Concept: Traffic Flow

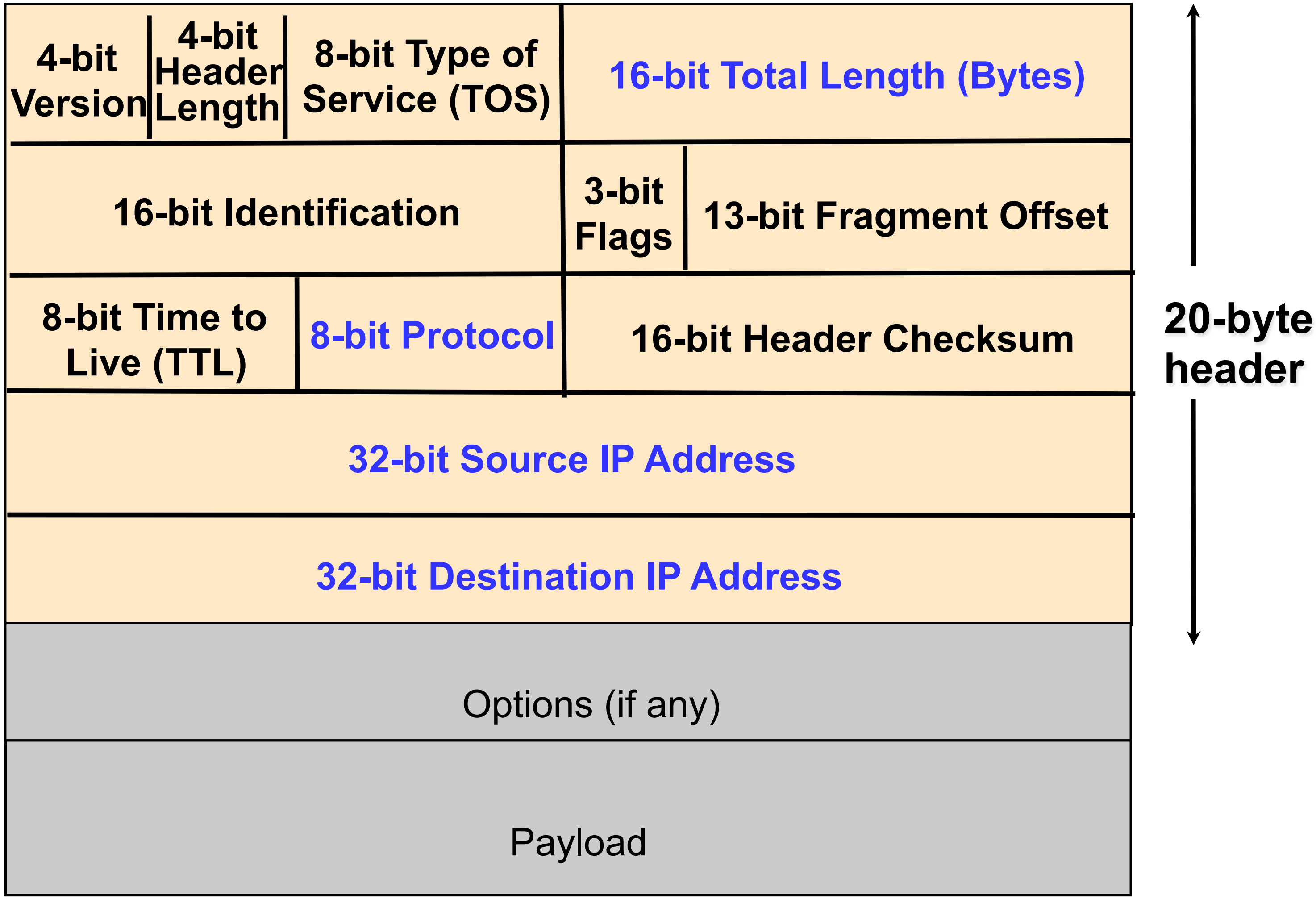
Prof. Anja Feldmann, Ph.D.
Dr. Nikolaos Chatzis

Passive measurement capabilities: Packet monitors

- Available data:
  - All protocol information
  - All content
- Possible analysis:
  - Application performance
  - User behavior
  - Application usage (e.g., P2P)
  - Abuse detection (intrusion detection system)
- Disadvantages:
  - Amount of data
  - Need for data aggregation
  - Needle in a haystack problem
  - Only captures on-network information
  - Usually needs fixed installations

