CLICK

A NEW PROGRAMMING LANGUAGE FOR DEVELOPING SOFTWARE ROUTERS

Under the Guidance of

Prof. D Manjunath and Prof. Anirudha Sahoo

MTech Seminar
Presented By

Vishal Prajapati
Indian Institute of Technology, Bombay

Acknowledgment

The graphs and figures are taken from Eddie Kohler's PhD thesis.

Network Diagram is created on "http://www.lovelycharts.com/"

Contents

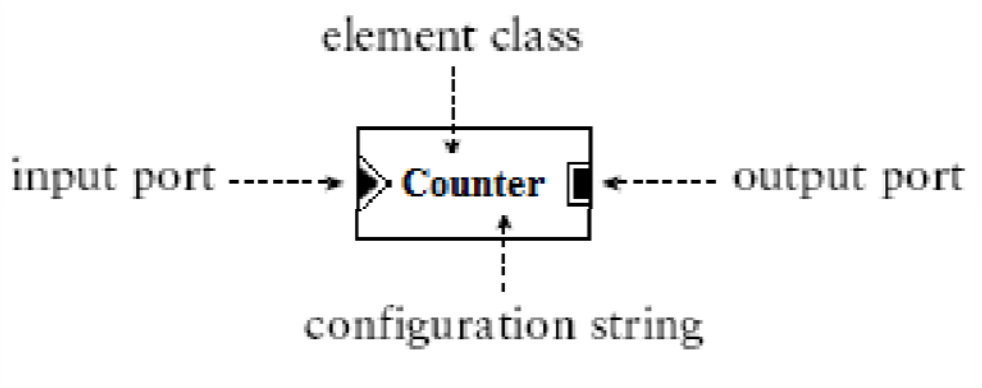
Introduction

- History - “Routers”
  - Packet forwarders => firewall, load balancer
  - Closed, Static, Inflexible Design.

- Solution – ***“Click”***
  - Simplicity
  - No. of lines code
  - Solution of problems simple then normal

Architecture

- Modular design
- Directed graph
  - Element Class
  - Ports
    - Push
    - Pull
    - Agnostic
  - Handlers



Click – The Programming Language

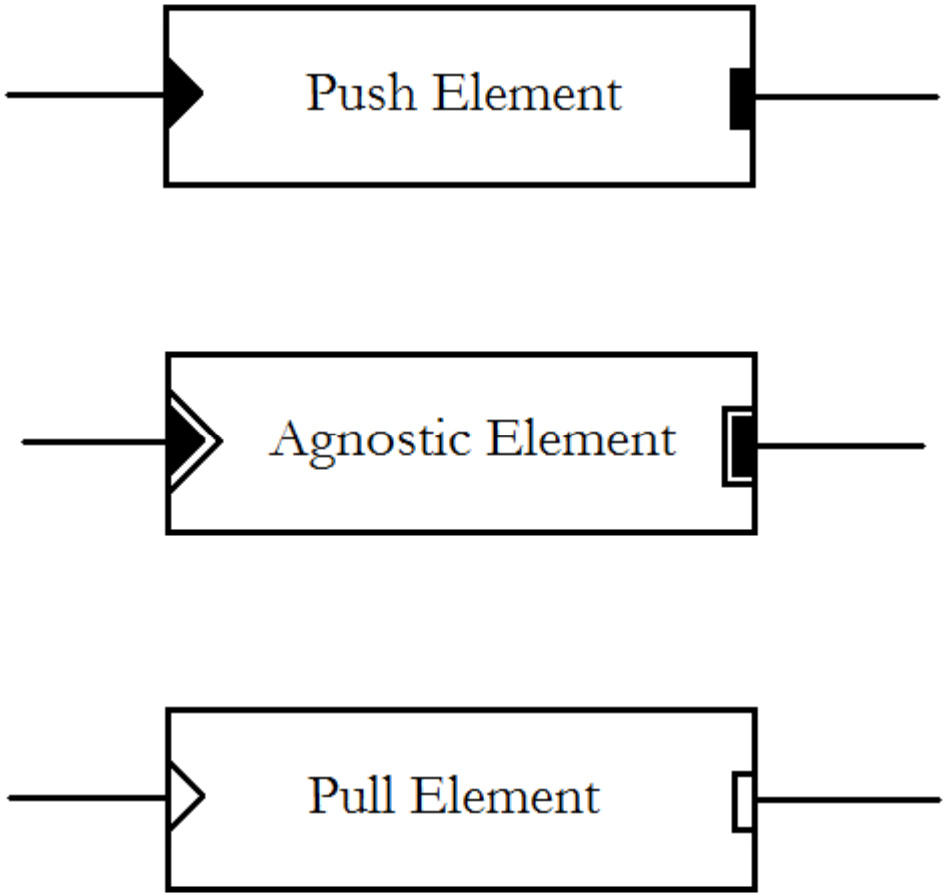
- Declarative Language
- Written in C++
- Syntax

```
h :: Hub
q :: Queue
FromDevice(eth0) -> [0]h;
FromDevice(eth1) -> [1]h;
h[0] -> q;
h[1] -> q;
q -> toDevice(eth0);
q -> toDevice(eth1);
```

