



IPv6 on DREN & Status of DoD Pilot

U.S. IPv6 Summit 2003
Arlington, VA

Ron Broersma
DREN Chief Engineer
High Performance Computing Modernization Program
ron@spawar.navy.mil



What is DREN?

- “Defense Research and Engineering Network”
- The DoD network that serves the High Performance Computing (HPC), Research and Development (R&D), Test and Evaluation (T&E), Modeling and Simulation (M&S), and related communities within the DoD.
 - Sites with DoD’s fastest supercomputers
 - Sites that were computing and networking pioneers
- Started in 1992 initially to provide IP and ATM connectivity to DoD’s HPC centers, but the scope has expanded since then.
 - Initially constructed out of the Army and Air Force Supercomputer Networks.
 - Commodity portions were later outsourced.
- Major thrusts are high performance, security, and new technologies.

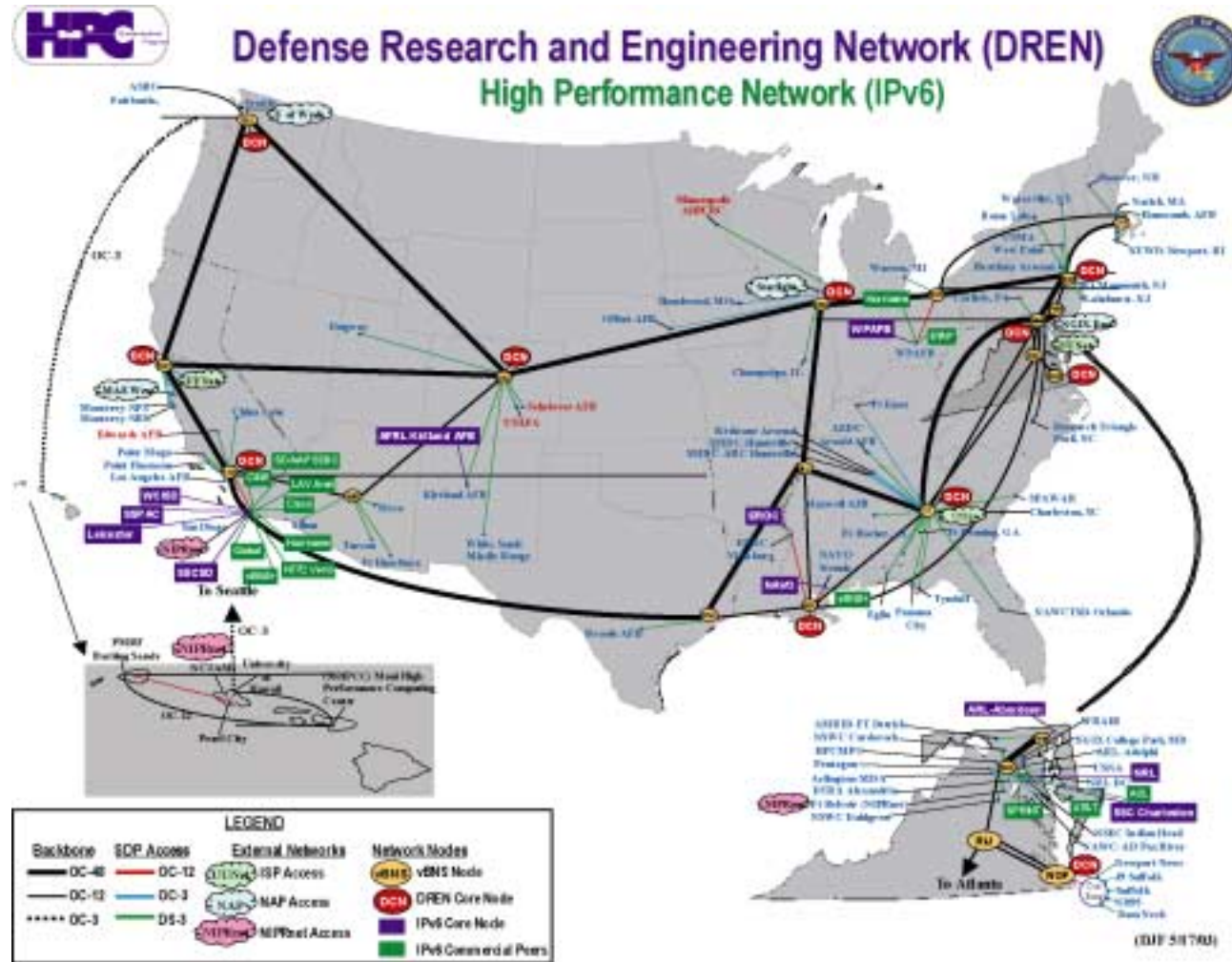


DREN Today

- 10 “core nodes” on OC-48 backbone (CONUS), with extensions to Hawaii and Alaska.
 - OC-192 in 2004
- About 100 sites (“Service Delivery Points”), connected at DS-3 to OC-12 rates.
 - Some upgrading to OC-48 in 2004
- IPv4 unicast and multicast, IPv6 unicast, and ATM services.
- Dual IPv6 networks (testbed, and production)
- “jumbo-clean” (i.e. 9K MTU everywhere)
- Multiple security levels.