Layer 3: IP

IP Header Format

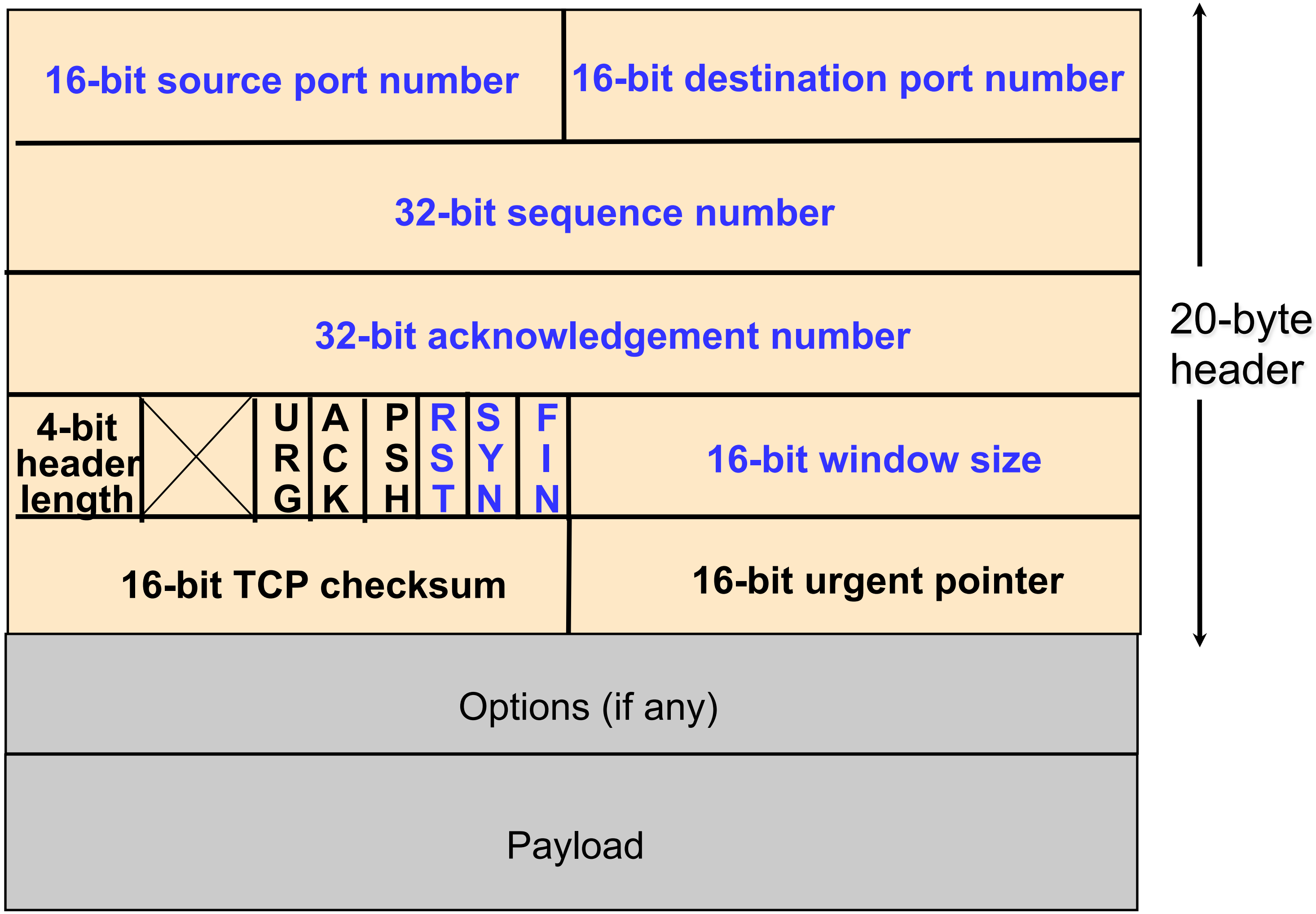


IP header analysis

- Source/destination addresses for traffic
  - Identity of popular Web servers & heavy customers
- Traffic breakdown by protocol (TCP/UDP/ICMP)
  - Amount of traffic not using congestion control
- Distribution of packet delay through the router
  - Identification of typical delays and anomalies
- Distribution of packet sizes
  - Workload models for routers and measurement devices
- Burstiness of the traffic on the link over time
  - Provisioning rules for allocating link capacity
- Throughput between each pair of src-dst addresses
  - Detection and diagnosis of performance problems

Layer 4: TCP

TCP header format



TCP header analysis

- Source and destination port numbers
  - Popular applications (HTTP, P2P, SMTP, DNS)
  - Number of parallel connections between src-dst pairs
- Sequence/ACK numbers and packet timestamps
  - Out-of-order/lost packets; violations of congestion control
  - Estimates of throughput and delay of Web downloads
- Number of packets/bytes per connection
  - Size of typical Web transfers; frequency of bulk transfers
- SYN flags from client machines
  - Unsuccessful connection requests; denial-of-service attacks
- FIN/RST flags from client machines
  - Frequency of Web transfers aborted by clients

Application layer

Packet contents: How much payload?

