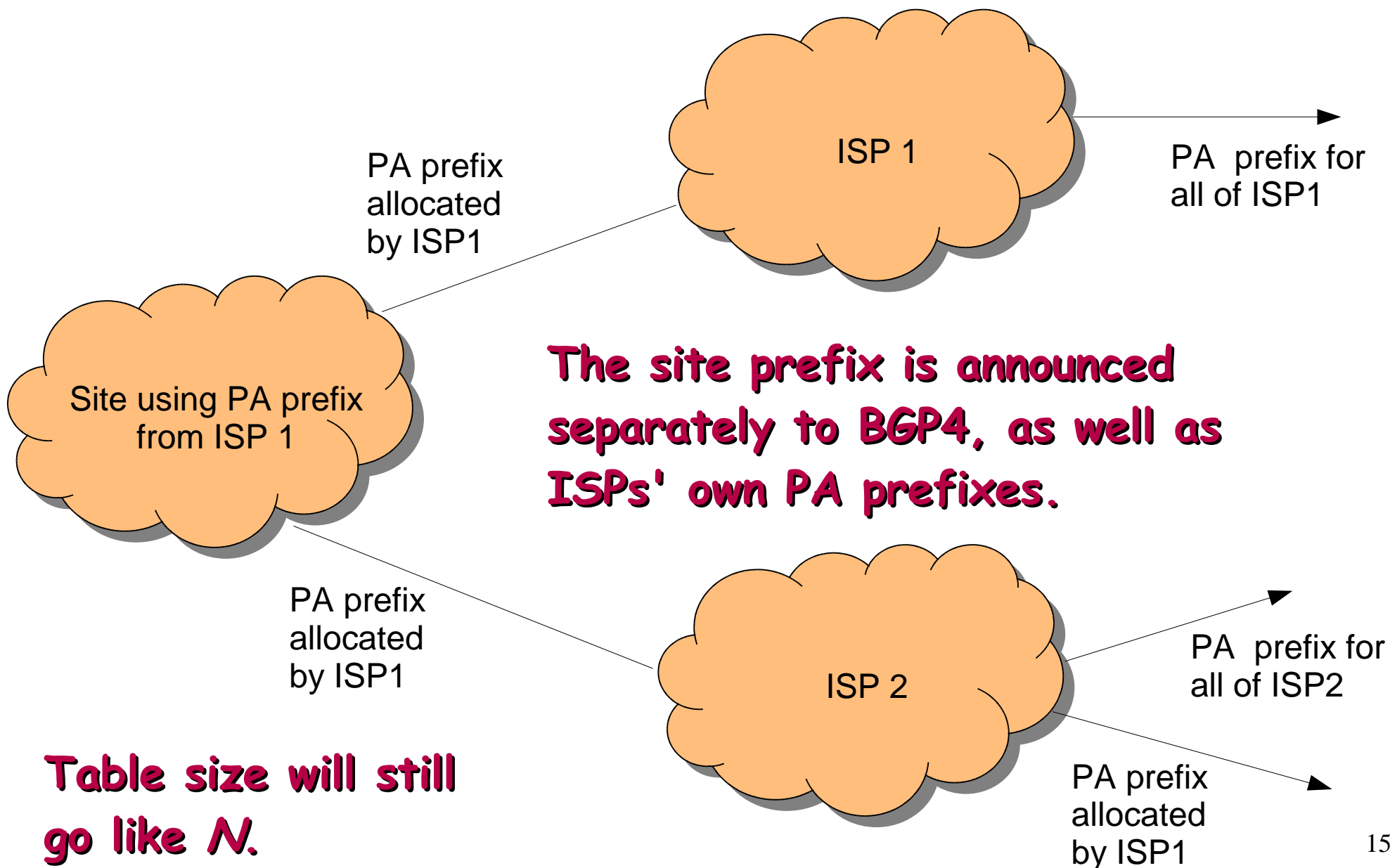
# Why PI hurts



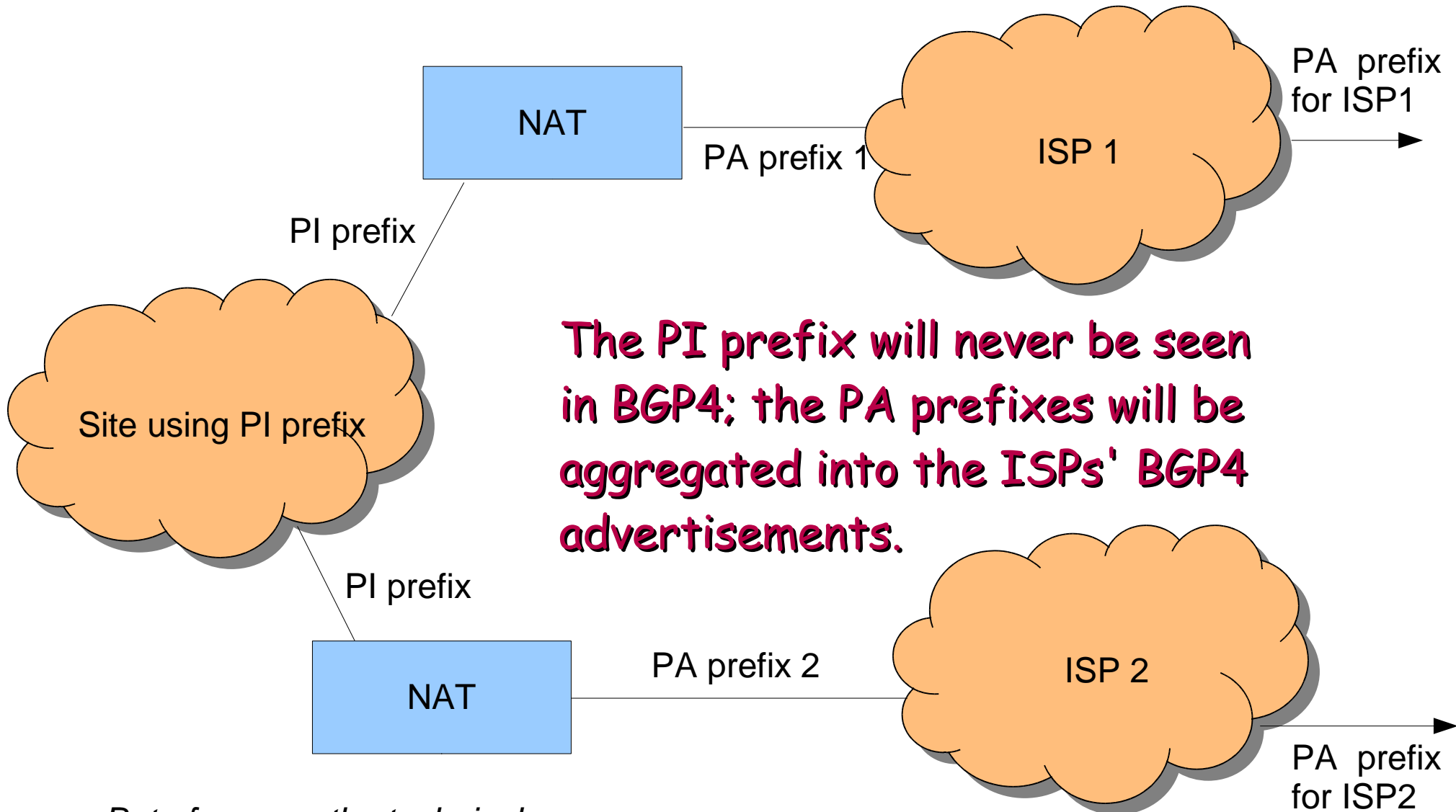
**The PI prefix is announced twice to BGP4, as well as ISPs' own PA prefixes.**