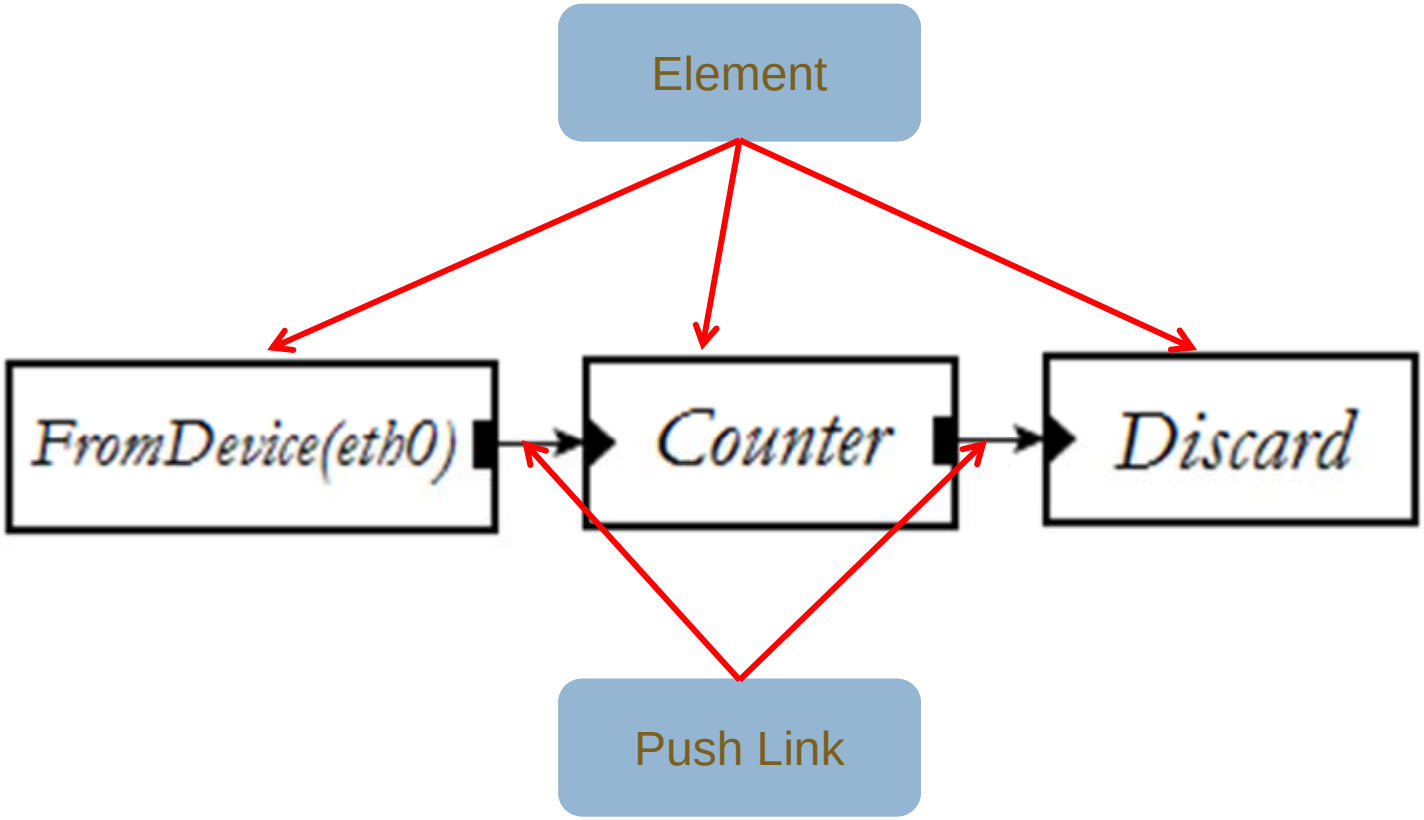
- Examples

```
----------------------- C++ -----------------------
class Hub : public Element { public:
    Hub();
    ~Hub();
    const char *class_name() const
      { return "Hub"; }
    const char *port_count() const
      { return "-/="; }
    const char *processing() const
      { return PUSH; }
    const char *flow_code() const
      { return "#/[^#]"; }
    void push(int port, Packet* p);
};
```

Symbols and Semantics



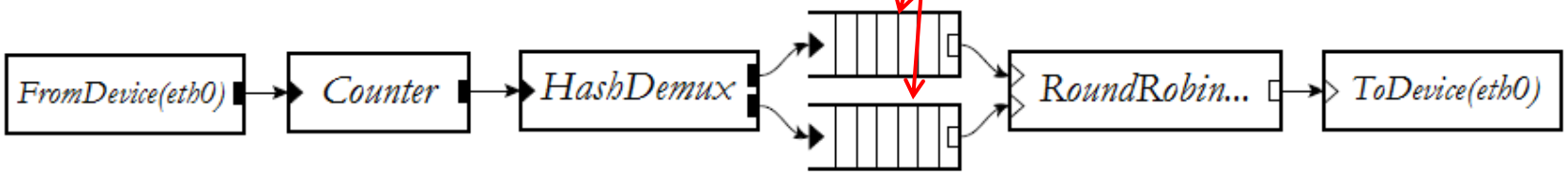
Example – 1



Simple Router

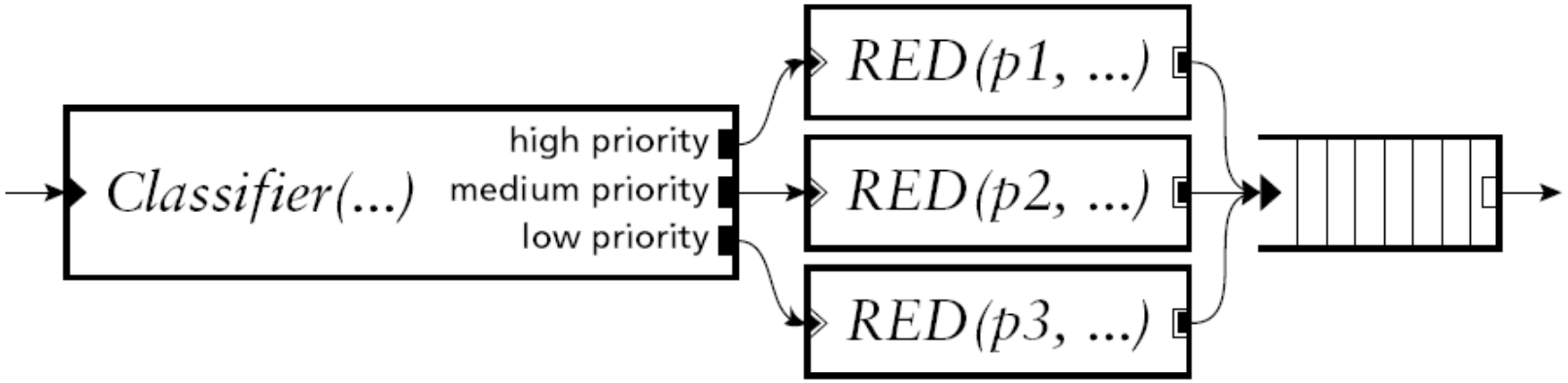
Example – 2

Queue is the element having both (PUSH and PULL) types of ports.



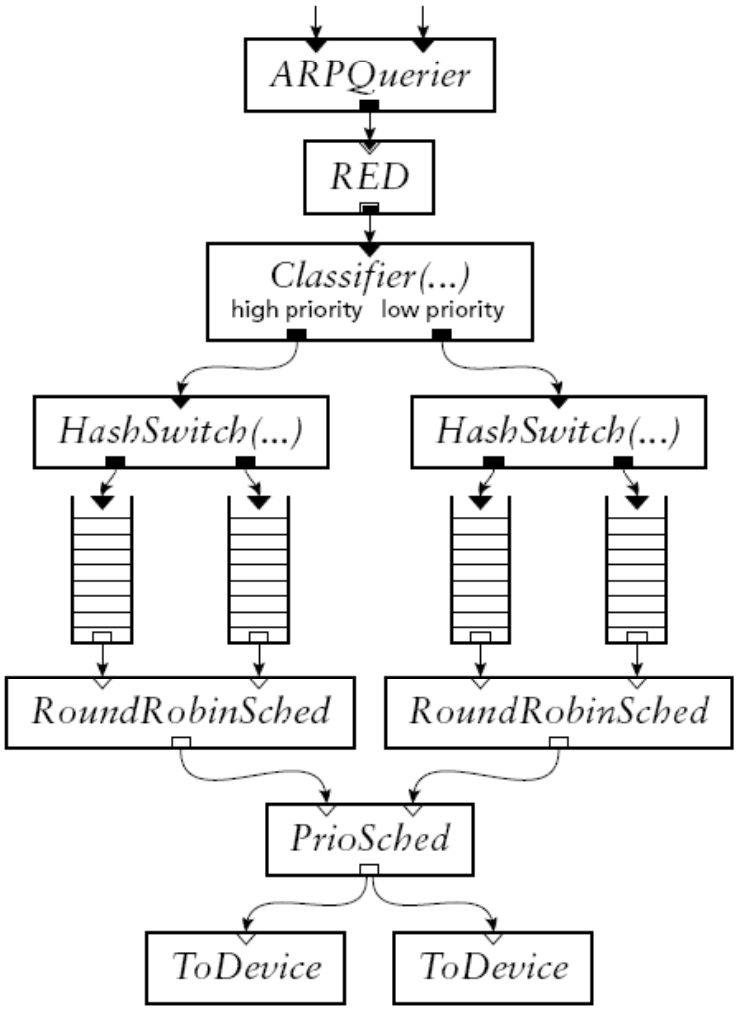
Implementing Stochastic Fairness Queuing