- Application-layer header
  - HTTP and RTSP request and response headers
  - FTP, NNTP, and SMTP commands and replies
  - DNS queries and responses; OSPF/BGP messages
- Application-layer body
  - HTTP resources (or checksums of the contents)
  - User keystrokes in Telnet/Rlogin sessions
- Security/privacy
  - Significant risk of violating user privacy
  - More sensitive for information from higher-level protocols
  - Traffic analysis thwarted by use of end-to-end encryption

HTTP request and response message

```
GET /tutorial.html HTTP/1.1
Date: Mon, 27 Aug 2001 08:09:01 GMT
From: jrex@research.att.com
User-Agent: Mozilla/4.03
CRLF
```

Request

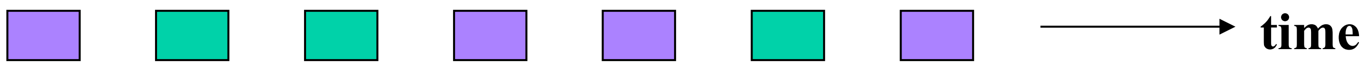
Response

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2001 10:09:03 GMT
Server: Netscape-Enterprise/3.5.1
Last-Modified: Sun, 12 Mar  2000 11:12:23 GMT
Content-Length: 23
CRLF
Traffic measurement talk
```

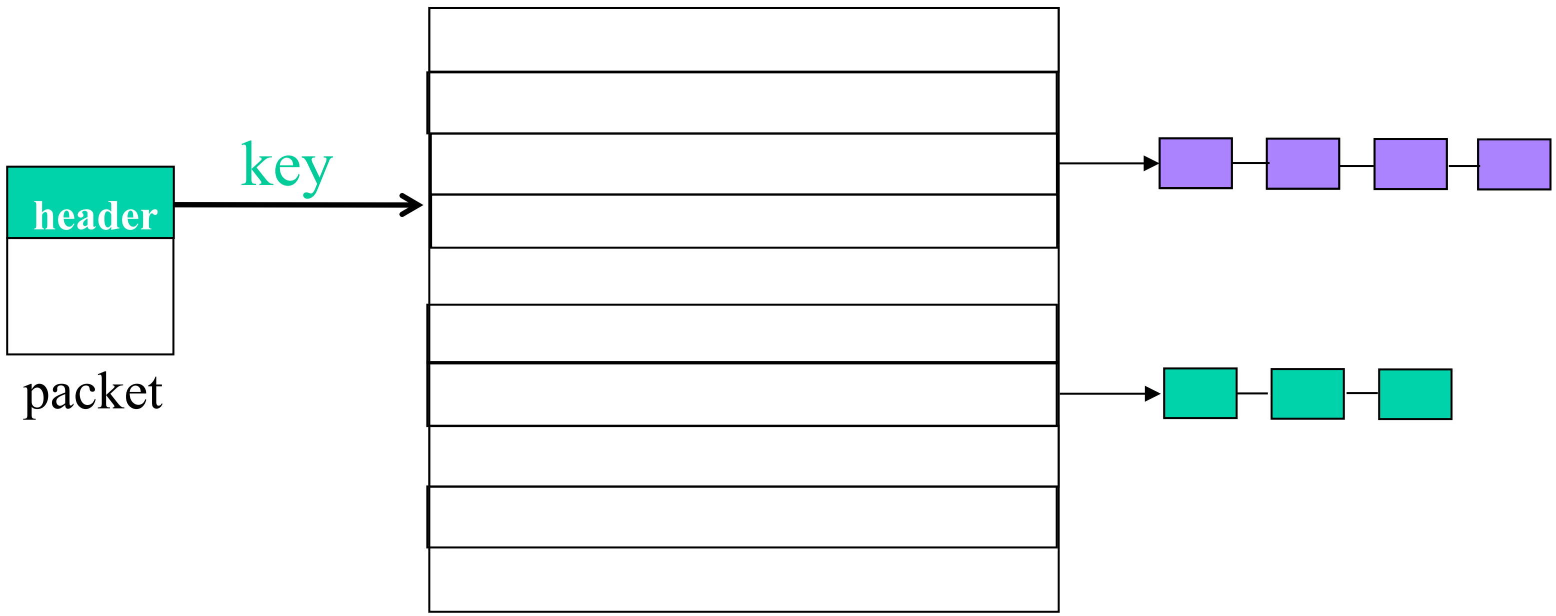
Application-layer analysis

- URLs from HTTP request messages
  - Popular resources/sites; potential benefits of caching
- Meta-data in HTTP request/response messages
  - Content type, cacheability, change frequency, etc.
  - Browsers, protocol versions, protocol features, etc.
- Contents of DNS messages
  - Common queries, frequency of errors, query latency
- Contents of Telnet/Rlogin sessions
  - Intrusion detection (break-ins, stepping stones)
- Routing protocol messages
  - Workload for routers; detection of routing anomalies
  - Tracking the current topology/routes in the backbone

Mechanics: Application-level messages

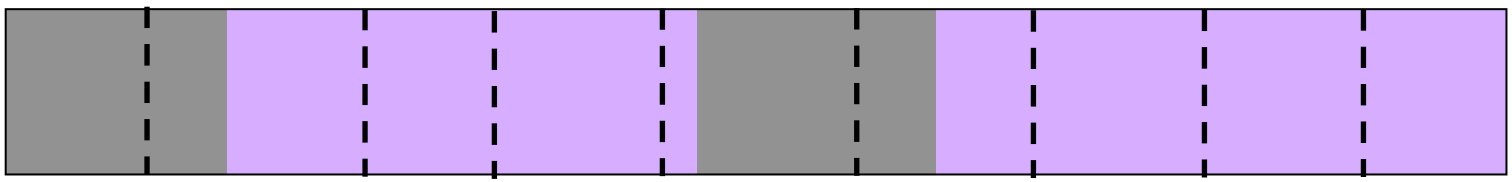


- Application-level transfer may span multiple packets
