Deploying New Internetworking Technologies with Linux

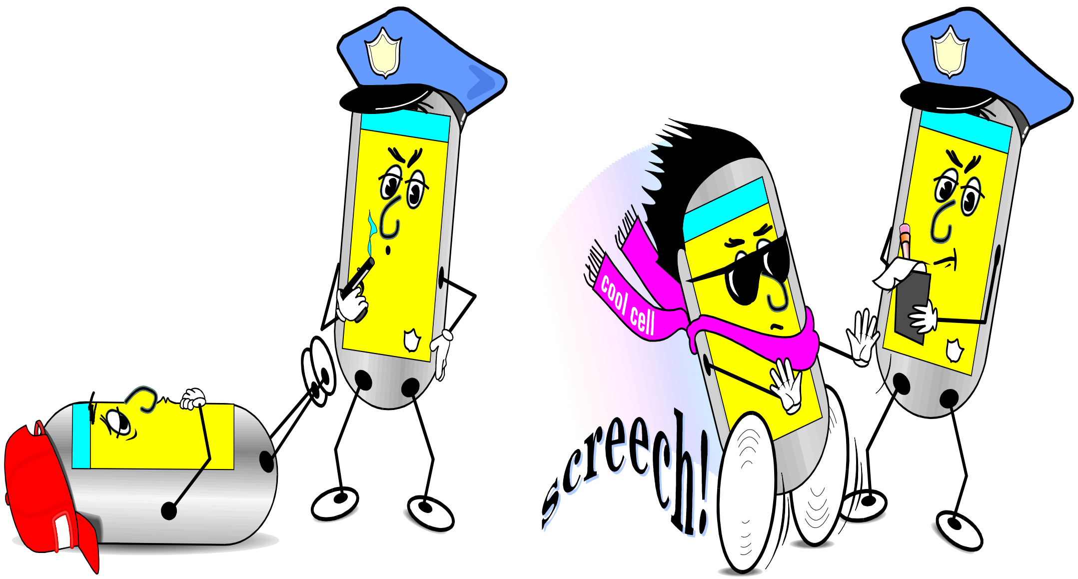
Karthik Prabhakar
karthik@corp.hp.com

HP WORLD 2002
Conference & Expo

Agenda

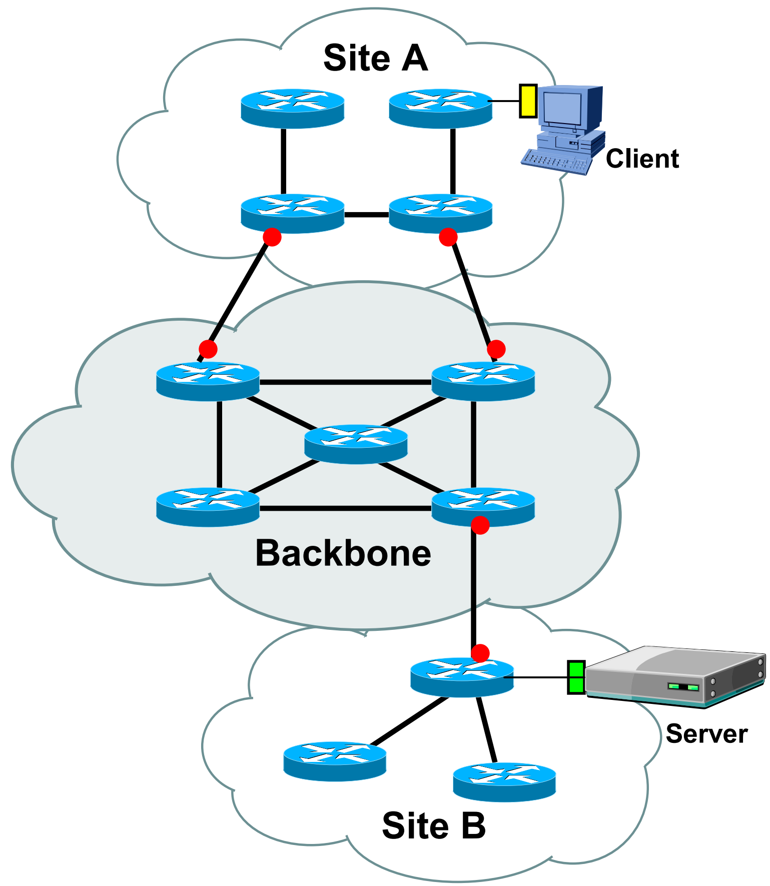
- QoS Technologies, Diffserv and ECN
  - Linux Policy Routing as a QoS-assist
- IP Security & IPSec Trends
- IP Multicast
- Enhancements to the DNS
- Moving to IPv6: Why, and How
  - Linux tools to assist with IPv4 -> IPv6 migration