**Table size will go like  $N$  instead of  $\log(N)$  or  $\sqrt{N}$ .**

# Even PA multihoming hurts



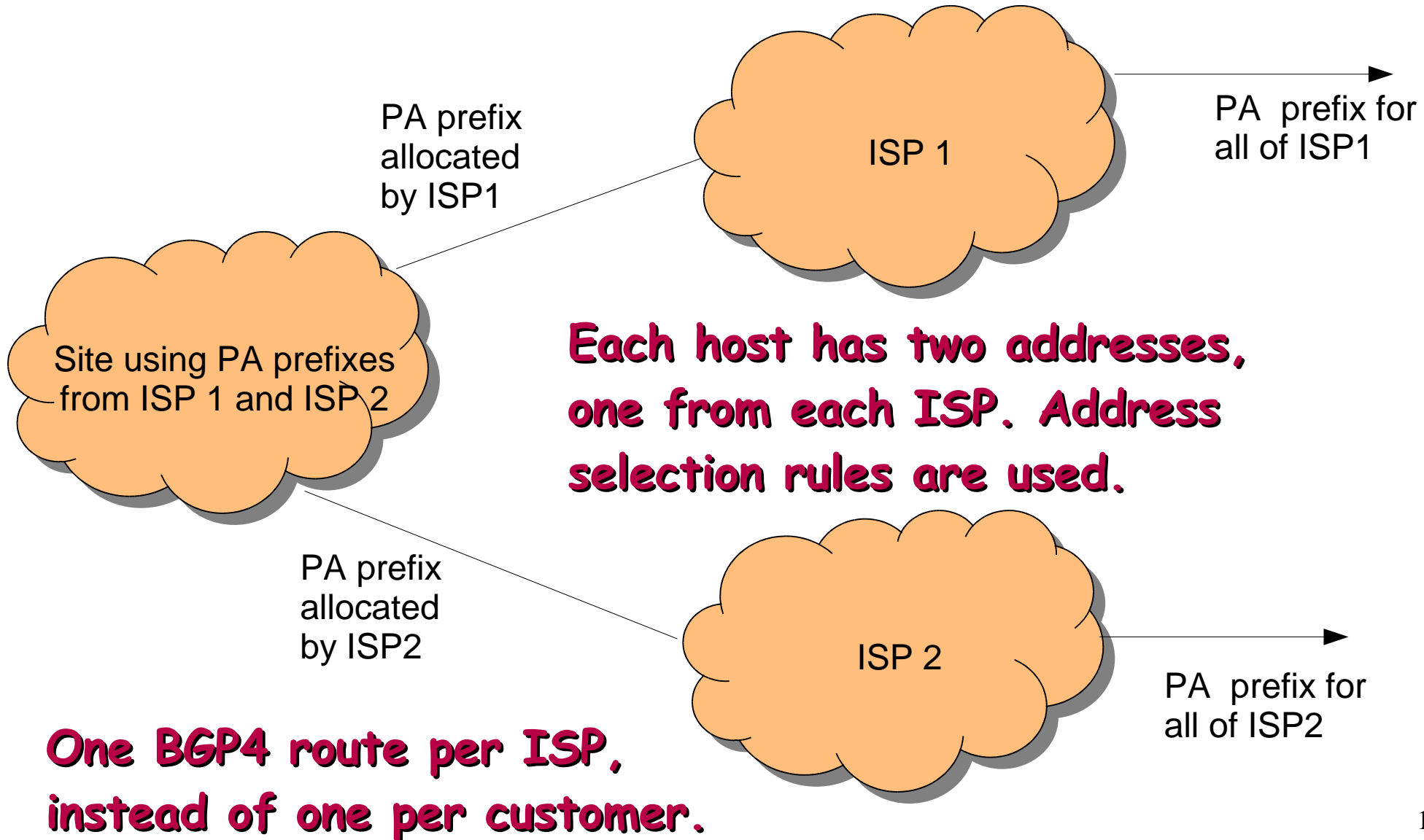
# Saved by the NAT?



