IPv6 - what is it,

what do we do with it?

Bob Fink

LBNL/ESnet

co-chair of the IETF IPv6 Transition WG

University of Auckland Seminar

5 April 2000

for the ppt of this talk:

< http://6bone.net/misc >

So what's a new Internet Protocol?

As the Internet sees it	*Example Internet Protocols*	*As OSI sees it*
Application	**FTP, TELNET, SMTP, NTP, SNMP, NFS, HTTP, etc.**	**Application**
		Presentation
		Session
Transport	**TCP, UDP, etc.**	**Transport**
Internet	**IPv4, ARP, ICMP, IGMP, etc.**	**Network**
Link Level	**Ethernet, Token Ring, FDDI, SONET, ATM, PPP, etc.**	**Data Link**
		Physical

Where IPv6 fits

A New Internet Protocol?

- As long ago as 1987, predictions and projections were made about the need for a next generation Internet protocol
- By July 1993, the IETF decided to take on the IPng job, versus letting the market decide
- By July 1994, IPv6 was chosen as IPng from a well planned and exhaustive process to evaluate, modify and select the winning design

Problems with IPv4

- Address depletion/exhaustion and its implications -
- Scaling problems with Inter-domain routing -
- Manual configuration required -
- Multicast, Security, Quality of Service and Mobility -
- Header and format limitations that limit future flexibility -

While we were contemplating an IPng...

- IPv4 work did not stand still…
- DHCP came along to make it (somewhat) easier to handle the user's and sys/net-admin's configuration
- Mcast evolved and is almost commonplace
- Security was added to IPv4 (by way of IPsec)
- QoS was added to IPv4 (by way of diffserv)
- Mobility was added to IPv4 and will grow in use

Address depletion & routing

- strong efforts were made to constrain address usage as anyone asking got what they wanted (not any more :-)
- strong efforts were also made to constrain routing table explosion which occurs when users are assigned random pieces of the address space ignoring what ISP is having to try to aggregate these blocks
- all this is known as CIDR (Classless Inter-Domain Routing)

And then there was NAT

- Network Address Translation allows a site to use private addresses behind a NAT gateway/firewall when communicating locally,
- and then automatically get a global IPv4 address assigned from a smaller pool when needed for Internet communication,
- which requires changing the IPv4 header's source address on the fly, which has a few problems...

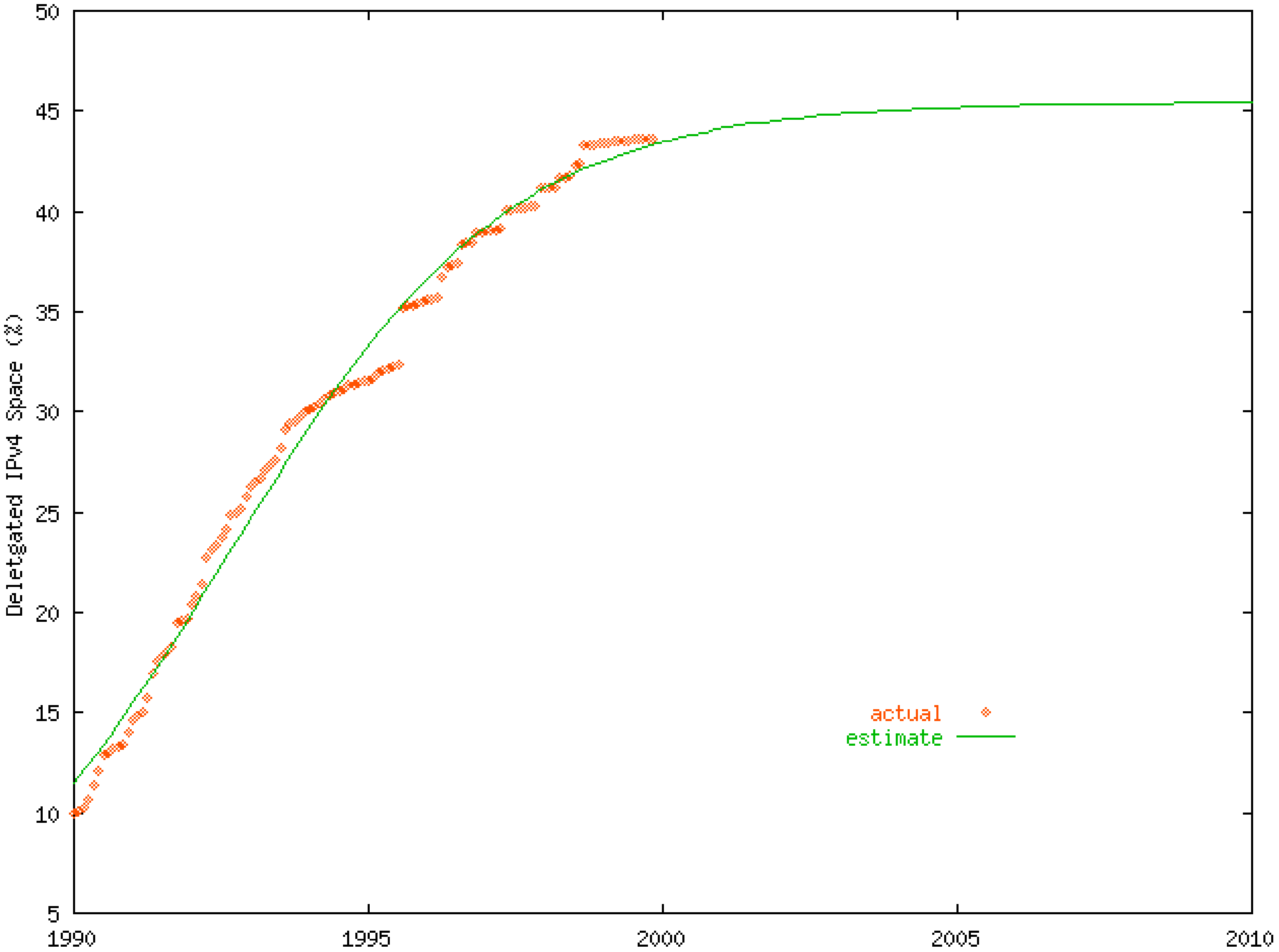
Problems with NAT

- Ruins the Internet end-to-end model by inserting an in-path transform of packet
- Much added operational complexity and cost, especially in large corporate environments, keeping up with what busts next
- Messes up the use of IPSEC, I.e., busts the end-to-end model
- …and you still eventually run out of addresses!