  - Demultiplex packets into separate "flows"
  - Key of source/dest IP addresses, port, and protocol
  - Hash table to store packets from different flows



Mechanics: Application-level messages

- Reconstructing ordered, reliable byte stream
  - Sequence number and segment length in TCP header
  - Heap to store packets in correct order & discard duplicates
- Extraction of application-level messages
  - Parsing the syntax of the application-level header
  - Identifying the start of the next message (if any)



- Logging or online analysis of message
  - Record URL, header, body, checksum, timestamps, etc.
  - Copy traces or analysis result to separate machine

System constraints

- High data rate
  - Bandwidth limits on CPU, I/O, memory, and disk/tape
  - Could monitor lower-speed links (near the edge of network)
- High data volume
  - Space limitations in main memory and on disk/tape
  - Could do online analysis to sample, filter, & aggregate
- High processing load
  - CPU/memory limits for extracting, counting, & analyzing
  - Could do offline processing for time-consuming analysis
- General solutions to system constraints
  - Sub-select the traffic (addresses/ports, first n bytes)
  - Kernel and interface card support for measurement
  - Efficient/robust software and hardware for the monitor

Passive measurement capabilities: Packet monitors (2.)

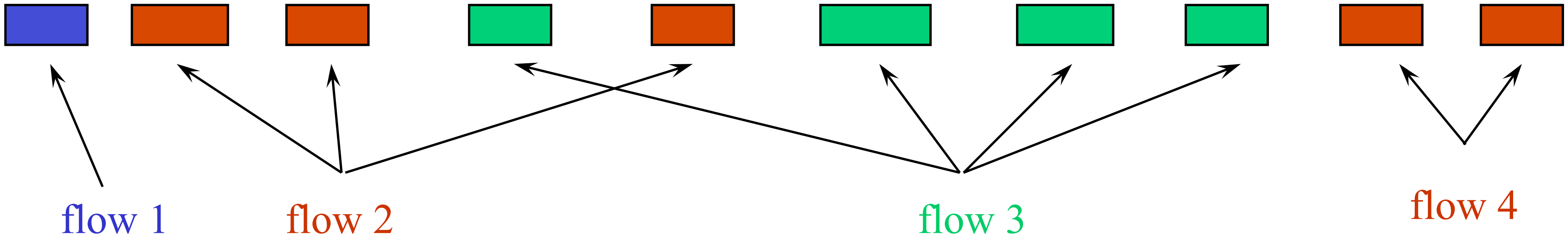
- Deployment scenarios:
  - Needs cooperation of the network operator
  - Limited number
  - Specialized hardware/software
  - Data collection / aggregation infrastructure

- Challenges
  - Data integrity
  - Incomplete data
  - User privacy & network security
  - Data correlation