DREN Map



December 17, 2003

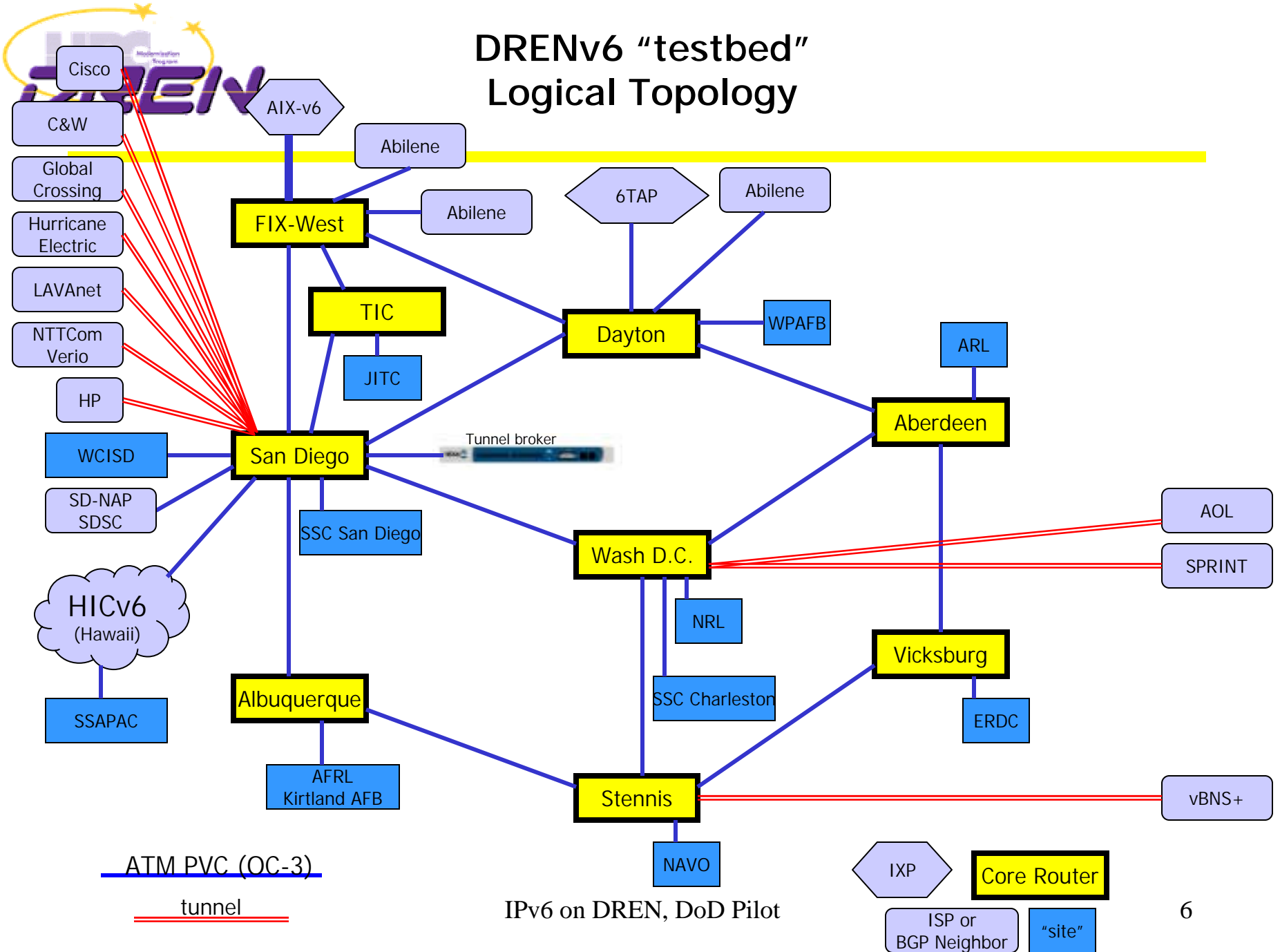
IPv6 on DREN, DoD Pilot



DREN IPv6 History

- 1995-2000
 - Ad-hoc tunnels, playing on 6bone.
 - Presentation at conferences
 - IPSEC (NRL)
 - Early implementations (NRL stack)
- Jan 2001 -
 - DREnv6 “testbed”
 - Native IPv6 (no tunnels)
 - Logically separate from DREN IPv4 backbone
 - OC-3 interconnects (ATM PVC mesh)
 - 8 core nodes (Cisco routers – dedicated to IPv6)
 - Sites connect via PVCs (native IPv6), or tunnels.
 - Peering with IPv6 enabled ISPs
 - DREN sites encouraged to connect and participate in testing and experimentation. Many tests conducted, many lessons learned.
 - “If you build it, they will come”
- 2002
 - New DREN2 backbone contract (MCI) includes IPv6
- Jul 2003
 - Selected as DoD IPv6 “pilot” (details below)
- Oct 2003
 - Added DREnv6 node at Ft Huachuca (TIC, JITC) for Moonv6 interconnect between DoD and Abilene (UNH)

DRENV6 "testbed" Logical Topology





Lessons (Challenges)

- Our customer sites find little or no incentive to run IPv6 (LAN administrator perspective).
 - There is no capability or feature of the Internet that you can't do today by not running IPv6.
 - Turning it on brings additional complexity, and has a learning curve.
 - Users aren't asking for IPv6.
