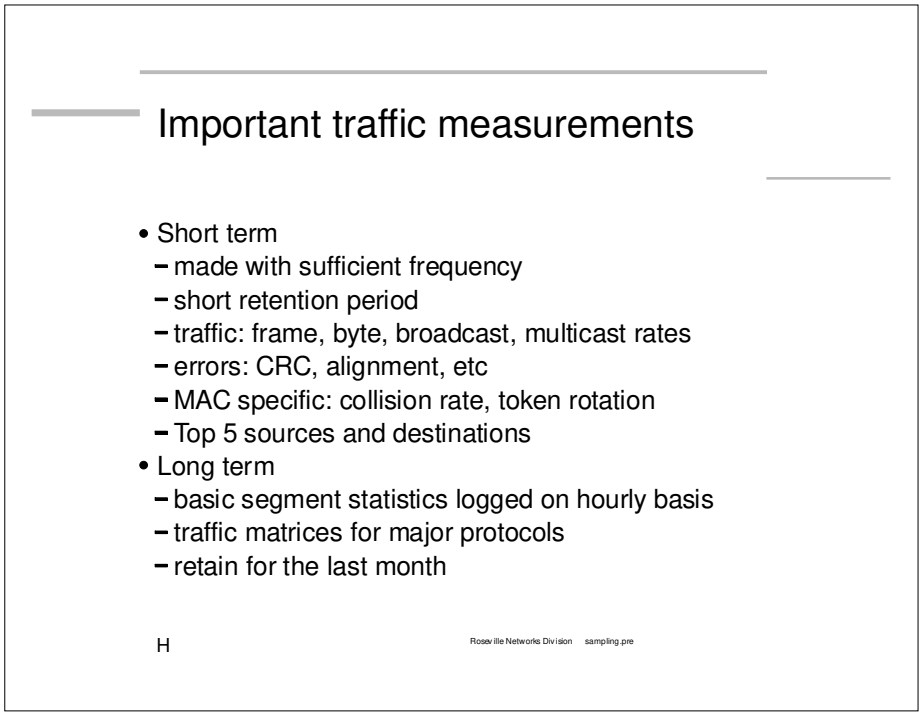
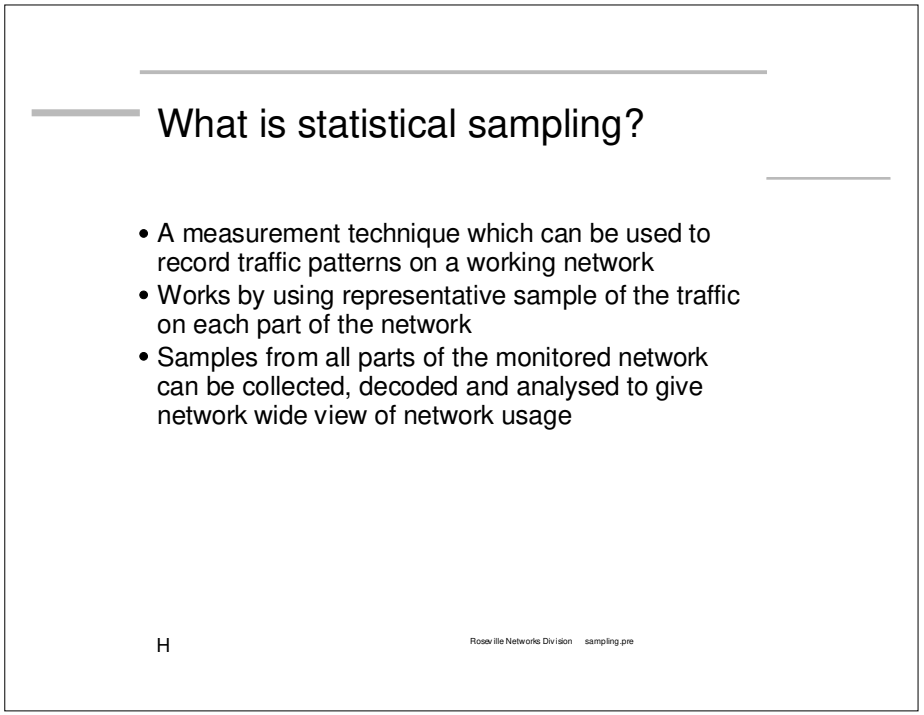
Statistical Sampling for Network Traffic Monitoring

