INTERNET TRAFFIC MEASUREMENT (PART II)

Gaia Maselli
maselli@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

Overview

- Basic concepts
- Characterization of traffic properties that are important to measure
- Challenges in traffic measuring
- Tools available
- The most important results regarding properties of Internet traffic

Tools

- *Packet capture*: The most fundamental tool for traffic analysis (also *passive* traffic measurement)
- *Data management*: online data analysis
- *Data reduction*: to reduce the volume of data requiring processing

Packet capture on general purpose systems

- An end system (within a LAN) can typically capture and record all packets that pass through its network interfaces
- The most common API is `libpcap` which is available on most operating systems
- The `libpcap` allows to specify the interfaces to be monitored and the types of packets to be collected
- An interface can be placed in *promiscuous* mode, meaning that it will report all packets received, not just those destined for the current host
- Packet filters are patterns specifying which packets should be returned to the application, and allow to specify how much of the packet should be captured (only headers or full packet data)
- `Libpcap` delivers raw packet data to the application
- Higher level software is needed for parsing packet header fields and interpreting protocols (public domain tools are `tcpdump` and `ethereal`)
- *Port mirroring* to capture all network traffic in switched Ethernet (copies of Ethernet frames sent to one port on the switch are also sent to the mirroring port)

Packet capture by special purpose systems

- Capturing packets from links in the Internet is more difficult
- Traffic is more representative and diverse
- Specialized hardware is required to the high link datarates
- Links must be tapped to provide physical access and copies of the data on the link (electrical or optical splitter)
- At high link speeds, packet capture with standard interface cards can overwhelm the CPU and bandwidth of a typical capture host
- Specialized network interfaces and interface drivers are generally required
- Non commercial products: OC3MON, OC12MON

Control plane traffic

- It is possible to obtain control packet traffic by directly participating in control plane communication (the exchange of routing messages)
- Obtaining local view of the BGP system
  - A standard PC (listening device) can be configured with software to speak BGP, and receive BGP updates by establishing a session with a BGP-speaking router
  - The BGP router then passes messages as if the listening device were another BGP router
  - The listening device is entirely passive and never generates BGP updates
- Collecting traffic from interior routing protocols in similar manner (OSFP, RIP)

Data management

- Specialized tools allows for management and analysis of immense amount of data
- They use sophisticated algorithms for operating on large streams of data
- Examples: Smacq, Windmill, Nprobe, FLAME, Coral Reef
- Problem similar to data stream management: very large dataset arrive incrementally over time and queries can be continuous or ongoing rather than one-time
- Gigascope: data stream management system specialized for network traffic analysis and used in conjunction with high performance packet capture
- Example

```
select tb, srcIP, sum(len)
from IPv4
where protocol = 6
group by time/60 as tb, srcIP
having count(*)>5;
```

Data reduction

- Reduce the *volume* of data requiring processing
- Several methods:
  - Counters
  - Flow capture
  - Sampling
  - Summarization
  - Probabilistic models

Data reduction: counters

- The most common form of traffic summarization is to use aggregation to form *time series of counts of traffic statistics* (bytes or packets per unit of time)
- This is supported in all current routers via MIB-II Management Information Base, accessed via SNMP
- Disadvantage: almost all traffic semantics are absent

Data reduction: flow capture

- To capture and store packet trains or flows
- Flow capture preserves information important to ISPs and traffic analysts by capturing and storing packet trains or flows
- ***Packet trains*** traces can be used for monitoring basic network activity, monitoring users and applications, network planning, security analysis, and accounting and billing
- A packet train record consists of the IP-header 5-tuple (source IP addr, source TCP or UDP port, destination IP addr, destination port, protocol id); start and end time, number of packets, number of bytes contained in the packet train.
- Cisco `NetFlow` is typically about 1.5% in volume compared to the actual traffic being summarized
- The end of the packet train is defined by
  - Timeout threshold applied to interpacket gaps
  - FIN / RST packets

Data reduction: flow capture (Cont)

