IPv4 and IPv6

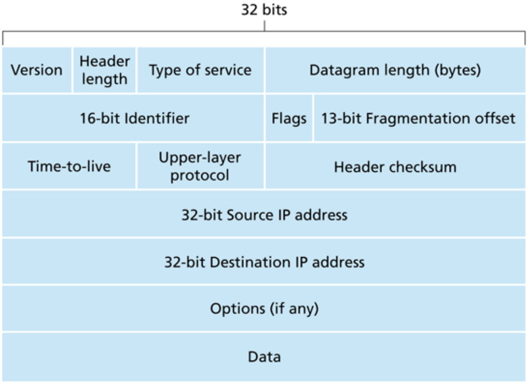
Daniel Zappala

CS 460 Computer Networking
Brigham Young University

Imagine Building IP

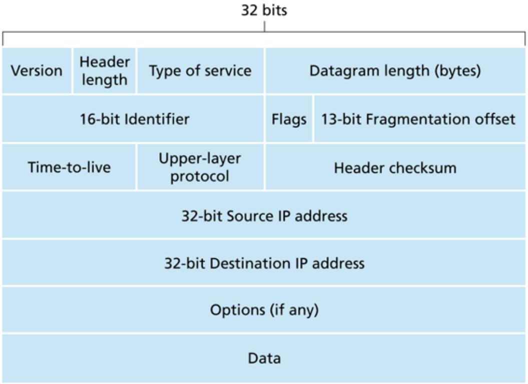
Common protocol for all networks
Must be very simple
Must last 40+ years

IPv4 Header Format



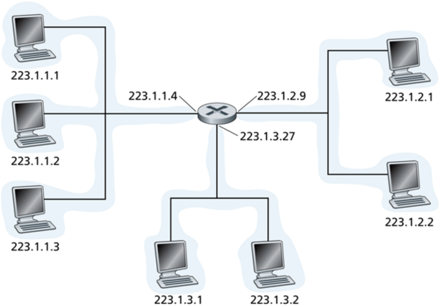
- header length: 20 bytes min.
- ToS: early attempt to route packets along paths with low delay or high bandwidth
- fragmentation: identifier, flags, offset

IPv4 Header Format



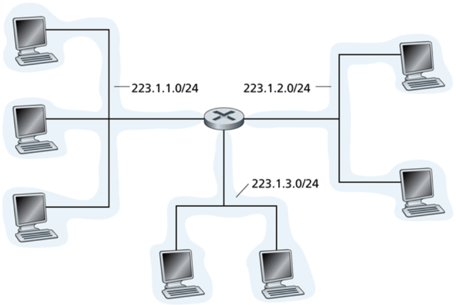
- TTL: used to literally be time (at least one second), now hops
- protocol: deliver to this protocol at destination
- options: includes timestamp, record route, source route

IPv4 Addresses



- 32 bits
- dotted-decimal notation: each part is 8 bits
- identifies an interface/link on a host or router

Subnets



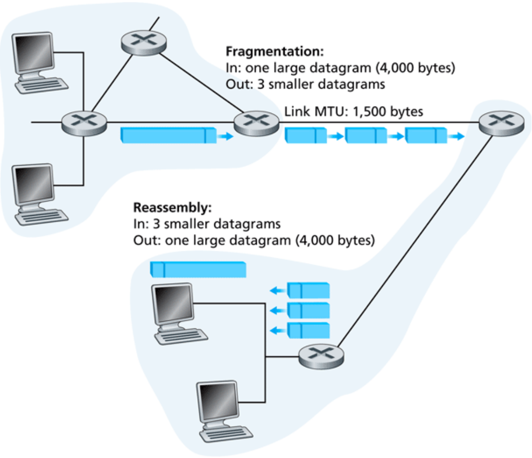
- IP address can be divided into subnet part (high-order bits) and host part (loworder bits)
- prefix notation: 223.1.1.0/24 indicates that the subnet is the high-order 24 bits
- interfaces whose IP addresses are on the same subnet can physically reach each other without a router

Forwarding Process

- check the destination address
  - is this one of my addresses?
    - send to next protocol specified in IP header
  - is this one of my subnets?
    - send to link layer to forward to the destination
  - do I have a route?
    - send to link layer to forward to next IP hop
  - destination unknown!
    - send an ICMP error to the source of the IP packet