So, can NAT solve its problems?

- NATs in every host (RSIP) - though this also (like IPv6) requires network software in every host system to change (though possibly with less impact)
- multiple NAT domains (again, RSIP) to solve global addressing by localizing sphere of globality (gee, is that a word?)
- It’s possible that the vendor community will believe this is worth pursuing instead of IPv6, but it will come at a big price

IPv4 Address Space Delegation - Fall 1999



Interesting comment on the trend line

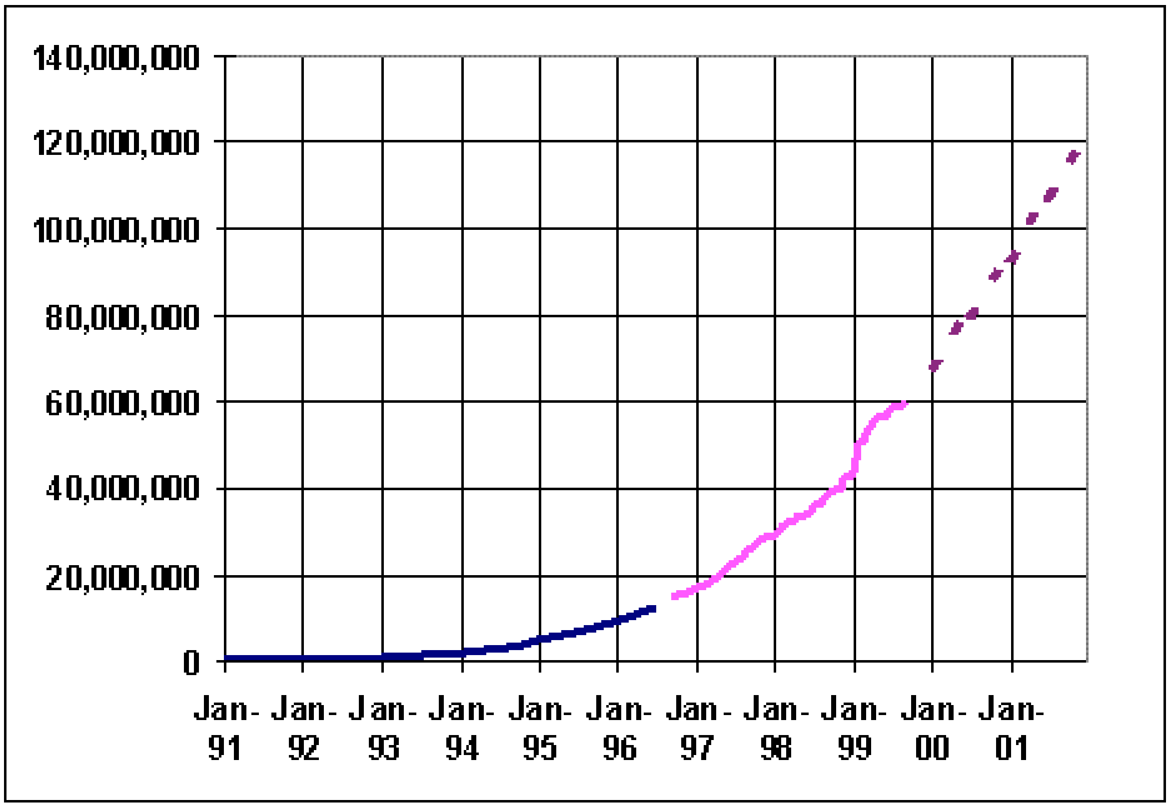
- **Frank Solensky: "…(my) estimates for maximum address space utilization have risen about 8% over the last 3 years. If one were to argue for extrapolating this over time as well, the resulting statement would be that "in the year 2019, the trend line will suggest that we will eventually run out of IPv4 addresses". "**
- **http://ipv4space.toplayer.com/**

Will we ever run out?

- sooner or later even these conservative global v4 address uses lead to … exhaustion of the global address space
- the trend is to more and more time spent online, beyond private network boundaries, … that is, using public/global addresses (Cable, DSL, etc.)
- newer applications (such as seamless global and mobile telephony) will lead to even more public/global address usage

Internet user growth

At some point, we will need more addresses



Telcordia Technologies

So looking at those IPv4 problems again

- **Address depletion/exhaustion and its implications -**
  IS A VERY BIG DEAL EVEN NOW WHICH CAN'T BE FIXED BY USING IPv4 or NATs IN THE LONG TERM
- **Scaling problems with Inter-domain routing -**
  A BIG DEAL WHICH MAY BE FIXABLE BY IPv4, MAYBE NOT, AND CERTAINLY ISN'T SOLVED YET
- **Manual configuration required -**
  A BIG DEAL WHICH IS MOSTLY FIXED BY DHCP, BUT MUCH REMAINS (small sites, mobility, site renumbering)
- **Multicast, Security, Quality of Service and Mobility -**
  A BIG DEAL THAT IS BEING RESOLVED NOW (but end-end security is in the tank due to NATs)
- **Header and format limitations that limit future flexibility -**
  NOT A BIG DEAL

and IPng proceeded ...

- Back at the IETF, the IPng Project proceeded at speed due to cries on impending disaster:
- A special IPng Directorate to guide the process was created,
- Whitepapers where called for,
- Criteria were generated, and
- Proposals for an IPng were called for.

IPng Criteria

- No. of networks - 10^9, preferably 10^{12}
- No. of end systems - 10^{12} , preferably 10^{15}
- Autoconfiguration - no manual configuring
- Straightforward transition plan from IPv4
- High performance
- Service classes
- Conservative routing schemes
- Secure operation
- Extensible format
- Tunneling built in
- Support for IPv4 features:
  - mobility, control protocol, multicasting, media independent,
  - datagram service, topologically flexible, robust service, etc.