Sonia Panchen

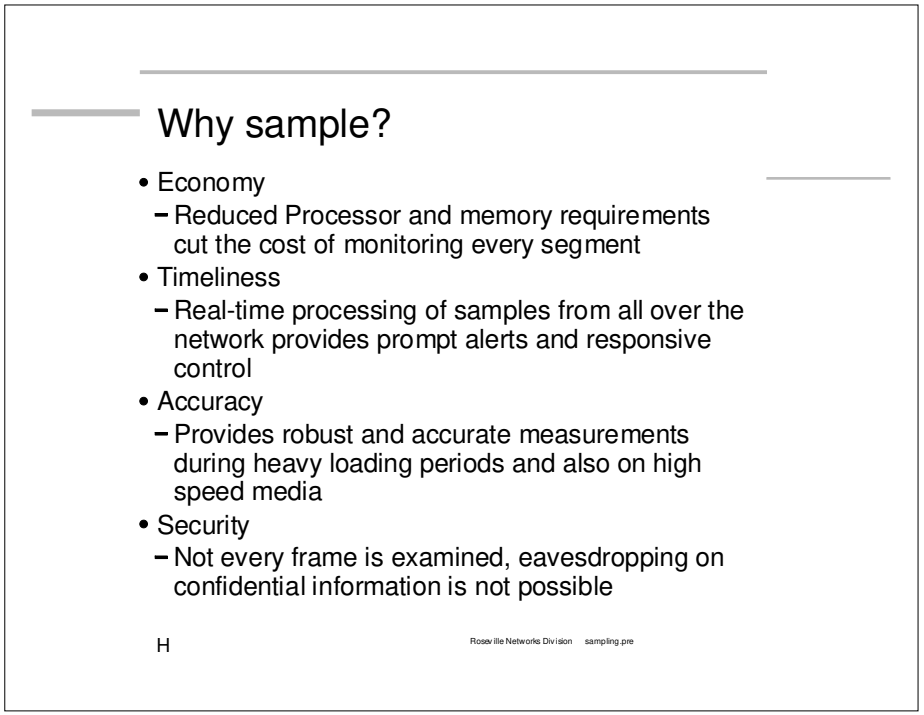
Hh



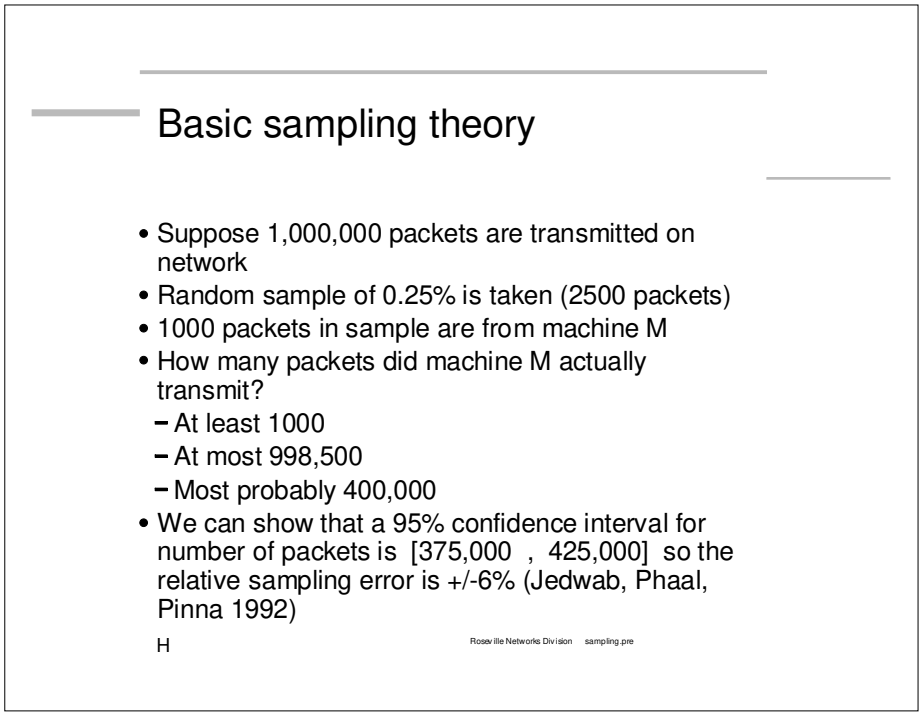
- SHORT TERM
- traffic and errors on segment and device basis
- frequent measurements so that can respond quickly to problems with no major loss of service (eg every minute)
- retain for hour to trace emergence of problem
- Top 5 sources of everything; destinations of frames and bytes; pairs of frames and bytes; connections of frames and bytes
- LONG TERM
- same basic stats as above but top n of little use over longer periods - changes in configuration changes traffic on each segment
- retain for one month, acivities occur on daily, weekly or monthly basis need to keep data long ebough to do this. Trending over long periods not very useful since past behaviour not good predictor of future - look at app deployment and kit purchasing plans



