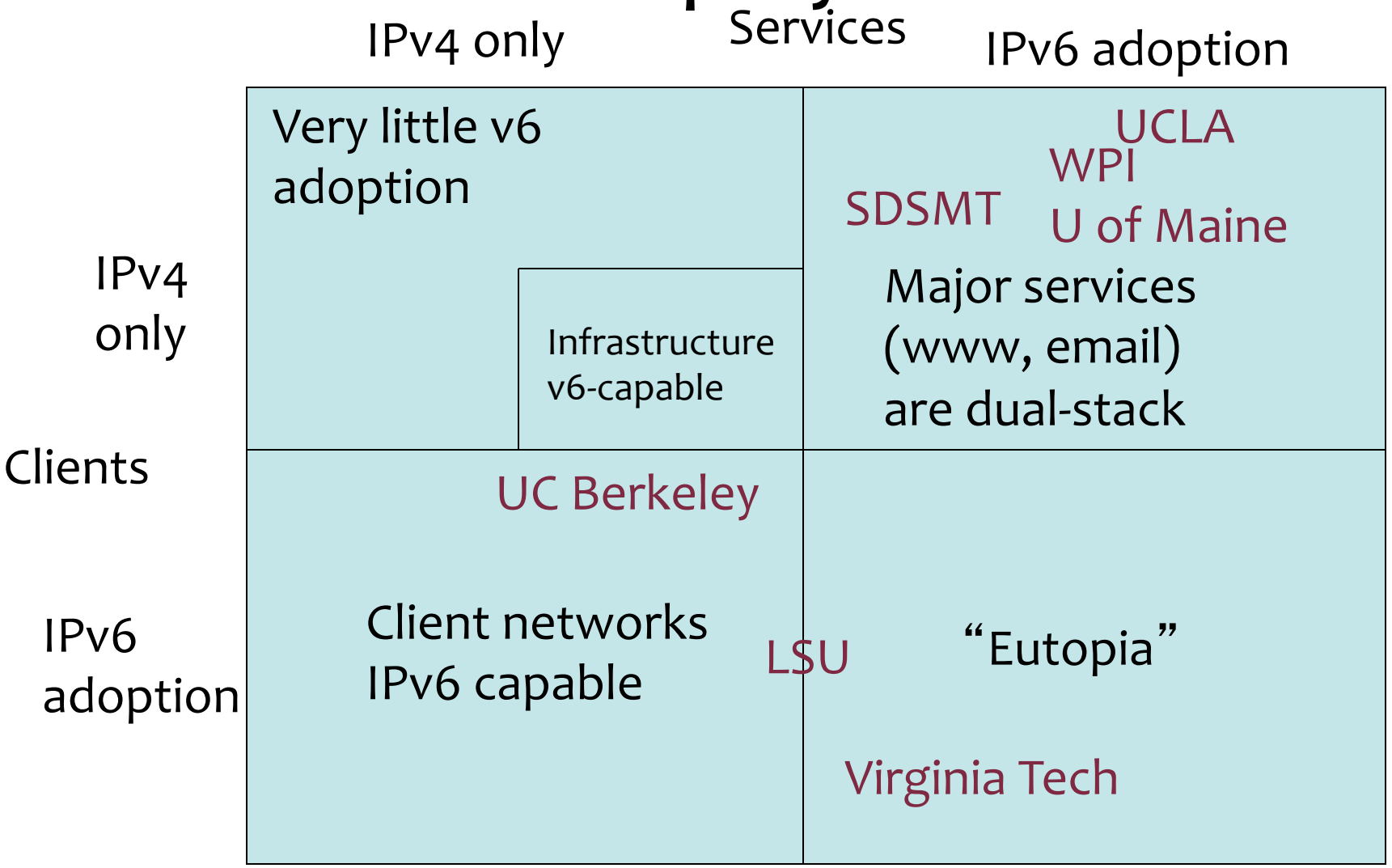
IPv6 Campus Wireless Deployment

Michael Sinatra, ESnet

Erik Klavon, UC Berkeley

Summer 2011 Joint Techs

EDU IPv6 deployment matrix



Why now?

- Issues with UCB's previous WLAN deployment:
  - Commercial vendor who was exiting the business. No more features, just lifeline support.
  - Burning through way too many IPv4 addresses, due to nature of the system.
  - No IPv6. Well, no *official* IPv6. We certainly had a bunch of rogue RAs.
  - No common, single-sign-on authentication.