IP Version Numbers

0 – 3 unassigned (to allow other uses in the future)

4 IPv4 (the current IP in TCP/IP)

5 Stream Protocol - ST (never an IPng)

6 SIP –> SIPP –> IPv6 (the next generation IP)

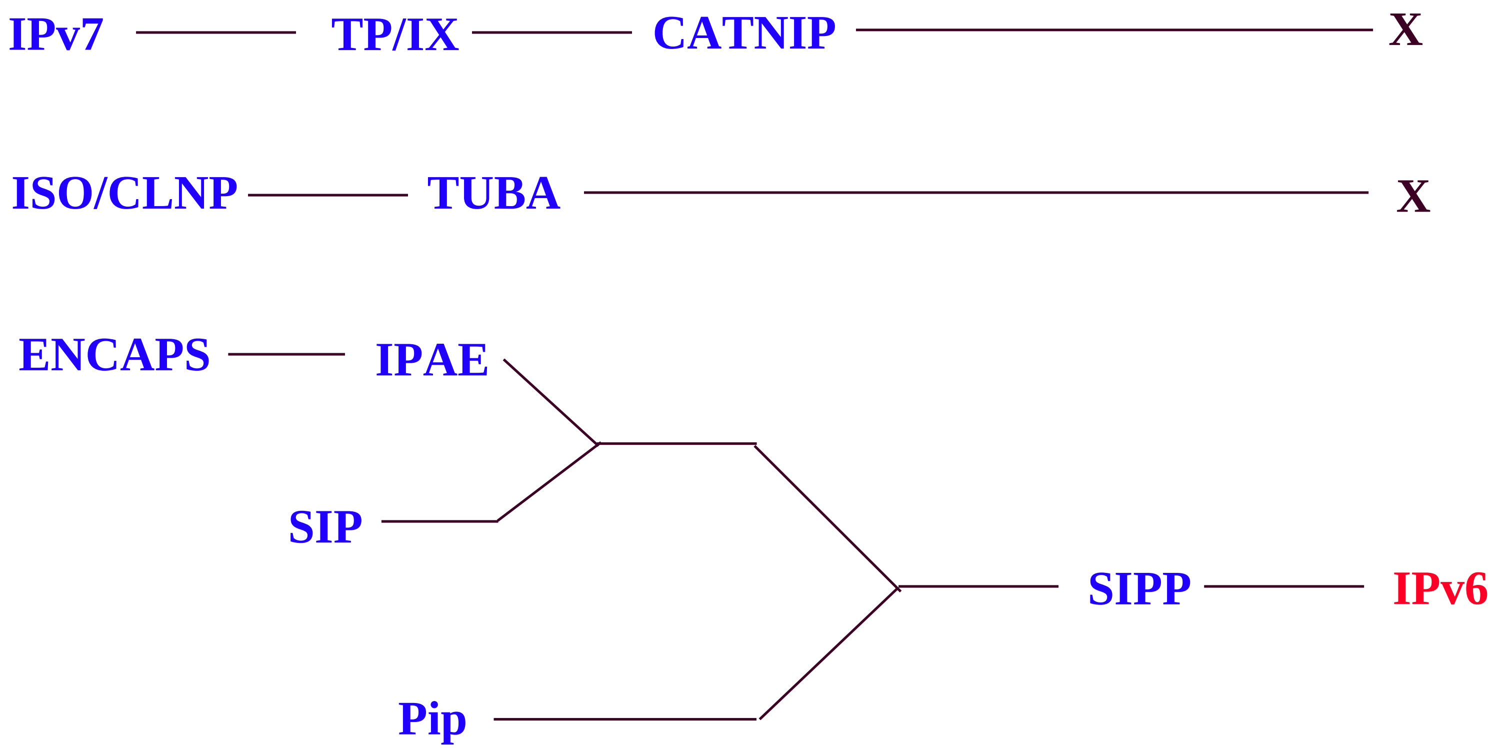
7 IPv7 –> TP/ IX –> CATNIP (died)

8 Pip (later joined with SIP)

9 TUBA (died)

10 – 15 unassigned

IPng Design Contenders



IPv6 Features

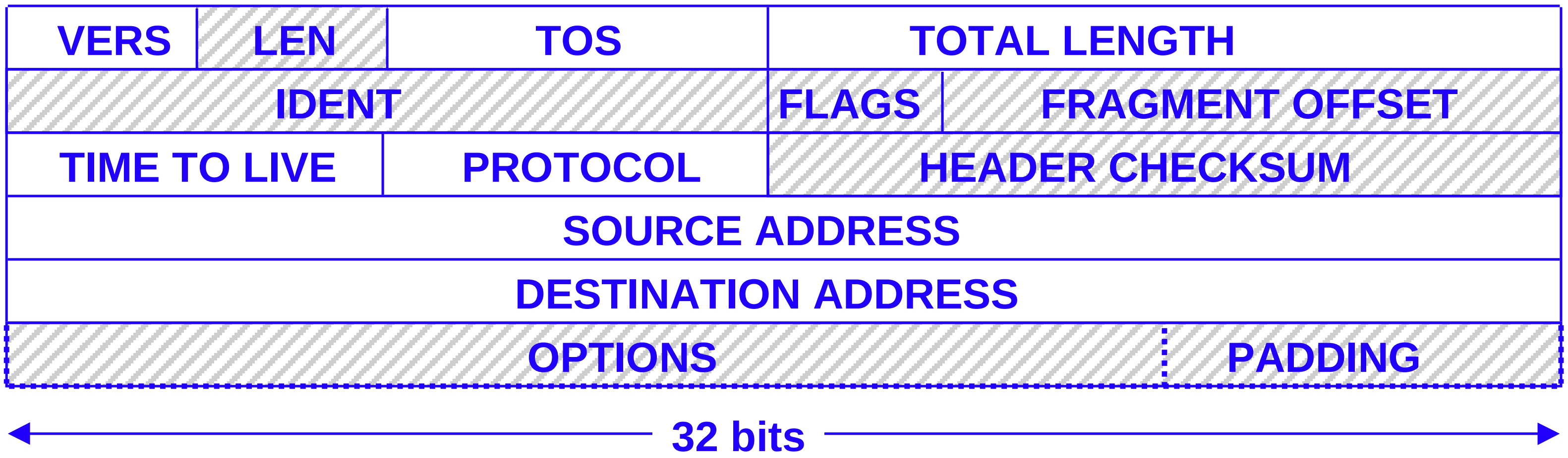
- Expanded addressing capabilities (for hosts, networks, organizations, providers, etc.)
- Streamlined header format for hi-performance
- Header support for options/extensions
- Stateless Autoconfiguration & Neighbor Discovery (real plug and play)
- Staged (incremental) deployment/transition from IPv4
- Support for security, quality of service, mobility and multicast built in from the start