QoS Technologies

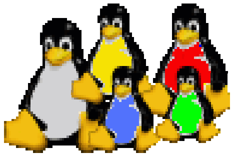


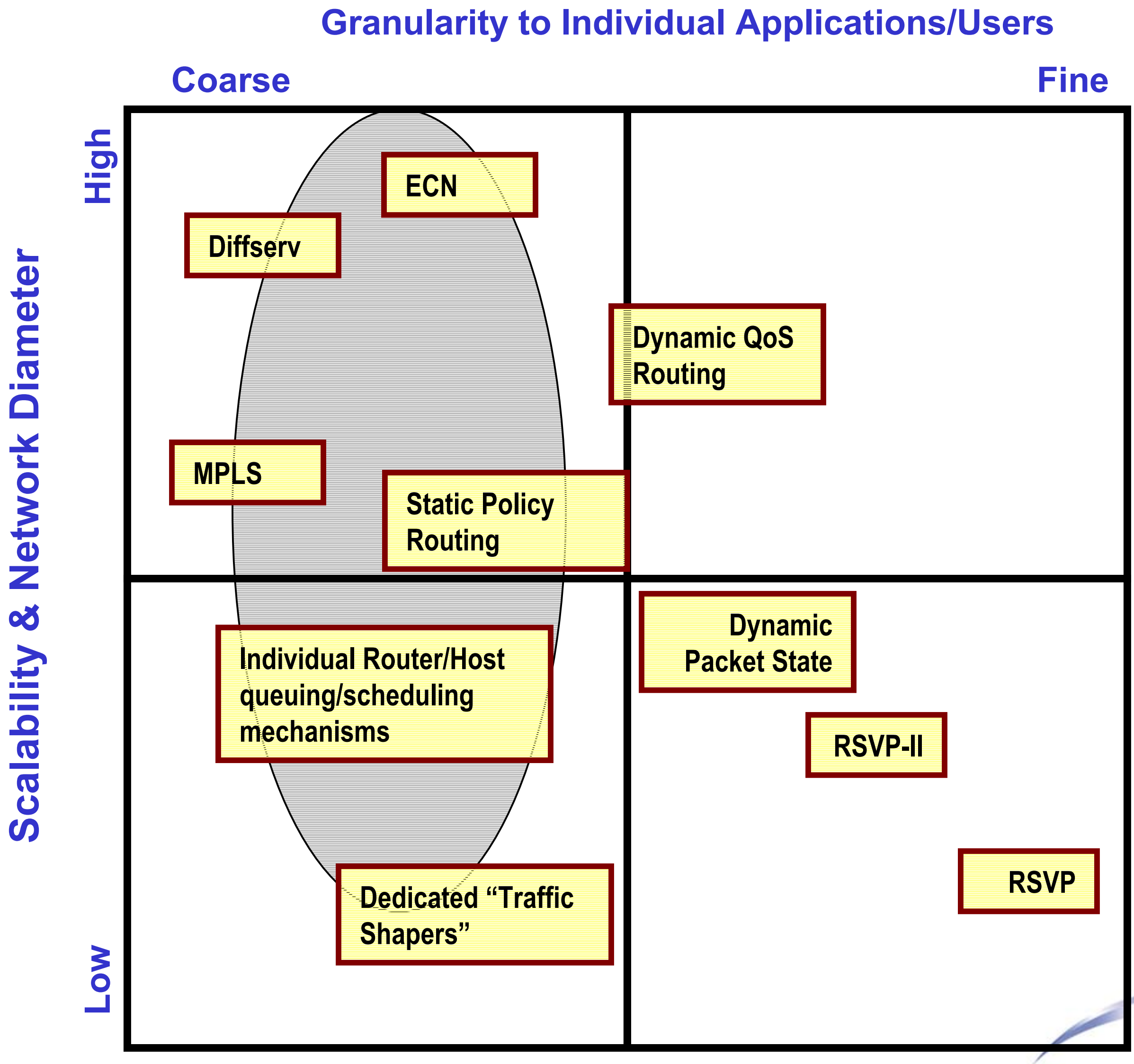
QoS: Why and Where?

- ”Quality” of Services is typically a subjective measure
- Usually, it is ineffective if not applied on an end-end basis
- Prefer standards-based approaches over the latest *snake-oil*
  - Watch out for violations of the end-end principle
  - Nudge vendors who do not support recent standards important for Internet evolution
- Both routers and hosts have a role