 - There is no immediate "win" to transitioning to the new protocol. The payoff is long-term. External incentives will be needed to encourage near term adoption and transition.
 - "If you build it, they won't necessarily come"
- Many commercial security components (like Intrusion Detection Systems, Firewalls, Security Scanners, etc.) don't yet support IPv6, so it is very difficult to deploy the technology to our sensitive DoD networks in a secure fashion.



DREN as DoD IPv6 Pilot

- DREN is in a unique position to serve as a DoD IPv6 pilot
 - Experience running IPv6 WAN.
 - R&D environment – familiar with technology insertion, and being a pioneer.
 - New contract includes IPv6 support (we just have to turn it on).
 - Management support.
 - Have the means to deal with the challenges.



FY04 DREN IPv6 Initiative

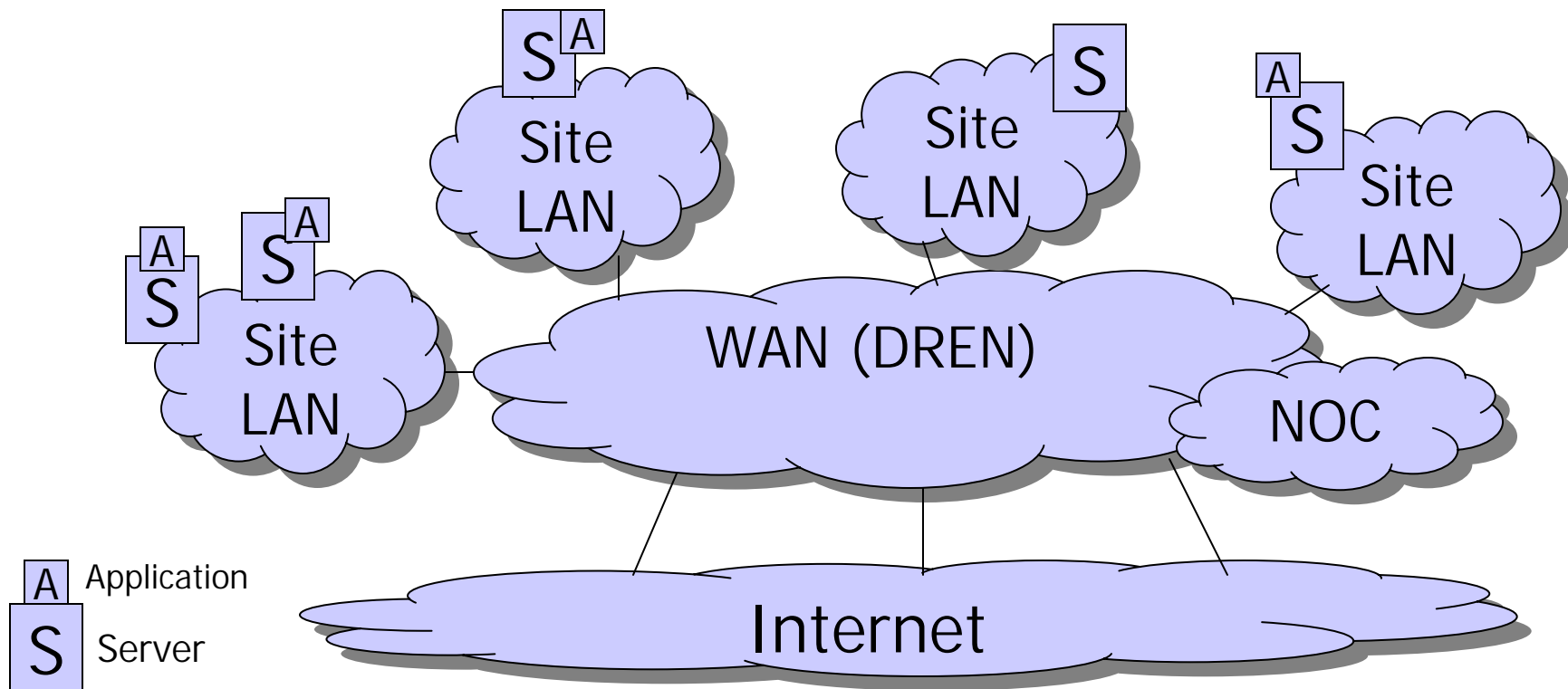
- DoD IPv6 Pilot network
- Goals – FY04
 - IPv6 enabled DREN infrastructure (all SDPs, the MCI virtual private network, the DREN NOC).
 - IPv6 enabled HPCMPO, HPCMP funded assets and services, HPCMP user community support applications, selected user application candidates.
 - Performance and Security as good as existing IPv4 service.
 - Facilitate IPv6 deployment into infrastructure at HPC user sites and DREN user sites.
 - Provide product feedback, lessons learned, published via web.
- Functional Areas in this project:

– IP transport and infrastructure	Ron Broersma, Navy
– Infrastructure services	Phil Dykstra, WCI
– Network Management	Tom Kile, Army
– Security	Doug Butler, OSD
– Applications	Ralph McEldowney, Air Force
– Planning for the Future	Ron Broersma, Navy
– HPC Community Involvement	John Baird, OSD



Transition Strategy (Notional)

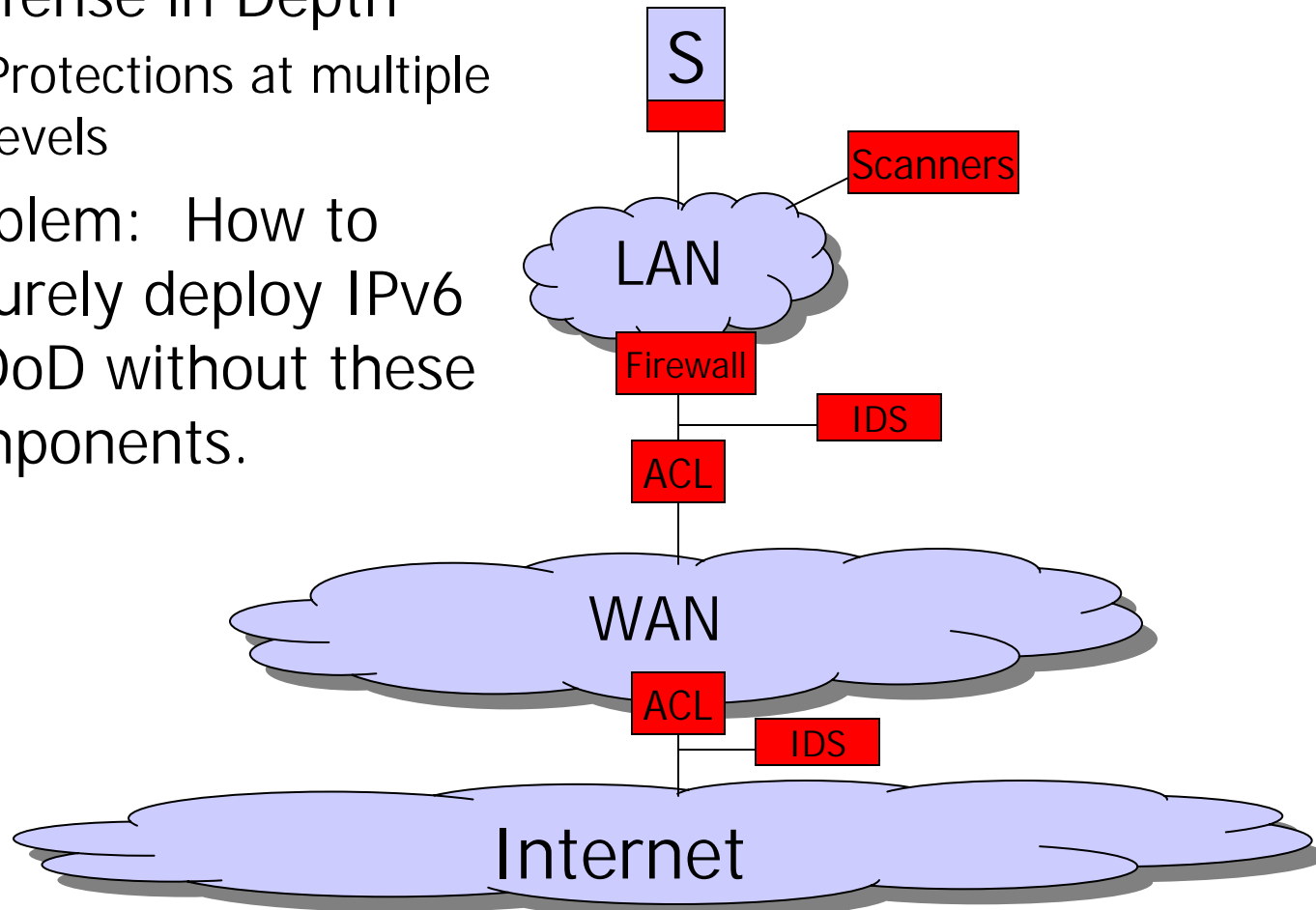
- Start with core, and work out to the edge
- Dual Stack throughout infrastructure
- Minimize need for tunnels, translators, and other transition schemes