- **Packet flows**: in some settings it is necessary to know only the number of bytes or packets that carry a particular 5-tuple value in a given interval
- Although most routers and switches can export some sort of flow records, higher–level software is generally needed for processing and interpreting the raw data
- `Flowtools` is a set of tools that can collect, send, process and generate reports from Cisco `NetFlow` or Juniper `cflowd` data
- IPFIX

Sampling

- A *subset of packets* are chosen for capture
- A sampling rate of 1% reduces the traffic size by a factor of 100

Questions:

- How should packets be chosen for sampling?
- how should one compensate for the sampling process when performing analyses?

Basic packet sampling: the sampling process is performed independently on each link being monitored

- Constant rate sampling
  - **Random** sampling: packets are chosen with some fixed probability
  - **Deterministic** sampling: packets are chosen periodically i.e., every *N*th packet is sampled
  - **Stratified** sampling: packets are divided into subsets and then sampling is applied within each subset (Ex. Sampling each packet train with fixed probability)

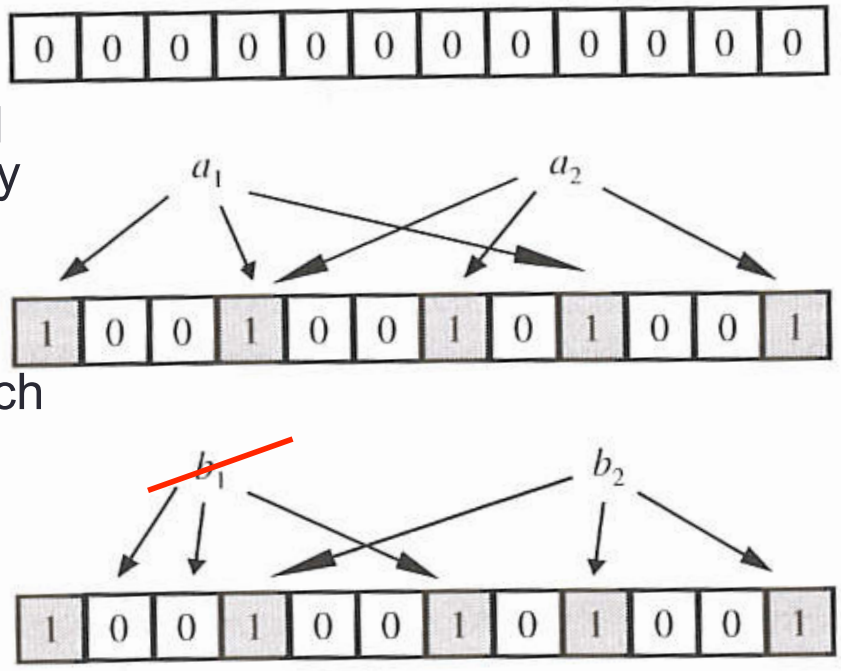
Trajectory sampling

- Basic sampling does not provide per packet delays, and flow track of traffic through the network
- Trajectory sampling: if a packet is chosen for sampling at any point in the network, it is chosen at all points in the network.
- Method: traffic measurement devices calculate a hash function for each packet based on the value of certain header fields (those that do not change as packets move through the network and are sufficient to identify the packet uniquely)
- The measurement device selects a packet for reporting if the hash value falls in a specified range
- By varying the range size the rate of packet sampling can be varied
- Metrics on customer performance (per-customer packet delays), detect routing loops and trace denial of service attacks

Summarization

- To form compact summaries that are useful for answering particular questions
- **Bloom Filters** is a summary of set that supports membership queries

- Given a set of keys $A=\{a_1,a_2,\ldots,a_n\}$ a Bloom filter provides a data structure and algorithm allowing one to ask whether key a_n is present in the set A.
- The data structure is a vector of m bits
- The algorithms used are based on k independent hash function $h_1,h_2,\ldots,h_k$ each having range $\{1,\ldots,m\}$
- The hash functions are chosen so as to map each item in A to a random number uniform over the range $\{1,\ldots,m\}$
- To check if an item b is A, check whether all bits specified by $h_i(b)$ are set to 1