  - Data privacy vs. data sharing
  - Data filtering
  - Data collection across network confederations

Passive measurement capabilities: Flow statistics

- Available data:
  - Summary information about traffic flows

IP flows: What is it?



- Set of packets that “belong together”
  - Source/destination IP addresses and port numbers
  - Same protocol, ToS bits, …
  - Same input/output interfaces at a router (if known)
- Packets that are “close” together in time
  - Maximum spacing between packets (e.g., 15 sec, 30 sec)
  - Example: Flows 2 and 4 are different flows due to time

Passive measurement capabilities: Flow statistics (2.)

- Available data:
  - Summary information about traffic flows
- Possible analysis:
  - (Application performance)
  - User behavior
  - Application usage (P2P usage)
  - Abuse detection (intrusion detection system)
- Disadvantages:
  - Coarser grain information
  - Data flood
  - Data aggregation
  - Needle in a haystack
  - Only captures on network information (no device info)
  - Usually needs to be configured on network devices

Passive measurement capabilities: Flow statistics (3.)

- Deployment scenarios:
  - Needs cooperation of the network operator
  - Larger number
  - Specialized hardware/software
  - Data collection/aggregation infrastructure
- Challenges
  - Lack of detail
  - Data integrity
  - Incomplete Data
  - Data correlation
  - Data privacy vs. data sharing
  - Data collection across network confederations

Collection of measurement data

- Need to transport measurement data
  - Produced and consumed in different systems
  - Usual scenario: large number of measurement devices, small number of aggregation points (databases)
