Final Presentation

Design and Prototypical Implementation of a User-based IP Accounting Module for Linux

Diploma Thesis Manuel Feier

Presentation Outline

- Introduction and requirements
- Software architecture
- Implementation
- Evaluation
- Conclusion / further work
- (Live demonstration of prototype)
- Questions

IP Accounting?

- Billing network traffic
- Network monitoring
- Administration
- Abuse detection
- Other applications

- Traditionally done per host / device
  - Assumes one user per computer / IP address
  - Usually done on routers / gateways (network layer)

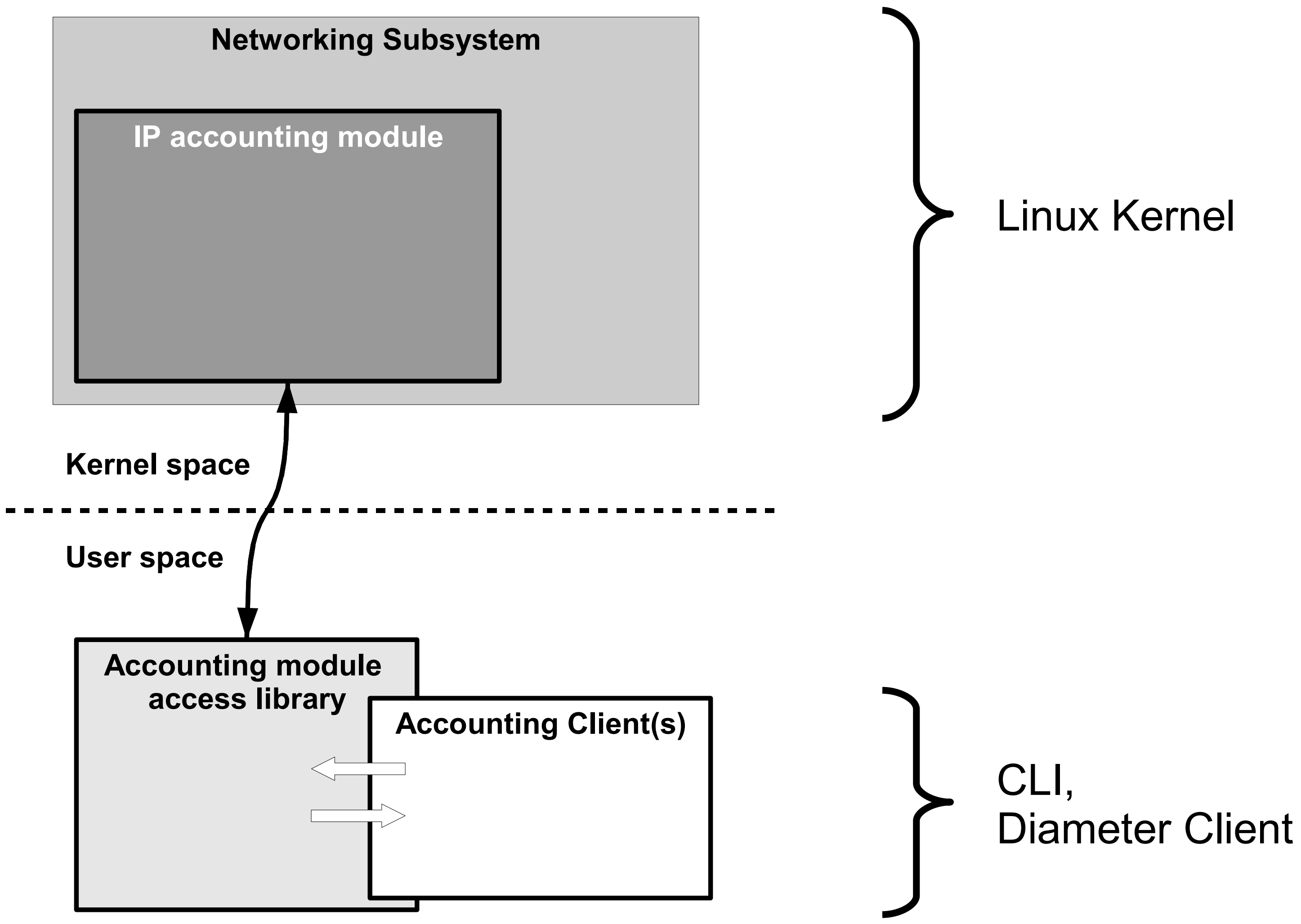
Problem Statement

Existing IP accounting techniques are not suitable for multiuser operating system environments.