**The PI prefix will never be seen in BGP4; the PA prefixes will be aggregated into the ISPs' BGP4 advertisements.**

*But of course, the technical community really wants to get rid of NAT.*

# Plan A (for IPv6): Multiple PA prefixes per site





## Him again

- There are no obvious technical defects in Plan A, but it's meeting operational resistance.
- It assumes that large user sites
  - have systematic address management and DNS generation
  - are willing to add and remove IPv6 prefixes (i.e., stepwise renumbering) when they switch ISPs
  - are prepared to update procedures for address management
- It also disturbs traffic management and business models for ISPs
- It doesn't help IPv4 at all (because we're running out of prefixes)
- Hence, the search is on for Plan B

# Routing (and addressing) issues (1)

[draft-narten-radir-problem-statement]

- Alignment of Incentives
- Pressures on Routing Table Size
  - Traffic Engineering
  - Multihoming
  - End Site Renumbering
  - Acquisitions and Mergers
  - Address Allocation Policies
  - Dual Stack Pressure on the Routing Table
  - Internal Customer Routes
  - IPv4 Address Exhaustion

# Routing (and addressing) issues (2)

- Additional Pressures on Control Plane Load
  - Interconnection Richness
- Questionable Operational Practices
  - Rapid shuffling of prefixes
  - Long prefixes to reduce Route Hijacking
  - Ignorance of effects on aggregation

# Solution criteria (1)

[draft-narten-radir-problem-statement]

- Provide sufficient benefits to the party bearing the costs of deploying and maintaining the technology to recover the cost for doing so.
- Reduce the growth rate of the DFZ control plane load. In the current architecture, this is dominated by the routing, which is dependent on:
  - The number of individual prefixes in the DFZ
  - The update rate associated with those prefixes.
- Any change to the control plane architecture must result in a reduction in the overall control plane load, and shouldn't simply shift the load from one place in the system to another, without reducing the overall load as a whole.

## Solution criteria (2)

- Allow any end site wishing to multihome to do so
- Support ISP and enterprise Traffic Engineering needs
- Allow end sites to switch providers while minimizing configuration changes to internal end site devices.
- Provide end-to-end convergence/restoration of service at least comparable to that provided by the current architecture
- *It goes without saying:* be deployable on the running Internet with adequate performance and scaling, at acceptable cost.

# Focus on Multihoming

- This is the key issue; if we can't solve this, nothing else can be solved\*
- After years of concern, we only know two approaches that avoid the PI heresy and NAT
  1. Ignore the routing system; solve the problem end to end between hosts (using multiple addresses per host).
  2. Split addressing into two layers: a locator used for routing and traffic engineering, and an identifier used between the hosts.

*\* Multicast and mobility issues are not discussed in this talk. It's implicit that the solution has to support traffic engineering at least as well as BGP4 today.*

# Host-based multihoming: SHIM6

- Inserts shim code at the top of the IPv6 stack
  - remote host has several IPv6 addresses (one PA address per ISP)
  - one of them is used as Upper Layer ID (i.e. the address used in socket calls, TCP checksums, IPsec, etc.)
  - the shim switches dynamically between the PA addresses (i.e. the addresses used in the packet headers vary)
  - zero visibility at routing level; only host software is touched
  - host sites must operate one PA prefix per ISP
  - a bit more complicated than it sounds, due to reachability and security issues
- Takes traffic engineering partly out of the hands of ISPs
  - ISPs would like control of ingress path selection, currently implemented by BGP4 policy

# Routing-based multihoming: research

- Basic idea is not new: split apart the functions of an address\*
  - *identifier* is used end-to-end (e.g. TCP checksum, IPsec)
  - *locator* is used for routing site-to-site (and for traffic engineering)
- Not clear how to make this change successfully on a running Internet
  - cut the IPv6 address in two halves (64 bit rewriteable locator and 64 bit fixed identifier)? Doesn't help IPv4.
  - encapsulate normal IP packets (with identifier-addresses) in tunnels (with rewriteable locator-addresses)?
  - add an explicit identifier layer?
- Ongoing work in the IRTF Routing Research Group

