Boosting Towards Active Nets

(a talk on network evolution....)

Jonathan M. Smith

University of Pennsylvania

Protocol Design (Today)

- □ Begins with problem to be solved, including assumptions
  - » e.g., TCP’ s “reliable bytestream”, over IP
- □ Optimization:
  - » Measure
  - » Identify common case
  - » Make it fast
  - » Repeat until satisfied.....

Critique of Methodology

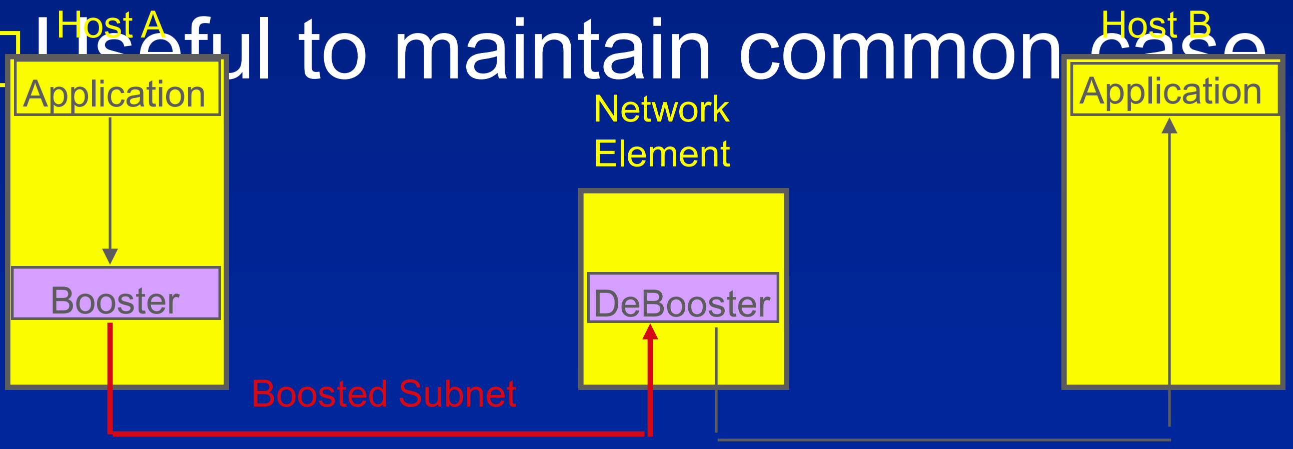
- □ Pessimistic Design Style
  - » Assume worst-case
  - » Pare away functions to get “fast-path”
- □ Optimizations Fragile
  - » Environment Changes (WWW)
  - » Common Cases Change (delay, loss, ...)
  - » Things can break BADLY! (try at home :-)

An alternative methodology

- □ Assume things are working well
- □ Detect when they are not (policy)
- □ Add functions (mechanism) to fix
- □ Functions are called “protocol boosters”
- □ An optimistic approach to transparently achieving high end-to-end performance

Protocol Boosters

- Protocol Elements added ‘ ‘as-needed’ ’
- Example of “optimistic” design method
- Useful to maintain common case