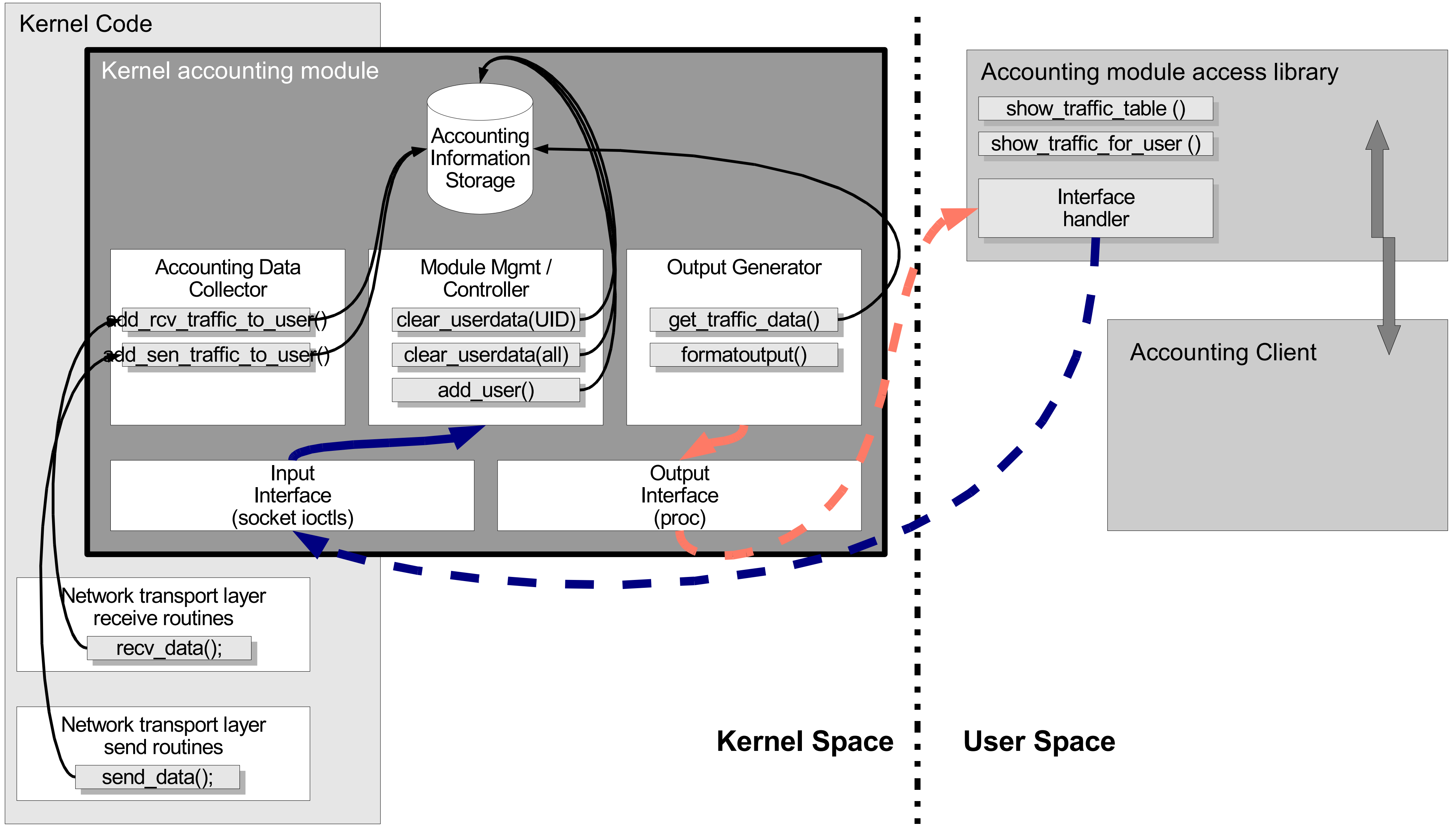
Core Requirements

- Measurement of IP network traffic per operating system user
  - Local system information
  - Support for both IPv4 and IPv6
- Should work on Linux 2.6 (kernel extension)
- Independence of transport layer protocols (e.g. TCP, UDP, ..) and applications
  - Distinction between transport protocols