  - Usually in-band transport of measurement data
    - Low cost & complexity
- Reliable vs. unreliable transport
  - Reliable
    - Better data quality
    - Measurement device needs to maintain state and be addressable
  - Unreliable
    - Additional measurement uncertainty due to lost measurement data
    - Measurement device can "shoot-and-forget"

Controlling measurement overhead

- Measurement overhead
  - In some areas, could measure everything
  - Information processing not the bottleneck
  - Examples: geology, stock market, ...
  - Networking: thinning is crucial!
- Three basic methods to reduce measurement traffic
  - Filtering
  - Aggregation
  - Sampling
  - ... and combinations thereof

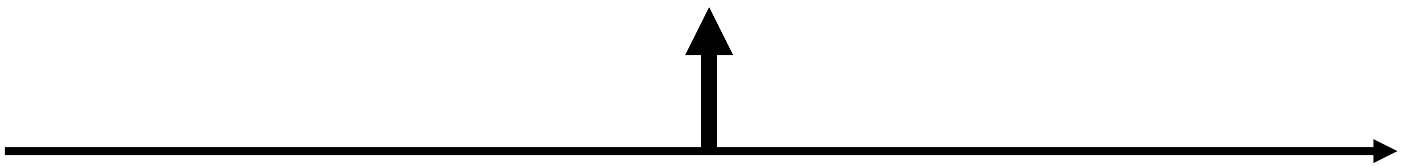
Filtering

- Examples:
  - Only record packets ...
    - matching a destination prefix (to a certain customer)
    - of a certain service class (e.g., expedited forwarding)
    - violating an ACL (access control list)
    - TCP SYN or RST packets (attacks, abandoned http download)

Aggregation

- Example: identify packet flows, i.e., sequence of packets close together in time between source-destination pairs [flow measurement]
  - Independent variable: source-destination
  - Metric of interest: total # pkts, total # bytes, max pkt size
  - Variables aggregated over: everything else

src	dest	# pkts	# bytes
a.b.c.d	m.n.o.p	374	85498
e.f.g.h	q.r.s.t	7	280
i.j.k.l	u.v.w.x	48	3465
....	....	....	

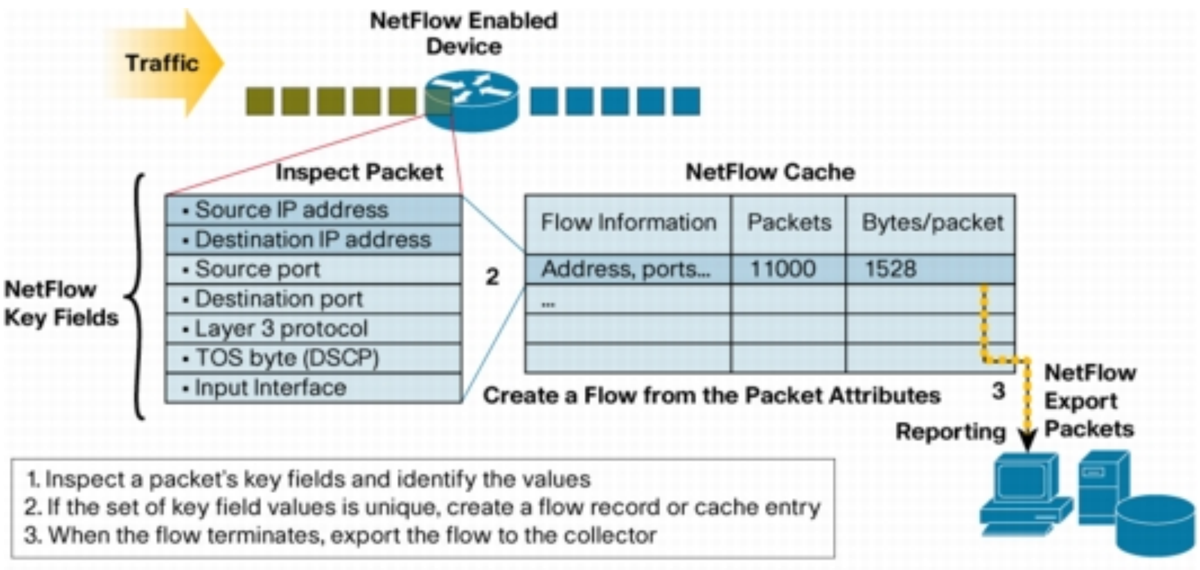


Aggregation (2.)

- Preemption: tradeoff space vs. capacity
  - Fix cache size
  - If a new aggregate (e.g., flow) arrives, preempt an existing aggregate
    - For example, least recently used (LRU)