Summarization (Cont)

- The Bloom filter may yield *false positive* with some probability that depends on *k*, *m*, and *n*
- After all elements of A are recorded in the Bloom filter, the probability that a randomly chosen bit is still 0 is

$$(1-1/m)^{kn}$$

- So the probability that k randomly chosen bits are set to 1 is

$$(1-(1-1/m)^{kn})^{k}$$

- This expression can be interpreted as the probability of a false positive (the probability that a key selected uniformly at random from the set of all keys not in A will in fact be assumed to be in A)

Summarization (Cont)

- Bloom filters allow to achieve a constant compression ratio regardless of the size of dataset being represented

Example: we want to construct a database of IP source/ destination pairs encountered in a traffic trace

- Each pair consists of 64 bits.
- If we employ a Bloom filter with false positive rate of 1% we should be able to store such a set of address pairs using only 9.58 bits per element

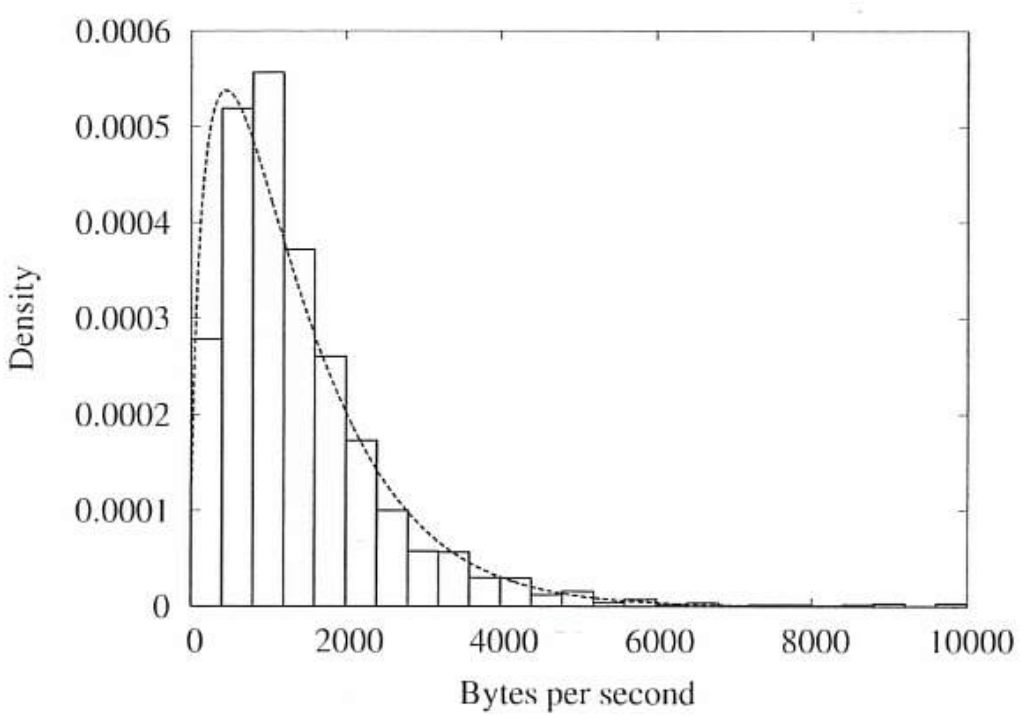
N.B. Bloom filters are not suited to storing sets whose members may depart over time (not possible to delete a key)

- To address the problem: *counting bloom filters* (each entry is a small counter rather than a single bit)

Probabilistic models

- Summarize the results in the form of probabilistic models
- A probabilistic model treats observations as if they were generated by a random process
- Features in a traffic trace may be summarized by providing a probability distribution function that matches the empirical distribution found (*analytic* model)