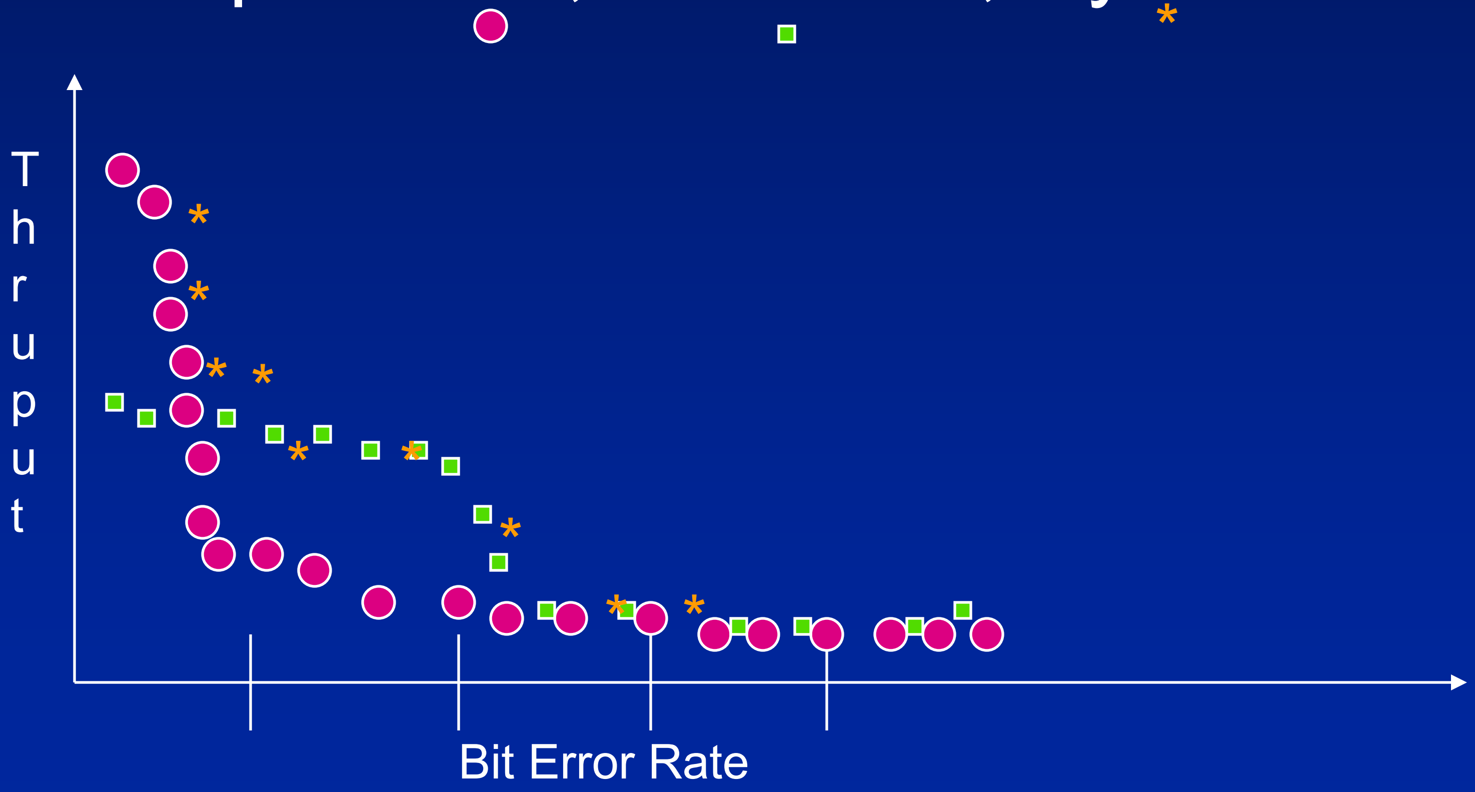
Examples

- Implemented over IP on FreeBSD
  - Encryption Booster
  - Compression Booster
- FEC Booster at Bellcore
- Hardware Support: The P4*

*see http://www.cis.upenn.edu/~boosters/boosters.html

Performance Potential:

□ Thruput: TCP, TCP/FEC, Hybrid

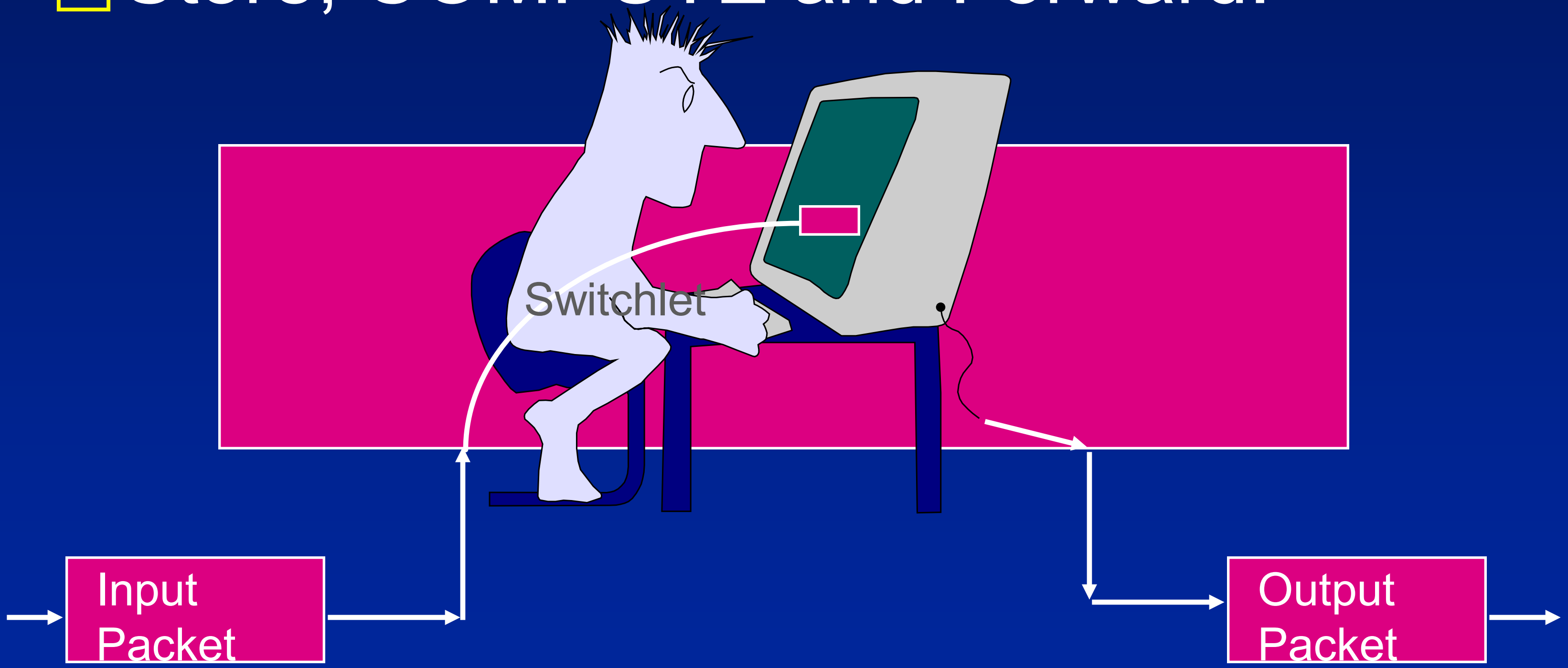


Thus, a question:

- □Q: What’s the network infrastructure needed to support this idea & others?
- □A: A *programmable* network infrastructure!

