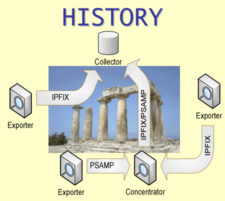
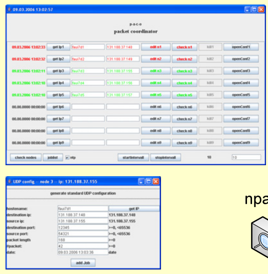


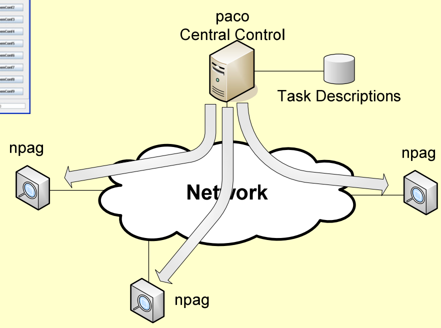
Policy-based traffic generation for IP-based networks

Falko Dressler
Autonomic Networking Group, Dept. of Computer Science 7
University of Erlangen-Nuremberg, Germany
dressler@informatik.uni-erlangen.de

Introduction & Overview

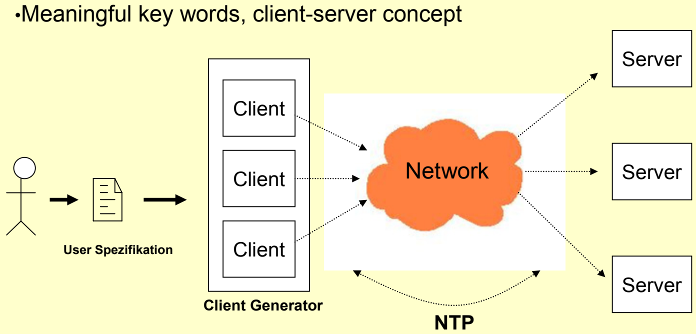
Abstract
Many testbeds and research projects rely on the generation of artificial network traffic. Such solutions are used for protocol or architecture verification, performance tests, and demonstrations. Even though a huge amount of traffic generators have been developed in the past, many of them still suffer from similar problems, e.g. they are either designed for throughput tests or protocol tests. In this paper, we present npag, a new generation traffic generator for IP-based networks. It features a policy-based configuration and a modular design making it feasible to employ it in many application scenarios ahead of simple throughput tests. For example, it allows the generation of packets with arbitrary header and payload information and to measure detailed quality of service parameters.





Policy-based packet generation

Traffic description
- Script language for specification, simple and concise
- Meaningful key words, client-server concept



Central Control	Probe
Probe communication • Availability • Req. for time synchr. • Submission of jobs	MU communication • State maintenance • Time synchr. • Job reception
Job management • Creation • Removal • Inspection • Termination	Job management • Execution • Termination

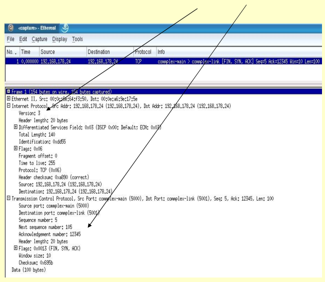
```
stream {
    tcp { # TCP protocol information
        sport = 5000;
        dport = 5001;
    }
    ip {  # IP protocol information
        src = 192.168.178.2;
        dst = 192.168.178.1;
    }
    traffic { # traffic behavior
        burst {
            packets = 100;
            size = 100;
        }
    }
}
```

Generation of TCP traffic

Definition of traffic characteristics

```
stream {
    ip {
        src = 192.168.178.24;
        dst = 192.168.178.24;
        version = 3; # IP version 3
        df = 1; # modify IP
        mf = 1; # header fields
        ttl = 255;
    }
    tcp {
        sport = 5000;
        dport = 5001;
        seq = 5;            # TCP header
        acknum = 12345; # fields
        fin = 1;            # and flags
        syn = 1;
        ack = 1;
    }
    traffic {
        burst {
            packets = 100;
            size = 100;
        }
    }
}
```

Generation of "invalid", i.e. not protocol-conform TCP/IP packets



Related Work & Conclusion

Related Work

Toolkit	Packet definition	Traffic model	Parallel operation	Packet types
ttcp	No	Through-put	No	TCP, UDP
NetPerf	No	Through-put	No	TCP, UDP
NetSpec	Limited	Variable	Yes	TCP, UDP
Iperf	No	Though-put	Yes	TCP, UDP
DBS	Limited	Variable	Yes	TCP, UDP
Harpoon	No	Variable	Yes	TCP, UDP
npag	Yes	Variable	Yes	TCP,UDP, ICMP, user-defined

Capabilities of npag and paco
- Variable traffic rates including bursty behavior
- Long and short term measurements
- Support for TCP/IP-conform and user-defined packet types (IP, ICMP, …)
- Quality of service measurements (loss, delay, …)
- Support for parallel data streams
- Flexible configuration scheme
- Coordination of distributed packet generators

Selected References
[1] P. Barford and M. E. Crovella, "Generating representative workloads for network and server performance evaluation," Proceedings of ACM SIGMETRICS, Madison, WI, June 1998, pp. 151-160.
[2] F. Dressler and G. Carle, "HISTORY - High Speed Network Monitoring and Analysis," Proceedings of 24th IEEE Conference on Computer Communications (IEEE INFOCOM 2005), Miami, FL, USA, March 2005.
[3] J. Sommers, H. Kim, and P. Barford, "Harpoon: A Flow-Level Traffic Generator for Router and Network Tests," ACM SIGMETRICS 2004, New York, NY, USA, Abstract and Poster, June 2004.

Further information: http://www7.informatik.uni-erlangen.de/~dressler/ and http://www.history-project.net/