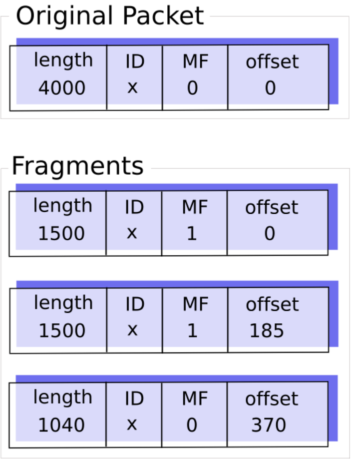
IPv4 Fragmentation and Reassembly

- each link has an MTU (maximum transfer unit) defining largest link-layer frame
- IP packets larger than the MTU must be fragmented
- reassembly only occurs at final destination
- uses IP fragmentation fields



IPv4 Fragmentation Example

- 4000 byte datagram, 1500 byte MTU
- how long will the fragmented packets be?
  - need 20 byte header
  - first two packets 1480 bytes
  - last packet is 3980 - 2*1480 + 20 = 1040
- what will the offsets be?
  - byte position in file / 8
- MF (more fragments) flag is set to 1 in all fragments except last
- ID must be unique to sender



ICMP

ICMP: Internet Control Message Protocol

- error reporting, ping
- network layer above IP: ICMP messages carried in IP datagrams
- ICMP message: type, code, checksum, message-specific data (RFC 792)

ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

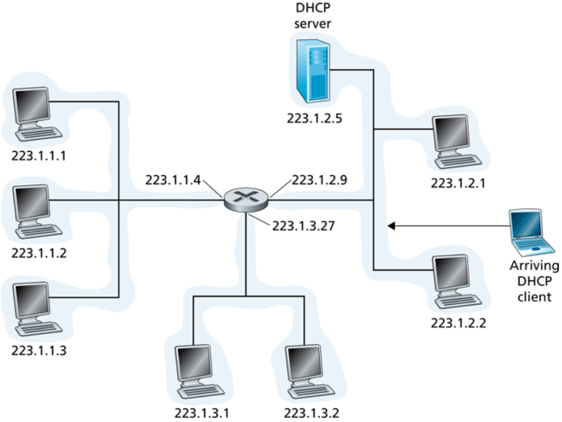
Traceroute and ICMP

- source sends UDP segments to destination
  - start with TTL = 1
  - increment TTL by one
  - use unlikely port number
- when nth datagram arrives at nth router
  - discard datagram (TTL expired)
  - send ICMP TTL expired message to source
  - message includes IP header, 64 bits of original datagram
- when ICMP message arrives, source calculates RTT for that hop
- traceroute takes 3 samples for each hop
- stop when ICMP returns a host unreachable packet, code 3 = port unreachable

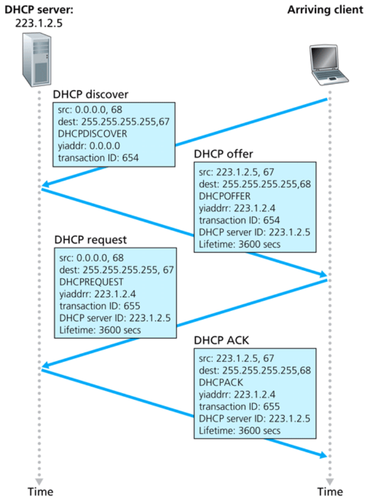
DHCP

DHCP

- IP addresses can be assigned manually
  - hard-coded into a configuration file
  - e.g. Gentoo: /etc/conf.d/net
- DHCP: dynamically get address from server



DHCP Transaction



NAT

NAT: Network Address Translation

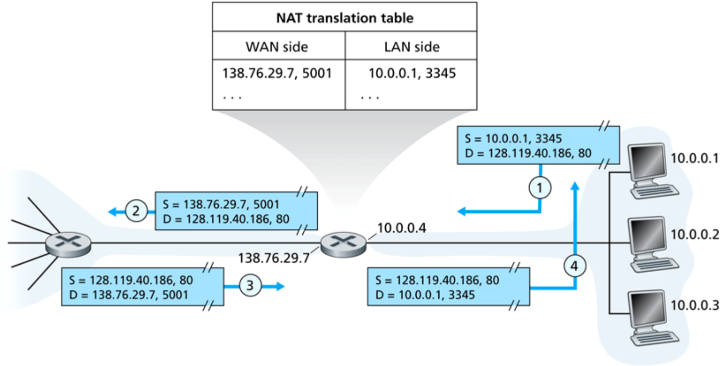


Figure 4.22 ♦ Network address translation

- use public port numbers to map to private connections
- can support 60,000+ connections with a single IP address

NAT: the Good ...

