



Active Signaling Protocol (ASP) Execution Environment

Bob Lindell
Computer Networks Division
USC/ISI

ASP Functionality

- Dynamic (remote) loading of AA code
- Security and Resource Protection
- Fine grain network I/O
- NodeOS extended Channel Interface
- User Application (UA) API
- Timing Services

AA Examples

- Ping and traceroute
- Ps, netstat, ifconfig, (remote) cat
- Management apps to map topology
- Routing protocols RIP and PIM
- RSVP and the AFSP variant
- Active Congestion Control (ACC)
- AA probes

ASP Execution Environment

- Hosts the execution of Active Applications (AA)
- User space operating system
- Protocol Programming Interface (PPI)
 - System call interface
- Written in Java with limitations
 - Performance and resource control

Dynamic Loading

- Based on an AASpec
 - AA name
 - Loading path
 - Initial Java class
- Remote loading using either TCP or RDP

Resource Protection

- AA Process Model
- Shared byte code (read only text)
- Separate data space (AA local data)
- Separate file space
- Simple Round Robin scheduling

Security

- Privileged AA(s) have capabilities
- Interface table modification
- Forwarding table modifications
- File space privileges
- Process termination
- VNET packet diversion

Virtual Network Stack (VNET)

- Extensible model including simple layer 2–4 implementations
- Link layers using UDP/IP or ANEP/UDP/IP encapsulation
- IP like unicast network layer with point to point interfaces
- UDP like transport service
- Hop by hop delivery (Router Alert)



Virtual Network Stack (VNET)

- Reliable Datagram Protocol (ACC)
- Zero copy stack
- Mb/s rates with RDP and no Java JIT

IP Network I/O

- UDP and TCP support
- Raw IP support migrating to Netiod
- Fine grain control
 - Header parameters
 - Outgoing/Incoming interface

EE Channel Interface

- VNET, UDP, and TCP
- Interface control "if[n]"
- Extensions
 - EE header processing "if/ip/udp/aaspec"
 - Recv/Send address specs

Timing Services

- Soft State containers with refresh timers
- Retransmission timers
- Can be used for hard state

User Application (UA) API

- Link between applications and the network
- Layered on TCP
- Uses AASpecs for naming and dynamic loading
- No user identity

Anatomy of an AA

- UA Interface
- Activity ID (Globally unique)
- Soft State

Future Work

- Versioning
- AA Interprocess Communication (IPC)
- Better management tools
- AA Java ClassLoader groups
- Network distribution of capabilities
- Raw IP with Netiod and Channel interface
- AA FlowIDs at the ANEP layer (Activity IDs)
- User Identities