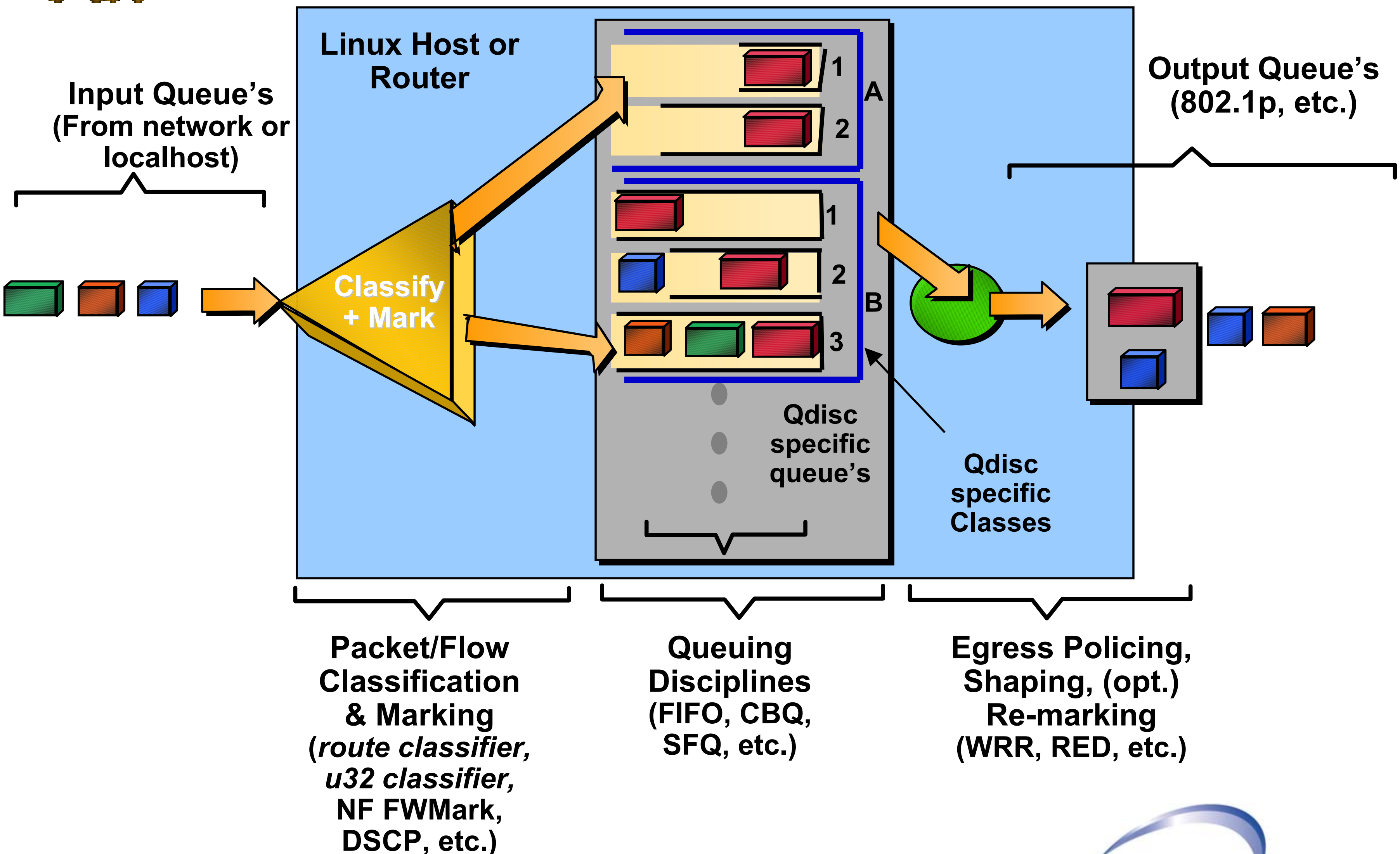


Classifying QoS & Flow-Control Techniques





Linux QoS Conceptual Overview

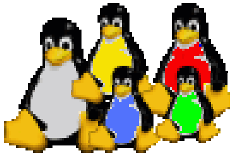


Example: Simple CBQ Configuration
(Limit outgoing email to 500 Kb/s using Netfilter+CBQ)

- **Enable necessary modules**
  - *modprobe iptable_mangle; modprobe ipt_mark; modprobe ipt_MARK*
  - *modprobe sch_cbq; modprobe cls_fw*

- **Filter on all tcp packets from port 25**
  - *iptables –I OUTPUT –t mangle –p tcp –d 10.0.0.0/8 –dport 25 –j MARK –set-mark 1*

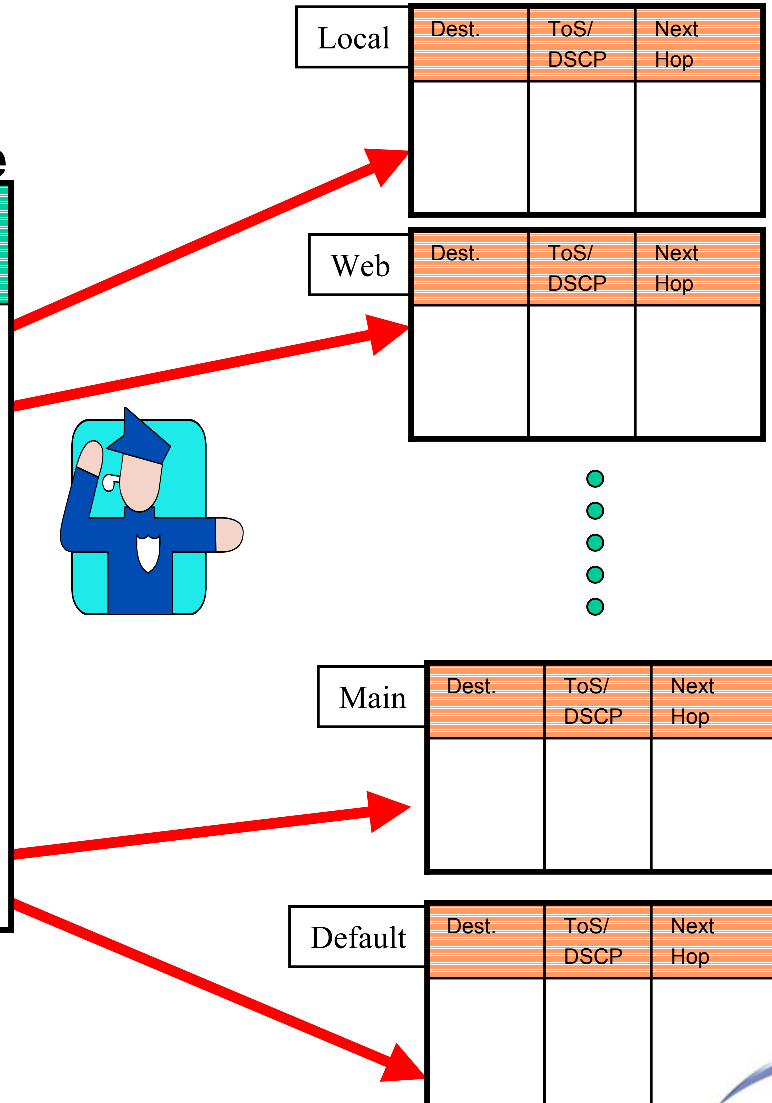
- **QoS Rules (CBQ) – limit flow to 500kb/s on 10Mb/s link**
  - *tc qdisc add dev eth0 root handle 10: cbq bandwidth 10Mbit avpkt 1200*
  - *tc class add dev eth0 parent 10:0 classid 10:1 cbq bandwidth 10Mbit rate 10Mbit allot 1514 weight 10Mbit prio 8 maxburst 20 avpkt 1200*
  - *tc class add dev eth0 parent 10:1 classid 10:200 cbq bandwidth 10Mbit rate 500Kbit allot 1514 weight 50Kbit prio 8 maxburst 20 avpkt 1200 bounded*
  - *tc filter add dev eth0 protocol ip parent 10:0 prio 8 handle 1 fw classid 10:200*



