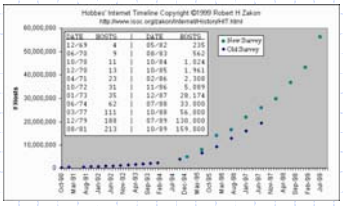
CHAMELEON - A System for Adaptive QoS Provisioning

Rajesh Krishna Balan

Centre for Internet Research

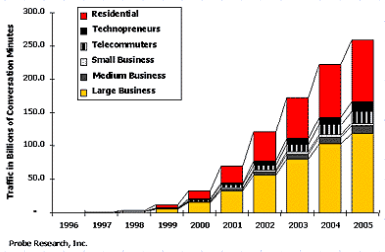
- **Internet has proliferated rapidly**



Date	Hosts
12/69	4
12/79	188
01/89	80,000
07/95	6,642,000
07/95	8,200,000
07/96	16,729,000
07/97	26,053,000
07/98	36,739,000
07/99	56,218,000
07/00	80,000,000

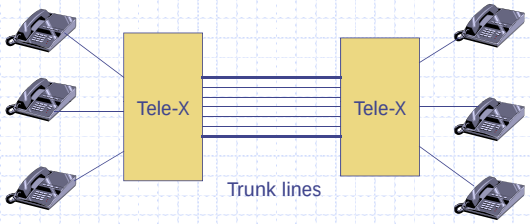
(projection)

- **Strong Interest in carrying Voice traffic over the Internet** - 250 billion conversation minutes by 2005

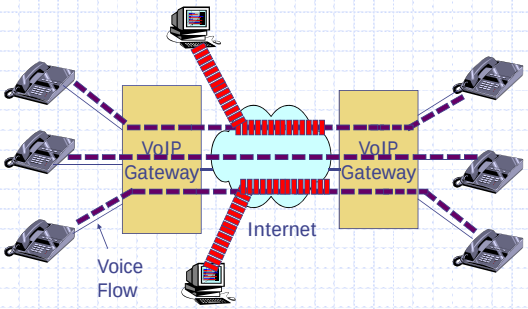


- **Internet is designed for non-real time data communications** - Assuring QoS is a problem.

Why there is no such problem in POTS?



- Statistical guarantees are provided

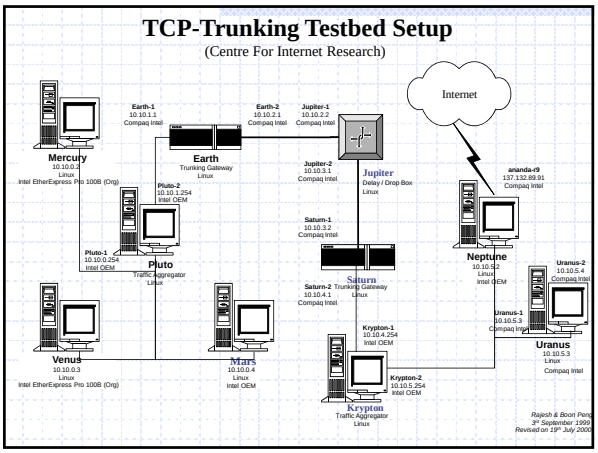


Technical challenges

- Lack of guarantees in terms of BW, packet loss, delay and jitter - Quality of Voice over Internet suffers. IP is only best effort!!
- QoS guarantees are primarily provided by two mechanisms:
  - RSVP (Resource Reservation Protocol)
  - Priority Queue Mechanism
- DiffServ framework initiative under IETF and QBone initiative under Internet2 or Next Generation Internet hold promise.

Objectives

- To search for and acquire various hardware and / or software tools which can generate application specific traffic which follow TCP dynamics.
- To verify and evaluate the properties and capabilities of TCP tunnels.
- To develop a complete system which uses TCP tunnels to provide QoS for various classes of traffic. This system should also be adaptive and thus be able to adjust itself to the traffic conditions on the network.
- To develop a mechanism by which the system can be used as an efficient yet reliable link layer protocol for transmitting TCP data over lossy links.



Traffic Generators

- Need for Realistic Traffic Generators
- Mathematical Models (Poisson, Erlang, Exponential etc.)
- Trace-based Models (collect usage statistics / data and replay etc.)
- Ad-Hoc Models (models with no realistic component. E.g. models which just generate packets and send them out as fast as possible etc.)

