# Master Course
# Computer Networks
# IN2097

**Prof. Dr.-Ing. Georg Carle**

**Chair for Network Architectures and Services**
**Department of Computer Science**
**Technische Universität München**
**http://www.net.in.tum.de**

# Welcome Back

- Welcome back after the christmas break!
- What remains to be achieved
  - Lectures
    - 7.1.; 13.1.; 20.1.; 21.1.; 27.1.; 3.2.; 4.2.
  - Complete projects
    - Remember the submission dates
      - 21 Jan: Evaluation
      - 4 Feb: Final Assessment
  - Exercieses
    - The following excercise dates remain 14.1.; 28.1.

# Preparation for the Examination

- Written exam: 13 Feb 2014, 8:30 - 9:30 in MI HS 1
- TUMonline registration will be closed on 15 Jan 2014

- The exam will consist of new questions, in a style that are similar to the questions of the exercises
- On the course page of last year, you find examples (trial exam, endterm exam):

  http://www.net.in.tum.de/de/lehre/ws1213/vorlesungen/master-course-computer-networks/

# Outline

- Network Measurements
- IETF Standardisation Process
- Discussion

# Network Measurements

# Network Measurements

- Introduction
- Architecture & Mechanisms
- Protocols
  - IPFIX (Netflow Accounting)
  - PSAMP (Packet Sampling)
- Scenarios

# Why do we measure the network?

- Network Provider View
  - Manage traffic
    - Predict future, model reality, plan network
    - Avoid bottlenecks in advance
  - Reduce cost
  - Accounting
- Service Provider View
  - Get information about the client
  - Adjust service to demands
  - Reduce load on service
  - Accounting
- Client View
  - Get the best possible service
  - Check the service („Do I get what I have paid for?)
- Researcher View
  - Performance evaluation (e.g., "could our new routing algorithm handle all this real-world traffic?")
- Security view
  - Detect malicious traffic, malicious hosts, malicious networks, …

# But why should we do it at all?

- Do we really have to?
  - The network is well engineered
  - Well documented protocols, mechanisms, …
  - Everything built by humans ⇨ no unknowns (compare this to, e.g., physics: cosmic inflation phase sound? etc.)
  - In theory, we can know everything that is going on
  - ⇨ There should/might be no need for measurements

- But:
  - Information in a distributed multidomain network only partly available
  - Moving target:
    - Requirements change
    - Growth, usage, structure changes
  - Highly interactive system
  - Heterogeneity in all directions
  - The total is more than the sum of its pieces

- And: The network is built, driven and used by humans
  - Detection of errors, misconfigurations, flaws, failures, misuse, …

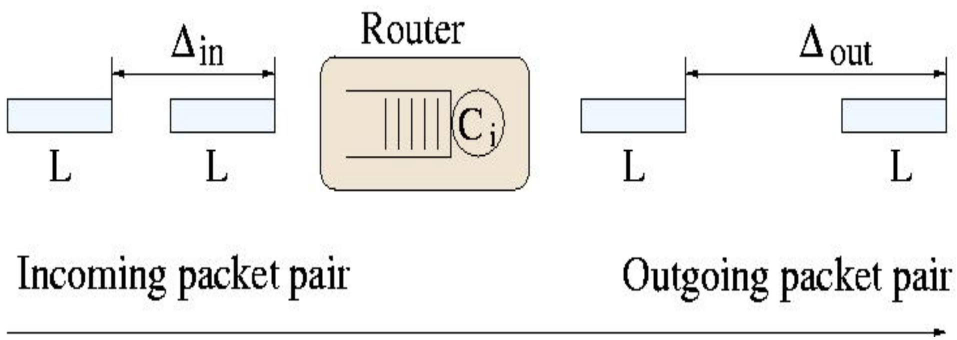
# Network Measurements

- Active measurements
  - "intrusive"
  - Measurement traffic is generated and sent via the operational network.
    (Examples: ping, traceroute)
  - Advantages
    - Straightforward
    - Does not depend on existing traffic by active applications
    - Allows measurement of specific parts of the network
  - Disadvantages
    - Additional load
    - Network traffic is affected by the measurement
    - Measurements are influenced by (possibly varying) network load