Example – 3



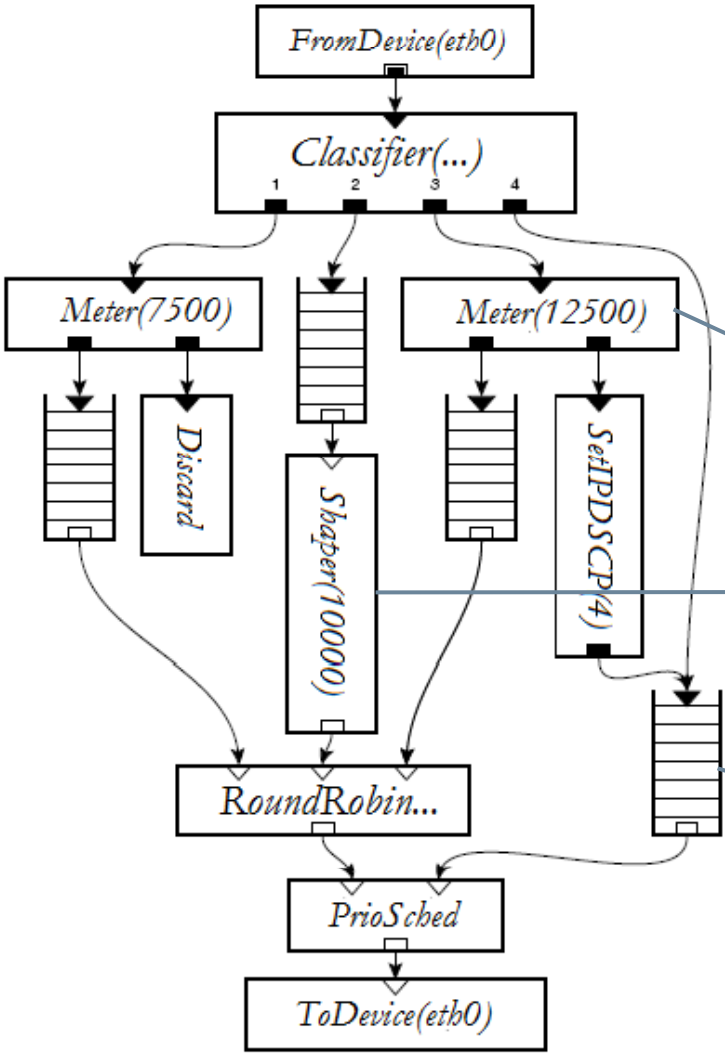
Implementing RED

Example – 4



Implementing Priority flow on the Stochastic Fairness Queuing with dropping mechanism

Example – 5



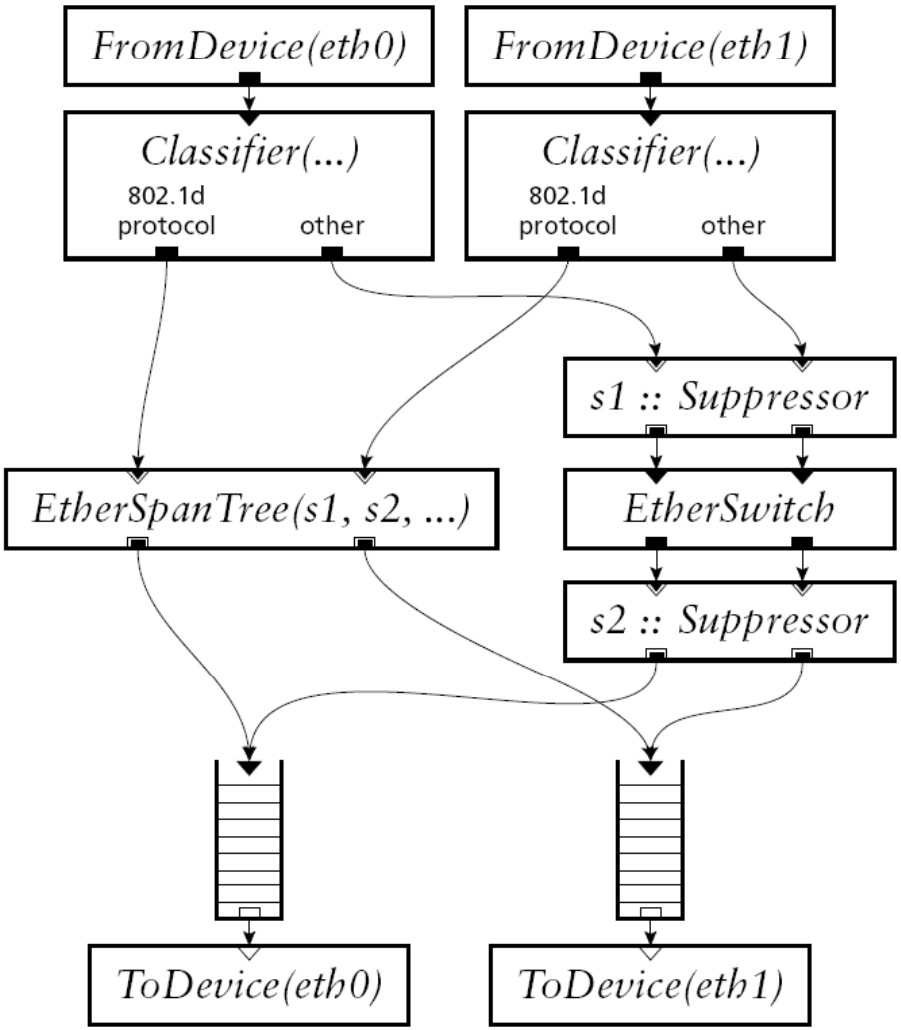
Differentiated Services Algorithm

Will send12,500 packets/second on its first output port other on second output port.

