



IPv6 Adaptation of EtherNet/IP

Evolution not Revolution

Technical Track

www.odva.org

Drivers for Migration

IPv4 Address Pool Depleted

- ▶ IANA issued last 5 blocks to the 5 RIRs on 3 February 2011
- ▶ 15 April 2011, AP runs out
- ▶ 14 Sept 2012, Europe runs out

US Government Mandates IPv6 ²

- ▶ USGv6-1.0 IPv6 Profiles
- ▶ No EtherNet/IP Device is "USGv6-1.0-Capable"

China National Development and Reform Commission to invest RMB 8bn (\$1.3bn) in IPv6 adoption over 5 years



Extending the Life of IPv4

PUBLIC IPv4 Addresses depleted

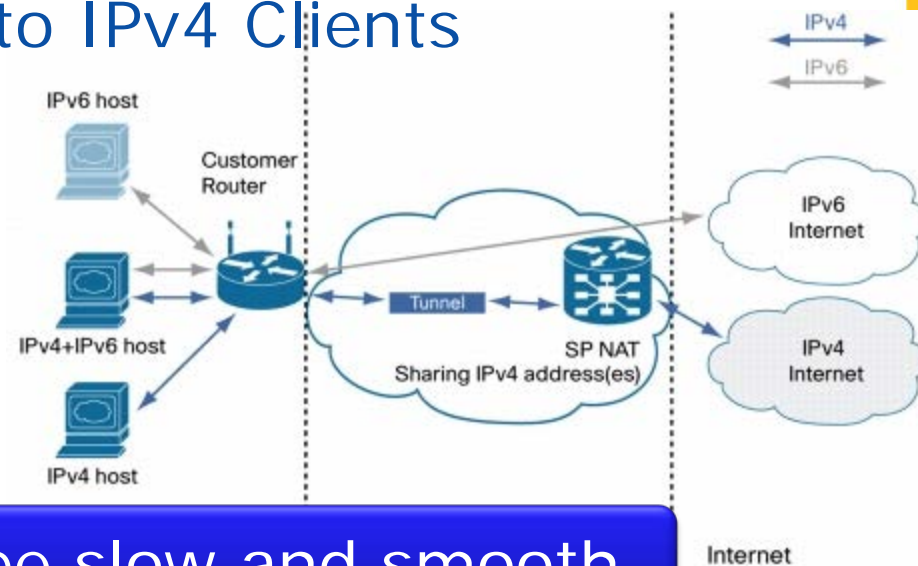
- ▶ Enterprise uses private IP addresses

Network Address Translation

- ▶ 192.168.1.x <> 10.10.y.y <> Internet
- ▶ IPv6 Internet available to IPv4 Clients

IPv4 & IPv6 Internet Coexistence

IPv4 & IPv6 Intranet Coexistence



IPv6 Migration will be slow and smooth

The obvious: bigger IP addresses

- ▶ 128 bit (IPv6) vs 32 bit (IPv4)
- ▶ Global Unicast Address format:

48 bits (or more)	16 (or fewer)	64 bits
<i>routing prefix</i>	<i>subnet id</i>	<i>interface identifier</i>

- ▶ Link local address format:

10 bits	54 bits	64 bits
<i>prefix</i>	<i>0</i>	<i>interface identifier</i>

Every node has a link local address assigned via Stateless Address Autoconfiguration

Simplified IP header

Offset	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	<i>Version</i>				<i>Traffic Class</i>				<i>Flow Label</i>																							
4	32	<i>Payload Length</i>																<i>Next Header</i>				<i>Hop Limit</i>											
8	64	<i>Source Address</i>																															
12	96																																
16	128																																
20	160																																
24	192																																
28	224	<i>Destination Address</i>																															
32	256																																
36	288																																

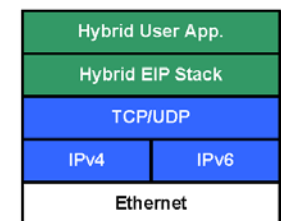
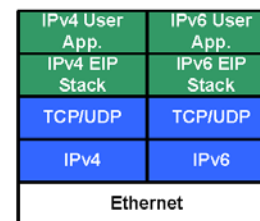
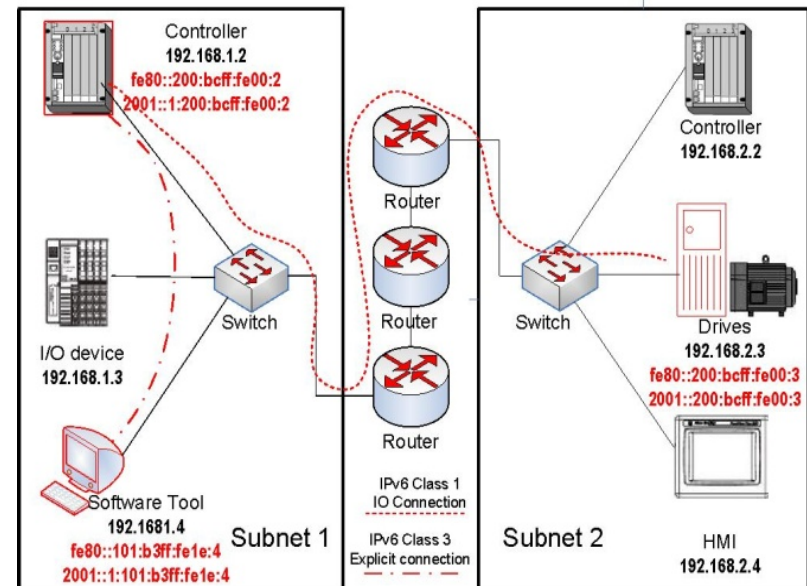
Note: TCP and UDP remain the same

Other new behaviors and features

- ▶ Neighbor Discovery protocol (enhance and replace ARP)
- ▶ New multicast address architecture; Multicast Listener Discovery (MLD) instead of IGMP
- ▶ Duplicate Address Detection (DAD) vs. IPv4 ACD (which is optional)
- ▶ DHCPv6

Holistic IPv6 and Hybrid IPv4/6

- ▶ Forward and backward compatibility needed
 - User support
 - Business case
- ▶ Dual stack critical
 - Controller
 - Device
- ▶ Abstract CIP from IP through TCP/UDP



Changes required to EtherNet/IP Spec

Is it bigger than a breadbox?