- Example: to fit a distribution to the time series of counts of bytes inter arrivals in a traffic trace
- A Gamma distribution matches the data
- The mass of information present in the traffic trace has been reduced to the small set of parameters needed to specify a particular instance of the Gamma distribution



Probabilistic models (Cont)

The benefits of using an analytic model include

1. Such models can be manipulated mathematically, leading to improved understanding (ex. Given an probabilistic model it is often possible to analytically compute its mean, variance, or other summary statistics)
2. Analytic models are concise and easily communicated (in most cases only the distribution type and two o three parameters suffice for a complete specification)
3. The particular values of a model's parameters can give insight into the nature of the underlying data. As model parameters vary, the nature of the underlying distribution varies in a predictable way

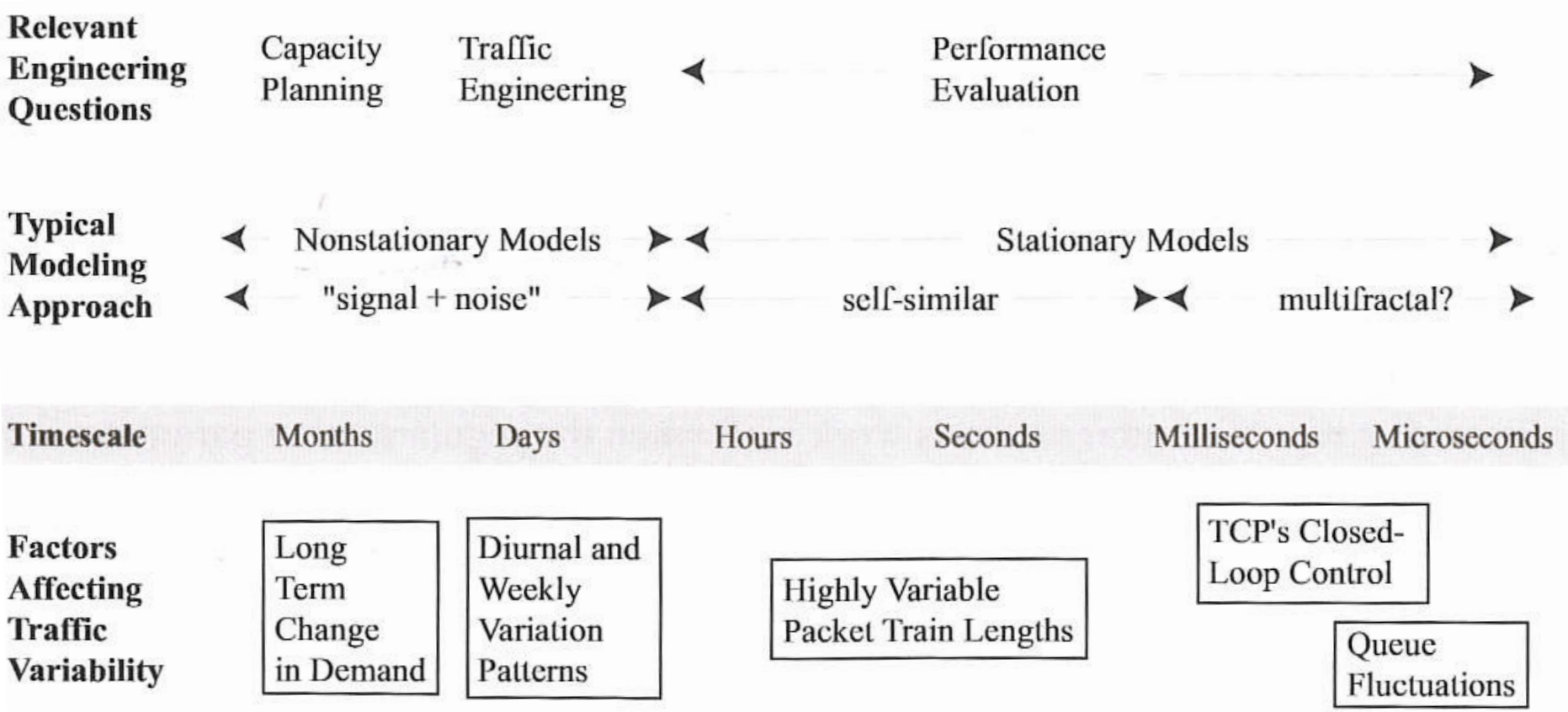
Probabilistic modeling involves two steps:

1. The choice of distribution function
2. The estimation of the relevant parameter values

- A general approach is to choose a collection of distribution functions, estimate appropriate parameter values for each and then chosse the resulting distribution that fits best

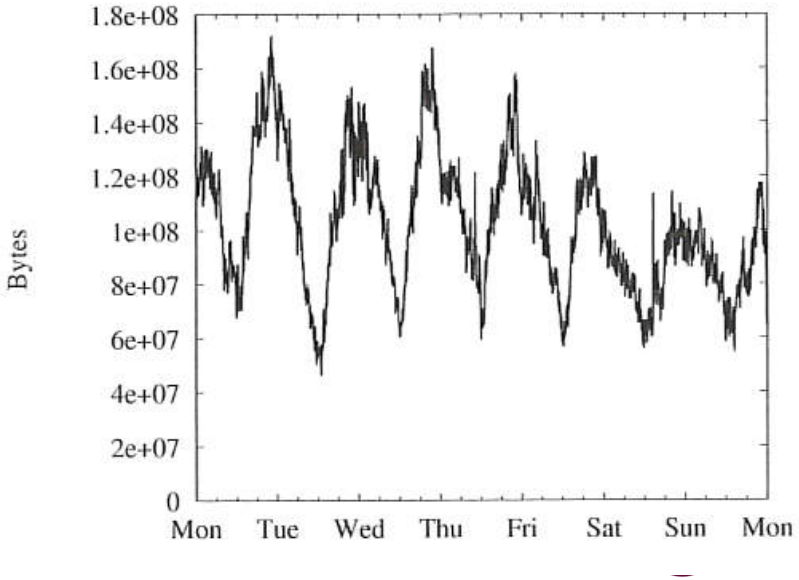
State of the art
(what is known about the various features of Internet traffic)

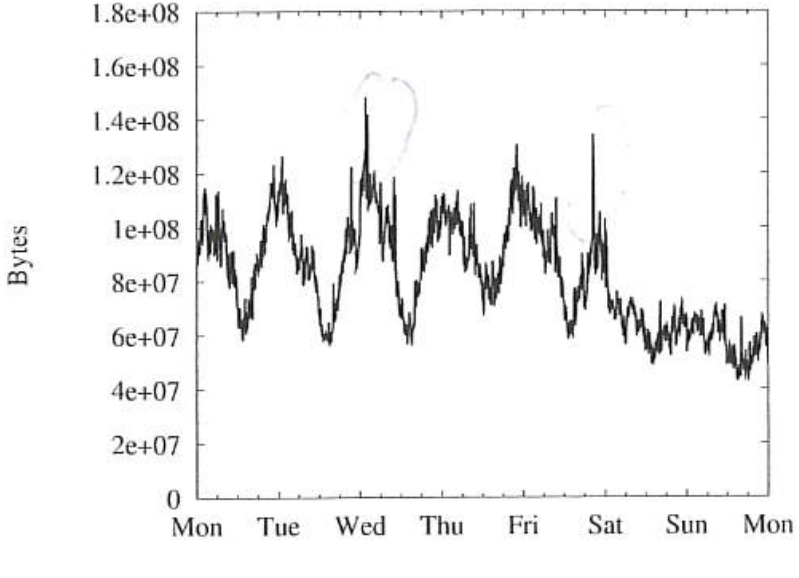
Overview for traffic analysis



Packets and bytes: long timescales (trends)