Policy Routing in Linux

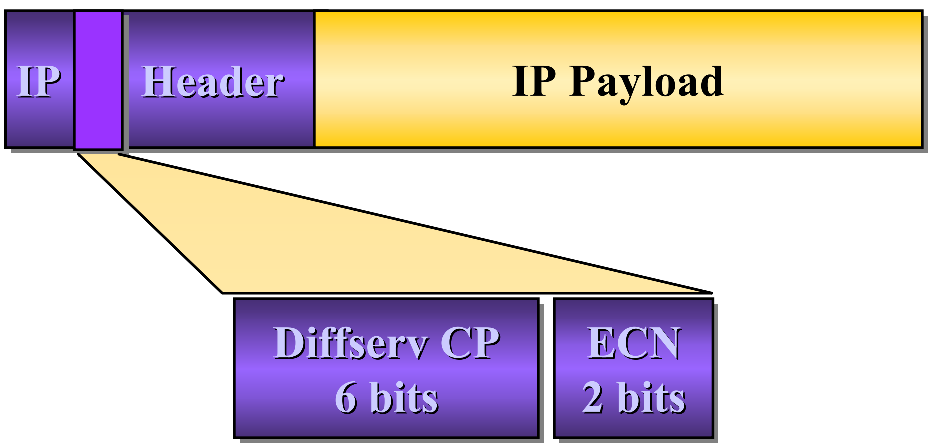
Routing Policy Database

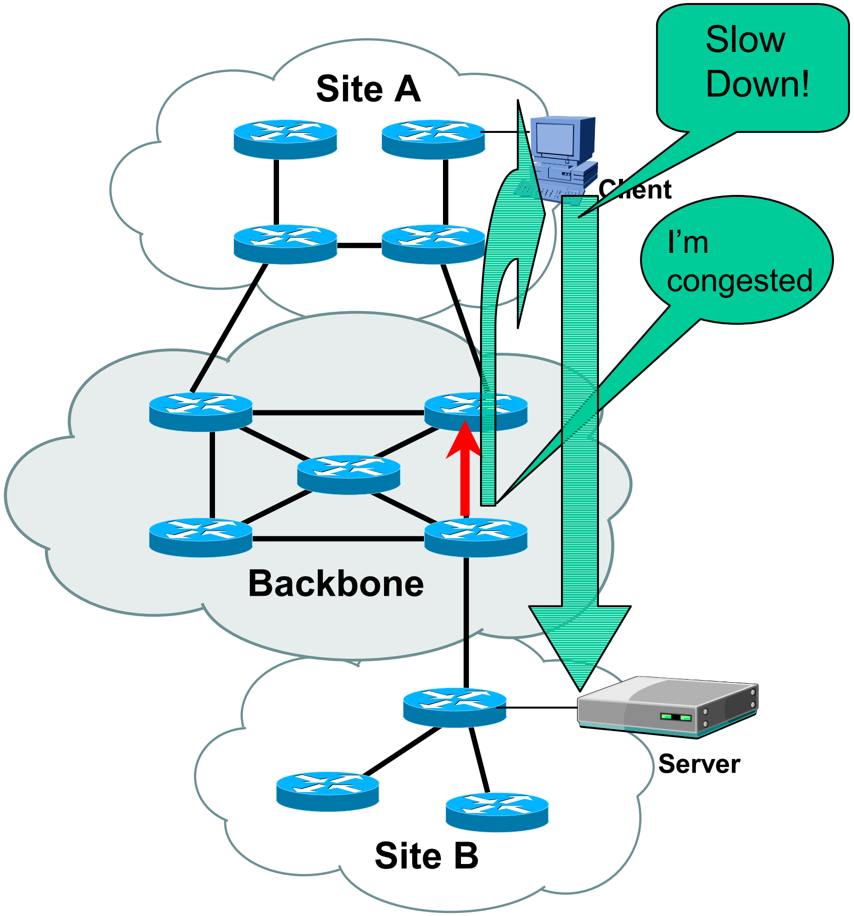
Priority	Rule to Apply	Table to Lookup	Realm
0	Any	Local	
2	fwmark=7	Web	5
6	src=10.7.4.1	Mgmt	
22	DSCP=5	Cust	8
.	.	.	
.	.	.	
.	.	.	
.	.	.	
.	.	.	
.	.	.	
.	.	.	
32766	Any	Main	
32767	Any	Default	



Explicit Congestion Notification (ECN)