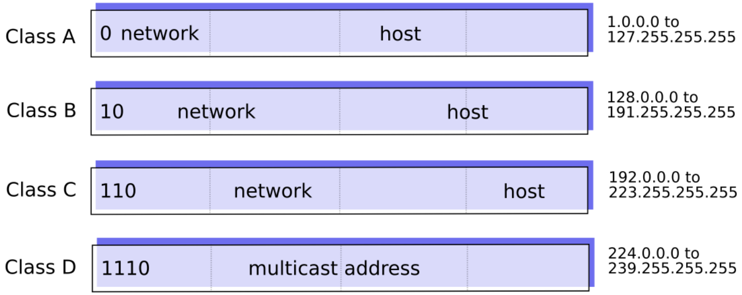
- conserves IP address space: private network only needs one IP address
- can change private IP addresses without notifying rest of Internet (DNS)
- can change ISP without changing IP addresses
- acts as a type of firewall - only reachable ports are those that you open first

...and the Bad

- layer violation: routers should only process IP, ports are in TCP/UDP
- violates end-to-end nature of Internet: any host can open a connection to any other host – makes running local servers and peer-to-peer applications hard
- address shortage should be resolved by IPv6
- individual computers should be made as secure as possible, rather than relying on firewalls or NAT boxes
- prevents many peer-to-peer applications from working
    - note: many emerging hacks and standards, including UPnP, that allow an application to create a mapping for a server running behind the NAT

CIDR

Classful IP Addressing



- used in early days of Internet to assign addresses to organizations
- led to waste: organizations want at least a B (65,000 addresses), even if they have 1000 machines (4 class Cs)
- early Internet users even got a class A (Stanford was 36.0.0.0)
- quickly ran out of addresses

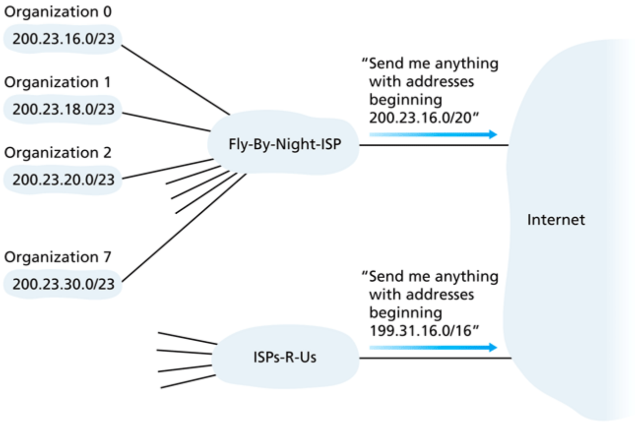
CIDR: Classless InterDomain Routing

- subnet portion of address has an arbitrary length
- address format: a.b.c.d/x, where x is number of bits in subnet portion
- example:
  - 11001000 00010111 00010000 00000000
  - 200.23.16.0/23
- enables conservation of IP address space, efficient routing
- IANA required organizations to return Class A, B addresses and re-number

Using CIDR Addresses

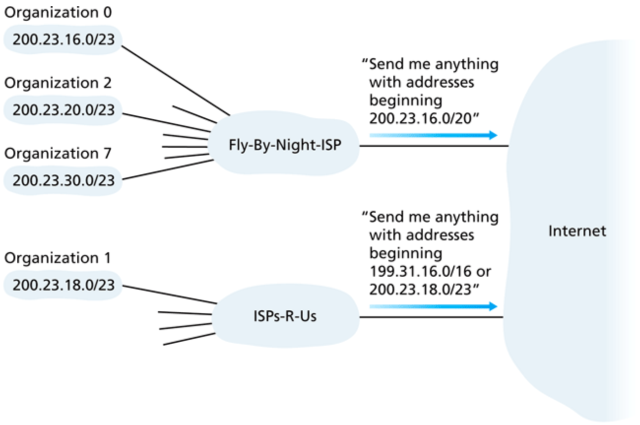
- each ISP has an assigned address space, from ICANN
- e.g 11001000 00010111 00010000 00000000 200.23.16.0/20
- can allocate to its customers
  - 11001000 00010111 00010000 00000000 200.23.16.0/23
  - 11001000 00010111 00010010 00000000 200.23.18.0/23
  - 11001000 00010111 00010100 00000000 200.23.20.0/23
  - ...
  - 11001000 00010111 00011110 00000000 200.23.30.0/23