- Timescale: longer than one hour
- Non stationary
- There are strong predictable variations present in traffic (trends)
- **Signal and noise**: long-timescale network traffic may be thought of as possessing a *predictable* component and a *stochastic* component ("signal and noise")
- The predictable component can be modeled as a time varying mean, while stochastic component shows Gaussian-like variability except for spikes





Packets and bytes: long timescale (trends)

- Growth of traffic
- Internet traffic **growth** has been **exponential**
- ARPANET
- 1980-1990
- Mid-1990 the fastest growth
- 1996-2000 it was doubling approximately annually

Exponential growth cannot continue indefinitely. For this reason understanding the growth of traffic on the Internet is an important ongoing question

Packets and bytes: short timescales (scaling)

- At shorter timescales, one can use stationary models to describe network traffic
- There is *no predictable variation* and **stochastic models are appropriate**
- However stochastic variation in Internet traffic generally shows some rather unusual characteristics, that can be described in terms of *scaling behavior* (the manner in which statistical properties vary as one observes traffic at different timescales)
- Given some time series of traffic counts $\{B_n, n=1,2,\ldots\}$ at a particular timescale T, one can form a rescaled view of the traffic $\{B_n^{(m)}, n=1,2,\ldots\}$ by defining

$$B_n^{(m)} = \sum_{i=nm}^{nm+m-1} B_i$$

For some positive integer m

- $\{B_n^{(m)}\}$ is just $\{B_n\}$ summed over non-overlapping blocks of size m, and so rescaling a traffic time series by m is equivalent to having observed the original traffic at the timescale T_m

Packets and bytes: very short timescales

- Timescale: hundreds of milliseconds (approximately the range of RTTs in many parts of the Internet)
- Packets arrival patterns are affected by
  - TCP
  - Time variation of queue lengths in routers
- Traffic is very **bursty** (*multifractal process* that shows different scaling behaviors at different instants of time)

Packets and bytes: other traffic measures

- Apart from byte and packet counts, an important metric of traffic is **packet size distribution**
- Important for router designers
- Unusual packet size distributions can signal the presence of anomalous traffic
- Packet size distributions are typically trimodal, clustered around a few particular sizes:
  - 40-44 bytes (TCP ack, control packets, TELNET packets)
  - 552-576 bytes (default maximum segment size for nonlocal destinations for host that do not use Path MTU Discovery)
  - 1500 bytes (Ethernet-connected hosts using Path MTU Discovery)

Higher-level structure: packet trains

- **Packet train duration** and **volume** show heavy tails: most packet trains are small, while most bytes are contained in long packet trains
- Elephant and mice phenomenon: a few ‘elephant’ packet trains mixed with many ‘mice’
- OFF times also show heavy-tailed distributions
- TCP connection interarrivals are highly variable and have been modeled using Weibull distributions
- **Packet train rate** (the speed at which bytes flow) is influenced by the transport protocol
- It has been shown that when packet loss is low (less than 2%) steady state throughput can be approximated by

$$R = \frac{MSS}{RTT \cdot \sqrt{p}}$$

where R is throughput in bytes per seconds, MSS is maximum segment size of a TCP packet and RTT is the round trip time of the connection, and p is the packet loss rate experienced by the connection

Higher-level structure: sessions

- The arrival process of sessions is generally well described as Poisson (at the highest level of network traffic structure an independence assumption seems valid)

Control traffic

- Experience with BGP showed that the failure of a BGP router can cause a “storm” of instability
- The volume of routing updates in BGP was found to be much higher than expected (typically by a factor of 10) and the majority of routing updates exchanged via BGP were redundant
- BGP message arrival is extremely bursty, it shows strong periodicities and autocorrelations

Wireless

- An emerging category of Internet traffic is wireless – packets that flow for some part of their path over a wireless link (wireless access points)
- As wired traffic, wireless traffic is **bursty** and shows daily variations
- Many measures of user activity show **high variability** (session durations)
- Results are consistent with the use of long-tailed distributions for wired networks
- One difference with wired traffic is spatial distribution: users tend to have irregular distribution in space and the load on access points tends to be highly uneven