What It Didn't Do

- Given the IPng project criteria of conservative routing schemes, IPv6 did not try to do anything radical
- thus IPv6 relies on current routing ideas and technology
- However, IPv6's new Aggregatable Global Unicast addressing format does provide a major step towards constraining routing table growth (more later)

IPv4 Header

20 bytes basic, expandable by 40 bytes



note: hatched fields absent in IPv6 header

note: recent IETF diff-serv work has led to a new TOS for IPv4 which is now the same for the IPv6 Class field

IPv6 Headers

40 bytes fixed… extension headers used for more than this

VERS	CLASS	FLOW LABEL
PAYLOAD LENGTH	NEXT HEADER	HOP LIMIT
SOURCE ADDRESS		
DESTINATION ADDRESS		

← 32 bits →

note: no header checksum, payload length doesn't include basic header, next header gives type of the header that follows, time to live replaced with real hop limit.

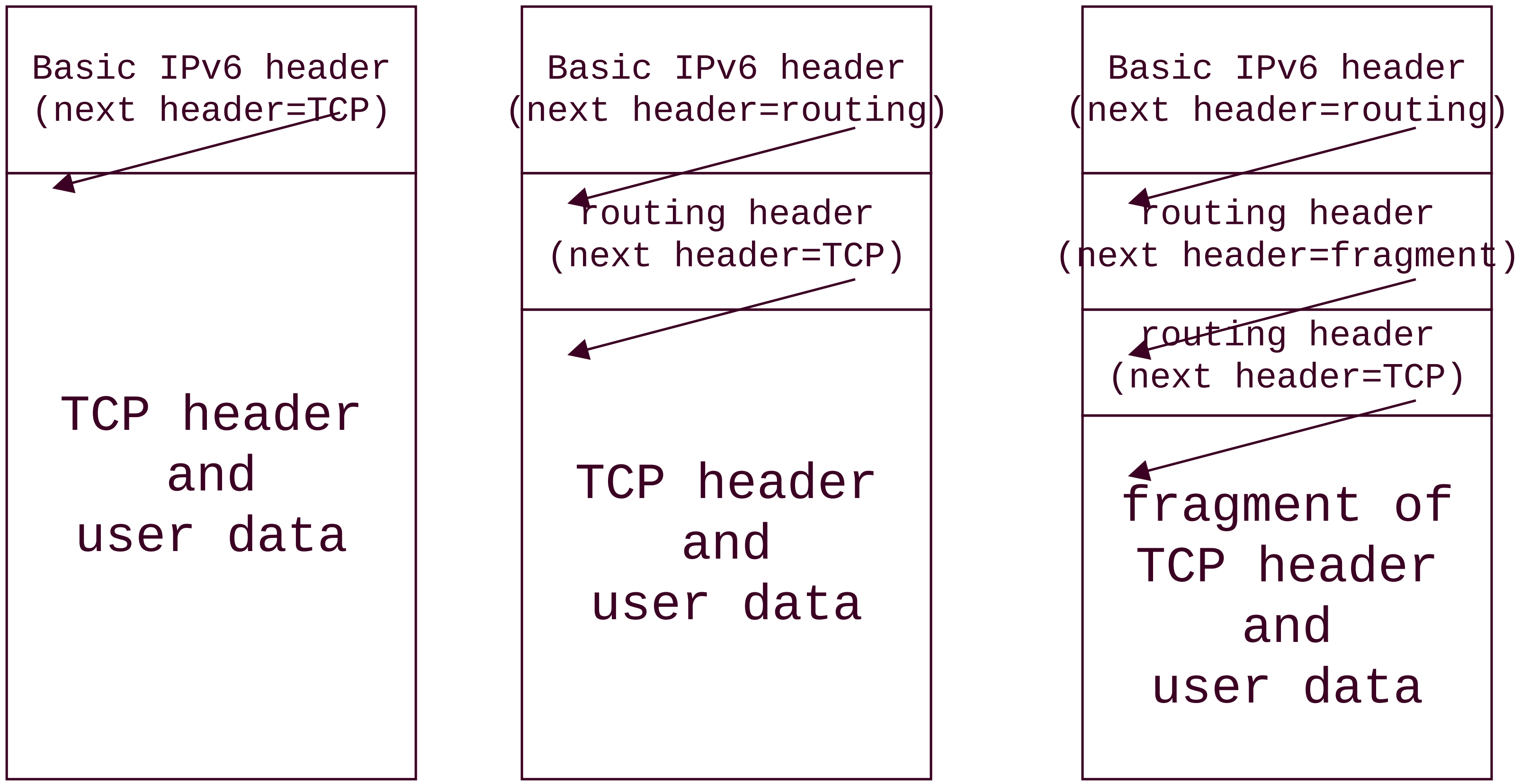
Changes in IPv6's Header

- addresses are bigger – was 32, now 128 bits
- no header checksum for efficiency - very little risk
- flow label added for fast processing in routers
- no network layer fragmentation
- length field excludes IPv6 header (as it is fixed)
- alignment changed from 32 to 64 bits for performance
- extraneous things removed altogether or moved to header extension which can be processed much faster

IPv6 Header Extensions

- IPv6 Headers are added on after the basic header
- Some are hop-by-hop (processed by routers), others are processed at final destination
- Are much lower overhead than IPv4 headers as usage is well defined, and are designed for fast processing
- Currently defined extension headers are:
  - hop-by-hop options (e.g., jumbo-grams, only one so far)
  - routing (source routing when required)
  - fragment (discouraged)
  - authentication (see Security)
  - encryption (see Security)
  - destination options (processed at final destination)

Header Extensions



Fragmentation To Be Avoided

- Experience has shown that network layer hop-by-hop fragmentation can impact performance and implementation complexity in routers
- Path MTU Discovery is the preferred way to avoid fragmentation by discovering the smallest MTU along the path
- When fragmentation is needed, e.g., for some old applications that still rely on it, it is best to perform on an end-end basis by hosts
- Thus fragmentation is now accomplished, if needed at all, by a special IPv6 header extension that keeps the router out of the process
- Default Minimum MTU has been raised to 1280 bytes to allow encrypted & tunneled IPv6 to operate by default over Ethernet links when Path MTU viewed as too much overhead