DoD Security Model

- “Defense in Depth”
 - Protections at multiple levels
- Problem: How to securely deploy IPv6 in DoD without these components.



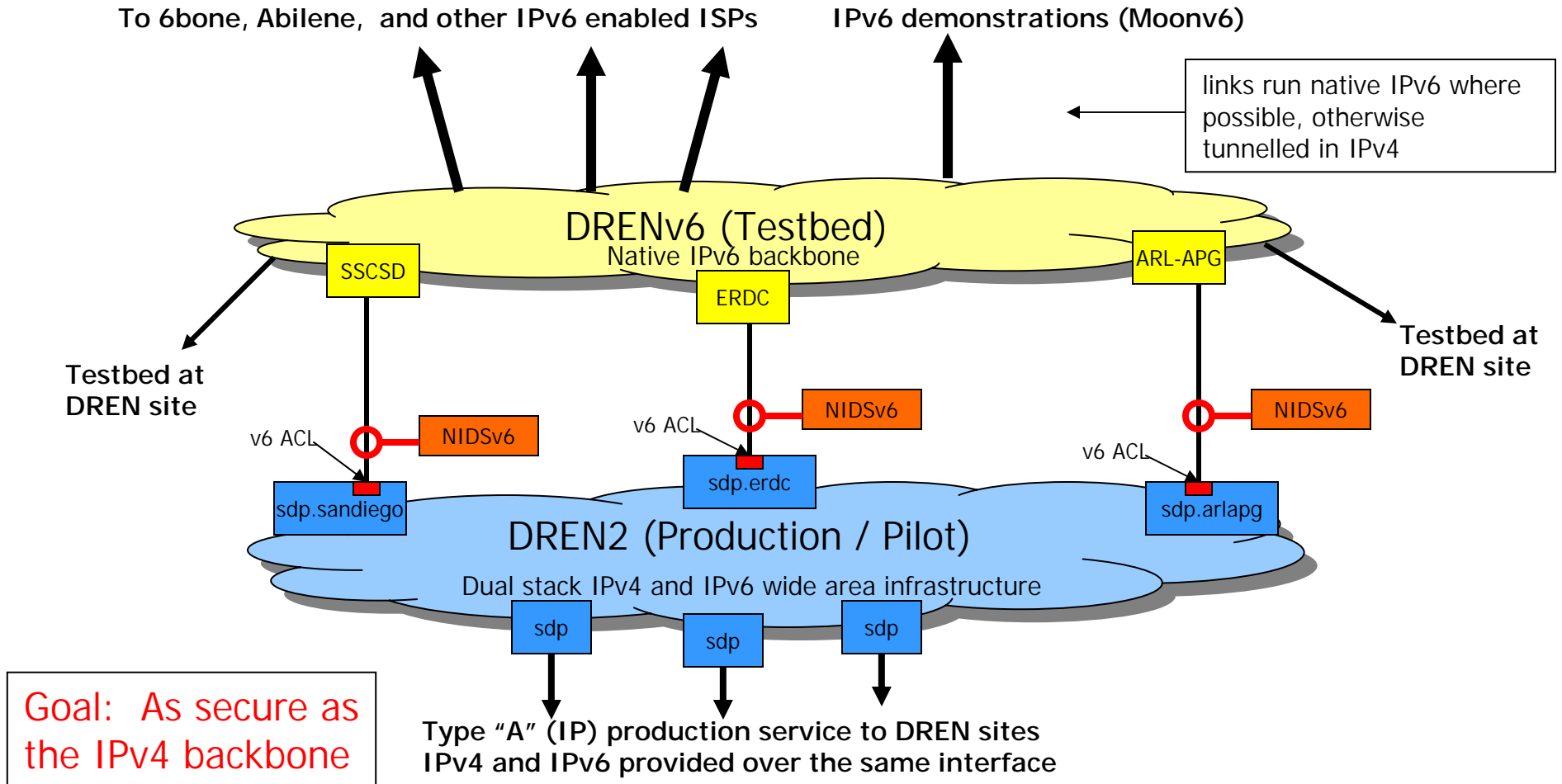


Overcoming the security issue

- Use DREnv6 testbed for transit to Internet
 - use to peer with rest of IPv6 enable Internet and other testbeds
 - continue to operate as an “untrusted” IPv6 network
- Enable IPv6 on new DREN2 (MCI) production network.
 - Dual stack everywhere.
- Establish trusted gateways between v6 enabled DREN2 and the DREnv6 testbed
 - Upgrade HPC Network Intrusion Detection Systems (NIDS) to be v6-compliant, monitored by the HPC Computer Emergency Response Team (CERT), and install at the trusted gateways.