Will pull10,000 packets/second queue will determine which packets to drop, if busty flow.

Best effort flow queue.

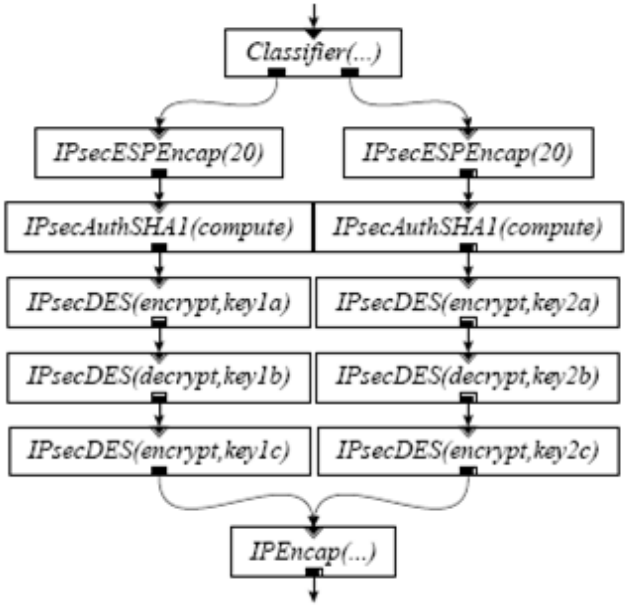
Example – 6



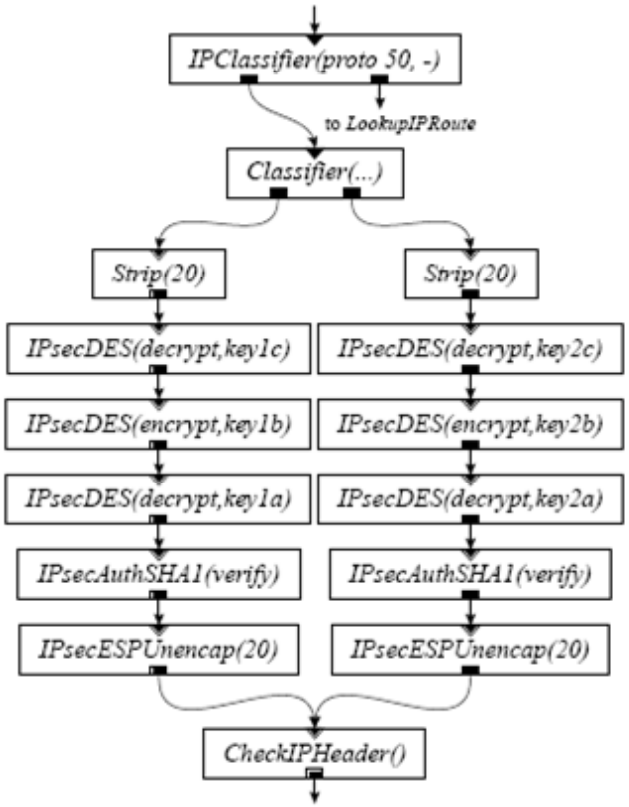
Ethernet Switch

This works as a 802.1d standard learning bridge between two networks connected to eth0 and eth1 ports.

Example – 7



IPSec VPN Tunnel Encryption configuration.



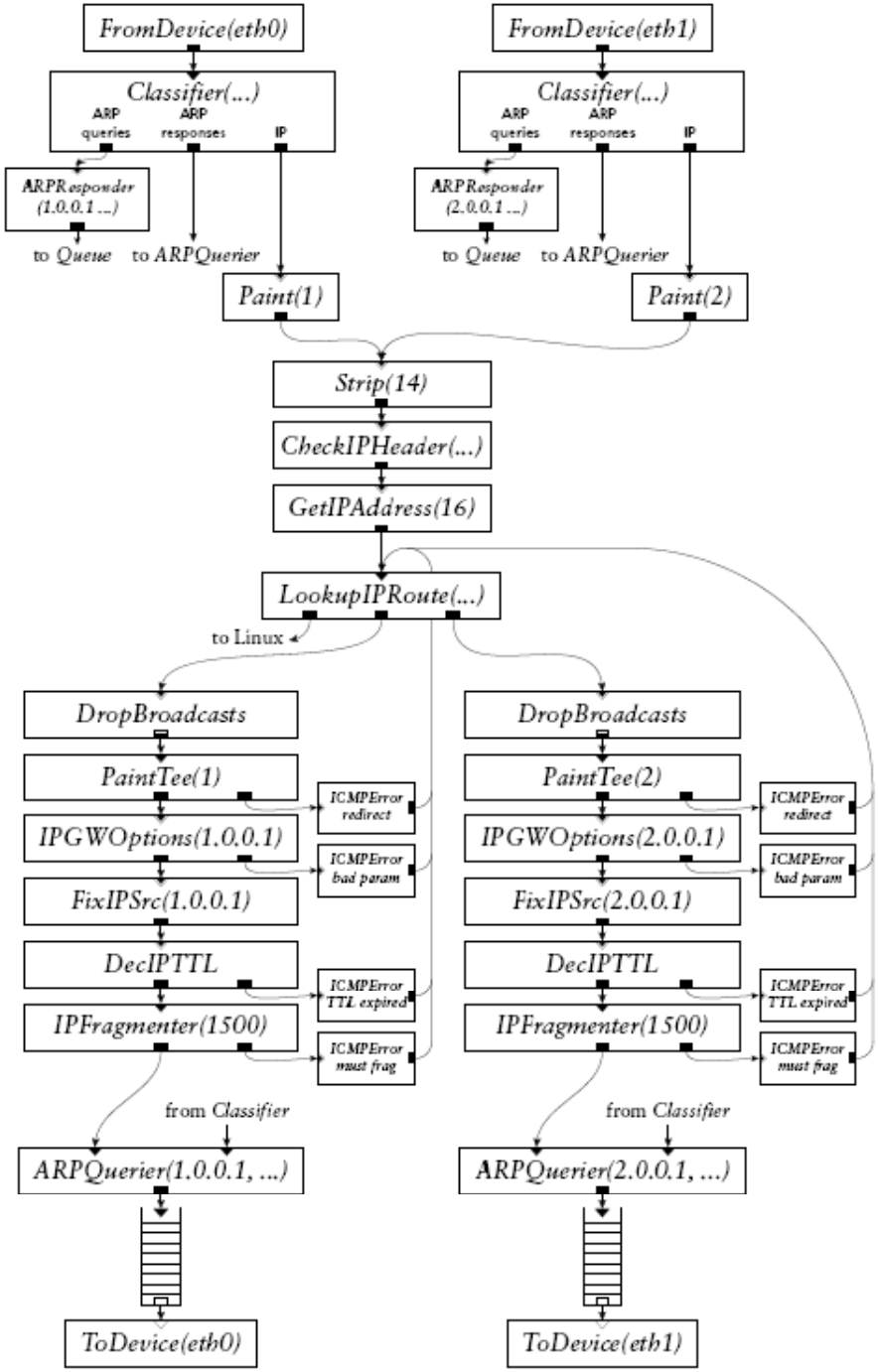
IPSec VPN Tunnel Decryption configuration.