Header Checksums Removed

- **Verification and update of header checksums can increase router processing time**
- **With very little risk IPv6 relies on two other levels of header protection rather than having its own specific header checksum**
  - **link layer checksums**
  - **transport layer (TCP & UDP) checksums used to protect headers as well**

IPv6 Address Format

IPv6 Address Format Prefix is left justified in 128-bit address field

Allocation	Prefix (binary)	Fraction of Address Space
Reserved	0000 0000	1/256
Unassigned	0000 0001	1/256
Reserved for NSAP Allocation	0000 001	1/128
Reserved for IPX Allocation	0000 010	1/128
Unassigned	0000 011	1/128
Unassigned	0000 1	1/32
Unassigned	0001	1/16
Aggregatable Global Unicast Addresses	001	1/8
Unassigned	010	1/8
Unassigned	011	1/8
Unassigned	100	1/8
Unassigned	101	1/8
Unassigned	110	1/8
Unassigned	1110	1/16
Unassigned	1111 0	1/32
Unassigned	1111 10	1/64
Unassigned	1111 110	1/128
Unassigned	1111 1110 0	1/512
Link-Local Unicast Addresses	1111 1110 10	1/1024
Site-Local Unicast Addresses	1111 1110 11	1/1024
Multicast Addresses	1111 1111	1/256

Only 15% of all IPv6 address space is specified for use so far, and almost none is assigned yet

Names and IPv6 addresses in DNS

- Systems supporting IPv6 are known by the same domain names we know today, but they will have IPv6 and IPv4 addresses registered to them
- AAAA records are used now to hold the IPv6 addresses alongside IPv4 A records
- most all DNS implementations have supported this for several years (via IPv4 access, not IPv6 access… that will come next year)

Aggregatable Unicast Addressing

3	13	8	24	16	64
001	TLA	Rsrv	NLA	SLA	Interface ID

48-bit Public Topology
Routing Prefix

80-bit end-site specific usage
ISP can not change this

TLA = Top-Level Aggregation ID - are assigned to ISPs and Exchanges that act in a default-free way with a routing table entry for every active TLA ID (this constrains the routing complexity)

Rsrv = Reserved for either TLA or NLA expansion

NLA = Next Level Aggregation ID - are assigned *by TLAs* to create a multi-level hierarchy underneath it as the ISP chooses
(i.e., multiple NLA levels allow more ISPs and then the end site)

SLA = Site Level Aggregation IDs are used to create local addressing hierarchy (e.g., a flat subnet space allowing 65K subnets)

Aggregation can assist routing

- Limits top level routing complexity - the top level only need to peer with, and carry routes for, others at the top level
- Allows top level ISPs to allocate addresses below them in their hierarchy with no affect on the routing complexity above them
- Provides a well defined 48-bit Public Routing Prefix that is more easily changed without disruption than in IPv4

Interface ID

- Interface IDs identify interfaces on a link
- derived from the newer IEEE EUI-64 format (which has an extended IEEE 48-bit "Ethernet" address embedded in it)
- helps automatic configuration while allowing expansion for future MAC address formats beyond Ethernet
  (e.g., USB, IEEE1394/Firewire, …)
- And the privacy fuss over the Interface ID

The privacy fuss

- **Some don’t like any (semi) consistent and traceable cookie, token, serial number or otherwise to be used and seen on the Internet, and certainly not without the user knowing and approving, and having an option to avoid its use**
- **Our answer to this was to allow implementers to assign both a permanent host ID (the EUI-64) for server use and a frequently changing random number host ID for browsing use**
  (Autoconfiguration supports this nicely)
- **This one turns out to be a win for IPv6 over IPv4**

Sub-TLA Assignment

3	13	13	19	16	64
001	TLA 0x0001	Sub TLA	NLA	SLA	Interface ID

- To assist in the slow start of TLA assignment, a Sub-TLA was defined which allows the international address registries to slow start TLA growth for all ISPs
- the ISP must demonstrate a high usage of its Sub-TLA space (80%) before qualifying for a TLA
- in practice, the IP registries are slow starting this by only assigning a /35 instead of the /29 we recommended!
- So when you see an IPv6 address that starts 2001::/16 it is one of these sub-TLA's
- Note, only 1 of the 8192 possible TLA's is being used this way, so very little IPv6 address space is committed to this

6bone (the IPv6 testbed) pTLAs

3	13	8 or 12	24 or 20	16	64
001	TLA 0x1FFE	pseudo TLA	NLA	SLA	Interface ID

- The 6bone uses a variation of this concept called pseudo-TLAs (pTLAs)

  `3FFE:0000::/24 to 3FFE:7F00::/24` **old 8-bit pTLA space**

  `3FFE:8000::/28 to 3FFE:FFF0::/28` **new 12-bit pTLA space**

- So when you see an IPv6 address that starts 3FFE::/16 it is one of these pseudo-TLA's

So where are we today?

- lots of excellent standards for IPv6
- lots of implementations (not enough mind you)
- an excellent testbed (the 6bone)
- production IPv6 addresses now available
- some really good mechanisms for transition
- very few applications vetted to run over IPv6
- little or no motivation to do IPv6 now

Standards

- **The IETF IPng WG now has a “core” or “base” set of IPv6 standards documents at DS (Draft Standard)**
- **if we had more (any) real production experience, the “core” standards could move to full standard**
- **in the IETF *(unlike most other standards organizations)* protocols must be in real use and proven to be interoperable to qualify for full “standard”, so draft is OK**

Implementations

- Sun Solaris 8 release includes IPv6
- Linux, BSD's, and Compaq have IPv6 in various forms
- Mac's and HP's can turn IPv6 on as they wish as they use the Mentat code-base
- Cisco has an excellent IPv6 version which will be released in 12.1(5)T this year
- Microsoft Windows 2000 next version will support IPv6; for now there is a pre-release

Testbed for IPv6 (6bone)

- mid 1996: the 6bone testbed starts up to test specs and implementations between US, EU & JP
- mid 1997: the 6bone restructures into a backbone core with leaf nets and sites
- late 1997: the 6bone converts to the new Aggregatable Unicast Addressing format
- Today: the 6bone is in 42 countries at 525 sites/networks, with 66 of these networks providing 6bone backbone service

IPv6 Address Allocation - a land rush!?

