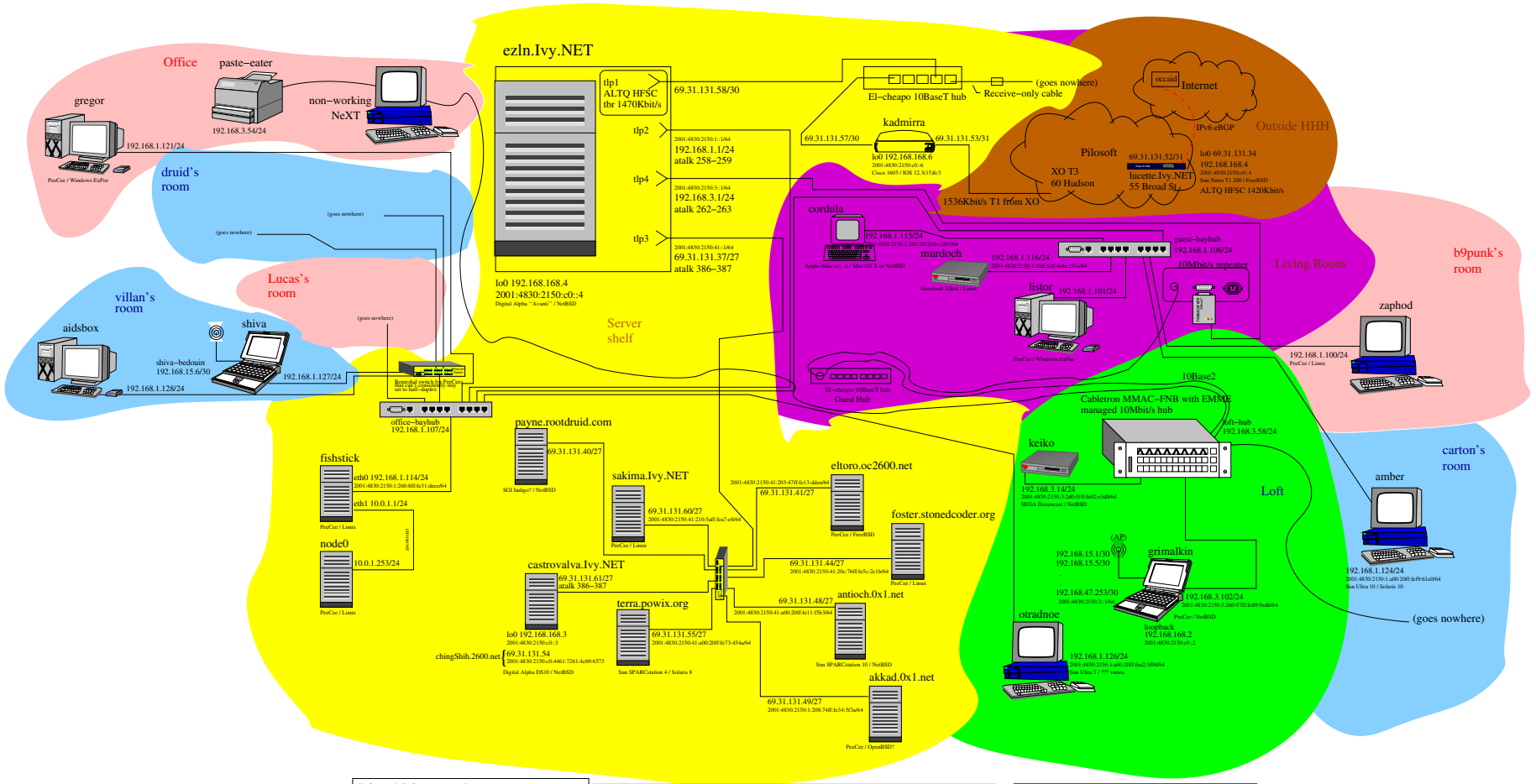

L2 and L3 protocols

all segments have IPv4, IPv6, and appletalk phase 2 except for the colocation segment, we use hubs only, no switches. I believe anything which can drop a packet needs to show up in traceroute, and eventuall yneeds to support things like multicast, RED/ECN/DiffServ, and needs to decrement IP ttl to avoid storms.

on lucette's tlp3 and grimalkin's wireless, we use dhcp servers that statically assign IPv4->L2 arp routes when clients register, to frustrate ettercap. on wireless, we also use /30 subnets so wireless clients are protected when talking amongs themselves. Since I think 802.11b repeats all traffic at the AP anyway, this should waste no airspace.

QoS

QoS is HFSC implementing three goals:

* bandwidth is shared equally among the seven different people who have machines colocated on the server shelf. Inside machines get about three shares' worth upstream.
* ssh traffic is separated from all other traffic, so if you aren't using scp to copy anything, within your share your ssh traffic will get transmitted before ftp or bittorrent or some other kind of upload. If you use scp it won't work——it's not stateful w.r.t. connections.
* there's one giant queue for peer-to-peer leeching programs. All machines known to run PtP have all traffic put to start into one queue shared by everyone. There are exceptions for well-known ports to let traffic back into the regular per-user queue.

Traffic shaping

A lot of people tell me I should use the ''traffic shaping'' feature of FreeBSD ipfw, or some Linux SOHO router or managed switch or Cisco CBQ. We don't do that here. HFSC is a ''work-conserving'' scheduler meaning if any class has backlogged traffic, our T1 will be full, no matter how low-''priority'' the class.

Where

QoS is done on ingress to the slow T1 pipe, so downstream QoS is done on lucette, and upstream on ezln. We do not do any ''shaping'' of traffic after it comes out of the pipe——we schedule output queues only. In both cases we schedule onto an Ethernet feeding a T1 router, but nothing else should go onto the T1 except the scheduled traffic we feed.

Routing

IGP

Although the number of wire segments at our site is small, all segments run OSPFv2, OSPFv3, and RTMP, so all the protocols we route are routed dynamically. This has advantages: I don't have to type in long IPv6 addresses quite so often, and I can do IPv6 routing properly using link-local addresses for the next-hop.

EGP

lucette keeps an IPv6 BGP4 session open to our IPv6 tunnel's endpoint at OCCAID in NJ. Because the session is running over a gif(4) tunnel, this is not multihop BGP. Along with the real /48 we're assigned a fake AS64582. We advertise the /48 to AS30071 and receive advertisements for the ~500 prefixes in the IPv6 default-free zone.

There is fully-meshed iBGP among lucette, ezln, castrovalva, and grimalkin. kadmirra doesn't have enough memory to do BGP, so it has default routes configured for IPv4 and IPv6. lucette advertises a default route over IPv4 BGP to the other iBGP systems, although there isn't much point to it yet.