Control-On-Demand
Current Approaches:

- Provision of an uniform interface to the network
- Placement of user-controllable computations inside a network

Active Networking
Control-O-Demand — Key Ideas

- Efficiency
- Flow Pinning
- Explicit Flow Labeling
- Use the IPv6 Flow Label
- Characteristic depending upon network conditions
- Each receiver in a multicast can specify different desired flows
- Flow specific control can be specified by the flow participants
- C-o-D enables networks to support different control needs
- Networks already support different transport needs

Stay in the fast path
Lightweight signaling (using extension headers)

- Frame Peeking

- Reducing bandwidth across the interface

  Controller does not have to process every packet

  - Best-effort control

  Flow semantics not modified by controller not being present

  - Enhancement control

  - Reducing on-line requirements

Staying in the fast path
Architecture

Controller

- Responsible for installing and initializing flow specific controller
- Interface between routing core and flow specific controller

Meta-Controller
core

synchronization between the per-flow control and the runtime

C-to-D interface is designed to minimize coupling and

- Frame Peeking

- Application level framing

- Controller performance enhanced by:

  - control

  - uses the Control on demand interface to customize per-flow

  - dynamically installed by the M-Controller

Flow-specific Controllers
Error condition, forward packet
Flow specific controller active, queue
PCB exists, keep statistics, forward packet
W-Controller knows about flow,
create PCB, cache output port, forward packet
set state to Initialize

Initialize

Copy packet to W-Controller
if C-o-D header present

NULL 0(1) Flow classification based upon flow id. into 1 of 4 states:

Flow Classification
Implementaiton Specifics

- Implementaiton Specifics

- Additions to Linux IP4G Implementation:

- Flow state caching and flow pruning

- Per flow output queue scheduling

- Per flow queuing

- Flow specific controllers installed using run-time dynamic

- Flow specific controllers installed using run-time dynamic

- M-Controller-Routeing-Core Interface through socket options

- M-Controller-Routeing-Core Interface through socket options

- Linux in-Kernel C-D Implementation

- Linux in-Kernel C-D Implementation
Virtual Topology — 10 Mbps Ethernet Implementation

### Diagram

- **Solaris** (Modified Kernel)
  - 5555::2
  - 5555::3

- **Linux** (IPv6 Extensions)
  - 1234::0 Net
  - 1234::2 Net
  - 5555::1 Net

- **Control On Demand Router**
  - 1234::1
  - 5555::1 Net

- **Solaris** (Modified Kernel)
  - 5555::2

---

### Topology

**Implementation Topology**
<table>
<thead>
<tr>
<th>Command</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET-PACKET-STATE</td>
<td>{{Packet? State?}}</td>
</tr>
<tr>
<td>SET-SCHEDULE</td>
<td>{{Byte? Rate?}}</td>
</tr>
<tr>
<td>EXPLODE-QUEUE</td>
<td>{{Queue Info, Start Offset, Length}}, {{matches</td>
</tr>
<tr>
<td>CHANGE-FLOW-STATE</td>
<td>{{Flow, State}}, Set Flow State</td>
</tr>
<tr>
<td>Effect</td>
<td>Effect</td>
</tr>
</tbody>
</table>

(e.g. Discard) Set packet state to Rate served, output Rate. After Byte? bytes, which matches # of matches, Start Offset, Length, Queue Info, EXPLODE-QUEUE, CHANGE-FLOW-STATE, SET-SCHEDULE, SET-PACKET-STATE, Command Parameters Effect.
Applications

- Variegated Multicast trees
- Reliable Multicast
- Application-specific congestion control
  - Video smoothing, stream thinning
  - Application-specific traffic shaping
Video smoothing: •

- Corresponding to frame is discarded.
- Frames are discarded first.
- All packets in frame cannot be fit into output buffer.
- Frames are discarded least important dataframes. &
- Upon congestion, application specified headers are parsed to

Stream thinning: •

- Application-specific traffic shaping
- Provides proper rates such that buffers are not over/underflowed
SET_PACKET_STATE

if PIN[0] < PIN.valid + 1:
    PIN.valid = PIN.valid + 1;
    if (Queued - Queued == 0):
        PACKET_DISCARD;
        PIN.PIN[PIN.valid][0] = state.id;
        PIN.PIN[PIN.valid][id][matches[?]];
        if (type i = I-PR and Queued < 0.3 * size)
            if (app-level-header) matches[?]
                for ? in matches :
                    if size < 0.8 * size
                        (Queued - Queued < 0)
                        EXPOSE-QUEUED;
                        NEW_PACKETS

Stream Thinning

Application of the GoD Interface
transmission rates over time intervals

- Given frame sizes, client buffer sizes — controller can compute

- Overflow/underflow

- Controller schedules frames to prevent client buffers from overflowing or undershooting.

**Video Smoothing**
Retransmission scheme

- Caching at intermediate nodes — best possible
- Reliable Multicast

Reliable Multicast — Controller provides at least one sub-stream

- Video coded into two sub-streams of equal importance
- Using a Multi-Description Coder (with Vinary Vals mismatch)

Applications in development
Behavior of C-o-D flows in heterogeneous networks

Wide-area dynamics of C-o-D

Simulation of control on demand systems (AN-Sin)

Intercontroller communication — Composition of policies

Analysis of contention at output queues

Vague gate multicast trees

Reliable multicast

Implementation of other policies — field test of C-o-D interface

Future Work