- Funny as it may seem, the biggest issue for the registries was/is to prevent a land rush that destroys the TLA address prefix usefulness for aggregation

- so the big emphasis is to only allocate IPv6 prefixes to networks, not end sites, that will really put up IPv6 service within 6-12 months

- the 6bone backbone is also being used as a prequalification step to help bootstrap this process

IPv6 Address Allocation - finally done!!

- **The big three Internet address registries, APNIC, ARIN & RIPE-NCC, started to assign IPv6 /35 ISP prefixes in July 1999**
- **As of this talk, 21 are assigned, mostly to R&E Networks:**

***APNIC*: CONNECT-AU, WIDE-JP, NUS-SG, KIX-KR, JENS-JP, ETRI-KRNIC-KR, NTT-JP, HINET-TW**

***ARIN*: ESNET, VBNS**

***RIPE-NCC*: DE-SPACE, EU-UUNET, UK-BT, CH-SWITCH, AT-ACONET, UK-JANET, DE-DFN, NL-SURFNET, RU-FREENET, GR-GRNET, DE-ECRC**

Site renumbering

- One very neat IPv6 feature enabled by 48-bit fixed external routing prefixes is site renumbering (remember that the ISP owns the prefix, not the site)
- Router Renumbering protocol lets a net manager multicast new and multiple prefixes to the site's routers, and reduce the time to live for old prefixes
- New prefixes get used via ND, old sessions die away, then the old prefix is removed
  - (please don't ask about long lived sessions :-)

Transition Mechanisms

- **The IETF NGtrans WG now has many mechanisms/tools to aid in transition**
- **From the start, dual IPv4/IPv6 stacks and IPv6 over IPv4 tunneling were built in**
- **Subsequently, we have been adding new tools to aid in the transition (lots of them as everyone has ideas, but few know for sure what will work)**
- ***6over4*** **and** ***6to4*** **are of special interest, so more on these later**

Applications and IPv6

- Remember, no application we know of needs IPv6 to run
- This is a feature, not a “bug”
- This feature is the way we guarantee that transitioning from IPv4 to IPv6 is transparent as possible
- But due to applications that use/store IPv4 addresses explicitly, a basic conversion to 128-bit addresses (new API) must be done

A (the?) First Scientific App on IPv6

- The UCSD/Osaka U. Telemicroscope Project is in the process of converting to IPv6
- The Argonne Lab Telemicroscope project is in hot pursuit to beat them
- these will be excellent demonstrations of how easy it is to run Internet apps over IPv6 as they demo over the web
- unfortunately, it's not easy (yet)
  - JAVA !!! and all kinds of little details

All this providing...

- Motivation to run real Internet applications over a native IPv6 infrastructure
- You didn’t think this was going to be easy did you?

And the motivation is low

- After all, there isn't a problem, is there? anyone that needs IPv4 addresses gets them (so the ISPs like to tell you)
- IPv6 hasn't made anyone a profit in the last 18 months and won't make anyone a profit in the next 18 months!
- NAT will solve all known problems!!
- it will be a very long time, if ever, that we run out of public IPv4 addresses!!!

Any problems with this?

- Justifying each piece of public address space based on your business plan
- operational and management costs for NAT are high
- non-traditional/newer uses of Internet technology just don’t qualify for large public address blocks (the mobile hand held voice and data folk are agonizing over this as we speak)

The price of not converting to IPv6?

or at least not starting to do it…

- **you can't get the bugs worked out, for both IPv6 specs and applications**

- **you can't get experience on how to really do a transition (no one knows enough to really determine this before hand!)**

- **and (this is a biggie) Internet technology will eventually be constrained to a much more limited problem space and**
  WILL BE REPLACED

Remember

- **it took IPv4 at least ten years to mature as a production useable specification with all the necessary surrounding junk to get the job done**
- **even then, Internet technology took a long time for real acceptance (there were competitors)**
- **this Internet “revolution” doesn’t just get to continue *automagically***
- **sometimes you have to decide something is ‘the right thing to do” and do it!**

So what is the right thing to do?

- finalize the basic standards
- finalize the initial transition tools/mechanisms
- get implementations released into production
- deploy early production networks
- and run production Internet apps run over them

So a little about transition

- Just how do we do this?
- Do we assume v6 in the core first,
- or v6 coming from the enterprise out?
- Do we assume v4/v6 dual stacks,
- or v6 only stacks, and the need for v6-v4 interconnection?
- or, or, or...Just what do we assume?

The let a thousand flowers bloom model

- With no clear idea of how a transition will happen, or when, ...
- the focus has been on inviting proposals, white papers if you will, and as many ideas as possible about assumptions, scenarios and tools/mechanisms to solve them
- hopefully most of the current phase of this work will be complete by the end of the year (note, this doesn't mean we know what will happen :-)