SwitchWare switching

- Store, COMPUTE and Forward!



‘ ‘Active’ ’ Networks

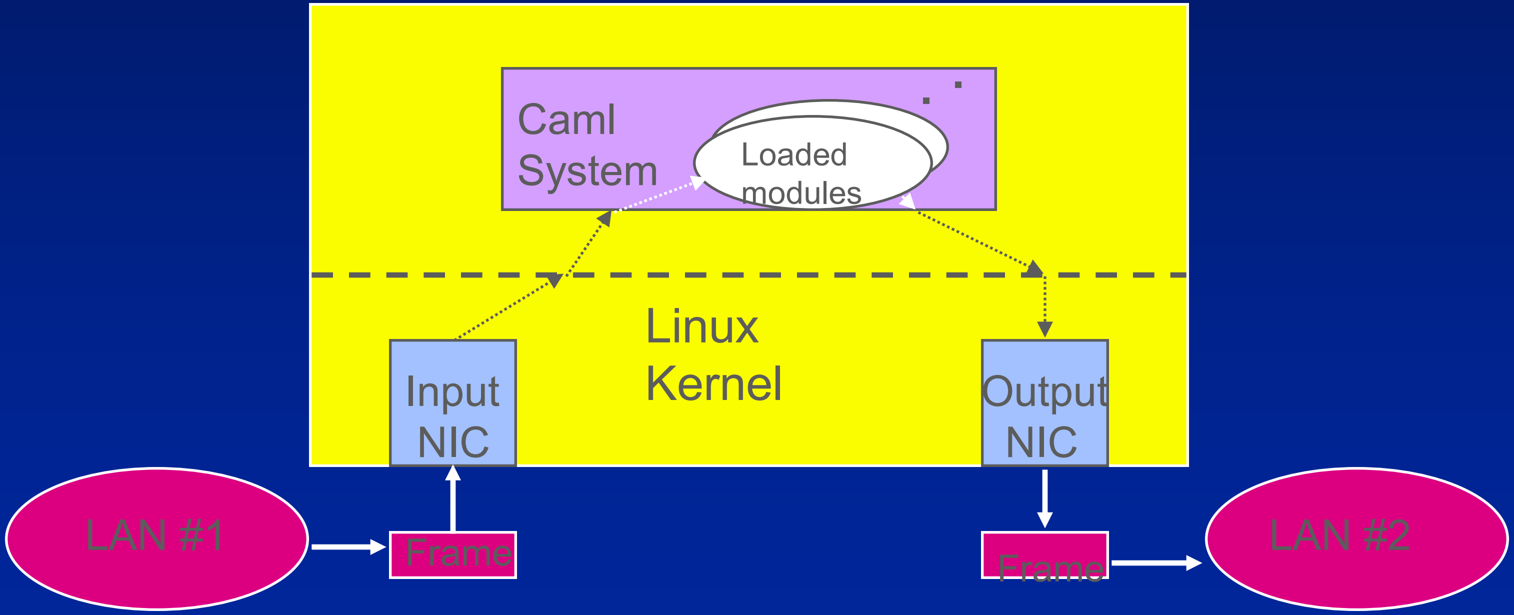
- □ Accelerate service creation with programmable network infrastructure
- □ Balancing flexibility and security
- □ Ad hoc architectures difficult to trust
- □ Is this just another O.S. problem?

The SwitchWare Language-Oriented Model

- □ Switchlet Language for users (SL)
  - » formal semantics restrict programs
  - » (Boosters make *fine* Switchlets :-)
- □ Wire Language for communicating (WL)
  - » formal semantics across boundaries
- □ Infrastructure Language for Virtual Machine (IL)
  - » formal semantics supported on metal: run-time

Current Software

- Active Bridging



Lessons from Bridge

- 16 Mbps vs. 32 Mbps for "C" equivalent
- Incremental Loads:
  - Buffered Repeater
  - Self-Learning
  - Spanning Tree Algs. (DEC & IEEE)
  - Automatic STA Transition in <0.1sec
- http://oilhead.cis.upenn.edu/~salex

Boosting Towards Active Nets

□*Trying to change the "tempo" of network evolution by design/architecture*

□*Protocol Boosters is a design *method**

   » Optimistic and as-needed functions

   » Consistent with "end-to-end" argument

□*Active Nets provide ideal infrastructure for Protocol Booster deployment*

http://www.cis.upenn.edu/~switchware

http://www.cis.upenn.edu/~boosters