  - Advantage: smaller cache
  - Disadvantage: more measurement traffic
  - Works well for processes with temporal locality
    - Because often, LRU aggregate will not be accessed in the future anyway → no penalty in preempting

Example: Cisco Netflow

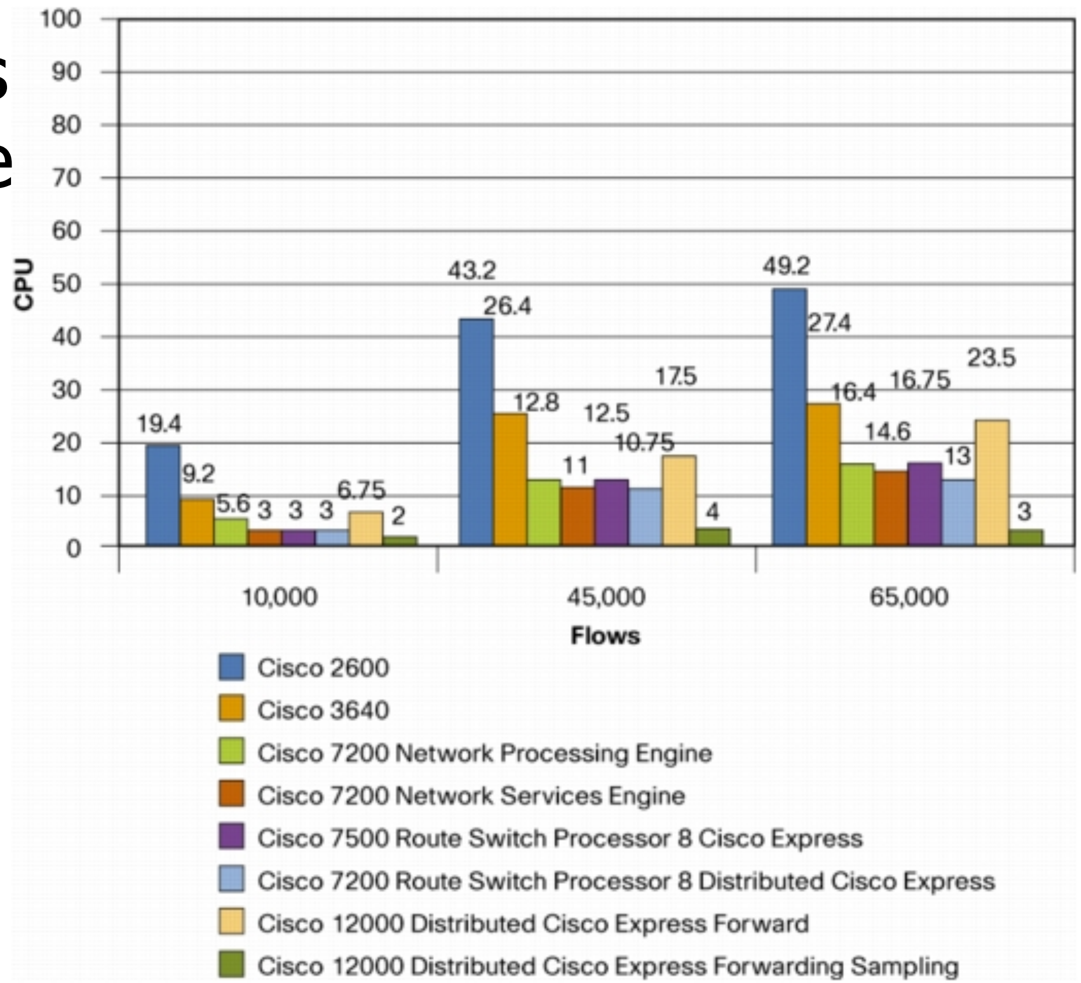
- Traffic monitoring system on switches and routers
  - Cache with 5-tuples: srcIP, srcPort, dstIP, dstPort, proto
  - Upon packet lookup, cache entry is created or updated
  - When cache full, flows are timed-out
  - Timers for flow time-outs



Example: Cisco Netflow (2)

- Impact of # of flows on router CPU usage

- Impact of sampling on average CPU utilization (Cisco 7505):
  - 1/100: -75%
  - 1/1000: -82%



Sampling

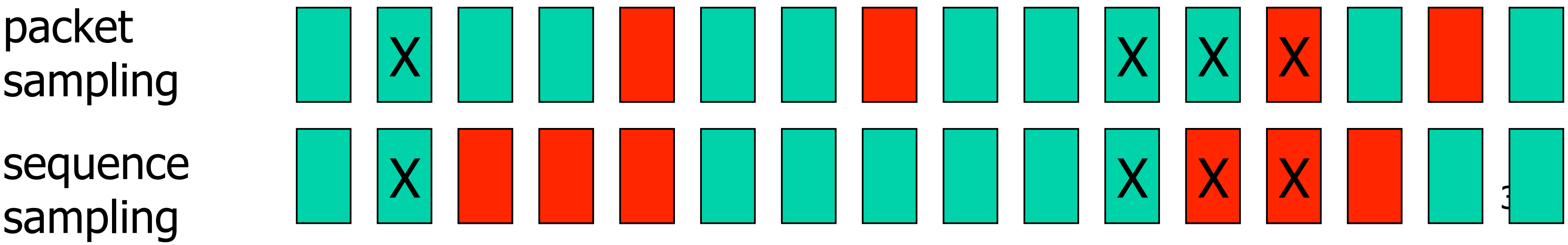
- At high speeds, traffic monitoring requires sampling
- How to sample?
  - Systematic sampling
    - Pick out every 100^{th} packet and record entire packet/record header, e.g., Netflow
    - Ok only if no periodic component in process
  - Random sampling
    - Flip a coin for every packet, sample with prob. 1/100
  - Record a link load every n seconds, e.g., SNMP

Sampling (2.)

- What can we infer from samples?
- Easy:
  - Metrics directly defined over variables of interest, e.g., mean, variance etc.
  - Confidence interval = “error bar”
    - Decreases as $1/\sqrt{n}$
- Hard:
  - Small probabilities:
    “Number of SYN packets sent from A to B”
  - Events such as: “Has X received any packets?”

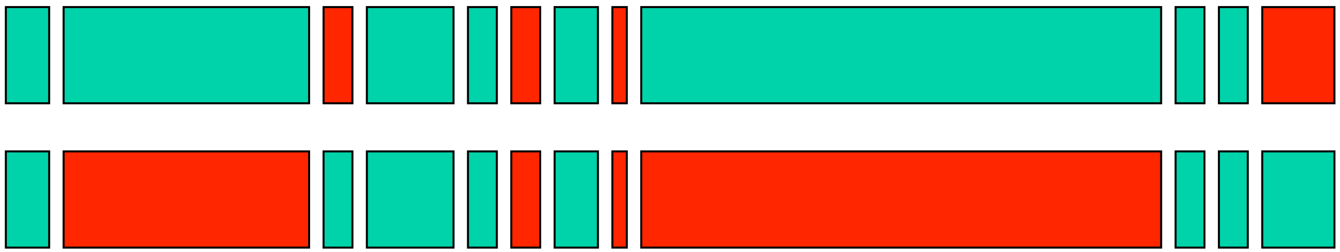
Sampling (3.)

- Hard:
  - Metrics over sequences
  - Example: “How often is a packet from X followed immediately by another packet from X?”