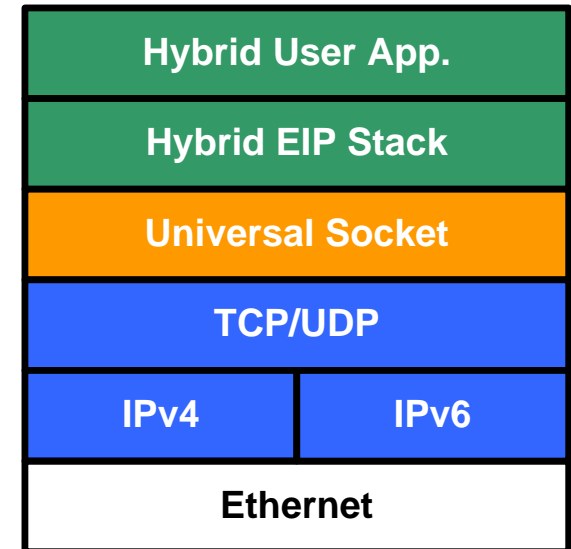
- ▶ General requirements for IPv6 nodes
- ▶ ListIdentity & network browsing
- ▶ ForwardOpen request / response
- ▶ Usage of multicast
- ▶ TCP/IP Interface Object
- ▶ DLR protocol and object
- ▶ CIP Safety on EtherNet/IP (UNID)
- ▶ QoS Object (minor)
- ▶ IPv6 Duplicate Address Detection
- ▶ Dual stack requirements
- ▶ IPv6 address selection and usage

Yes, but it's smaller than CIP Motion

Porting an Application from IPv4 to IPv6

Independent of CIP Stack – how to migrate TCP/IP stack

- ▶ Select dual stack
- ▶ Develop hybrid application
 - IP protocol selection
- ▶ Handle multiple unicast IPv6 addresses
 - IPv6 address selection
- ▶ Broadcast to link-local all-node multicast



Porting an Application from IPv4 to IPv6

Independent of CIP Stack – how to migrate TCP/IP stack

- ▶ Address related interfaces changed
 - IP address
 - Socket data structure
 - Address conversion function
 - Address resolution function
 - Multicast interfaces
- ▶ Socket functions for TCP/UDP data flow control not changed
 - Socket creation, TCP/UDP receiving and sending

TCP/IP Interface Object

Problem:

- ▶ Current object supports IPv4 address config only
- ▶ Need config method allowing for multiple IPv6 addresses
- ▶ Opportunity to “clean up” the current object

Proposed Solution:

- ▶ Define new object supporting IPv4 & IPv6
- ▶ Eventually deprecate current object
- ▶ Ongoing discussion on design of new object

List Identity Command

Problem:

- ▶ Currently sent to IPv4 broadcast addr. No broadcast in IPv6
- ▶ Response contains embedded IPv4 address
- ▶ Needs to account for multiple IPv6 addresses

Proposed Solution:

- ▶ For IPv6 send to link-local all-nodes multicast
- ▶ Define rules for responding when multiple addresses in use
- ▶ Structure of the ListIdentity response still TBD

Forward Open/Close

Problem:

- ▶ IPv4 multicast address embedded in Originator-Target ForwardOpen exchange
- ▶ Needs to allow for IPv6 as well as IPv4

Proposed Solution:

- ▶ Allow Sockaddr Info item to include IPv6 address. Sin_Family indicates IPv4 or IPv6
- ▶ Backwards compatible with current IPv4 implementations

CIP Adaptation of IPv6

A slim volume....

Minimimise impact on users and vendors

Make it as simple as possible to understand

Ensure that no action impacts:

- ▶ IPv4 Devices
- ▶ Ability of an IPv4 host to communicate with a hybrid device

Video using Camtasia or similar tool to show:

- ▶ Single Stack IPv6 Devices Interacting
- ▶ Wireshark traces
- ▶ Dual stack originator simultaneously interacting with IPv4 and IPv6 devices
- ▶ Wireshark traces
- ▶ Communication with a ControlLogix controller using IPv6

Video using Camtasia or similar tool to show:

- ▶ Network browsing
- ▶ UCMM (SW tool to Scanner)
- ▶ Class 3 (SW tool to adapter)
- ▶ Class 1 (Scanner to adapter)

- ▶ Demo and Wireshark traces
- ▶ IPv4 and IPv6

The ODVA Roadmap

Goal: Position EtherNet/IP as “IPv6-Ready”

- ▶ Ongoing prototyping / interoperability demo
 - Continue current work. Demo at Implementor Workshop or Plug Fest
- ▶ EtherNet/IP System Architecture SIG develops specification content
 - Work has started. Currently working on major technical items and evaluating options and alternatives
 - After initial work, begin creating specification content
 - Completion Target: Late Spring 2014

The ODVA Roadmap

Goal: Position EtherNet/IP as “IPv6-Ready”

- ▶ Approval by TRB; publication in Volume 2
 - Target: Late 2014
- ▶ Development of Conformance Tests
 - Test with initial prototype implementation based on spec
- ▶ White paper to assist end users in IPv6 adoption
 - ?????

Questions