# Example: Packet pair probing

- Packet Pair (P-P) technique
  c.f. work by Jacobson & Keshav
- Send two equal-sized packets back-to-back
  - Packet size: L
  - Packet TX time at link i: $L/C_i$
- P-P dispersion = time interval between last bit of two packets
- Without any cross traffic, the dispersion at receiver is determined by bottleneck link (i.e., slowest link):

$$\Delta_{out} = \max\left(\Delta_{in}, \frac{L}{C_i}\right)$$




$$\Delta_R = \max_{i=1,\ldots,H}\left(\frac{L}{C_i}\right) = \frac{L}{C}$$

- C.f paper „“ in Usenix, Kevin Lai, Mary Baker
  Nettimer: A Tool for Measuring Bottleneck Link Bandwidth
  https://www.usenix.org/conference/usits-01/nettimer-tool-measuring-bottleneck-link-bandwidth –
  „If two packets are sent close enough together in time to cause the packets to queue together at the bottleneck link, then the packets will arrive at the destination with the same spacing as when they exited.“

# Network Measurements II

- Passive measurements (or **Network Monitoring**)
  - "non-intrusive"
  - Monitoring of existing traffic
  - Establishing of packet traces at different locations
  - Identification of packets, e.g. using hash values
  - Advantages
    - Does not affect applications
    - Does not modify the network behavior
  - Disadvantages
    - Requires suitable active network traffic
    - Limited to analysis of existing / current network behavior, situations of high load, etc. cannot be simulated/enforced
    - Does not allow the transport of additional information (time stamps, etc.) within measured traffic

# Network Measurements III

- Hybrid measurements
  - Modification of packet flows
    - Piggybacking
    - Header modification
  - Advantages
    - Same as for “passive”
    - additional information can be included (time-stamps, etc.)
  - Disadvantages
    - Modifying of data packets may cause problems if not used carefully

# Measurement types (summary)

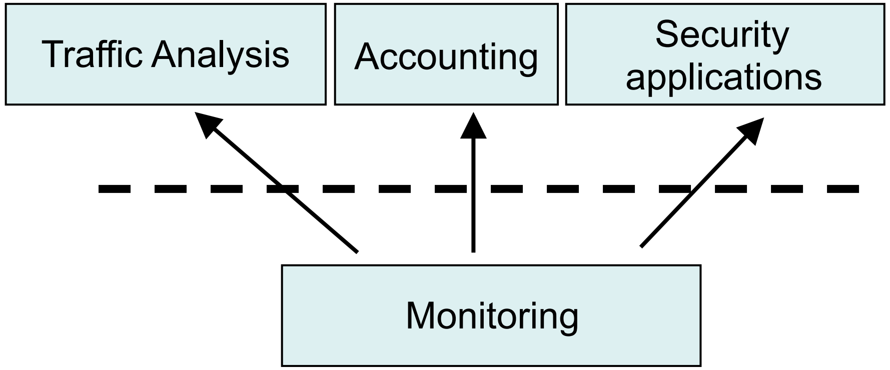
- Active Measurements
  - Intrusive
  - Find out what the network is capable of
  - Changes the network state

- Passive Measurements (or network monitoring)
  - Non-intrusive
  - Find out what the current situation is
  - Does not influence the network state (more or less)

- Hybrid
  - Alter actual traffic
  - Reduce the impact of active measurements
  - Might introduce new bias for applications

# Network Monitoring

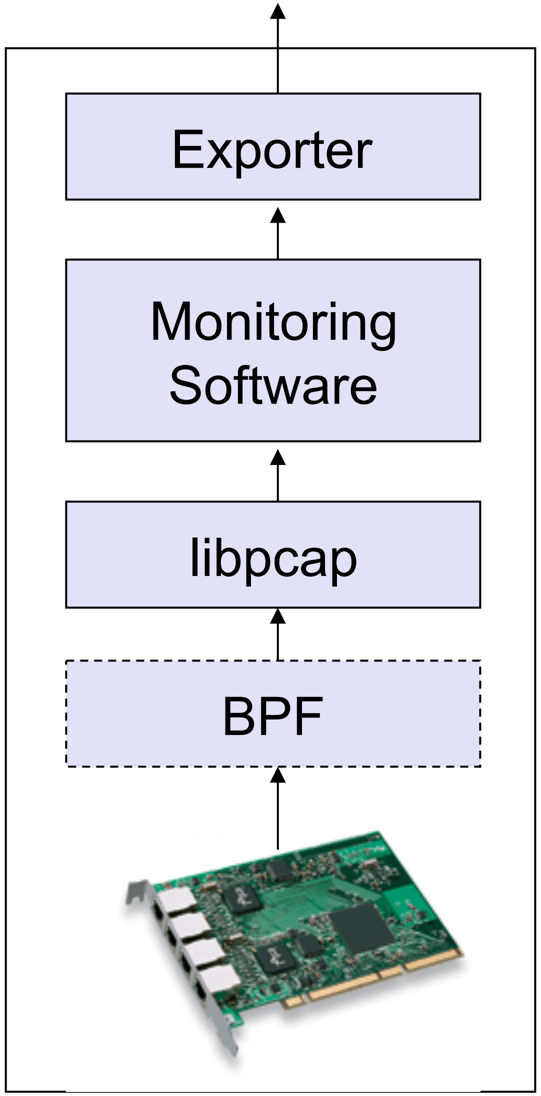
- Applications of network monitoring
  - Traffic analysis
    - Traffic engineering
    - Anomaly detection
  - Accounting
    - Resource utilization
    - Accounting and charging
  - Security
    - Intrusion detection
    - Detection of prohibited data transfers (e.g., P2P applications)
  - Research