Hardware Traffic Generators

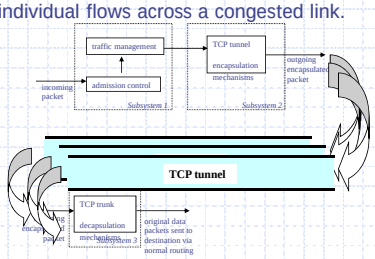
- Very expensive (> $70,000 for a full system)
- Used for stress testing production networks
- Generally unable to generate application level traffic (telnet, HTTP etc.) which follow TCP dynamics (re-transmissions, window size negotiations etc.)

Software Traffic Generators

	Ttcp	Iperf	Surge	IPB	AB	Tcplib	TG	Netperf	Netspec	Rtpplay
Telnet	NO	NO	NO	NO	NO	YES	YES	NO	YES	NO
ftp *(1)* **(TCP bulk)**	YES	YES	NO	NO	NO	NO	NO	YES	YES	NO
ftp *(2)* **(realistic)**	NO	NO	NO	NO	NO	YES	YES	NO	NO	NO
HTTP	NO	NO	YES	YES	YES	NO	NO	NO	YES	NO
UDP voice *(3)* **(UDP bulk)**	NO	NO	NO	NO	NO	NO	YES	NO	NO	NO
UDP video *(4)* **(UDP bulk)**	NO	NO	NO	NO	NO	NO	YES	NO	NO	NO
RTP voice *(5)* **(realistic)**	NO	NO	NO	NO	NO	NO	NO	NO	NO	YES
RTP video *(6)* **(realistic)**	NO	NO	NO	NO	NO	NO	NO	NO	NO	YES
Works properly?	YES	YES	YES	YES	YES	Only a library *(7)*	NO *(8)*	YES	NO *(9)*	YES

Principle of TCP Tunnels

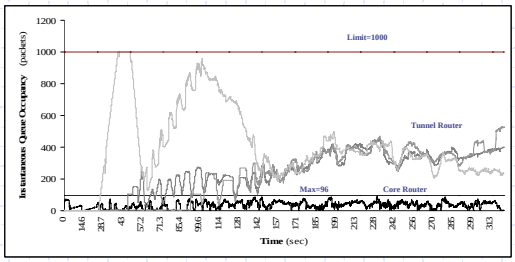
- TCP tunnels are virtual circuits through which all traffic belonging to a class of application are aggregated.
- A single aggregated flow in a tunnel can be managed better than individual flows across a congested link.



Principle of TCP Tunnels (cont)

- Provides statistical guarantees to classes of traffic (eg., voice traffic should get at least 5% of the BW; Video traffic should get at least 10% of the BW, but no more than 20%).
- Guarantees can be assured because of the traffic distribution pattern (eg., all flows do not peak at the same time)
  - Aggregation of traffic
  - Reliable link level protocol
  - Protection of flows
  - Congestion handling is moved from core to edge of the network

Back Pressure Effect of TCP Tunnels



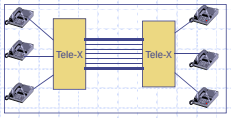
Instantaneous Queue occupancies for core and tunnel routers using TCP tunnels

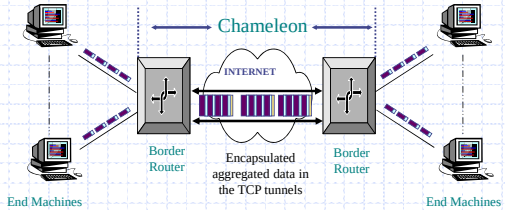
The Chameleon

- A complete solution that can help in the design of Voice over IP Gateways with QoS guarantees.
- Traffic -Modeling of VoIP.
- Research and development at the network protocol layer (TCP/IP etc.).
- Started in co-operation with Harvard University.

Chameleon

Designed to support multiple classes of traffic over the Internet with explicit minimum statistical guarantees.



