LBL DSD Group Presentation

pktd: *A Packet Capture and Injection Daemon*

José María González, chema@cs.berkeley.edu

Vern Paxson, vern@icir.org

ICSI Center for Internet Research / EECS Department, U.C. Berkeley

Overview

- **measurement infrastructures**
- **deployment issues**
- ***pktd***
- **control mechanisms**
- **details**

Measurement Infrastructures

- **measurement infrastructure (MI) definition**
  - **compare to single-point measurements**
- **active vs. passive**
- **examples**
  - **NPD**
  - **NIMI**
  - **Surveyor**

Deployment Issues

- **trust**
  - **scalable?**

Deployment Issues

- **trust**
- **client requirements vs. host owner concerns**

Deployment Issues

- **trust**
- **client requirements vs. host owner concerns**
- **mechanisms (granularity)**
  - **user/group/world, rwx mechanism (BPF on BSD)**
  - **coarser (Linux)**

Deployment Issues

- **trust**
- **client requirements vs. host owner concerns**
- **mechanisms (granularity)**
- **administrative hassle**

Deployment Issues

- **trust**
- **client requirements vs. host owner concerns**
- **mechanisms (granularity)**
- **administrative hassle**
- **resource control**
  - **need enforcement point**

Deployment Issues

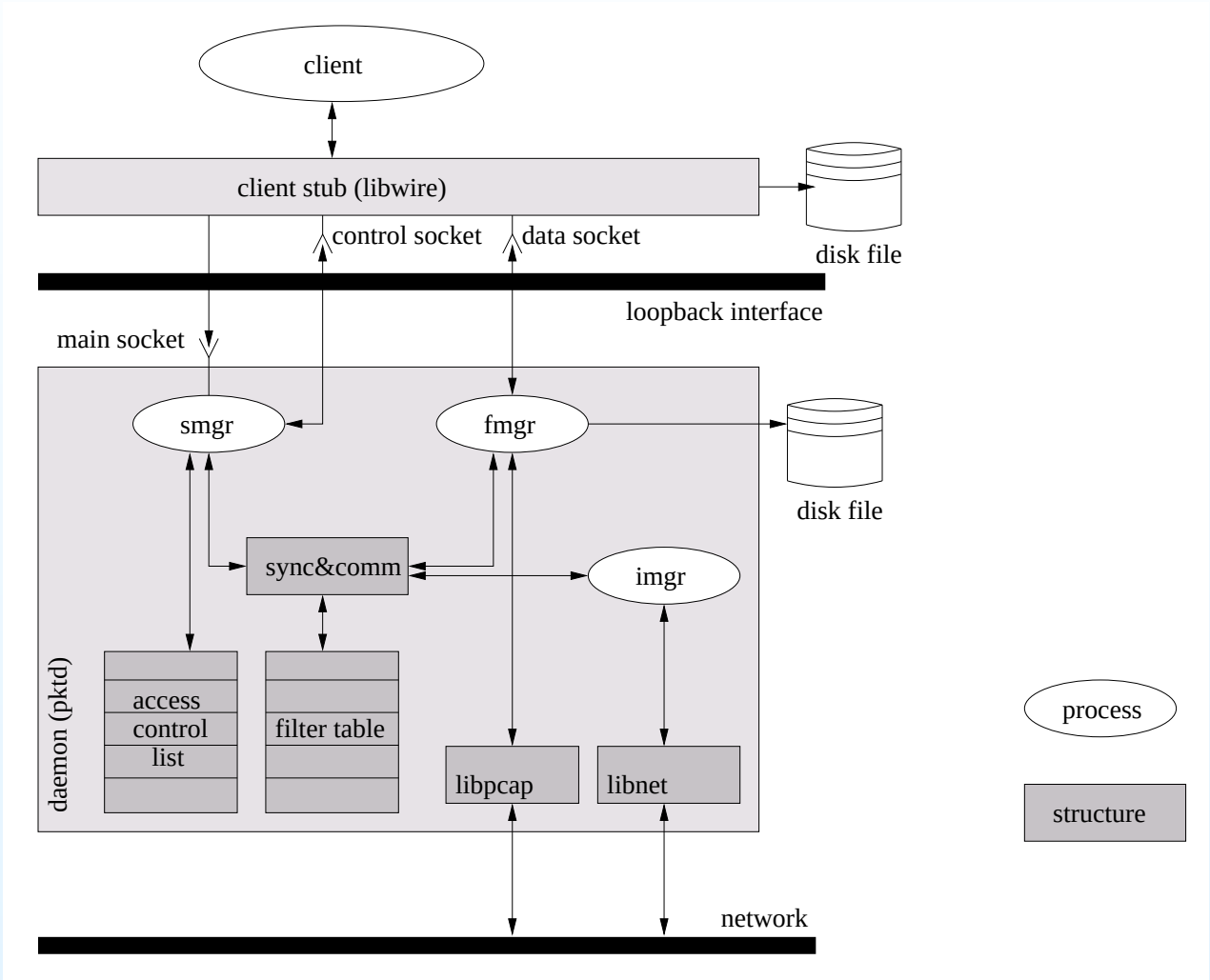
- **trust**
- **client requirements vs. host owner concerns**
- **mechanisms (granularity)**
- **administrative hassle**
- **resource control**