- Issues
  - Protection of measurement data against illegitimate use (encryption, …)
  - Applicable law ("lawful interception", privacy laws, …)

# Monitoring Probe

- Standardized data export
- Monitoring Software
- HW adaptation, [filtering]
- OS interface
- Network interface card



# High-Speed Network Monitoring

- Typical requirements
  - Multi-Gigabit/s Links
  - Cheap hardware and software → standard PC
  - Simple deployment

- Problems
  - Several possible bottlenecks in the path from capturing to final analysis

Bottlenecks?

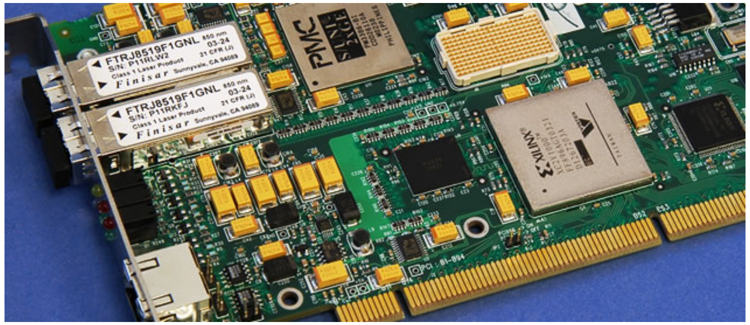
# High-Speed Network Monitoring II

- Approaches
  - High-end (intelligent) network adapters
    - Large amounts of memory
    - Can do filtering, timestamping etc. on their own
  - Sophisticated algorithms/techniques in OS stack for
    - Maintaining packet queues
    - Elimination of packet copy operations
    - Maintaining state (e.g., managing hash tables describing packet flows; sophisticated packet classification algorithms)
  - Sampling
  - Filtering
  - Aggregation

⇨ more on subsequent slides

# Special Network Adapters

- Server NICs (Network Interface Cards)
  - Direct access to main memory (without CPU assistance)
  - Processing of multiple packets in a single block (reduction of copy operations)
    → Reduced interrupt rates



- Monitoring interface cards
  - Dedicated monitoring hardware (usually only RX, no TX)
  - Programmable, i.e. certain processing (filtering, high-precision timestamps, ...) can be performed on the network interface card

# Memory Management I

- Reduction of copy operations
  - Copy operations can be reduced by only transferring references pointing to memory positions holding the packet
  - Management of the memory is complex, garbage collection required
- Aggregation
  - If aggregated results are sufficient, only counters have to be maintained

# Memory Management II

- Hash tables
  - Allow fast access to previously stored information
  - Depending on the requirements, different sections of a packet can be used as input to the hash function
- Multi-dimensional packet classification algorithms (e.g., HiPac)
  - Allow to test large # of complex filtering rules within one lookup operation (e.g., "all TCP packets from network 131.159.14.0/24, but not 131.159.14.0/27, and with source port 80, 443 or 6666–6670, but not with destination address 192.168.69.96–192.168.69.99 → Apply rule 34")
  - Mostly tree-based → Lookups fast, but tree alterations costly

# Packet Sampling

- Goals
  - Reduction of the number of packets to analyze
  - Statistically dropping packets
- Sampling algorithms
  - Systematic sampling
    - Periodic selection of every n-th element of a trace
    - Selection of packets that arrive at pre-defined time intervals
  - Random sampling
    - n-out-of-N
    - Probabilistic
  - "Time machine" sampling: Sample first N bytes of every flow