IPSec Implementation

Figure Source [4]

Example – 8

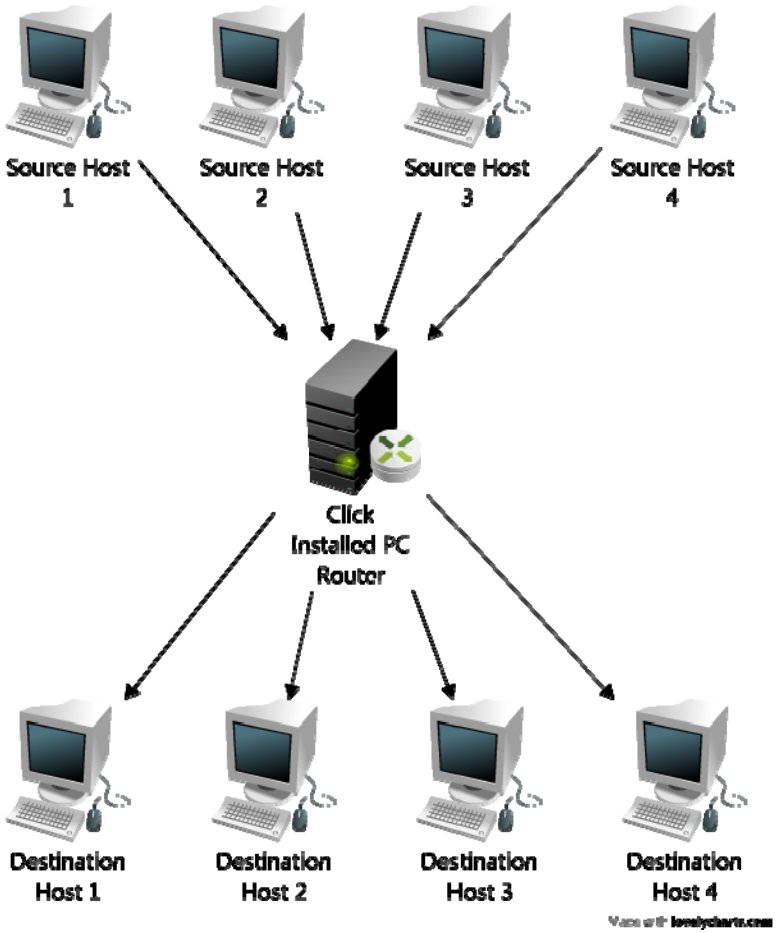
Full fledged
IP Router



Evaluation

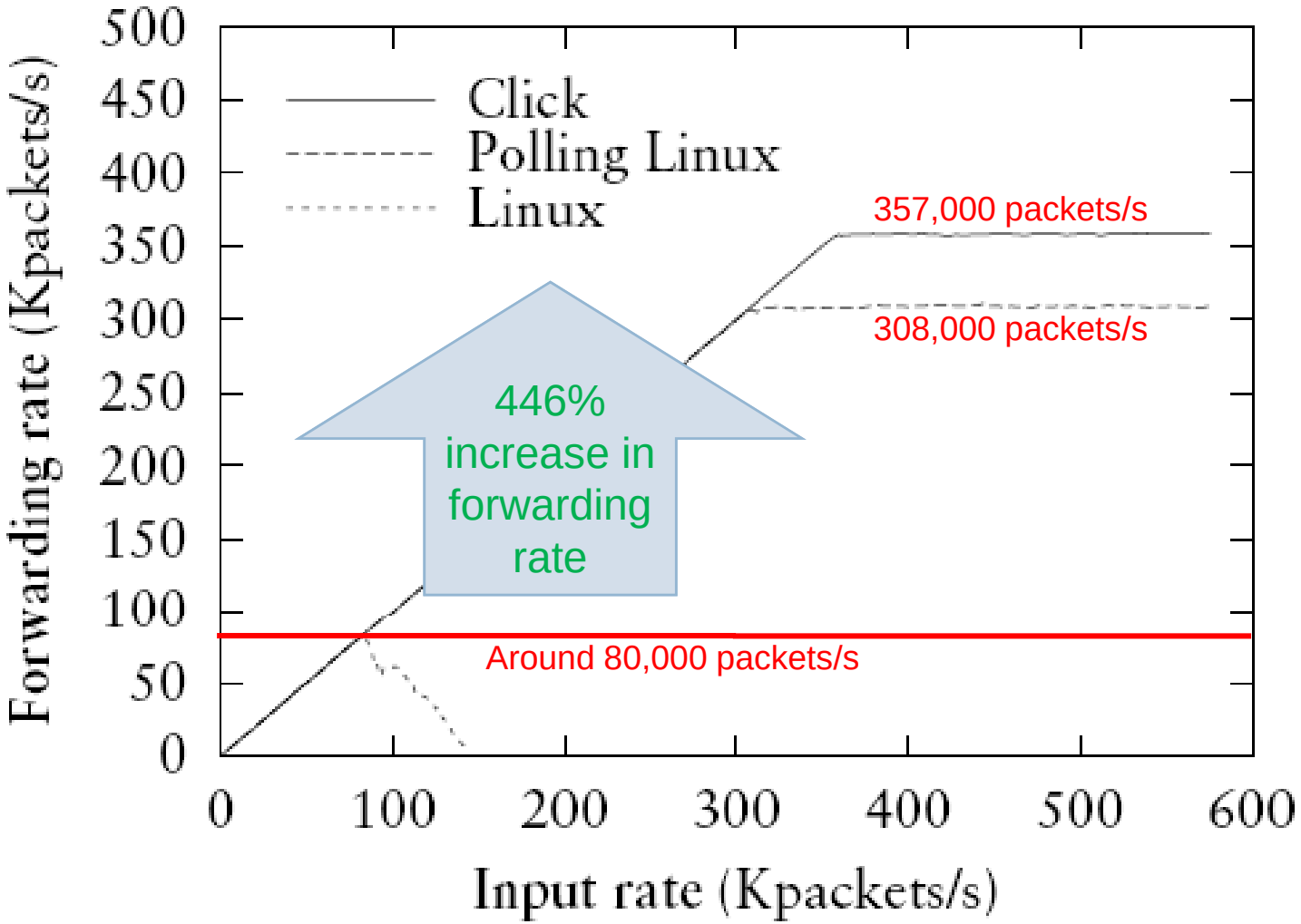
- Forwarding path
- Performance of the **“*Click*”** diffserv configuration
- Comparison of IP router and non-IP router created with click
- CPU time breakdown