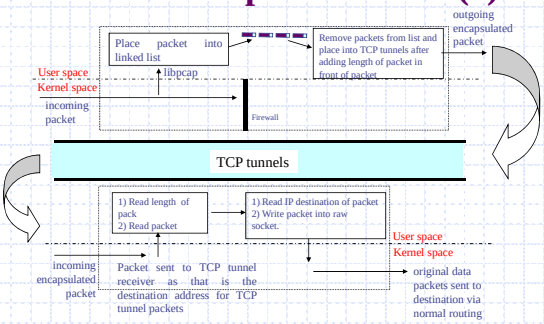


Aggregation of Incoming Data into TCP Tunnels by Chameleon

Chameleon Implementation

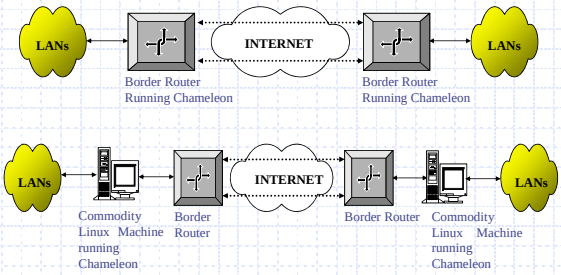
- QoS routines are compiled into the Linux kernel to enable the various QoS providing buffer and scheduling mechanisms (like RED, CBQ etc.) in the Linux kernel.
- User level program, TC used to activate the various buffer and scheduling mechanisms. This requires different parameters to be passed into the program TC.

Chameleon Implementation (2)

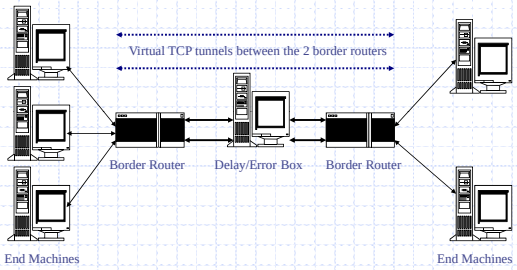


Data Flow Diagram of TCP Tunnel Software

Deployment



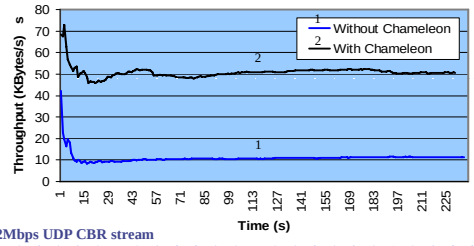
Experimental Setup



Logical setup of Experimental Testbed

Results - TCP Bulk Traffic

Shows effectiveness in protecting TCP bulk traffic

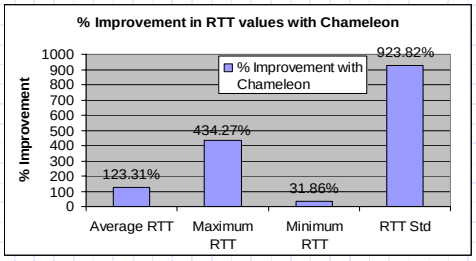


One 2Mbps UDP CBR stream
One 2Mbps UDP exponential on/off stream
Ten TTCP bulk transfer

Results - Web Traffic

Traffic Mix
- 100 web sources
- 2 video streams

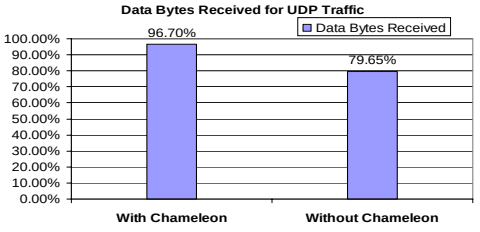
- Jitter is less
- Substantial improvement in End-to-end response time



Results - UDP Packet Loss

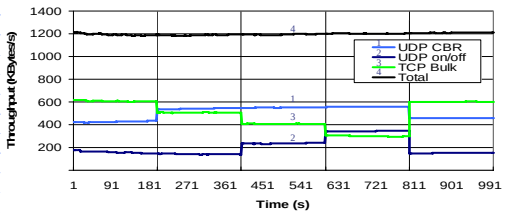
Traffic Mix
- 10 TCP bulk sources
- 2 video stream