Reminder: Rogue RAs are bad

- In IPv6, rogue RAs (usually windows or mac hosts with connection sharing turned on) can wreak havoc, especially on wireless nets.
  - All IPv6 hosts that grab onto that RA push their IPv6 traffic through that host.
  - What happens when host shuts down? What if it's misconfigured?
  - http://events.internet2.edu/2008/jt-hawaii/sessionDetails.cfm?session=3638&event=278
  - This even happens at conferences/meetings sometimes!

But I digress…

- MR. SINATRA: Michael Sinatra, UC Berkeley. I agree with Cathy in just about all of her sentiment. And I have one other thing to say, which is that -- this is slightly off topic, but it's kind of an operational issue -- if you have the IP address 192.35.164.158, you're announcing yourself as an IPv6 router, please stop doing that.

- You're running 6 to 4; you're running yourself as a to 4 relay, please stop doing that because it's breaking our IPv6 connectivity. And yes, I do support the idea of the proposal. Thanks.

- SPEAKER: What was the last octet?

- MR. CURRAN: Yes, repeat the information -- point of information.

- SPEAKER: 164.158, those were the last two octets. 158 was the last octet.
  - ARIN XXII, Los Angeles, October 2008

Solution?

- What would the solution look like?
  - Should support IPv6 from the beginning.
  - Deal with rogue RAs as best as possible.
  - Needs to support CAS authentication. http://www.jasig.org/cas
  - Reduce the *needless* consumption of IPv4 address space.
  - Allow mobile clients more flexibility in authentication.
  - Maybe not be dependent on a particular vendor? Open-source? Interoperable?

Solution!

- UCB's residence halls had been developing and using captive portal solutions for many years.
  - 1990s: PANDA. SNMP-based system with no web-based sink, initially.
  - late 1990s/early aughts: PAPANDA (Passive-aggressive PANDA). Had more features and included early captive portal functionality.
  - late aughts: Circe
  - Circe had been modified to work with WLANs in the residence halls. Central IT organization decided to steal it^H^H^H^H^H^H^H^Hcollaborate on the new solution.

Gearing up

- Network Engineer Erik Klavon had worked in both central IT and ResNet and had recently moved back into central IT (speaking of stealing).
  - Had extensive knowledge of the system and was able to recruit and supervise students to work on project.
  - Erik also worked with a demanding, loud-mouthed network engineer who called himself the “IPv6 compliance officer” and insisted that we do IPv6 from the beginning. He needed little convincing.

Implementation

- For a variety of reasons, the IPv4 implementation had to be done somewhat differently than IPv6.
  - IPv4 addresses are scarce; IPv6 addresses are practically limitless, even on a single /64.
  - IPv4 has DHCP, IPv6 has… uh… DHCPv6.
  - IPv4 has NAT, IPv6 doesn't (see above).

Implementation

- Dual-core AMD64 system, with multiple GigE interfaces.
- FreeBSD 7.x or 8.x. No, it doesn't run on Linux.
- squid, apache
- OS built-ins: ipfw; previously also used netgraph (ng_nat).
- perl and Bourne-compatible shell scripts to glue it all together.
- Some kernel hacking (fairly minimal).

Implementation

- For IPv4:
  - Initially you are assigned to a vlan which is set up as a many-to-one NAT. The captive portal uses squid to redirect you to the captive portal web, where you use CAS to log in.
  - Once you authenticate, you are switched to a one-to-one NAT instance (same vlan as before). There's various goo that makes this work…
  - Some kernel hacks to increase the number of sessions.
  - You MUST be running IPv4 in order to get on the network.