CIDR and Route Aggregation



- using CIDR allows routes to be aggregated

Breaking Route Aggregation



- changing ISPs (and keeping IP addresses) breaks route aggregation

IPv6

Motivation: 32-bit address space running out

- short-term solutions
  - CIDR, reclaim class A addresses
  - NAT
- IETF coordinated design process, many proposals
  - discussion on big-internet and IPng lists
  - CATNIP - variable length addresses, interoperability among many protocols
  - NIMROD - variable length, hierarchical addresses, separate host identification (naming) from host location (routing)
  - TUBA - use CLNP for network layer, with OSI-specified big addresses
  - SIPP: Simple Internet Protocol Plus (Steve Deering) - 64-bit addresses, remove unneeded functionality
- and the winner is ...

Motivation: 32-bit address space running out

- short-term solutions
  - CIDR, reclaim class A addresses
  - NAT
- IETF coordinated design process, many proposals
  - discussion on big-internet and IPng lists
  - CATNIP - variable length addresses, interoperability among many protocols
  - NIMROD - variable length, hierarchical addresses, separate host identification (naming) from host location (routing)
  - TUBA - use CLNP for network layer, with OSI-specified big addresses
  - SIPP: Simple Internet Protocol Plus (Steve Deering) - 64-bit addresses, remove unneeded functionality
- and the winner is ...
  - SIP $\Rightarrow$ SIPP (SIP + PIP + IPAE) $\Rightarrow$ IPv6

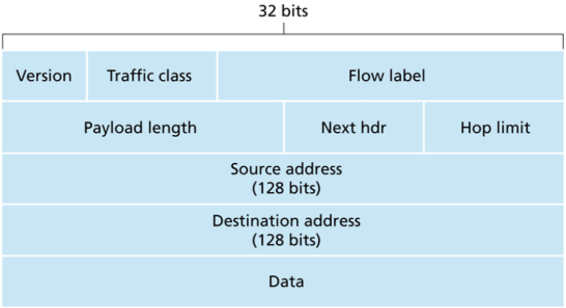
How Big Is Big Enough?

- proposals
  - fixed length, 64 bits
  - variable length, up to 160 bits
  - compromise: 128 bits
- theoretically perfect allocation
  - 128 bits $= 3.4 * 10^{38}$ addresses
  - $7 * 10^{27}$ atoms in your body, so $4.86 * 10^{10}$ addresses per atom
  - 6 billion people in the world
  - 8 billion addresses per atom in your body
- in reality ...
  - prefix (address type): 3 bits
  - registry ID: n bits
  - provider ID: m bits
  - subscriber ID: o bits
  - intra-subscriber ID : $125 - n - m - o$ bits
- address space can always be wasted

As Long As We're Designing a New Version of IP ...

- simplify IP header
  - get rid of functionality not used or needed in IPv4
  - speed processing/forwarding
  - no checksum
  - no fragmentation
  - fixed 40-byte header, no options
- support emerging QoS proposals
  - traffic class, flow label

IPv6 Header



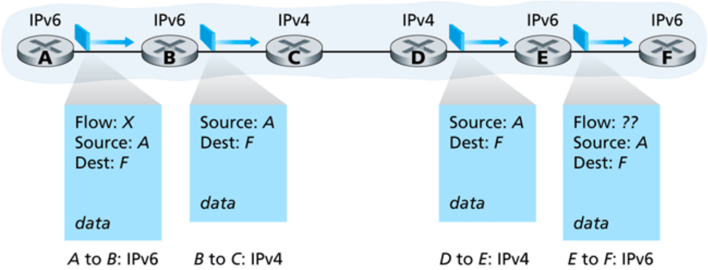
- traffic class: enable routers to map traffic into classes (delay, loss guarantees, etc)
- flow label: uniquely identify all packets for a particular flow/application, used for QoS
- Next Header: upper layer protocol or option

What Happened to IPv5?