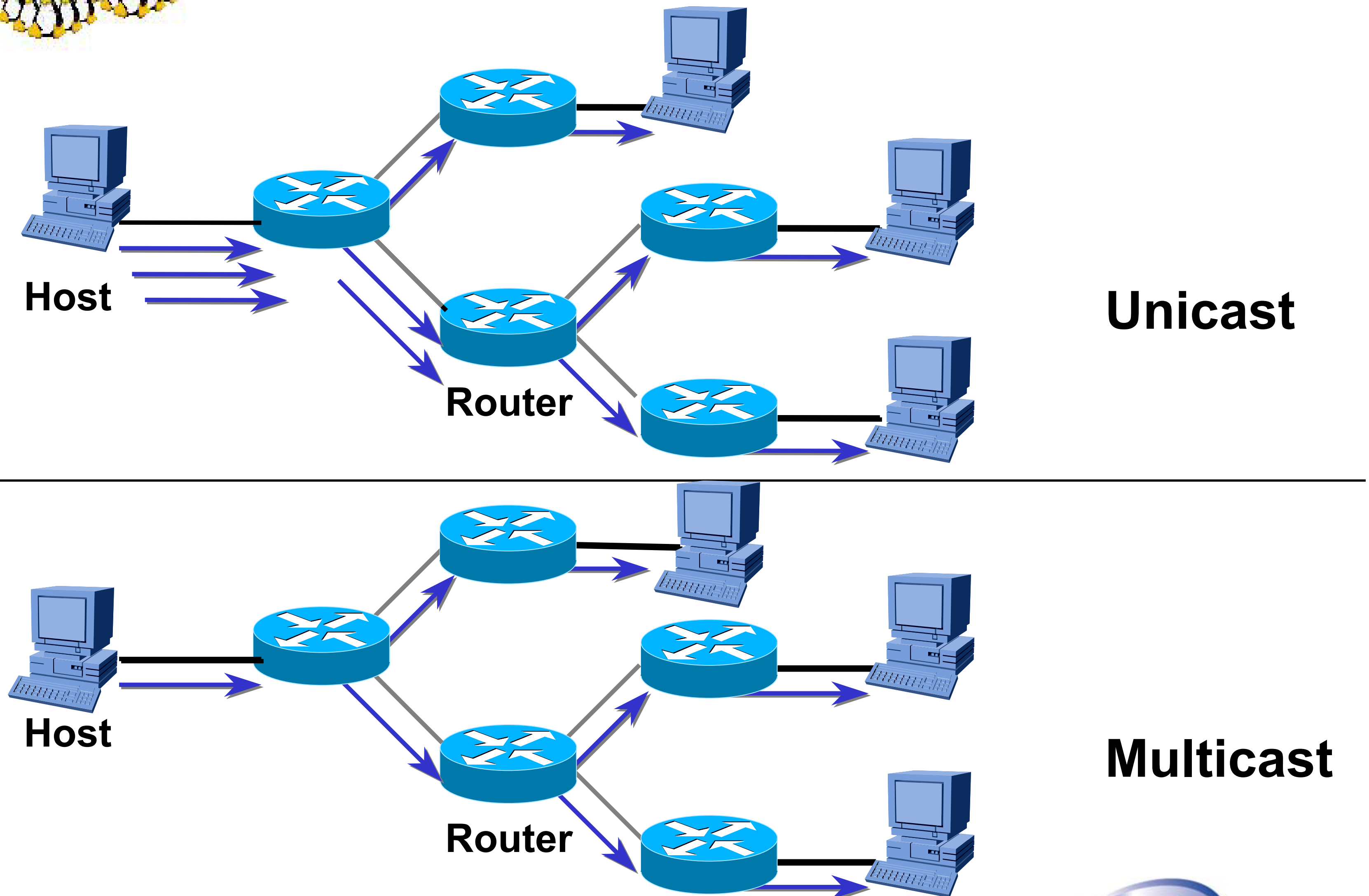
- New standard for Flow control in the Internet
  - Allows routers to signal congestion to endpoints
  - Allows hosts to slow down before packet loss begins
- Not all host vendors have implemented ECN yet

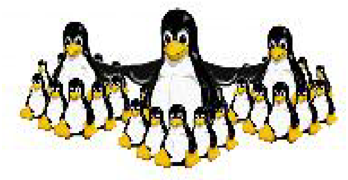




IP Multicast on Linux

Unicast Vs Multicast





IP Multicast Protocols

- Internet Group Management Protocol (IGMP) – v2, v3
- Older Multicast Routing Protocols
  - DVMRP
  - MOSPF
  - CBT
- Protocol Independent Multicast (PIM)
  - PIM Dense-Mode
  - PIM Sparse-Mode
  - PIM Source-specific Mode (SSM)
- Multicast Source Discovery Protocol (MSDP)
- Border Gateway Multicast Protocol (BGMP)

Internet Group Management Protocol (IGMP)

- Routers use *Membership query* to solicit multicast group membership from directly connected hosts
- IGMPv2 adds message type for hosts to *Leave Group,* so that prunes can be sent immediately (if no other receivers express interest)
  - IGMPv1 simply allowed membership query to timeout
- IGMPv2 is the most commonly deployed version in most current host stacks

IGMPv3

- Adds ability for host to indicate which groups it is interested in, and from which source IP addresses
  - Two options:
    - Include (S1,S2,….Sn, G)
    - Exclude (S1, S2, ….Sn, G)
  - IGMPv2 Equivalents are Include (*,G) or Exclude(NULL,G)
- Implementation for linux
  - http://www.sprintlabs.com/Department/IP-Interworking/multicast/linux-igmpv3/ (not maintained anymore)
  - ftp://ftpeng.cisco.com/igmpv3linux/ (More recent port of an implementation for FreeBSD)
- Requirement for *PIM-Source Specific Multicast*

Internet Multicast Backbone (MBone)

- Still commonly tunnels between providers & enterprises running DVMRP
  - Some providers migrating to native multicast routing protocols (PIM for intra-domain, and MSDP for inter-domain)
- Mrouted is trivial to configure, and is the dominant DVMRP implementation on the MBone
  - v3.9 beta 3 preferred to v3.8 (works well on linux)
    - Simple *mrouted.conf:*

```
Tunnel 10.5.12.34 192.168.17.35 ratelimit 256 metric 1
phyint eth0 ratelimit 512
phyint eth1
```