# Packet Filtering

- Goals
  - Reduction of the number of packets to analyze
  - Possibility to look for particular packet flows in more detail, or to completely ignore other packet flows
- Filter algorithms (explained subsequently)
  - Mask/match filtering
  - Router state filtering
  - Hash-based selection

# Packet Filtering – Algorithms

- Mask/match filtering
  - Based on a given mask and value
  - Simple case: selection range is single packet header value (e.g., mask out least significant 6 bits of source IP address; match against 192.0.2.0)
  - In general: can be a sequence of non-overlapping intervals of the packet
- Router state filtering
  - Selection based on one or more of the following conditions
    - Ingress/egress interface is of a specific value
    - Packet violated ACL of router
    - Failed RPF (Reverse Path Forwarding)
    - Failed RSVP
    - No route found for packet
    - Origin/destination AS equals specific value or list of values

# Packet Filtering – Algorithms II

- Hash-based filtering
  - Hash function h maps the packet content c, or some portion of it, to a range R
  - The packet is selected if h(c) is an element of S, which is a subset of R called the selection range
  - Required statistical properties of the hash function h
    - h must have good mixing properties
      - Small changes in the input cause large changes in the output
      - Any local clump of values of c is spread widely over R by h
      - Distribution of h(c) is fairly uniform even if the distribution of c is not

# Packet Filtering – Algorithms III

- Hash-based filtering (cont.)
  - Usage
    - Random sampling emulation
      – Hash function (normalized) is a pseudorandom variable in the interval [0,1]
    - Consistent packet selection and its application
      – If packets are selected quasi-randomly using identical hash function and identical selection range at different points in the network, and are exported to a collector, the latter can reconstruct the trajectories of the selected packets
      → Technique also known as *trajectory sampling*
      – Applications: network path matrix, detection of routing loops, passive performance measurement, network attack tracing

# IPFIX: IP Flow Information Export

- IPFIX (IP Flow Information eXport) IETF Working Group
  - Standard track protocol based on Cisco Netflow v5…v9
- Goals
  - Collect usage information of individual data flows
  - Accumulate packet and byte counter to reduce the size of the monitored data
- Approach
  - Each flow is represented by its IP 5-tuple (protocol, srcIP, dstIP, srcPort, dstPort)
  - For each arriving packet, the statistic counters of the appropriate flow are modified
  - Whenever a flow is terminated (TCP FIN, TCP RST, timeout), its record is exported
  - Sampling algorithms can reduce the # of flows to be analyzed
- Benefits
  - Allows high-speed operation (standard PC: several Gbps)
  - Flow information can simply be used for accounting purposes, as well as to detect attack signatures (e.g. increasing # of flows / time)

# IPFIX – Work Principles

- Identification of individual traffic flows
  - 5-tuple: Protocol, Source IP, Destination IP, Source Port, Destination-Port
  - Example: TCP, 134.2.11.157, 134.2.11.159, 2711, 22
- Collection of statistics for each traffic flow
  - # bytes
  - # packets
- Periodical statistic export for further analysis

| Flow | Packets | Bytes |
|---|---|---|
| TCP, 134.2.11.157,134.2.11.159, 4711, 22 | 10 | 5888 |
| TCP, 134.2.11.157,134.2.11.159, 4712, 25 | 7899 | 520.202 |

# IPFIX – IP Flow Information Export Protocol

- Quite a number of RFCs
  - Requirements for IP Flow Information Export (RFC 3917)
  - Evaluation of Candidate Protocols for IP Flow Information Export (RFC3955)
  - Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information (RFC 5101)
  - Information Model for IP Flow Information Export (RFC 5102)
  - Bidirectional Flow Export using IP Flow Information Export (IPFIX) (RFC 5103)
  - IPFIX Implementation Guidelines (RFC 5153)

- Transport protocol: Transport of exported IPFIX information records
  - SCTP must be implemented, TCP and UDP may be implemented
  - SCTP should be used
  - TCP may be used
  - UDP may be used (with restrictions – congestion control!)