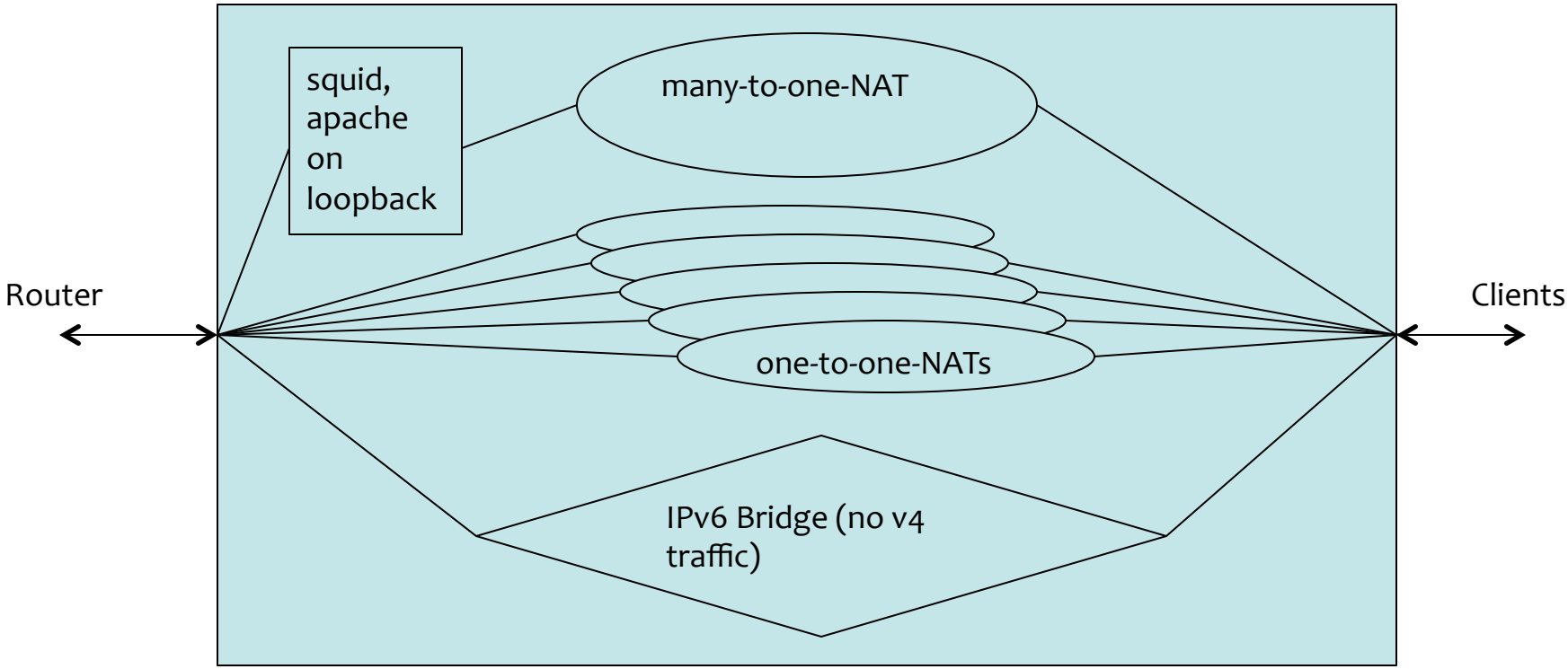
Implementation

- For IPv6:
  - A bridge is set up in the OS between the "inside" and "outside" vlans.
  - All IPv4 traffic is always blocked. IPv4 traffic must take the NAT route.
  - MAC learning *and* flooding disabled!
  - Upon successful authentication, the system adds user's MAC address to the bridge MAC table so that user can send/receive traffic.
  - Kernel hack to prevent user from *sending* traffic if the *source* MAC address isn't in the bridge table. → RPF for Layer 2!