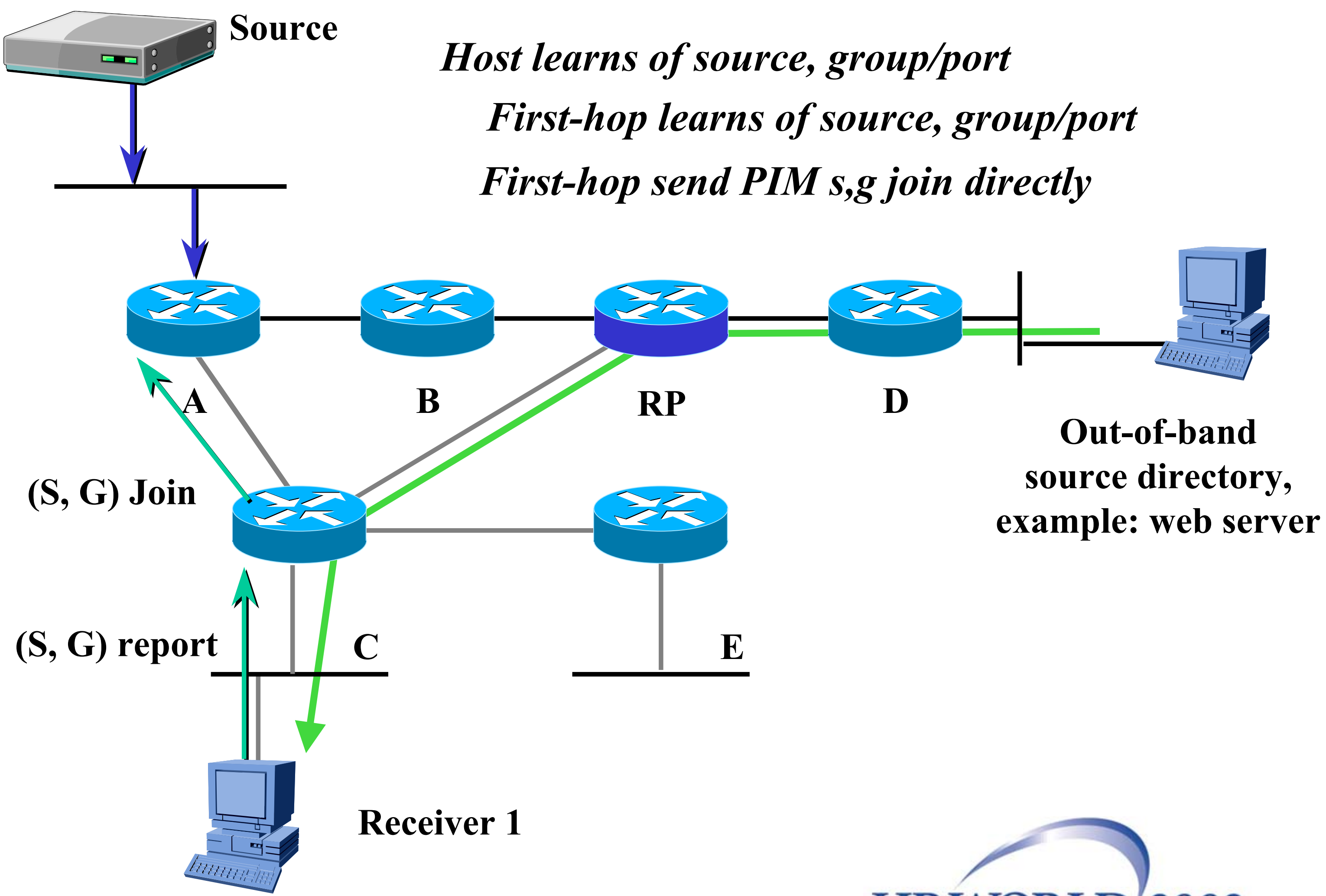
- Due to complexity of other alternatives, PIM-Source Specific Mode is gaining popularity for One->Many multicast transmissions
  - IGMPv3 is a pre-requisite