# IPFIX – Applications

- Usage-based accounting
  - For non-flat-rate services
  - Accounting as input for billing
  - Time or volume based tariffs
  - For future services, accounting per class of service, per time of day, etc.
- Traffic profiling
  - Process of characterizing IP flows by using a model that represents key parameters such as flow duration, volume, time, and burstiness
  - Prerequisite for network planning, network dimensioning, etc.
  - Requires high flexibility of the measurement infrastructure
- Traffic engineering
  - Comprises methods for measurement, modeling, characterization, and control of a network
  - The goal is the optimization of network resource utilization

# IPFIX – Applications II

- Attack/intrusion detection
  - Capturing flow information plays an important role for network security
  - Detection of security violation
    1) Detection of unusual situations or suspicious flows
    2) Flow analysis in order to get information about the attacking flows
- QoS monitoring
  - Useful for passive measurement of quality parameters for IP flows
  - Validation of QoS parameters negotiated in a service level specification
  - Often, correlation of data from multiple observation points is required
  - This required clock synchronization of the involved monitoring probes

# IETF Structure and Internet Standards Process


*Scott Bradner*

*Harvard University*
*http://www.sobco.com/sob/sob.html*


*77th IETF - March 2010*
Anaheim, California, USA

# The IETF - Internet Engineering Task Force

- Formed in 1986
  - evolved out of US government activities
  - ARPA's Internet Configuration Control Board (ICCB) (1979) and Internet Activities Board (1983)
- Was not considered important for a long time - good!!
- Not government approved - great!!
  - but funding support from U.S. Government until 1997
- Specifications always available without charge (vs. ITU-T, IEEE)
- People ***not*** companies

"We reject kings, presidents and voting.

We believe in rough consensus and running code"

Dave Clark (1992)

# IETF Organisation

- 1K to 2K people at 3/year meetings (many more on mail lists)
- >100 working groups with working group chairs
- 8 areas with Area Directors (ADs):
  GEN, APS, INT, O&M, RAI, RTG, SEC, TSV:
  - IETF Chair & AD for General Area (gen) - 0 WGs
  - Applications (app) - 15 WGs
  - Internet (int) - 28 WGs
  - Operations & Management (ops) - 15 WGs
  - Real-time Applications and Infrastructure (rai) - 19 WGs
  - Routing (rtg) - 16 WGs
  - Security (sec) - 17 WGs
  - Transport Services (tsv) - 14 WGs
- Internet Enginnering Steering Group (IESG): ADs + IETF Chair
- Internet Architecture Board (IAB): architectural guidance, liaisons
- IETF produces standards and other documents

# Working Groups

- No defined membership
  - just participants
- "***Rough consensus*** *and running code...*"
  - no formal voting - can not define constituency
    - can do show of hands or hum - but **no** count
  - does **not** require unanimity
  - chair determines if there is consensus
  - disputes resolved by discussion
  - mailing list and face-to-face meetings
  - final decisions must be verified on mailing list
    - to ensure those not present are included
      - but taking into account face-to-face discussion
- Sessions are being streamed & recorded

# IETF Standardisation Procedure

- Proposals published as Internet Drafts (ID)
- Worked on in a Working Group (WG)
- WG sends to IESG request to publish an ID ‘when ready’
- proposal reviewed by AD
  - can be sent back to working group for more work
- IETF Last-Call
- IESG review
  - last call comments + own technical review
  - can be sent back to Working Group for more work
- publication as RFC

# RFC Repository Contains:

- Standards track
  - OSPF, IPv6, IPsec ...
- Obsolete Standards
  - RIPv1
- Requirements
  - Host Requirements
- Policies
  - Classless Inter-Domain Routing
- April fool's day jokes
  - IP on Avian Carriers ...
  - ... updated for QoS
- Poetry
  - 'Twas the night before startup
- White papers
  - On packet switches with infinite storage
- Corporate documentation
  - Ascend multilink protocol (mp+)
- Experimental history
  - Netblt
- Process documents
  - IETF Standards Process

# Standards Track RFCs

- Best Current Practices (BCP)
  - policies or procedures (best way we know how)
- 3-stage standards track (not all that well followed)
  - Proposed Standard (PS)
    - good idea, no known problems
  - Draft Standard (DS)
    - PS + stable
    - multiple interoperable implementations
    - note: interoperability not conformance
  - Internet Standard (STD)
    - DS + wide use
- *"The Internet runs on proposed standards"*
  – perhaps first said by Fred Baker, Cisco Fellow, IETF Chair 1996-2001

# Challenge Interoperability

Example:
IPFIX Interoperability Test Event, 63rd IETF

- Participants
  - CISCO
  - IBM Research Zürich
  - NEC Laboratories Heidelberg
  - Fraunhofer FOKUS, Berlin
  - University team of Prof. Carle
    - c.f. RFC 3333, 5477, 5815
- Lession learned:
  Organisation of interoperability activities is useful. We do not necessarily need to organize joint meetings, but should make more of a habit of organizing joint testing, e.g. combined with chat sessions.