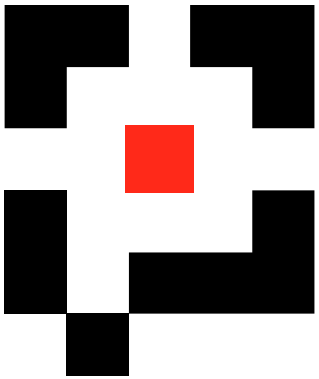
INDIANA UNIVERSITY

Presentation by ANML
June 2004



pervasivetechnologylabs

AT INDIANA UNIVERSITY

# Porting Applications to IPv6

INDIANA UNIVERSITY

# About the Presenter

- Mark Meiss
- Academic Background:
  - B.S. Mathematics, B.S. Computer Science
  - Ph.D. student in Department of Computer Science
- Research interests:
  - Structural analysis of network traffic data
  - High-performance file transfer protocols
  - Autonomous information retrieval agents

pervasivetechnologylabs
AT INDIANA UNIVERSITY

INDIANA UNIVERSITY

# About the Presenter

- Professional Experience:
  - Over 10 years in software development
  - With IU IT Services since 1997
  - Worked with Bloomington NOC
  - First employee of ANML
  - Developed Animated Traffic Map, Router Proxy, Tsunami file transfer protocol, etc.

pervasivetechnologylabs
AT INDIANA UNIVERSITY

INDIANA UNIVERSITY

# Overview

Application porting issues under IPv6

- Parsing
- Data types and structures
- URLs
- Databases
- Other Issues

(We've already covered DNS issues…)

pervasivetechnologylabs
AT INDIANA UNIVERSITY

INDIANA UNIVERSITY

# Address Parsing Issues

- IPv4 dotted decimal addresses were trivial to parse
- Legacy code may contain common I/O metaphors
  - `sscanf(buf, "%u.%u.%u.%u", &a[0], &a[1], &a[2], &a[3]);`
  - `printf("%u.%u.%u.%u", a>>24, a<<8>>24, a<<16>>24, a<<24>>24);`

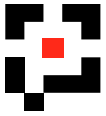
pervasivetechnologylabs
AT INDIANA UNIVERSITY

# Address Parsing Issues

- IPv6 hex-colon addresses require library support for input or output
  - A complete input parser can be a few hundred lines of code
  - Rendering an address in canonical form involves complex analysis of the address
- This is one wheel you don't want to invent several times

# Storing Addresses in Integers

- Legacy code often stores IPv4 addresses in 32-bit unsigned integer variables
  - Native data type
  - Makes masking operations easy
  - Fewer details to remember
- Few machines have a native 128-bit data type
  - Must change all code to use the appropriate *struct* or an array of native data types



# `struct sockaddr_in`

- The original BSD sockets interface recognized that future addresses might be bigger than 4 bytes
  - So they left space for 12 bytes
  - But an IPv6 address is 16 bytes
- `struct sockaddr_in` has to be replaced with `struct sockaddr_in6`
- This makes some existing libraries impossible to use with IPv6

# Endian Issues

- IPv6 does not introduce any new endian issues
  - Everything is still in network byte order
- When we do have 128-bit machines, we will need `htonllll()` and `ntohllll()`
  - Or maybe little-endian machines will explode

# URL Embedding Issues

- Original standards for URLs and URIs do not allow IPv6 addresses in URLs
  - Problem is the colons in hex-colon notation
- RFC 2732 modifies standard to allow IPv6 addresses in brackets
  - http://[2004:5b::f23b:10ff:fe87:559d]/
- Some browsers accept this (IE, Mozilla), but a lot of legacy code does not

# Binding to Selected Addresses

- In the IPv4 world, the vast majority of systems have one address per interface
  - Most legacy code is sloppy about specifying an address for binding server sockets
- IPv6 systems usually have multiple addresses per interface (link-local, etc.)
  - May need to introduce more specific socket bindings in existing code

INDIANA UNIVERSITY

# Database Renormalization

- On most current systems, pointers are 4 bytes
  - Applications *lose* memory by pointing to IPv4 addresses instead of storing them in place
- With 16-byte IPv6 addresses, pointing to shared addresses may save a huge amount of memory

pervasivetechnologylabs
AT INDIANA UNIVERSITY

INDIANA UNIVERSITY

# Fragmentation Issues

- In IPv4, fragmentation is rarely an application-level concern
  - Fragmentation can be done by any router
- In IPv6, fragmentation is done *only* by the endpoints of a connection
  - In theory, MTU discovery is performed
  - But what if the path changes during the life of the socket…?

pervasivetechnologylabs
AT INDIANA UNIVERSITY

INDIANA UNIVERSITY

# Problems with Tunneling

- There is not yet ubiquitous native IPv6 connectivity
  - Most IPv6 applications will sometimes pass through some sort of tunnel
- Tunnels should be transparent
  - "Should be" and "are" can differ dramatically
  - Network debugging tools and tunnels don't mix well

pervasivetechnologylabs
AT INDIANA UNIVERSITY

# Problems with Support Tools

- Developers require support tools for porting and testing network applications
  - These support tools may not be IPv6-capable
  - The IPv6 support may be experimental or limited
- Most IPv6 stacks have *ping6*, *traceroute6*, etc., but what about packet capture tools?

INDIANA UNIVERSITY

# Less Mature Code Paths

- IPv4 has been around for over 20 years
- Most IPv6 implementations are much newer and much less tested
  - What do you do when your application doesn't work because the operating system is broken?
  - How much time can you invest in discovering other people's bugs?

pervasivetechnologylabs
AT INDIANA UNIVERSITY

INDIANA UNIVERSITY

# Conclusion

- The difficulty of porting applications will vary greatly from application to application
  - Depends on the choice of coding metaphors, memory management, etc.
  - Possibility of library interface problems
  - Time and cost estimates are likely to be inaccurate because of this
- GOOD LUCK!

pervasivetechnologylabs
AT INDIANA UNIVERSITY