\*can arguably be traced as far back as a paper by Louis Pouzin in 1974

# Menagerie of proposals

(not all in RRG)

- LISP - Cisco-driven "map and encap" approach
  - for IPv4 (until addresses really run out) and IPv6
- AIRA
- APT
- CRIO
- BGP hierarchy
- HIP
- HRA
- ILNP
- IPvLX
- Ivip
- NIRA
- Six/One
- TAMARA
- TRRP
- V6DH
- Virtual Aggregation
- SiMIA

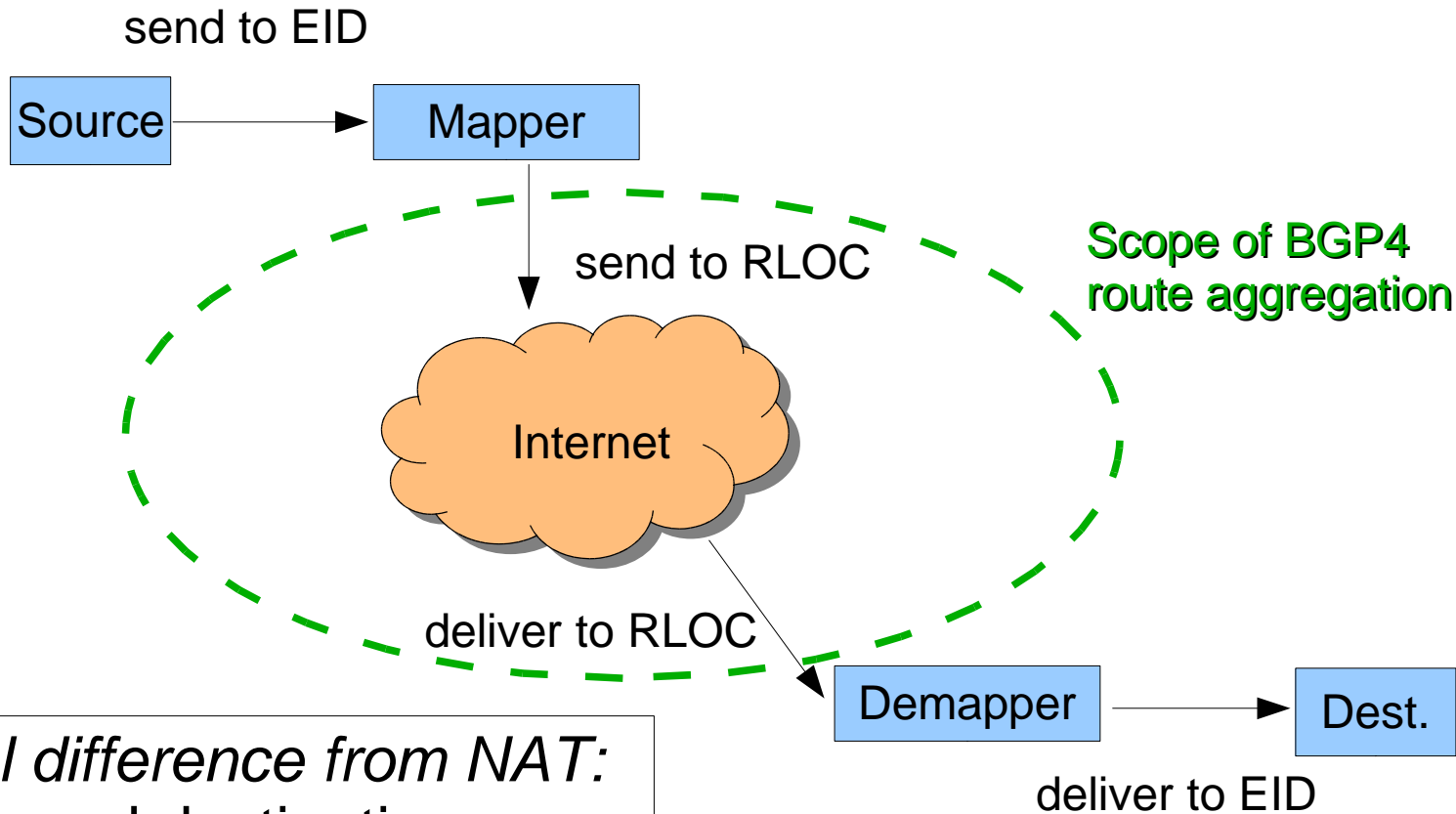
# Commonality

- Almost all have some form of distinction between *locator* and *identifier*
  - but when used locally, identifiers may become locators
- All therefore need some end-to-end mapping between locator space and identifier space
  - effectively, we've inserted either a layer of routing hierarchy or an extra layer of addressing, or both.
- Many need a globally visible mapping database of some kind
  - some ride on DNS, but that raises scale and performance concerns

# Mapping

EID = End-system identifier - not assumed to aggregate in BGP4

RLOC = Routing locator - aggregates in BGP4



*Critical difference from NAT:*  
Source and destination see exactly the same EID.

# Taxonomy (1)

[draft-halpern-rrg-taxonomy]

- Possible end system identifications:
  - Name, such as DNS
  - globally unique location insensitive bit string
  - globally unique location sensitive bit string
  - purely local identifiers [not considered useful in the Internet]
- Possible maps:
  - DNS to EID
  - DNS to RLOC
  - EID to RLOC
  - RLOC to EID

# Taxonomy (2)

- Possible map distribution models:
  - Push - source of an EID->RLOC mapping entry pushes it out to all mapping boxes (border routers) in the world
  - Pull - a mapping box requests an EID->RLOC mapping entry when it needs to send to a new EID
    - on demand
    - from a cached distributed database (like DNS)
  - Hybrid - selective push
- These issues have critical impact on scalability and performance.

# Taxonomy (3)

- Possible implementation mechanisms:
  - Encapsulation and tunnelling
  - Rewrite address at each end
  - End-system based management of separate ID layer

# Compatibility and deployment

- RLOCs are clearly PA addresses.
- Are EIDs PI addresses or something new?
- For IPv4, IPv6, or both?
- Do A and AAAA records deliver EIDs?
- Do upper layers use EIDs in socket calls and 3<sup>rd</sup> party references?
- Are host software changes needed?
- How does the solution interwork with existing BGP4 deployment and existing hosts and routers?
  - Is stepwise deployment OK?
- Impact on MTU size and fragmentation?

# Preserving the API?

- We've learnt from trying to deploy IPv6 that the routers and the IP stacks are the *easy* part. Anything that appears above the socket API creates a Y2K-like problem.
- In my opinion, any new solution that invalidates the socket API again, or even requires noticeable changes to TCP code, is undeployable.
- In my opinion, the routing community tends to underestimate the importance of this.

# BTW

- If you think that compact routing research might be relevant, see *On Compact Routing for the Internet*, Krioukov et al, ACM SIGCOMM CCR **37(3)** 43-52, July 2007.
  - Some pragmatic work on compressing tables by "virtual aggregation" may help. See *A White Paper on Reducing FIB Size through Virtual Aggregation*, Francis et al, Cornell University, work in progress, 2008.  
([www.cs.cornell.edu/People/francis/va-wp.pdf](http://www.cs.cornell.edu/People/francis/va-wp.pdf))

# Sources

- <http://www.potaroo.net/>
- <http://www.irtf.org/charter?gtype=rg&group=rrg>
- draft-narten-radir-problem-statement
- draft-halpern-rrg-taxonomy
- *Quantifying Path Exploration in the Internet*, Oliveira, R. et al, UCLA, ACM IMC'06, 2006.
- *Modeling BGP Table Fluctuations*, Flavel, A. et al, University of Adelaide, in *Managing Traffic Performance in Converged Networks*, Springer-Link, 2007.