IPv6 Address Autoconfiguration

Dave Katz (cisco Systems)

Co-Chair, Address Autoconfiguration Working Group (Pending)

The Problem:

System bootstrap (“plug ‘n’ play”)
Painless renumbering

Working Group Scope:

Only address autoconfiguration (first part of bootstrap)
Other types of autoconfiguration outside of the scope

Goals

A single mechanism that works across the range of environments

- “Dentist Office”
- Large enterprise
- Full range of administrative controls (fully automatic to highly paranoid)
- “1000 PCs on the loading dock at 4:30 Friday”

Determinism

Single host protocol for all environments

Architectural Positioning

Address autoconfiguration is closely tied to routing

May be sufficient for very lightweight systems

Should be distinct from “kitchen sink” of configuration parameters (DHCPng)

Routing and address structure should be as opaque as possible to hosts

Taxonomy

Manual

Address configured by hand

Autonomous

Host synthesizes address with no external information or interaction

Semiautonomous

Host synthesizes address by combining a priori information and other information gleaned externally

Stateless Server

Host queries a server; the server provides an address without maintaining nonvolatile state

Taxonomy

Stateful Server

Host queries a server; the server provides an address and maintains nonvolatile state

Address Server Deployment

Every wire must have an address server present if hosts are going to talk off of the wire

Local server may very likely be colocated with router

Local server may annotate request and forward it to a more distant server (stateful, integrated with management, etc.)

Basic Mechanism (Host View)

Host autonomously configures an address usable only on the local wire (well-known prefix plus suffix unique on wire)

[Perhaps use PPP negotiation in point-to-point case to deal with two hosts with no unique suffix squaring off on a point-to-point link]

Host sends an address request packet to a well-known multicast address containing:

- Unique (on the wire) token
- Addresses currently in use
- IPv4 address (for v4 transition)
- Domain name
- Authentication information

Basic Mechanism (Host View)

If no response, retry for awhile; may use local-wire-scope address in the mean time (perhaps forever if on isolated network)

Reply contains:

- Fully specified address(es)
- Holding time
- "Stop using address X"

Local Server View

For local (usually stateless, if this is a router) operation, server uses some combination of host-provided information, local topology information, and other a priori information to synthesize an address

Possibilities:

- Topologically-significant prefix + host suffix
- Pool of addresses
- SIT-compatible embedded IPv4 address
- ...

Local Server View

For more complex operation, local server annotates request with topology information and forwards request to remote server

Remote server can do almost anything to create an address (or deny it) so long as uniqueness rules are satisfied:

- Hash host suffix (for compactness)
- Count from 1
- Perform database lookup
- ...

Address Lifetime

Addresses need to die eventually (to aid in renumbering)

Given address size, addresses are unlikely to be reused quickly except in special circumstances (address pools)

Addresses renewable by re-requesting before lifetime expires (providing list of addresses in use aids in determinism)

Often it may be better to continue to use an expired address for existing connections (address server may be temporarily unavailable)

Address Lifetime

Addresses have three states:

- Active (still within holding time of assignment; routability guaranteed)
- Stale (holding time expired or address no longer mentioned by address server in address assignments, routability no longer guaranteed)
- Invalid (address server explicitly telling host to drop the address)

Hosts may continue to use addresses indefinitely so long as they are not explicitly told to stop using them

Renumbering

Servers configured with new topology information

Routers announce that addressing has changed

Hosts request addresses immediately and receive new ones

Routing to old addresses is turned off

Hosts eventually give up on old addresses, or are told to drop them

IPv4 Transition

SIT transition scheme calls for embedded IPv4 address

Two scenarios:

- Dual stack host already knows its v4 address (manually or via DHCP), includes it in address request; easily dealt with in local server

- IPv6-only host makes regular request; server (probably remote) determines appropriate v4-compatible address (perhaps via lookup in DHCP database)

Protocol Interactions

Dynamic DNS updates obviously a requirement

DHCPng needs to be defined for other bootstrapping functions

Status

Architecture proposal presented at BOF in Toronto

Protocol draft under discussion on mailing list

Internet-Draft version of protocol by end of October

VERY agressive timetable mandated by critical path

E-mail

Discussion list:
    addrconf@cisco.com

Request list:
    addrconf-request@cisco.com

Me:
    dkatz@cisco.com