Experimental Setup

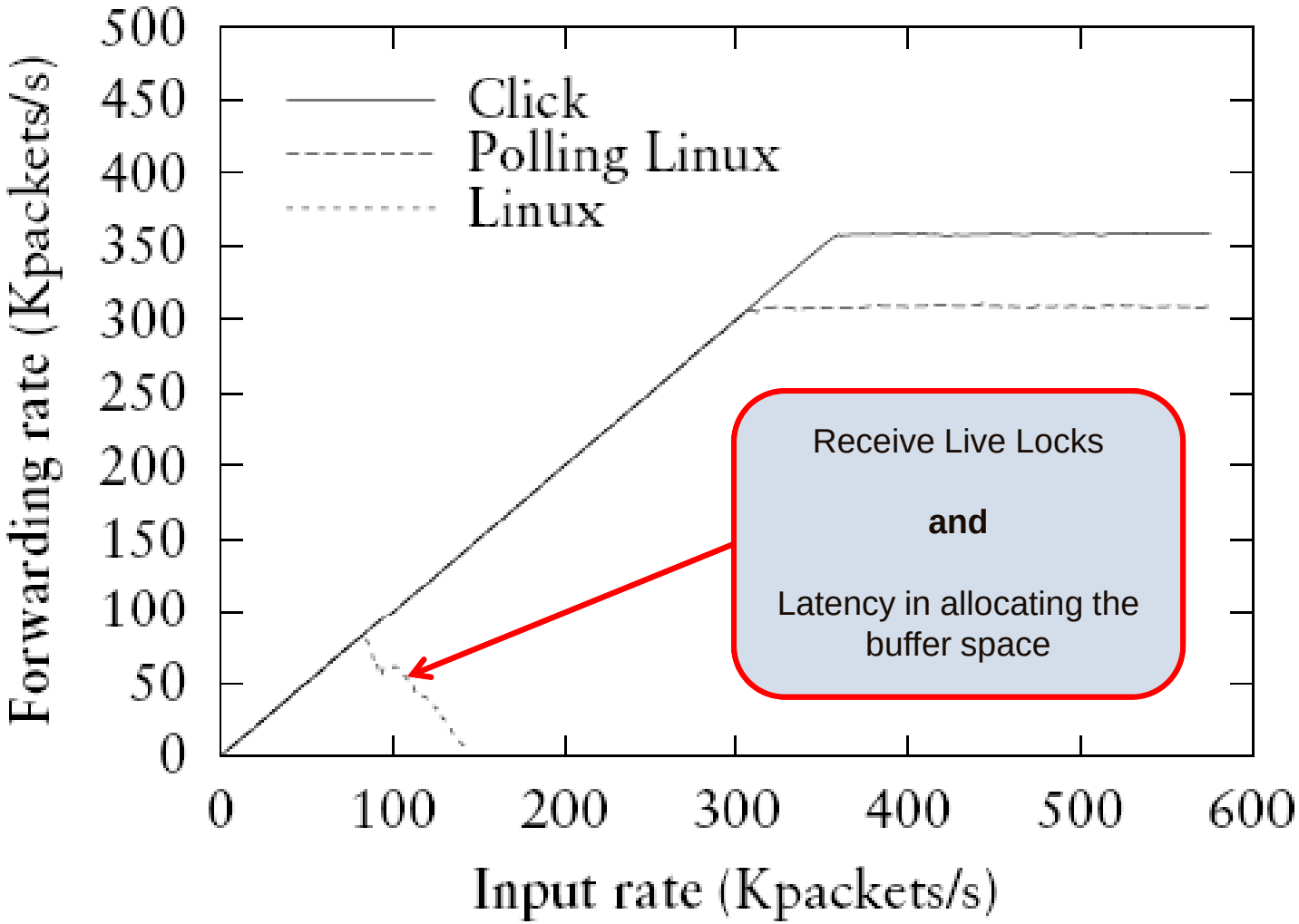


- All are having DEC 21140 Tulip 100 Mbit/s PCI Ethernet Controllers.
- Router Host
  - 700 MHz Intel Pentium III CPU
  - Eight Ethernet Controllers are on multi-port cards split across the motherboard's two independent PCI buses.
- Source Hosts
  - 733 MHz Pentium III CPUs
  - Generates the 64-byte packets per second of UDP flow.
- Destination Hosts
  - 200 MHz Pentium Pro CPUs
  - Counts and Discards the packets.