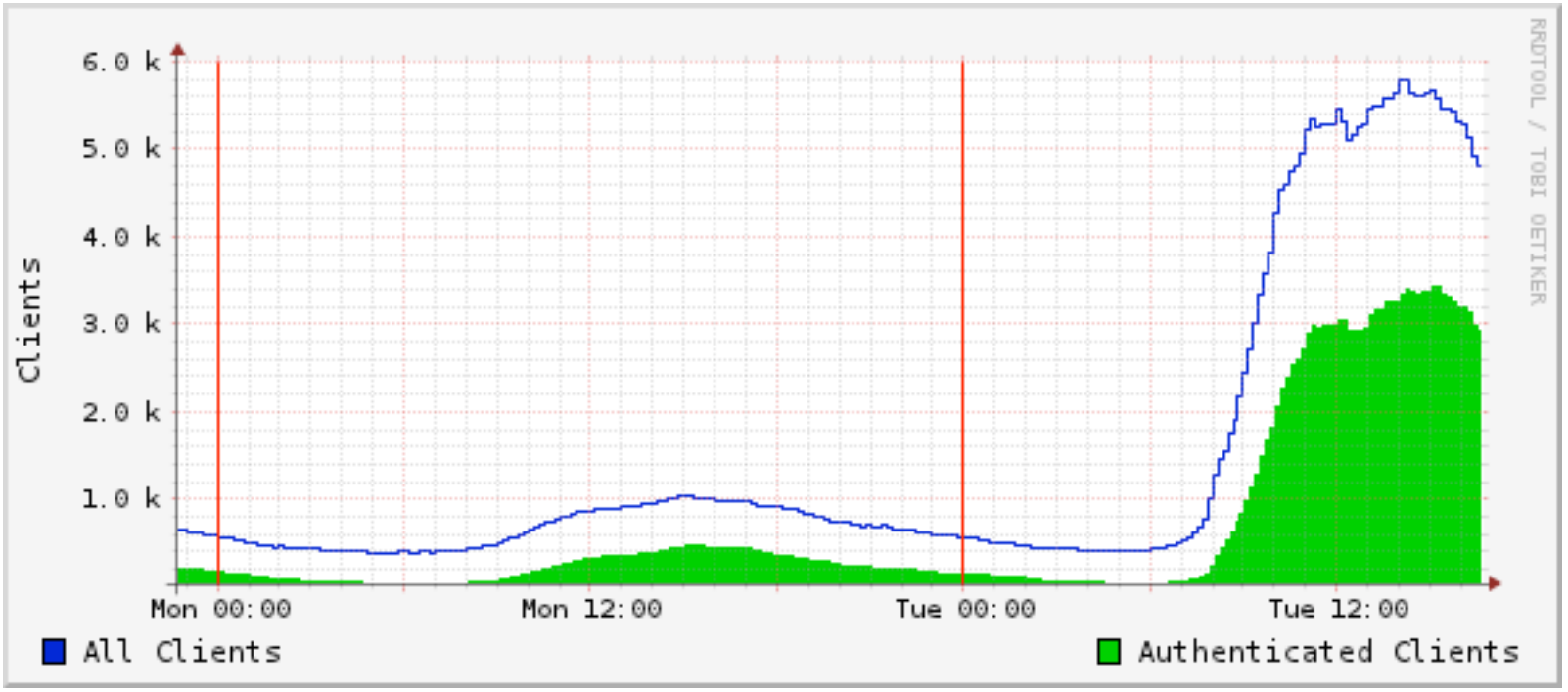
Implementation

- For IPv6:
  - Globally routable IPv6 address is configured on the router upstream from the captive portal.
  - RAs from this router are set with the preference of “high.”
  - RAs get bridged via the captive portal once the user authenticates so that the client machine will autoconfig.
  - Increase the frequency of the RAs, since the initial solicitation will probably get lost.