 - Install v6 version of standard DREN v4 Access Control Lists (ACLs) to protect pilot network to same level as IPv4 production network.
- DREN customers receive “safe” native IPv6 service via existing service delivery point (SDP), in parallel with IPv4 service.



DREN IPv6 transition architecture – FY04



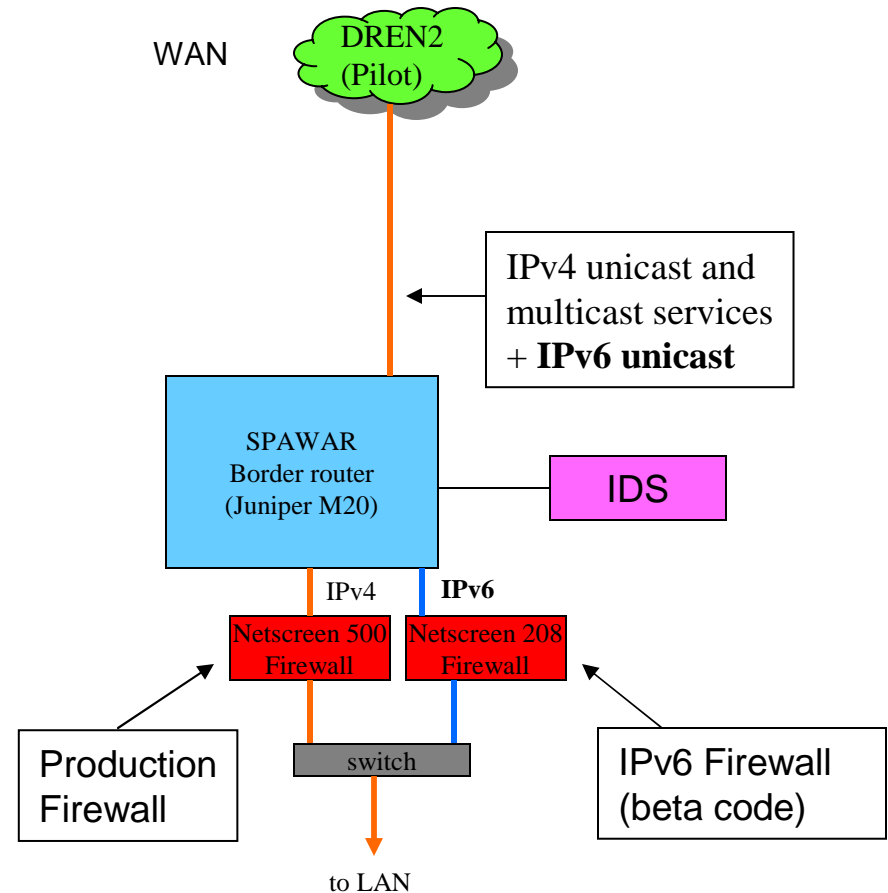
December 17, 2003

IPv6 on DREN, DoD Pilot



Site Security Solution (Example – SPAWAR)

- SPAWAR Intrusion Detection System (IDS) modified to support IPv6
- Netscreen Firewall operating “beta” release with IPv6 support in parallel with production firewall.