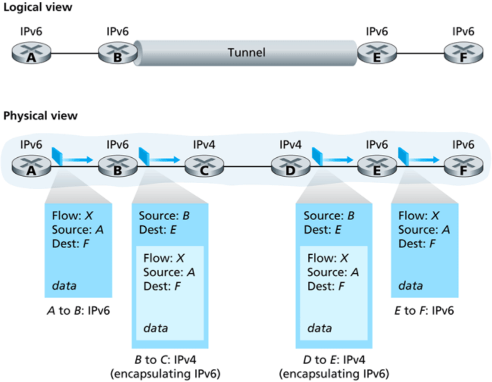
- 0-3 : unassigned
- 4 : IPv4
- 5 : ST (Stream Protocol), not used
- 6 : IPv6 (was SIP, then SIPP)
- 7 : CATNIP
- 8 : PIP
- 9 : TUBA
- 10-15 : not assigned

IPv6 Transition

- can't upgrade all routers at the same time or on the same day
- must interoperate between IPv4 and IPv6
- dual-stack: support both IPv4 and IPv6 in a single host/router
  - can deliver native IPv6 traffic where supported
  - loses IPv6 information when translating to IPv4



IPv6 Tunnels

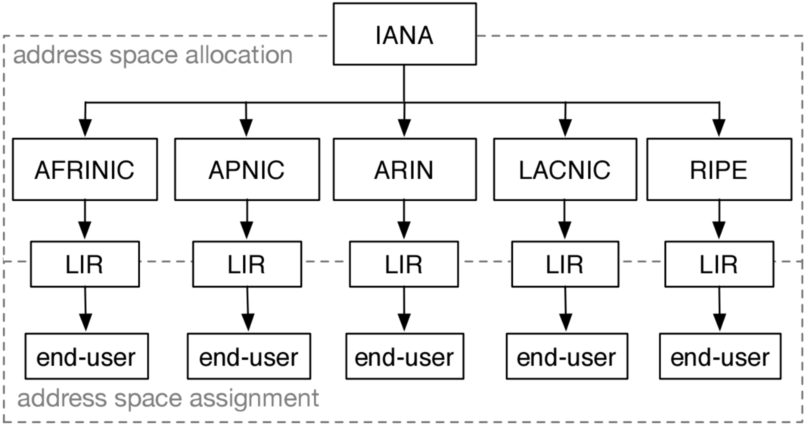


- tunnels: IPv6 carried as payload in IPv4 packet
  - can carry IPv6 packets end-to-end
  - requires configuration