- **provide tools that automate trust**
  - **provides trust to both parties**
  - **increase MI value**

pktd

- **sole trusted, privileged entity**
  - **full NIC access**
- **multiplexes the resource among clients**
  - **clients must request measurements to *pktd***
  - ***pktd* implements mechanisms to grant/deny per-client access**
  - **host owner decides policies**

pktd (cont.)


client
client stub (libwire)
control socket
data socket
disk file
loopback interface
main socket
smgr
fmgr
disk file
sync&comm
imgr
daemon (pktd)
access
control
list
filter table
libpcap
libnet
process
structure
network

pktd (cont.)

API call	Parameters
`wire_init`	filter, snaplen, other
`wire_done`	pdd, ps
`wire_setfilter`	pdd, filter, other
`wire_activity`	pdd, cb, user_data
`wire_inject`	pdd, ip

pktd (cont.)

- **advantages**
  - **only entity host owners need trust**
  - **static**
  - **finer granularity**
  - **more efficient use of resources (packet filter access)**

pktd Control Mechanisms

- **per-client tuning**
- **access type**
  - **capture vs. injection**
- **traffic type**
- **traffic contents**
- **resource control**

Traffic Type Mechanisms

- **which packets can be accessed**
  - **SSH may be OK**
  - **telnet is a no-no**
- **implementation**
  - **use per-client *tcpdump* expressions**
    - `port ssh and not port telnet`
  - **all captured traffic must match the client filter**

Traffic Type Mechanisms (cont.)

- **advantages**
  - **convenient**
  - **easy to implement**
    - `installed expression = client expression AND requested expression`

Traffic Contents Mechanisms

- **which packet headers/fields can be accessed**
- **protocols**
  - **IP/TCP may be OK**
  - **HTTP is more risky**
- **protocol fields**
  - **IP TTL is normally OK**
  - **IP src/dst address poses privacy concerns**

Traffic Contents Mechanisms (cont.)

- **implementation**
  - **per-client snaplen control**
    - **40 bytes to ensure IP/TCP, and no HTTP**
  - **per-client, per-protocol mask**
    - **mask defines accessible fields**
    - **want finer access (trace anonymizing)**

Implementation

- **privileged daemon**
- **12K lines of C code over pcap/libnet**
- **Linux, FreeBSD, Solaris**
- **heavy use (SCNM project)**

Performance Issues

- **compare model: *tcpdump***
  - **1 client over *pktd* = *tcpdump***
  - **full 90-byte headers in 832 Mbps streams**
- **interrupt coalescence**
- **careful buffer management**
  - **standard C stdio is not optimized for small writes**
- **BPF kernel buffer size**