DREN2 (Pilot) backbone status

- All 100 Juniper routers recently upgraded to support IPv6
 - JunOS 5.6 and 6.1
 - DCNs and SDPs
- WAN Interconnects with security solution deployed at 3 locations.
- Backbone is now IPv6 enabled and ready to bring production sites online.
 - Sites already turned up: HPCMO, SSC San Diego, ARL, ERDC, Indian Head, Quantico, DREN NOC.



Performance Results

- Phil Dykstra (on DREN2 “pilot” net):
 - “Using iperf, SSC [San Diego, CA] to ARL [Aberdeen, Maryland], MTU 9k, I get about 567 Mbps with IPv4, 565 Mbps with IPv6. So at first glance, performance seems nearly identical (minus the extra header overhead of course).”
 - Done between 2 Linux machines on opposite coasts connected to DREN OC-12 sites.
- 10Gb-E testing at HPC Center, sending a 4 Gb/s stream from Linux with 10Gb-E NIC.
 - 3939.8044 Mbps UDP single stream (IPv4)
 - 3930.6234 Mbps UDP single stream (IPv6)



DREN performance measurement tools

- DREN "AMP"
 - Active Performance Measurement system
 - IPv6 updates – Phil Dykstra
- nuttcp 4.0 (NRL)
 - TCP performance tester (client/server)
 - IPv6 updates – Rob Scott (NRL)
 - <ftp://ftp.lcp.nrl.navy.mil/pub/nuttcp>



DREN Security Tools

- Network Intrusion Detection System (NIDS)
 - Based on LLNL "JIDS"
 - Upgraded to IPv6 – Ken Renard
- SPAWAR IDS
 - Upgraded to IPv6 – Dr. Mark Shensa
- Snort 2.0.1
 - Upgraded to IPv6 – Ken Renard
 - Need testers
- Kerberos v1.3 (MIT)
 - IPv6 updates for DREN release by Ken Hornstein (NRL)
- Working on IPv6 for...
 - DoD CAC with OpenSSL, PKI, OCSP, LDAP



Addressing

- 2001:480::/32
- /44 reserved for each SDP
- Sites get a /48
- All subnets are /64
 - No tiny subnets for point-to-points



Large projects with interest in IPv6, using DREN

- Global Information Grid (GIG) related experiments (NRL, SPAWAR)
- Future Combat System (FCS) (Army)
 - Existing DREN sites, plus 8 new Boeing sites
- Fleet global unified routing architecture (Navy)



Backup (Some Lessons Learned)



Initial Lessons Learned

- Mainstream operating systems now come with fairly mature IPv6 stacks.
- Many routers now include production IPv6 in their latest releases.
- Many important components are either not IPv6 aware or cannot operate using IPv6 protocols
 - (network management tools, security components, peripherals and appliances like printers/scanners).
- It is not really practical today to run purely IPv6 without also supporting IPv4 in the network.
 - IPv4 is going to be around for a long time in our networks.
- There are quite a few new concepts that need to be learned before a network administrator can comfortably and properly implement IPv6 in the network.
 - It is important to get our network engineers and technicians up to speed on the technology well in advance of implementation and transition.



Initial Lessons Learned

- Since commercial security components (like IDS and Firewalls) don't understand IPv6, it is very difficult to deploy the technology to our sensitive networks in a secure fashion.
- There is little or no incentive to running IPv6 in most networks today.
 - There is no capability or feature of the Internet that you can't do today due to not running IPv6.
 - Turning it on brings additional complexity, and has a learning curve.
 - Users aren't asking for IPv6.
 - There is no immediate "win" to transitioning to the new protocol. The payoff is long-term. This contributes to the lack of incentive to transition today. External incentives will be needed to encourage near term adoption and transition.
- Performance of IPv6 implementations vary.
 - We have verified that recent Linux 2.4.x kernels handle IPv6 as efficiently as IPv4, i.e. without throughput degradation or increased CPU load.
 - We have also verified that Juniper routers can forward IPv6 at line speed (tested through OC12).- Many traditional network performance testing and diagnostic tools are still being ported to IPv6.
- There is a tendency by network engineers to apply IPv4 constraints to IPv6 situations.
 - Overly conservative with address space in developing addressing plans (i.e. tiny subnets).
 - Desire to use NAT and private address space.