Software Architecture (Survey)



Software Architecture (Details)



Implementation: Survey

- Implementation in C
- Kernel coding: A different world
  - No standard C libraries
  - Debugging more difficult („kernel panic“)
- Networking-related kernel documentation very scarse
  - „Read the source“
  - „Trial and error“-method

Implementation: Retrieving Data

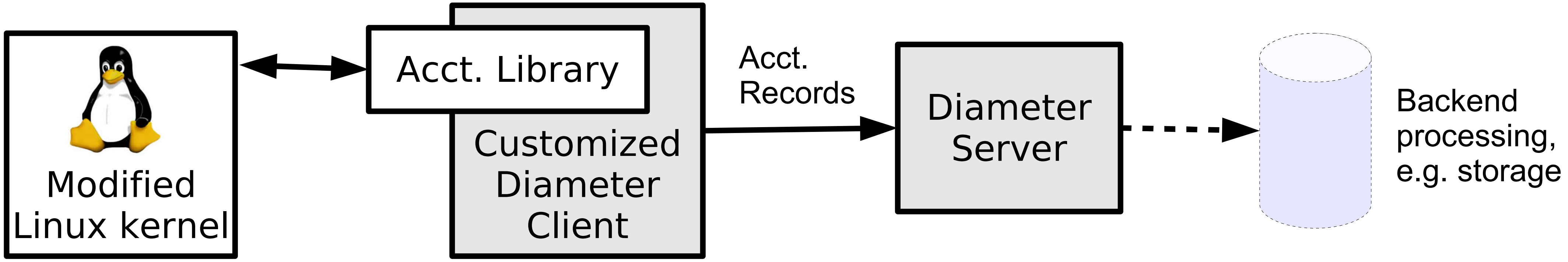
- Localization of suitable send() and receive() routines in the networking subsystem
  - Partially dependent on transport protocols
- Extract information from every packet
  - user (socket owner), size, transport protocol
- Send extracted information to storage component

Implementation: Interfaces

- Export of accounting information via procfs
  - User space applications can read a virtual file with accounting information table
- Control mechanisms via ioctl
  - User space applications can call kernel functions and pass arguments to them
- Integration in a user space acct. library
  - Command line client, Diameter client

Impl.: Integration with Diameter

- Diameter is an AAA protocol (Authentication, Authorization and Accounting)
  - Exchange of predefined data records („AVP's“) between Diameter clients and server
  - Using Linux Kernel accounting module as local data source



Evaluation Survey

- Does the implementation comply with the requirements?
  - Systematic check of requirements
  - Test of IP accounting accuracy
  - Test of IP accounting performance
- Formative evaluation process

Evaluation Results

- The prototype fulfils the requirements
- Accuracy very precise*
- Little impact on TCP/IP throughput

	Throughput (Mbit/s) for IPv4			***Throughput (Mbit/s) for IPv6***		
Test method	*Reference Kernel*	*Extended Kernel*	*Difference %*	*Reference Kernel*	*Extended Kernel*	*Difference %*
Manual	93.880	93.099	0.839	92.661	92.281	0.412
Iperf	94.08	91.7	2.595	92.88	92.87	0.012

Evaluation Discoveries

- Handling of traffic without corresponding network sockets
  - Who is the „owner“ of this traffic?
  - Accounting traffic to a special user „nobody“
- Limitations with ICMP and IPv6 (extension headers)
  - ICMP subsys. scattered around both user- and kernel space, only works in root context
  - IPv6 support requires additional work (due to EH field ambiguity)

Future Work

- Refactoring and optimization of code
- Introduction of facilties to manage individual users (traffic quota, IP white / blacklists, ...)
- Extension from „traffic per user“ to „traffic per process“
  - Application-specific IP access mechanisms
- Framework for integrating local system information with network traffic

Live Presentation

Thanks

- Thanks for your attention!
- Thanks to my supervisor for the support!
- Questions?