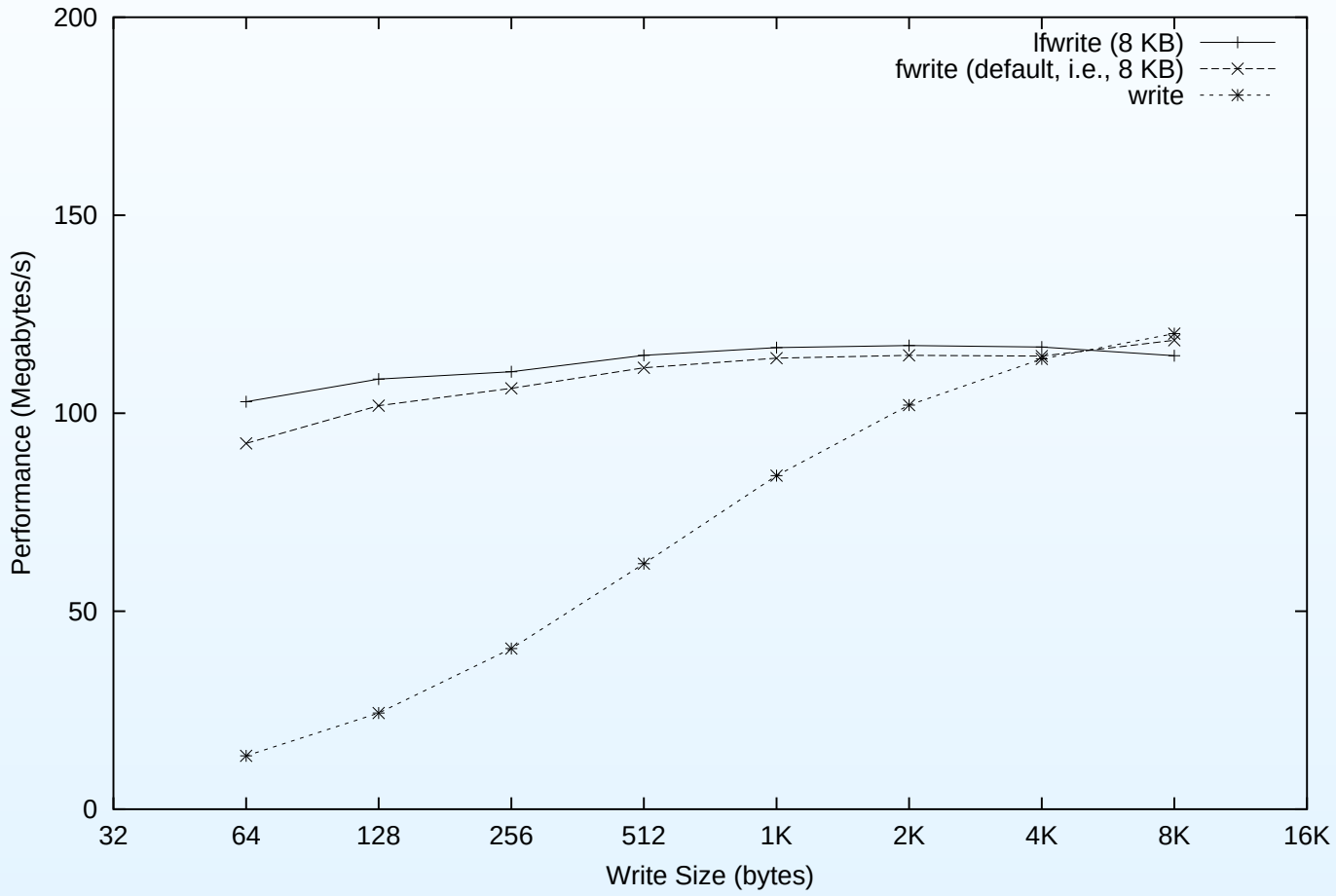
Performance Issues (cont.)

- **compression *à la* CSLIP**
  - **self-contained (no link-layer info)**
  - **need compress timestamps**
  - **cycles more important than space**
  - **may be lossy!**

Summary

- ***pktd* encourages safe participation in MIs**
- ***pktd* offers fine-privilege granularity and permits host-specific policies**
- **clients benefit from a greater MI availability**
- **performance is not degraded**

lstdio Performance



tcpdump vs. *pktd* Performance