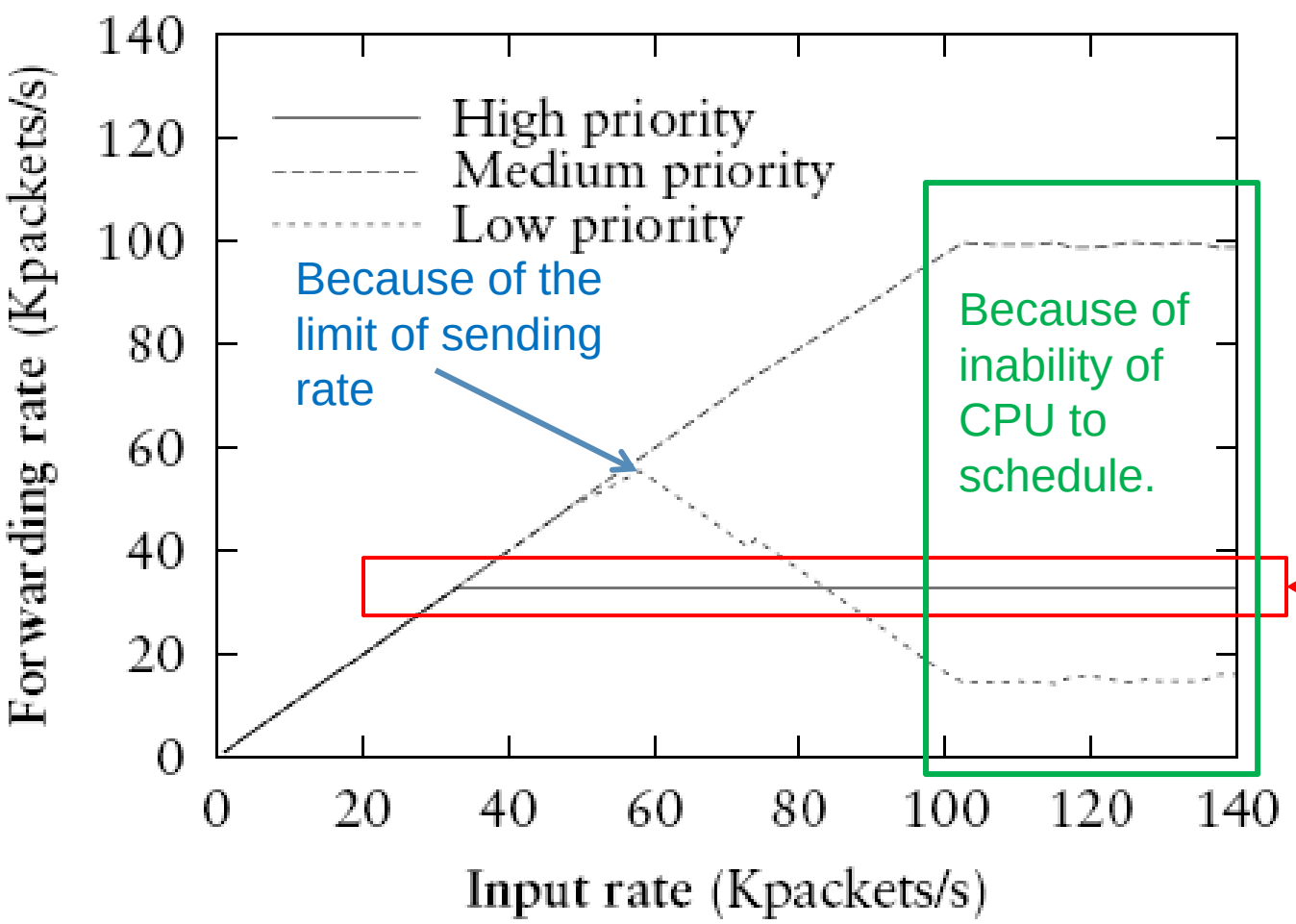
Forwarding Path

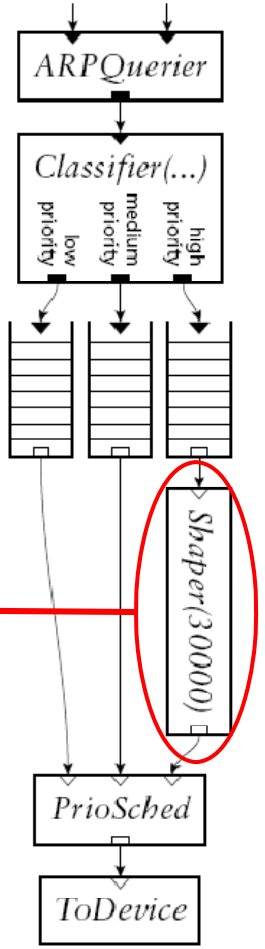


Forwarding Path

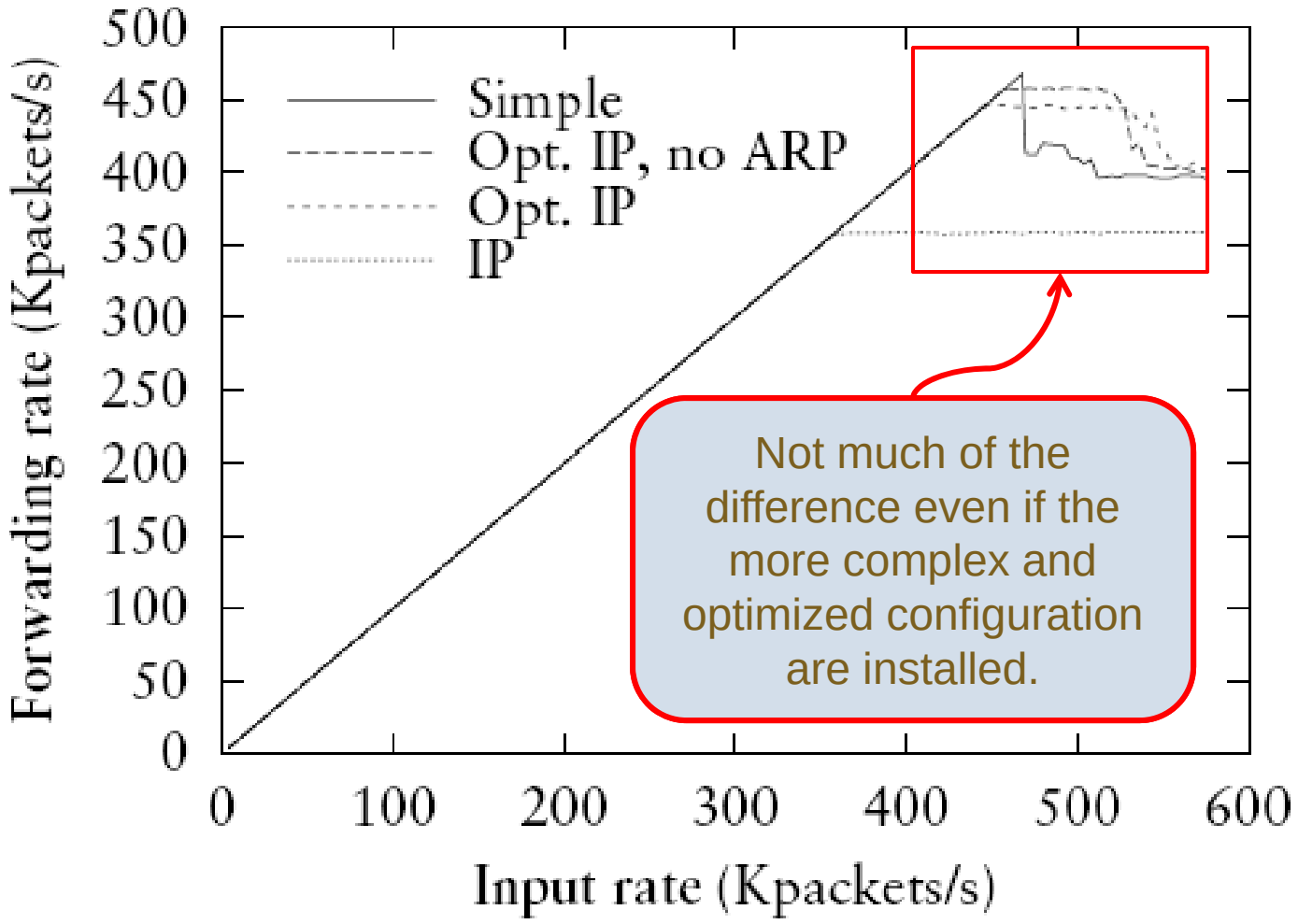


Performance of the Click diffserv configuration

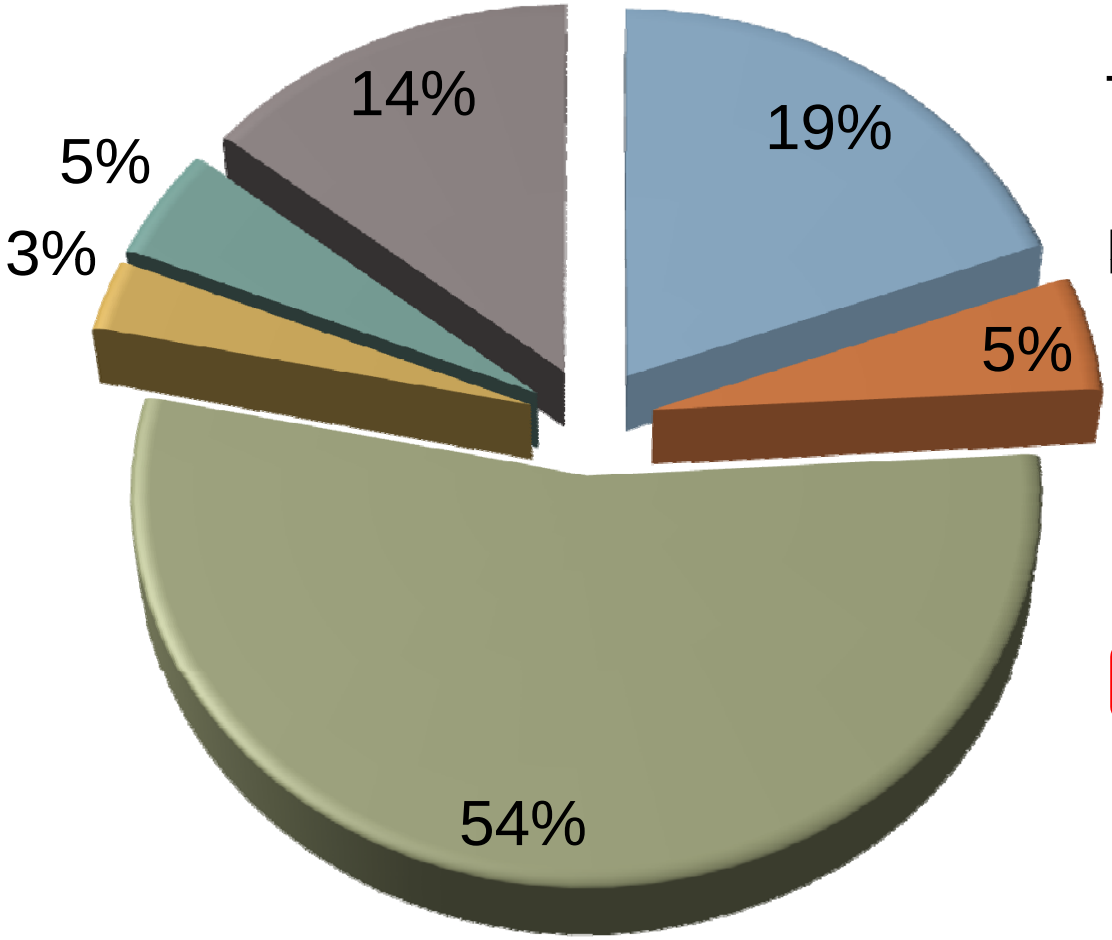




Comparison of IP router and non-IP router created with click



CPU Time Breakdown



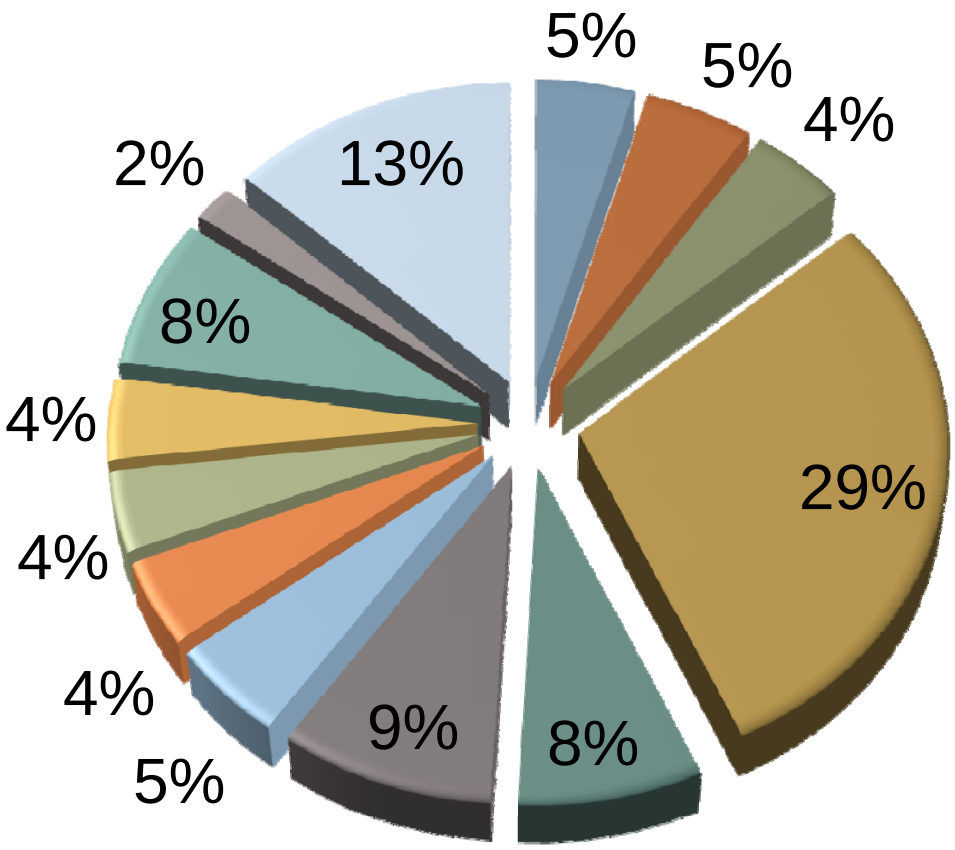
Total Time : 2905 ns

Push Time : 1572 ns

- Polling packet
- Refill receive DMA ring
- **Click forwarding path: push**
- Click forwarding path: pull
- Enqueue packet for transmit
- Clean transmit DMA ring

CPU Time Breakdown (Cont…)

Per element execution time



Total Time : 1455 ns
CheckIPHeader Time : 457 ns

Limitations

- Limitations of the tool
  - ***"Click"*** cannot be used to create the coarse-grained elements.
    - Coarse-grained elements are required when control or data flow doesn't match the flow of packets.
      - Ex. BGP.
  - ***"Click"*** cannot schedule the CPU per individual flow.
    - It is using stride algorithm for CPU scheduling.

- Limitations of the language
  - Compound element classes
    - Ex. The customized handlers implementation is not possible.
  - Compound elements are an imperfect abstraction mechanism.
    - They do not hide themselves from the user.
    - They are strictly less powerful than native element classes.

Conclusion

- Simple design
- Readable language configurations
- Better performance in smaller networks
- Easy to find faults because of modularity

References

[1] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The Click modular router. *ACM Transactions on Computer Systems, 18(4),* November 2000.

[2] The Click Modular Router Project: *http://read.cs.ucla.edu/click/*

[3] Eddie Kohler, Click for Measurement**,** *UCLA Computer Science Department Technical Report TR060010,* February 2006

[4] Benjie Chen and Robert Morris, Flexible Control of Parallelism in a Multiprocessor PC Router, *USENIX 2001 Annual Technical Conference,* June 2001.