Lessons Learned (Vendors)

- Be aware that there is often a large difference between the marketing and technical definitions of "IPv6 support" by vendors. Marketing definitions usually mean that the product won't crash if you send it an IPv6 packet. The technical meaning includes all RFC-based requirements and full product functionality in an IPv6-only environment.
 - Many routers that do packet switching in hardware, only do so for IPv4. The IPv6 packet forwarding is done in software (slower, higher CPU loads).
 - You generally don't find all the missing pieces until you try to implement it in a real network.
 - Most (90%+) products that claim IPv6 support do not provide ANY IPsec support for IPv6. The <10% of products that do support IPsec for IPv6 are not as functionally complete as their IPv4 counterparts (like Solaris 9 and the lack of IKE support for IPv6 IPsec)
- If you query vendors about IPv6 support, you must query each individual product function and IPv6 feature. The best way is to obtain an evaluation unit/copy and test it yourself (which is, of course, very time consuming).



Lessons Learned (DNS)

- Currently 4 root DNS servers allow native IPv6 queries (B, F, H, M). H is on DREN. None of them have officially advertised IPv6 addresses yet (AAAA records). DNS responses are limited to 512 bytes, so only two AAAA records can be added to the root zone without making any changes to the protocol, servers, or client resolver libraries.
- Army's current DNS architecture uses Nortel's NetID. This does not support IPv6 records.
 - On September 30, Lucent was awarded a contract to replace NetID with VitalQIP, which will support IPv6 records in the next release (December?). The transition to VitalQIP is supposed to be completed within 10 months of the award date.
 - No idea if/when Army plans to allow native IPv6 queries (i.e. have IPv6 addresses on the Army name servers).
- A naming convention can be very helpful when you are unsure what address is being chosen. We adopted the following convention:
 - Canonical host name is given "A" and "AAAA" resource records.
 - Add "-4" suffix to name, and get just the "A" record.
 - Add "-6" suffix to name, and get just the "AAAA" record.
 - PTR records point to the canonical name.



Lessons Learned, cont'd

- There are no commercially available IPv6 NTP (Network Time Protocol) servers.
 - The first production release of NTP 4.x code with IPv6 support was on 15 Oct 2003.
- IPSec support in products is generally weak.
 - Most implementations are functionally incomplete with respect to IPv6 IPSec and even IPv4 implementations leave a lot of room for improvement.
 - Documentation is especially lacking. Documentation is only for the most common cases and generally does not cover functionality of implementation, but instead, a cookbook for setting up a few specific configurations (e.g. one OS only explains which configuration steps are needed to set up IPSec with another machine of the same OS. If they explained the functionality, then you could figure out how to configure with any other OS). This is based on IPv4 IPsec evaluation, since IPv6 IPSec implementations are rare.



Lessons Learned (Applications and Software Development)

- Litmus test for v6-enabled Applications is that there should be no loss of functionality if you turn off IPv4.
- There are several IPv6 porting guides in the Internet. These are lessons learned while porting network-centric applications (network and security tools). This should go beyond typical IPv6 porting guides for those applications which deal heavily with network packets and interfaces. These recommendations are based on "C" code.
 - Try to be as OS-agnostic as possible. Most OSes have very different APIs for dealing with IPv6 interface operations and this can be difficult. For a good example of generic, OS-agnostic code, look at MIT Kerberos 5 v1.3.
 - Build application-specific address structure for your code. This would typically be a structure that includes the address type, address data, and optionally address size. This allows a single structure for dealing with multiple address types.
 - Hostname lookups: expect multiple addresses to be returned. This should be obvious for hosts with multiple IPv4 addresses, but you need to account for several IP addresses (at least 2) per interface. Also, you need to take link-local and site-local addresses into consideration (should you ignore them?).
 - When replacing IPv4 addresses in code, rename variables or structure members so that compiler can help you find all instances of the address variable that need to be adjusted.
 - Take IPv4-compatible, and IPv4-mapped addresses into account (e.g. when comparing addresses).
 - Use "struct sockaddr_storage" for sockaddrs and cast to the appropriate sockaddr_* for the address family.



Lessons Learned (misc)

- Loading a version of code on a Cisco router which supports IPv6 often requires memory or other hardware upgrades.
- Port mirroring for IPv6 traffic not supported in Juniper.
- Juniper ACLs don't support "tcp-established".
- For SGI systems with Marconi ATM interfaces, enabling IPv6 on IRIX without the latest ATM drivers may cause the kernel to crash.
- Most testing and assessment tools do not support IPv6
 - Security scanners (like ISS)
 - Performance testing tools and appliances
 - nuttcp (NRL) has been v6-enabled recently