- Packet loss is less



Results - Adaptive Nature

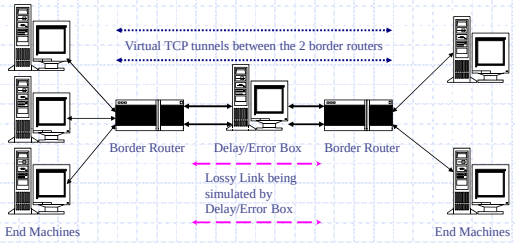
Traffic Mix
- 10 TCP bulk sources
- 2 UDP streams



Reliable Data Link Layer

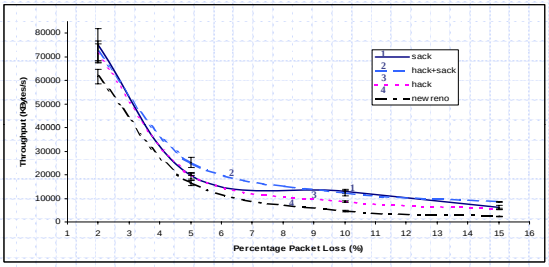
- Lossy / Wireless links are becoming increasingly common in todays networks
- However, TCP performs poorly in cases where packets are lost due to corruption
- Chameleon can be used to “isolate” the lossy links from the rest of the network
- Versions of TCP optimised for lossy environments, like TCP HACK, can be used by the Chameleon

Reliable Data Link Layer (2)



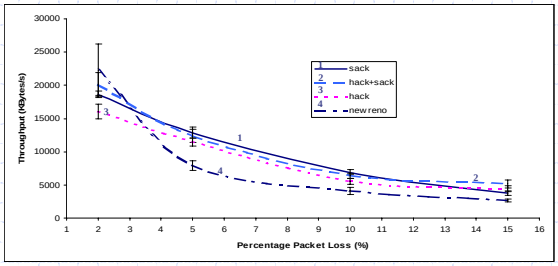
Logical Setup of Experimental Testbed for lossy link experiments

Reliable Data Link Layer - Results



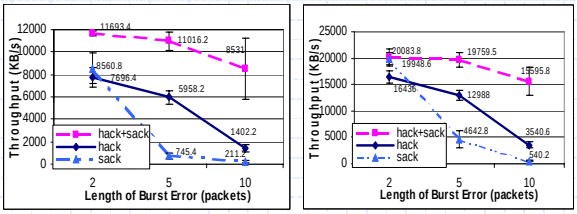
Throughput of Chameleon versus percentage packet loss for short latency (10 ms) link with random single packet errors

Reliable Data Link Layer - Results (2)



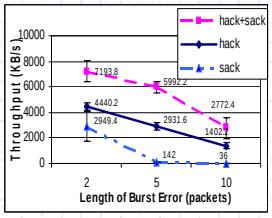
Throughput of Chameleon versus percentage packet loss for long latency (300 ms) link with random single packet errors

Reliable Data Link Layer - Results Burst Errors

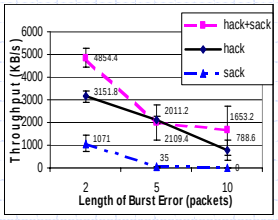


5% burst error **2% burst error**

Reliable Data Link Layer - Results Burst Errors (2)



10% burst error



15% burst error

Future Work

- Port Chameleon fully into kernel space
- Test Chameleon in different network conditions
- Optimise Chameleon for UDP traffic
- Optimise Chameleon as a VoIP gateway
- More work needs to be done on the use of the Chameleon as a reliable data link layer
- Fully automate the Chameleon

Summary

- Chameleon enables the provisioning of QoS for various classes of traffic in an easy yet effective manner
- No modifications to existing protocols and applications and is totally transparent to the end users
- The Chameleon is also adaptive and reacts seamlessly to changes in network bandwidth. It will strive to satisfy all statistical QoS contracts
  - **Applicable** not only to voice, but **other multimedia traffic** as well.
- Expertise in **traffic generation** and analysis:
  - Poisson - telnet; Heavy-tail (Pareto) - web traffic; exponential on-off - voice

Acknowledgements

- A/P A. L. Ananda
- Mr. Lee Boon Peng
- TCP Trunk Project Group
  - Dr. Lillykutty Jacob
  - Dr Winston Seah Khoon Guan
  - Renjish Kumar
  - Liu Yong Xiang
  - Chan Lee Lee
  - Yeo Ann Kian

Questions?

THANK YOU