    - Higher-order events: probability of sampling i successive records is p^i
    - Would have to sample different events, e.g., flip coin, then record k packets



Sampling (4.)

- Sampling objects with different weights
- Example:
  - Weight = flow size
  - Estimate average flow size
  - Problem: a small number of large flows can contribute very significantly to the estimator
- Stratified sampling: make sampling probability depend on weight
  - Sample "per byte" rather than "per flow"
  - Try not to miss the "heavy hitters" (heavy-tailed size distribution!)

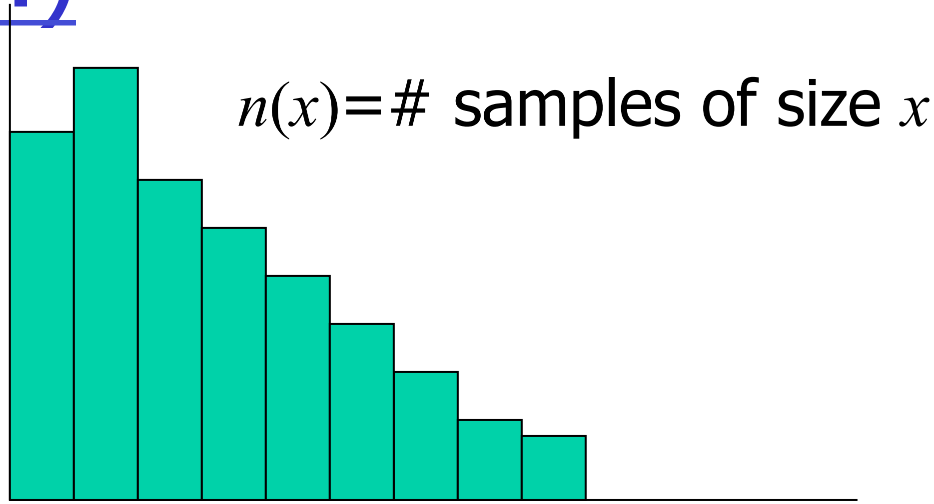


$p(x)$ constant

$p(x)$ increasing

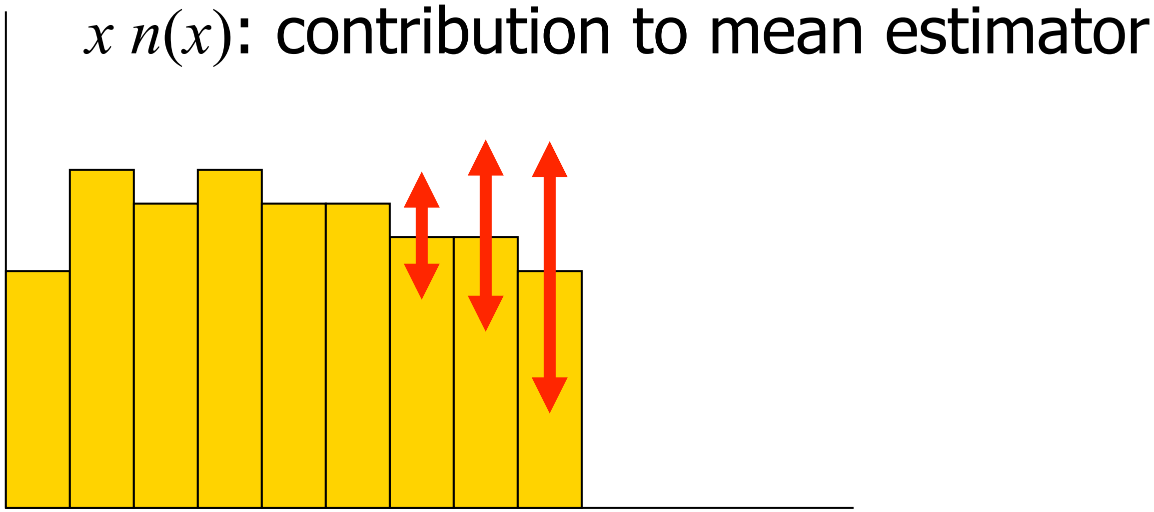
Sampling (5.)

Object size distribution



Estimated mean :

$$\hat{\mu} = \frac{1}{n} \sum_x x \cdot n(x)$$



Variance mainly due to large x

Better estimator: reduce variance by increasing # samples of large objects

Basic Properties

	Filtering	Aggregation	Sampling
Precision	exact	exact	approximate
Generality	constrained a-priori	constrained a-priori	general
Local Processing	filter criterion for every object	table update for every object	only sampling decision
Local memory	none	one bin per value of interest	none
Compression	depends on data	depends on data	controlled

Combinations

- In practice, rich set of combinations of filtering, aggregation, sampling
- Examples:
  - Filter traffic of a particular type (ACLs), then sample packets, e.g., Netflow
  - Sample packets, then filter
  - Aggregate packets between different source-destination pairs (e.g., subnet), sample resulting records
  - When sampling a packet, sample also the next k packets, compute some aggregate metric over these k packets
  - etc.