The Click Modular Router

Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti and M. Frans Kaashoek
Laboratory for Computer Science, MIT

Presentation By
Bryan D. Payne
University of Maryland
Element Code

Class NullElement : public Element {
  public:
    NullElement()                  { add_input(); add_output();}
    const char *class_name() const { return "Null";}
    NullElement *clone() const     { return new NullElement;}
    const char *processing() const { return AGNOSTIC;}
    void push(int port, Packet *p)  { output(0).push(p);}
    Packet *pull(int port)         { return input(0).pull();}
}

- Easy to write, most take about 120 lines of code
- New element classes can be added at run-time
- Express a single, simple idea:
  - CheckIPHeader
  - DropBroadcasts
  - DecIPTTL
Configuration Language

// Declare three elements
src :: FromDevice(eth0);
ctr :: Counter;
sink :: Discard;

// Connect them together
src -> ctr;
ctr -> sink;

- Language allows for compound elements to simplify syntax
- Only specifies connection between elements
Extensions...Endless Options

- Scheduling & dropping policies
- Queuing requirements
- Differentiated services
- IP tunneling
- Ethernet switching
- IP header compression / decompression
- IPsec
- WLAN communication
- Network address translation (NAT)
- Firewalls
- etc
How Click Runs

- 2 options: **kernel** or user-space driver
- Loading new config...
  - Normally destroys state and drops all packets
  - Hotswap possible for some changes
- Handlers (used to modify local element config)...  
  - `/proc/click/<element>/<handler>`
  - View statistics
  - Change queue lengths
Optimizations

- Polling versus Interrupts yields huge gains
Optimizations (part 2)

- Avoiding virtual function call overhead
- Combining elements doesn't increase performance
  - Yet Knit claims big performance gain with flattening
  - Recall Clack from the Knit paper
- Some optimizations limit functionality
Pros & Cons

➢ Pros
  • Clean, modular design and configuration
  • High degree of flexibility
  • Good performance on commodity hardware
  • Potential platform for Active Network research

➢ Cons
  • Performance comparison with commercial routers
  • Element configuration may be harder than advertised
Conclusions

- Flexible, open, modular router design
- Performance appears good, but could use further study
- Project still under active development (linux-2.4.20)