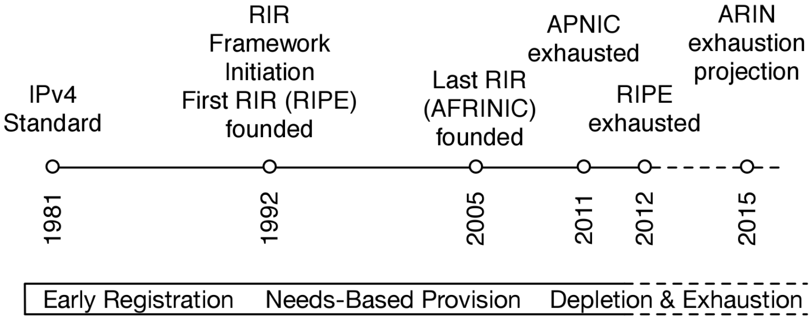
A Primer on IPv4 Scarcity

Philipp Richter, Mark Allman, Randy Bush, and Vern Paxson,
CCR, April 2015

Regional Internet Registry System

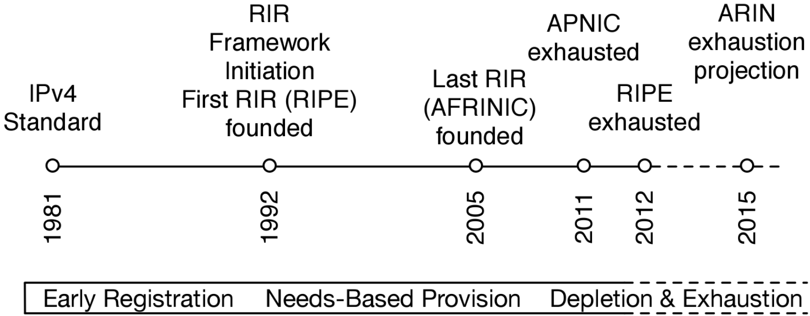


Evolution of Address Management



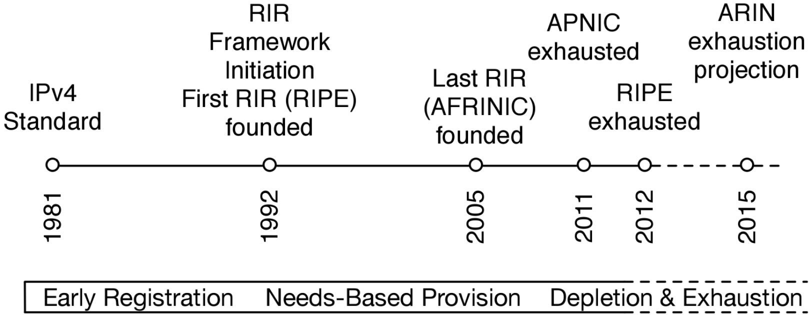
Initially, address blocks were allocated quite informally, with Jon Postel serving as the czar personally attending to each allocation. Postel periodically re-published RFCs enumerating the current address assignments ("please contact Jon to receive a number assignment").

Evolution of Address Management



The discussion at that time included the need to distribute the administration of IP address blocks to regional registries, covering distinct geographic regions to better serve the respective local communityconsciously fragmenting the registry. In addition, classless inter-domain routing (CIDR) and private address space arose in 1993-4 to further conserve publicly routable space.

Evolution of Address Management

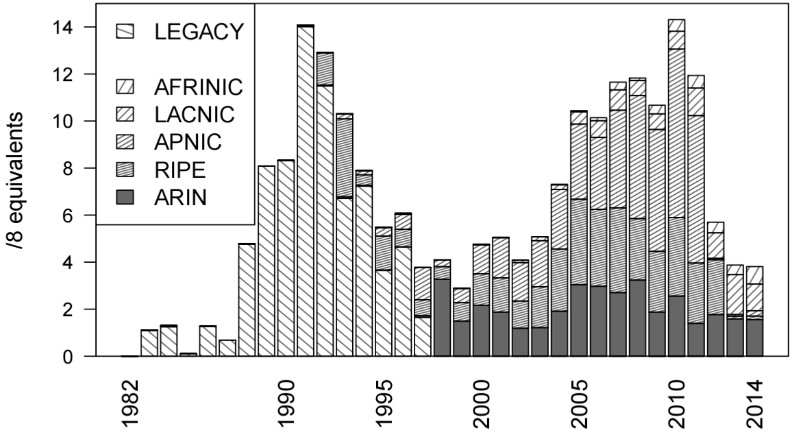


The five RIR communities agreed to a policy regarding address block allocation upon the onset of exhaustion, which ICANN ratified in 2009. The policy dictated that when the IANAs IPv4 free pool reached five remaining /8 blocks, the IANA would distribute these blocks simultaneously and equally to the five RIRs. In February 2011, the IANA allocated its last five free /8 address in accordance with the policy, one to each RIR. After that point, from a global perspective the pool of available IPv4 addresses was fully depleted.

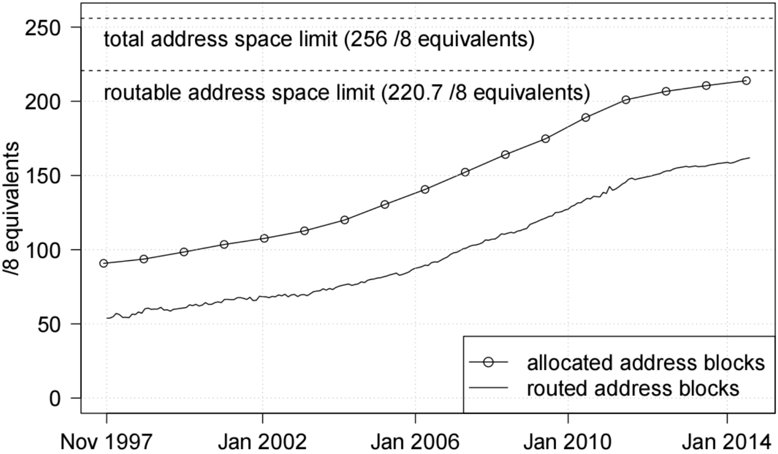
Address Space Statistics (February 2015)