PIM Source Specific Mode

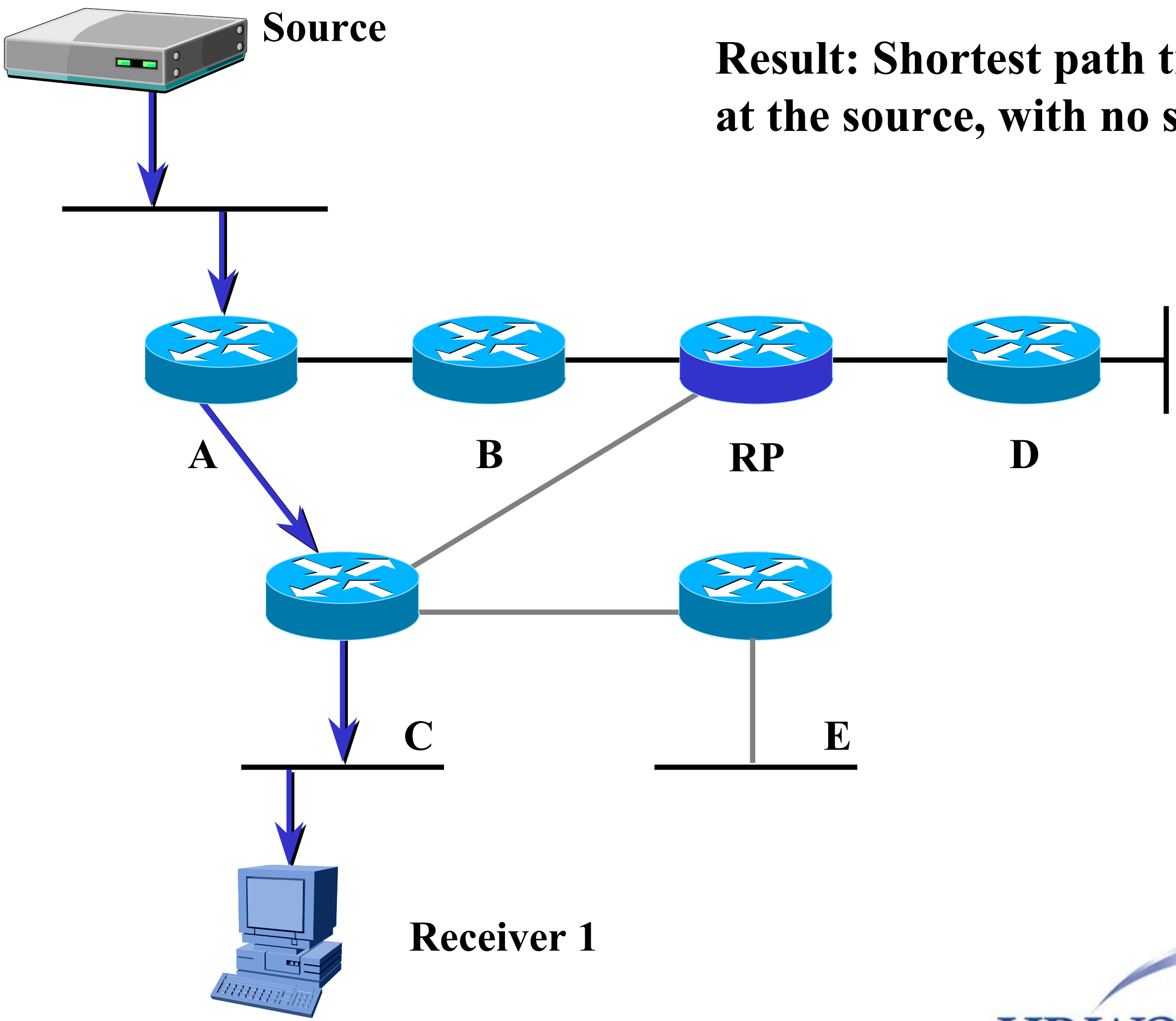
Host learns of source, group/port

First-hop learns of source, group/port

First-hop send PIM s,g join directly



PIM Source Specific Mode



Result: Shortest path tree rooted at the source, with no shared tree.

Hot Topics in DNS

- DNSSec
  - Delegation Signer
- The TSIG skirmish
- AAAA vs. A6 battle
- EDNS
- Protecting the DNS against attacks

DNSSec Performance

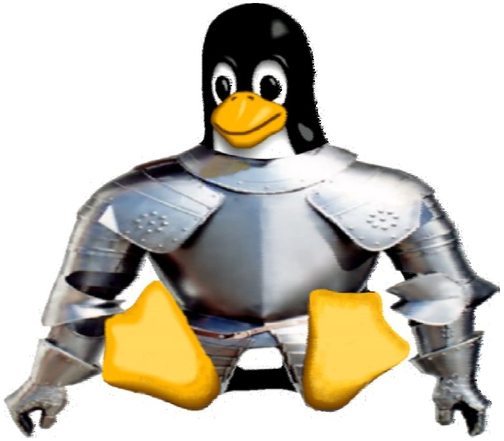
<Placeholder for DNSSec signing performance on Linux/IA-32 and Linux/IPF>

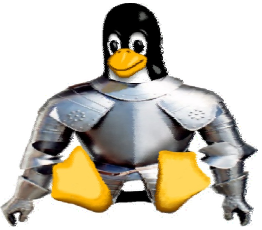
- Use of optimized assembly code for openssl & DNSSec performance on IPF
- DNSSec signing performance (and quality) can be improved by the use of */dev/random* in Linux
  - Possible sources of entropy include attached peripheral devices, network traffic/noise, etc.
  - Use of Intel Random Number Generator if motherboard supports it

Simple Tricks to limit DNS attack damage

- In addition to usual DNS/Bind precautions:
  - Rate limit tcp connections to DNS ports
    - Limit damage from DNS syn attacks
  - Rate limit outgoing UDP traffic
    - Limit damage caused by DoS attacks through DNS traffic amplification
      - DNS responses are typically much larger than DNS requests
      - Especially in conjunction with DNSSec
  - Prioritize traffic from known/trusted servers over unknown ones
    - For e.g., traffic from trusted slaves or clients gets classified into a different class from other DNS traffic
  - Use (with caution) IP anycast or similar mechanisms to spread DNS query load amongst many servers

IPSec and IP Security Trends





FreeS/Wan:

FreeS/WAN

- Continued support of kernel IPSec (KLIPS) and Internet Key Exchange/IKE (Pluto)
  - Slow evolution for IPSec for IPv6 (USAGI IPSec implementation is a much better option for Linux IPSec for IPv6)
  - Support for IPComp (pre-encryption compression support)

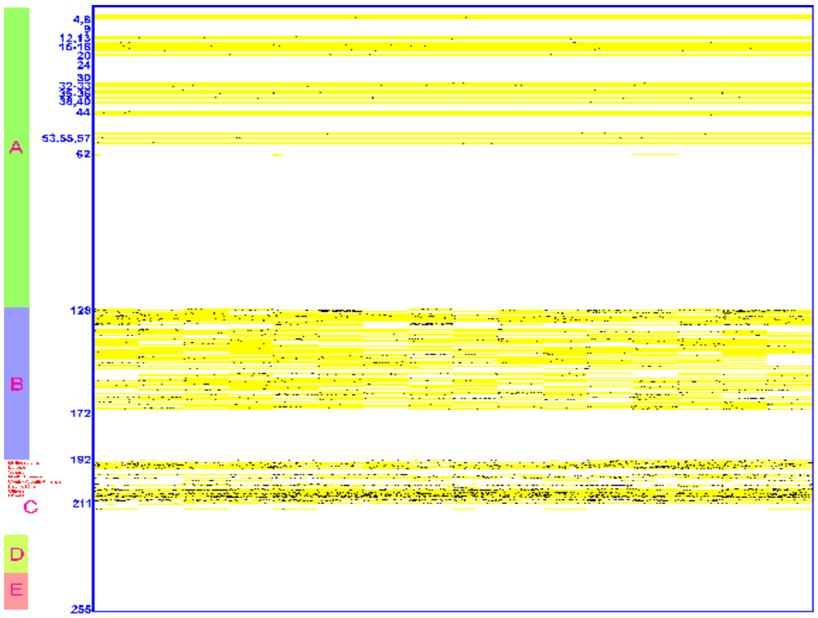
- Work in progress at the IETF on “Son of IKE” (SOI)
  - Various proposals under consideration to streamline/simplify IKE

- Frees/wan not part of main kernel tree yet
  - Many distributions (for e.g., Mandrake 8.2) include freeswan.