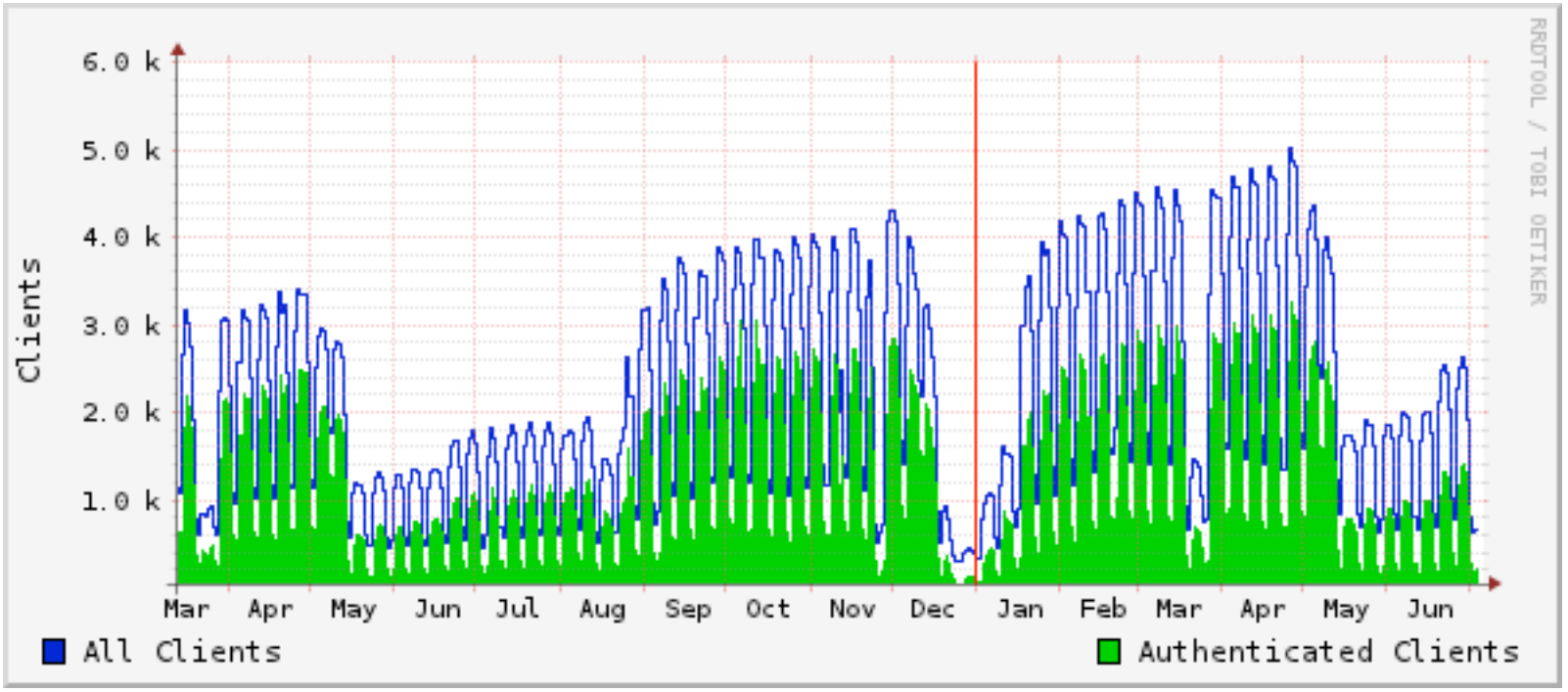
Implementation



Some results



Some results



Some results

- Fits and starts:
  - Our test platform turned out not to be a good production platform.
  - We needed to get a good feel for how the system would scale. → A lot of trial and error.
  - Made some modifications: Ditched ng_nat for in-kernel NAT via IPFW. This required some kernel hacks to increase the number of one-to-one NAT sessions.
  - One-to-one NAT prevents unauthenticated devices from consuming globally-routable IPv4 addresses, but still allows for easy connection tracking.

Some results

- IPv6 has been pretty much of a success so far.
  - Rogue RA problem has diminished radically.
  - IPv6 does “just work.”
  - So does uTorrent, probably.
  - One of our AP vendors had broken IPv6 forwarding (or really good RA Guard, depending on how you look at it). We did get them to fix that.
  - The other vendor passes IPv6 traffic, but that’s about it.
  - We still need RA-guard functionality for smaller switches and APs, but we’re in better shape than before.

Futures

- Funding models on campus have changed and WLAN is now funded properly.
  - Already causing a significant additional deployment in APs. → Blanket coverage.
  - Currently 10, moving to 20 captive portal boxes on campus, and more will probably be coming. New machines being ordered. New machines are scaling much better than previous ones.
  - Usage varies from area to area (and therefore from CP to CP) throughout the day.
  - Globally-routable IPv4 space still needs to be over-provisioned somewhat.

Futures

  - We could go to many-to-one NAT and… well, let's just say "blecch" and be done with it.
  - Another possibility: Have CP run an OSPF instance and announce /32s to the upstream router which will announce a covering route in the campus table. IPv4 subnets can be pooled into larger subnets with less slack. Proof of concept underway.
- EDUroam and 802.1x are being considered.
- Support for IPv6-only clients.
- Greater redundancy.
- Traffic (including NAT) offload.

Conclusions

- Open-source captive portals can be done. A talented student labor pool helps.
- IPv6 on wireless is a good thing.
  - Native IPv6 can be configured to trump rogue RAs.
  - Improve service and security with IPv6. No more tunnels hiding v6 traffic—now it's out in the open.
- Scaling always an issue—you may be on uncharted territory, especially if you are exercising v6 features to any great degree.
- WLAN IPv6 is a great way to increase IPv6 eyeballs.