	handed out /8s	**of which legacy /8s**	**available /8s**
ARIN	100.5	$\sim$ 64.9	0.35
RIPE	47.6	$\sim$ 11.93	0.97
APNIC	51.0	$\sim$ 4.40	0.74
LACNIC	10.9	$\sim$ 0.58	0.20
AFRINIC	4.5	$\sim$ 0.02	2.63
total	**214.5**	$\sim$ **81.83**	**4.89**
% of routable	**97.2%**	$\sim$ **37.1%**	**2.2%**

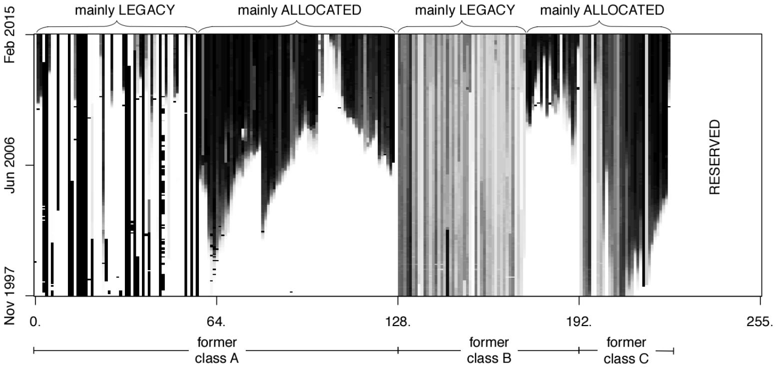
Yearly Allocations



Allocated and Routed Address Blocks



Evolution of the Distribution of Routed Address Blocks



Address Markets

- All but AFRNIC allow address space transfers, and some offer listing services
  - RIPE had 1 million available addresses listed, 17 million requested addresses
  - trades possible between ARIN and APNIC
  - prices up for negotiation
- possible to transfer outside of RIR control
  - not entirely under RIR control
  - legacy addresses particularly vague regarding control
  - Nortel bankruptcy -¿ 660K addresses to Microsoft
- *not much prevents any party from unofficially transferring an address block to another*
- not sure this can even be detected

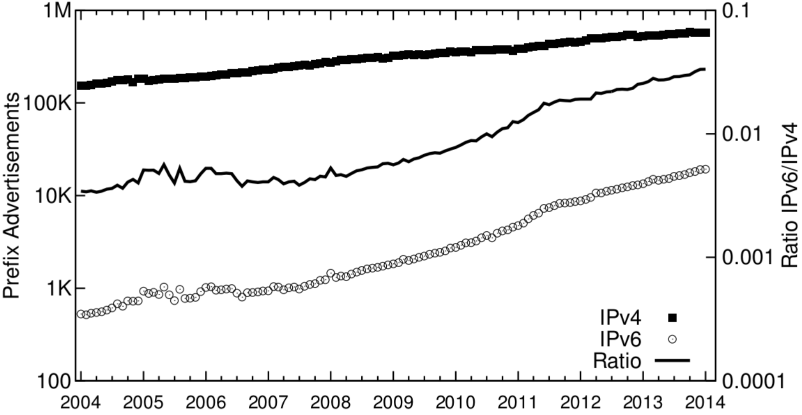
Overcoming Scarcity

1. IPv6!
   - *As of February 2015, Google reports some 4.5% of clients accessing Google to be IPv6 enabled, with adoption rates as high as 28% in Belgium, around 10 to 15% in the US and Germany, and increasing support in other European countries. Nonetheless, the per-host adoption rate still ranges at or below 1% for most countries, including China, India and Russia*
2. ISPS beginning to use carrier-grade NAT – 3% of Internet users
3. use address space more efficiently – reallocate/sell addresses