IPv6: Why and How

Why IPv6?

- Address Space and Growth
  - Allocations have slowed, but not due to lack of demand
- IP attached devices and systems continue to burgeon



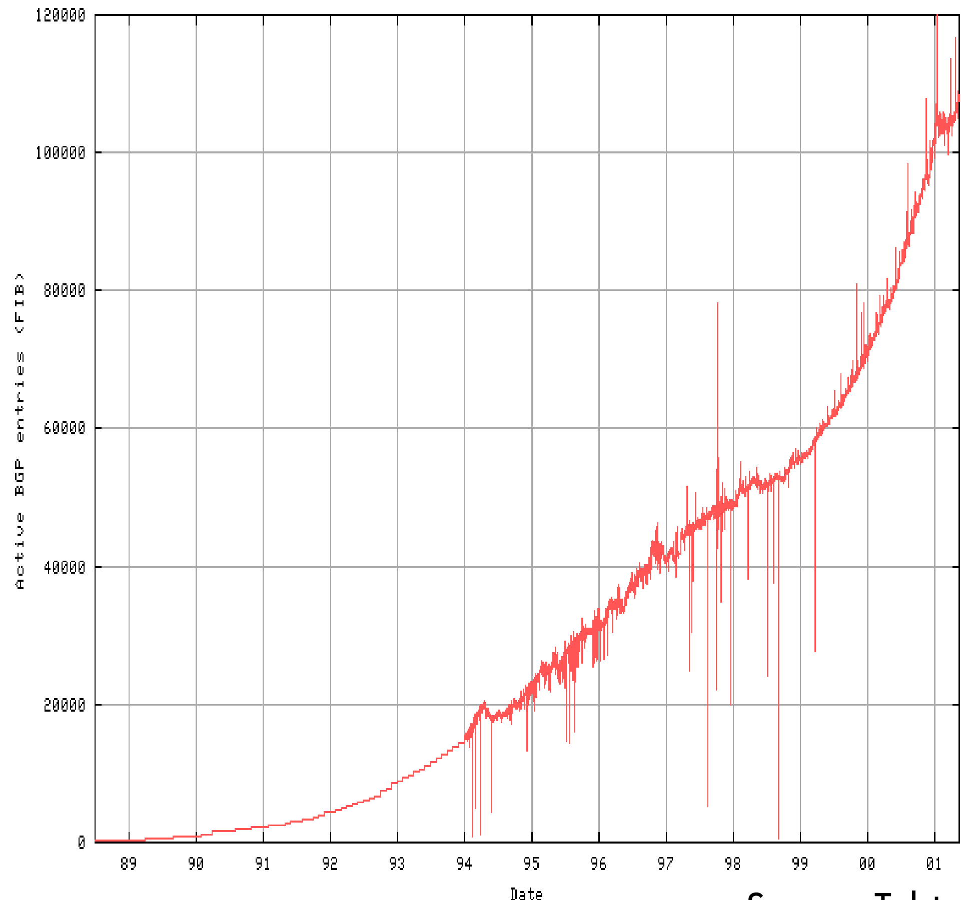
IPv4 Address Allocations

IPv4: 4,294,967,296 unique addresses

IPv6: 340,282,366,920,938,463,463,374,607,431,768,211,456 unique addresses

Why IPv6?

- Address Space and Growth
- Internet Routing Table Growth
  - Increased multihoming of networks
  - Increasing richness of interconnectivity
- IPv6 uses the same fundamental routing protocols as IPv4
  - But IPv6 and 6Bone Routing policy tries to encourage aggregation



Source: Telstra

Why IPv6?

- Address Space and Growth
- Internet Routing Table Growth
- Internet Features integrated into the base protocol
  - Security (IPSec)
  - Mobility (Mobile IPv6)
  - Manageability (Address auto-configuration, support for renumbering, etc.)
  - Extensibility

Linux IPv6: History & Status

- IPv6 has been part of the Linux kernel since November ’96 (2.1.8)
  - Historically, distributions had to be manually upgraded to support the necessary libraries, commands and utilities
- Most recent versions of Linux distributions are IPv6 ready
  - Some include kernel IPv6 support in the form of a loadable module
  - Debian-based Gibraltar (http://www.gibraltar.at) provides bootable ISO images for CD-ROM based linux IPv6 gateways
- Many independent efforts to add features and improve robustness of Linux IPv6
  - USAGI project: general improvements to the kernel, updates for recent standards changes (source/destination address selection, etc.), IPSec implementation, etc.
  - Mobile IPv6 implementation at MIPL
  - IP6Tables through Netfilter project

Linux IPv4-to-IPv6 Transition Tools

- v6-to-v6 over IPv4 clouds
  - Configured Tunnels (IPv4-in-IPv6) – “sit” tunnels
  - Automatic Tunnels with 6to4
  - Automatic Tunnels with ISATAP
  - Automatic Tunnels (and free address!) with Freenet6
- v6-to-v4 and vice-versa
  - Socks-based Translator
  - NAT Protocol Translation (pTRT, etc.)