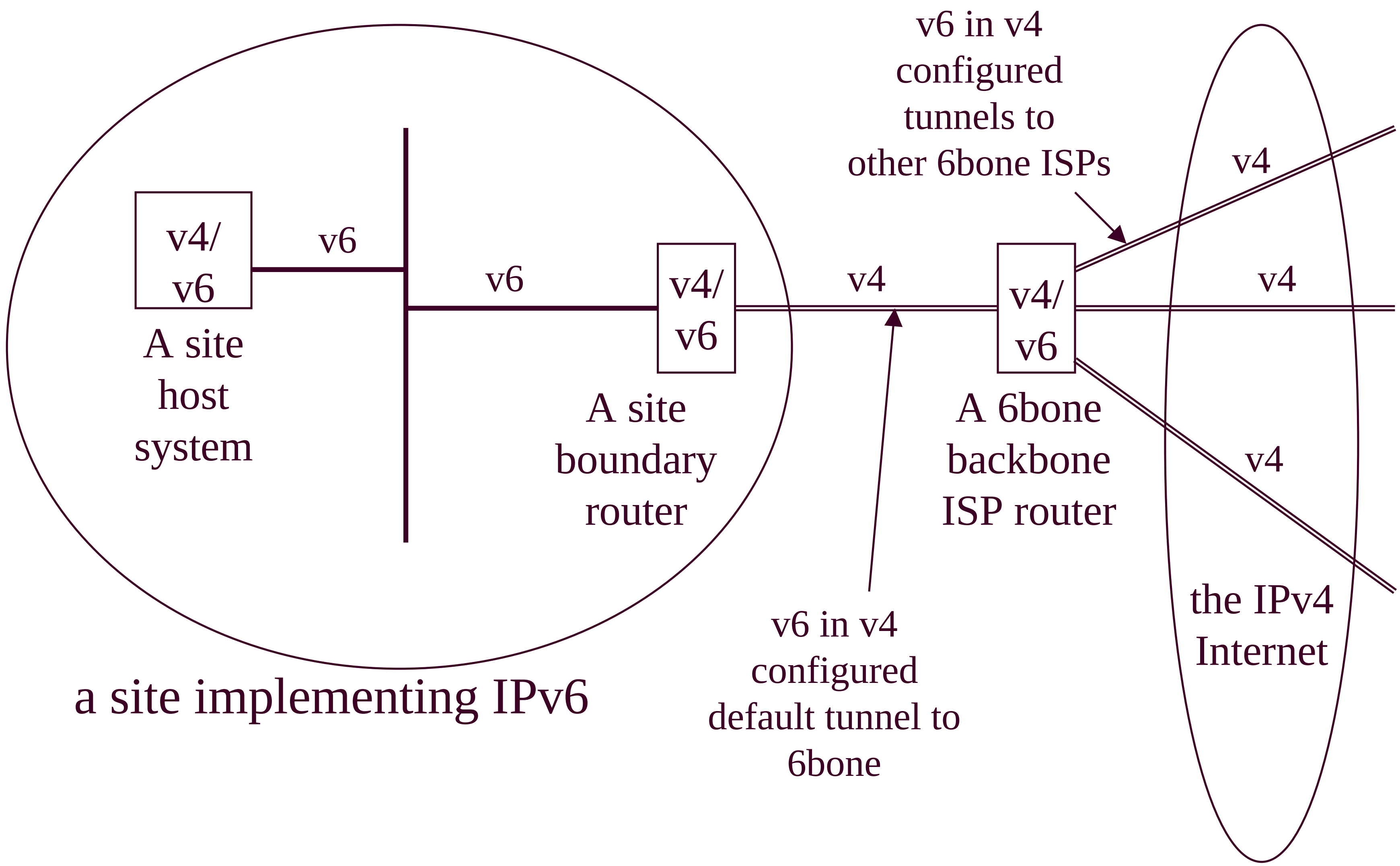
Basic Transition Mechanisms

The basic IPv6 transition strategy from RFC1933 specifies:

(this has been around for four+ years)

- **dual-stacks (in routers and hosts)**
- **IPv6 tunneled over IPv4 (protocol type 41)**
- **registration of an IPv6 address record in the DNS means the site has IPv6 connectivity somehow**

How these are used in the 6bone



Other good ideas

- **There are many transition mechanisms, but a few are more likely to prevail than others, at least for the traditional Internet world all of us know and love:**

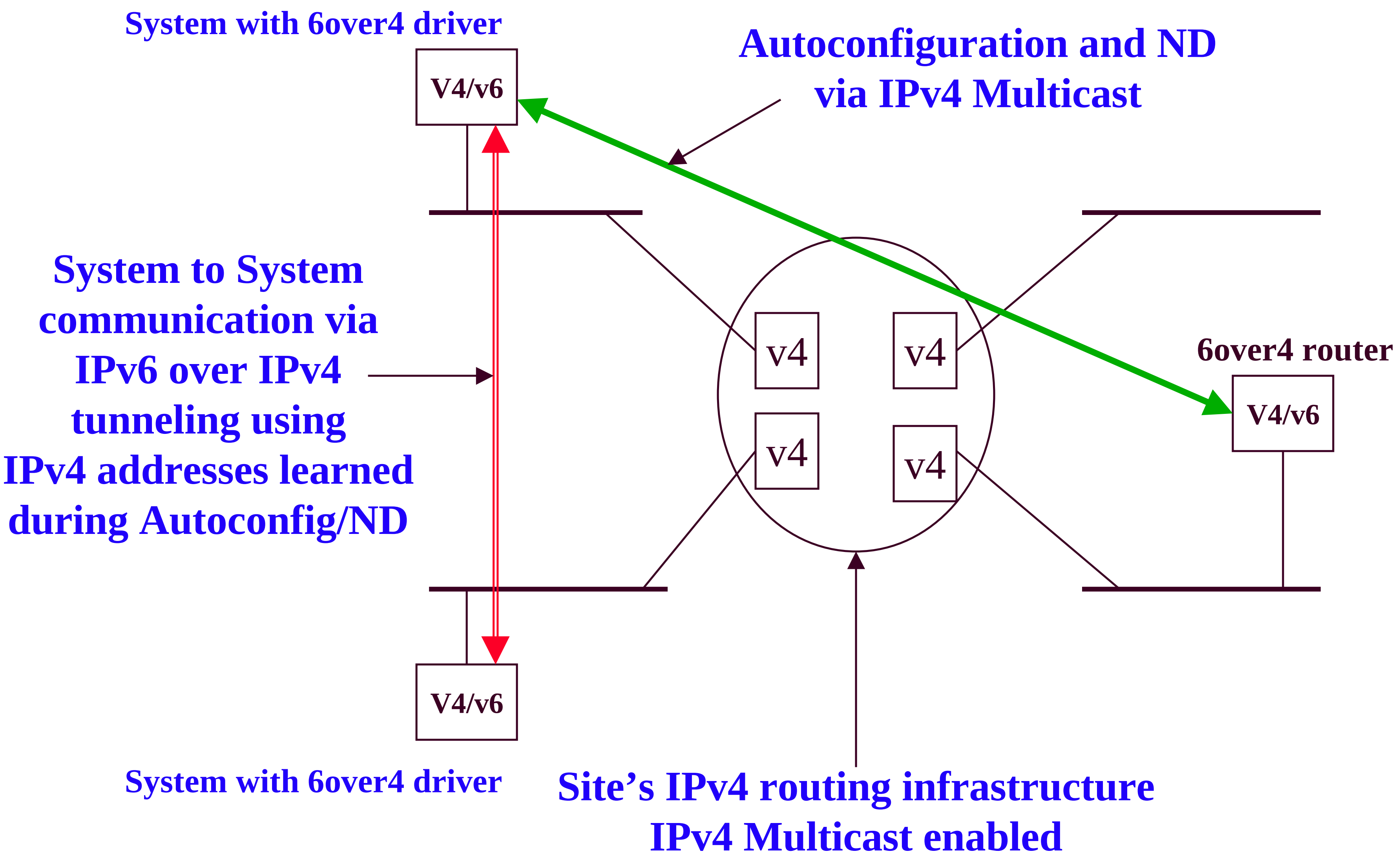
  6over4 **and** ***6to4***

- **New, non-traditional, uses of IPv6 may use other tricks from various types of translators to various application gateways**