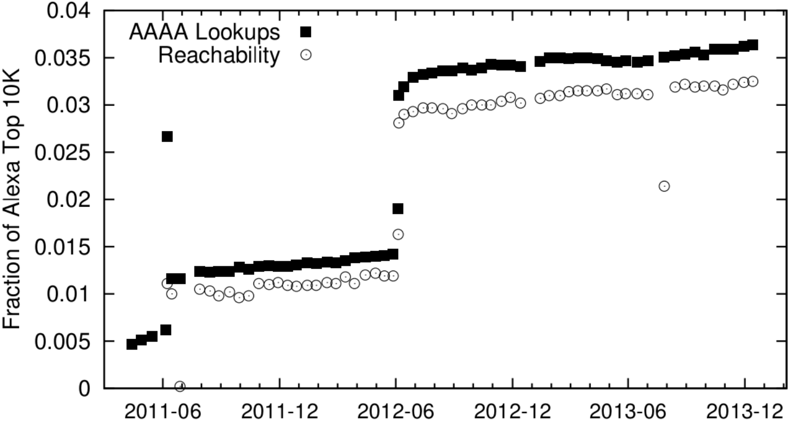
Measuring IPv6 Adoption

Jakub Czyz, Mark Allman, Jing Zhang, Scott Iekel-Johnson, Eric Osterweil, and Michael Bailey, SIGCOMM, 2014

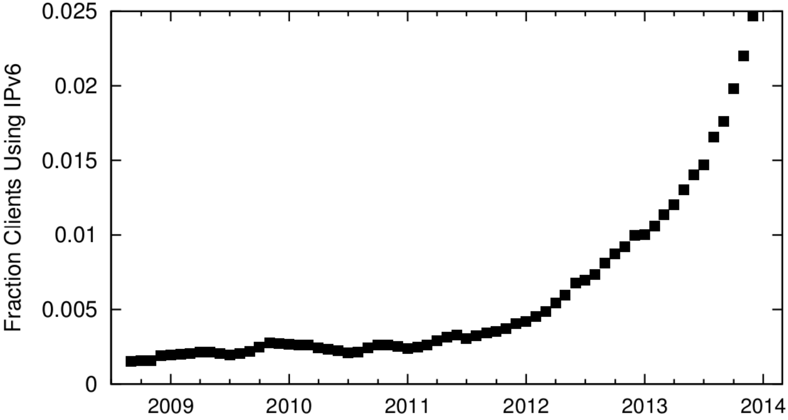
Advertised Prefixes



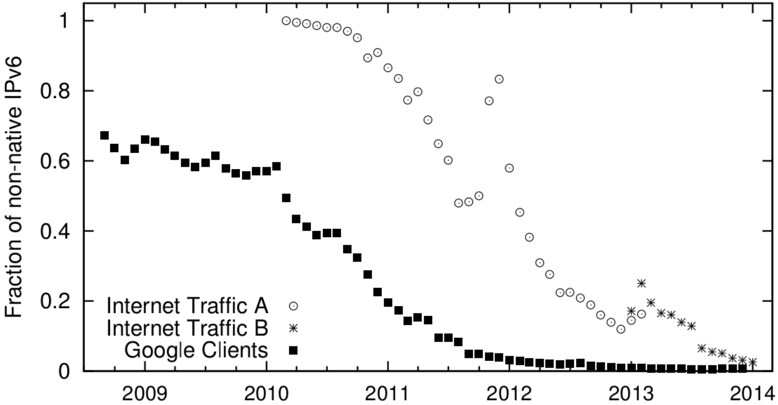
Alexa Top 10K Deployment



Google Clients Reachable via IPv6



IPv6 Native Traffic



IPv6 Usage

Table 6: Measures of actual operational characteristics of IPv6, recently and three years ago. These suggest that IPv6 is now mature. We contend that IPv6, as a real, production protocol, has finally come of age.

Metric: Operational Aspect Measured	IPv6 Status at End of:	
	2010	2013
U1: IPv6 Percent of Internet Traffic	0.03%	0.64%
U1: 1-yr. Growth vs. IPv4 (*Mar-2010 – Mar-2011)	−12%*	+433%
U2: Content's Portion of Traffic (HTTP+HTTPS)	6%	95%
U3: Native IPv6 Packets vs. All IPv6	9%	97%
U3: Native IPv6 Google Clients	78%	99%
P1: Performance: 10-hop RTT^{-1} vs. IPv4	75%	95%