6over4

- Developed as a basic v6 over v4 media driver
- it assumes a v4 multicast-capable domain exists between multiple dual v6/v4 stack hosts, and at least one v4/v6 6over4 capable router
- v4 multicast is used to discover the v4 tunnel endpoint of another v6/v4 host
- and a v6 over v4 tunnel is dynamically established as needed, host to host
- Biggest issue for usage is whether sites implement IPv4 multicast, but might be useful

How 6over4 works



6to4

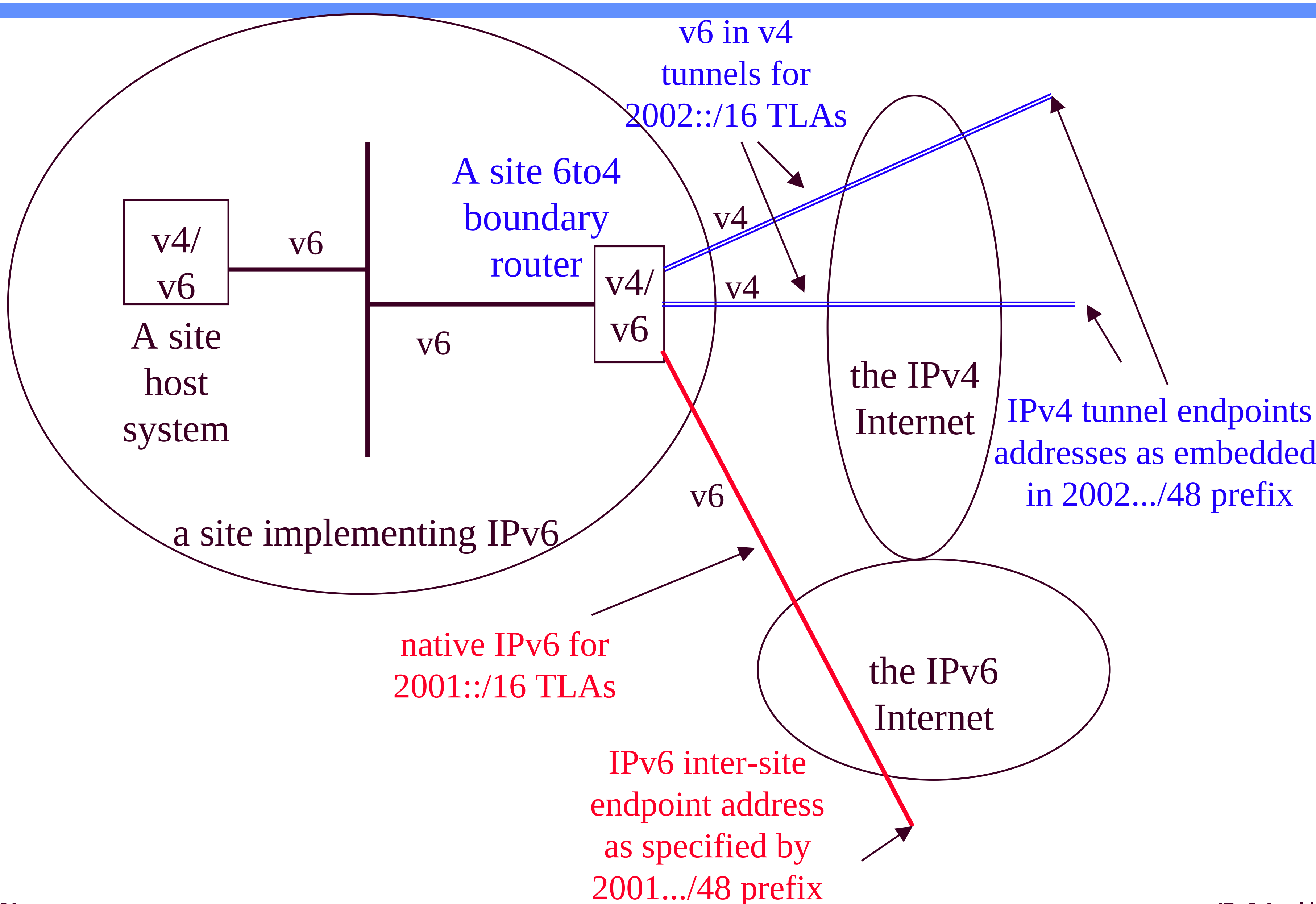
- **Specifies the 16-bit TLA prefix 2002::/16 as a “6to4” flag indicating that the 32-bit sized NLA below it carries an IPv4 Tunnel Endpoint Address of the site’s egress router**

3	13	32	16	64
001	TLA 0x002	IPv4 TEA	SLA	Interface ID

6to4

- thus an IPv6 host at a “6to4” site sending to another site with a ”6to4” TLA will have an IPv6 over IPv4 tunnel automatically constructed as the IPv6 packet passes through the site’s egress router
- requires careful source address selection so return traffic can make it back, but is considered as way to interconnect v6 clouds

A 6to4 example



My transition model for all this

- As more system and router vendors support IPv6 (including apps conversion and testing) we can start the real transition to IPv6 ...
- site net managers just turn on IPv6 in their site routers, and use “6to4” to reach other IPv6 sites
- site sys admins just turn on the IPv6 stack in their system boot configs they distribute for their site
- and systems speak IPv6 to those that support it, and IPv4 to those that don’t (DNS helps here)
- … and eventually IPv4 fades (all this 10-15 years)

An appropriate role for R&E folk

- Act in a leadership role to support IPv6 happening
- get some IPv6 going on your Campus to enable some application trials
- help some science program get a well chosen application that is both flashy to demo and reasonably easy to convert
- …keep the faith, and the Internet architecture

Thanks for listening

Pointer to everything IPv6: [6bone.net]()

This talk: [6bone.net/misc]()

Questions: fink@es.net