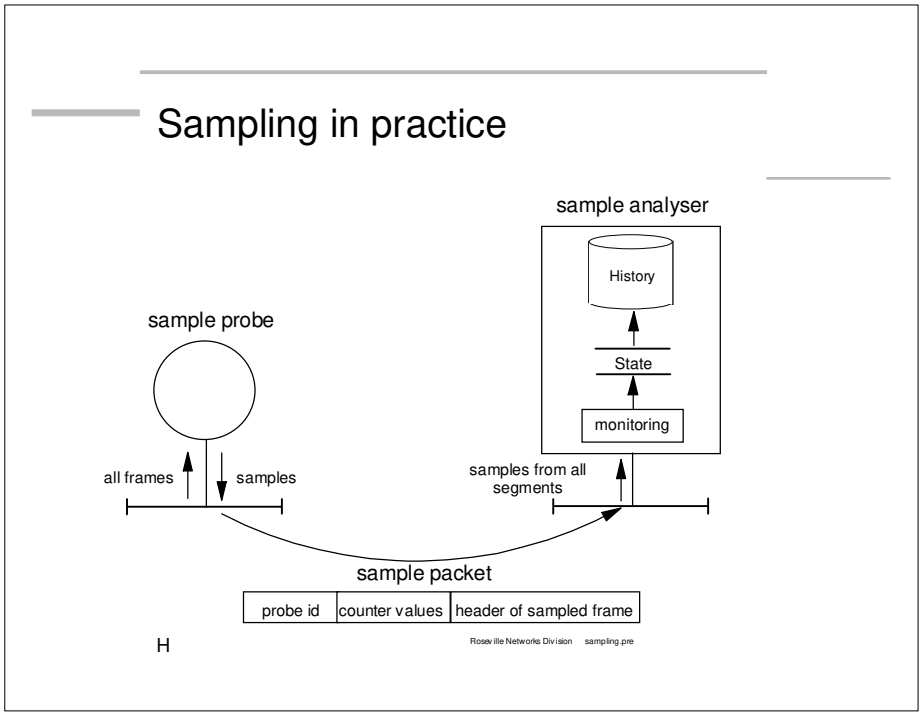
- Statistical sampling is widely used in industrial quality control, financial auditing, epidemiology etc where exhaustive analysis is impossible or too expensive
- Sampling invovles selecting and testing small representative group and inferring information about all items in group
- Applied to constructing traffic matrices and other traffic data: count every frame, but only select small proportion (eg 0.25%) to decode and analyse
- Forward samples to central location to be decoded and analysed to reduce memory requirements of sampling instrument
- Network wide traffic matrices easily produced at central location by examining samples from all over network



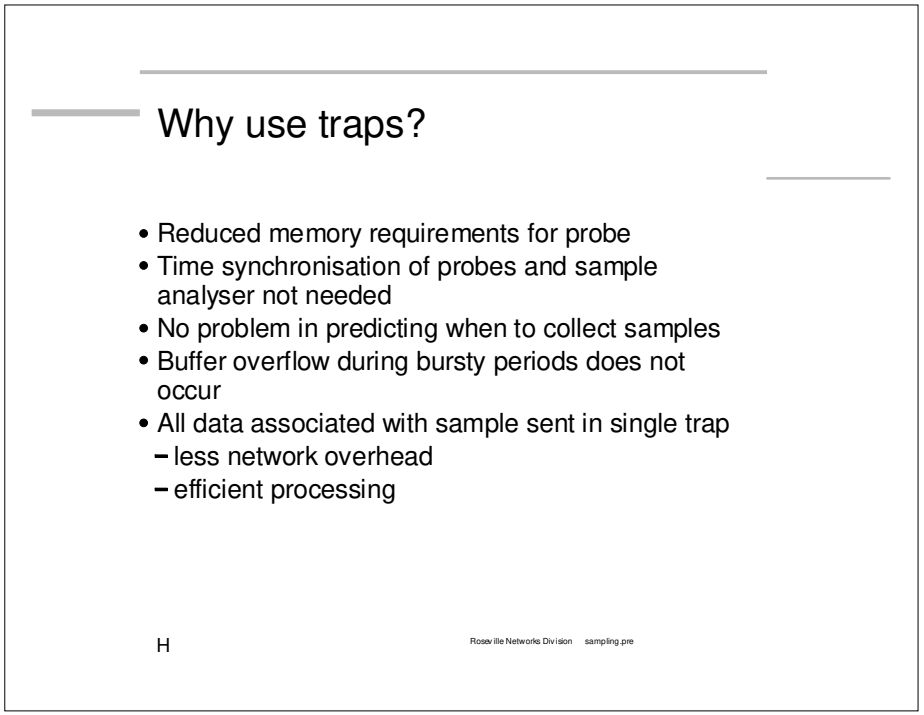
- As networking speed increase to 100Mb/s and beyond and the number of protocols decode grows, sampling becomes the only way to produce traffic matrices
- Other key benefits are:
- Economy - reduced processor and memory requirements (easily embedded)
- Timeliness - prompt alerts and reponsive control
- Accuracy - accurate measurements during heavy traffic; samplers are so simple that they can keep up at high traffic rates
- Security - Do not look at every packet, no eavesdropping on passwords, confidential information etc, no misuse of traffic patterns
- Limited traffic overhead on network



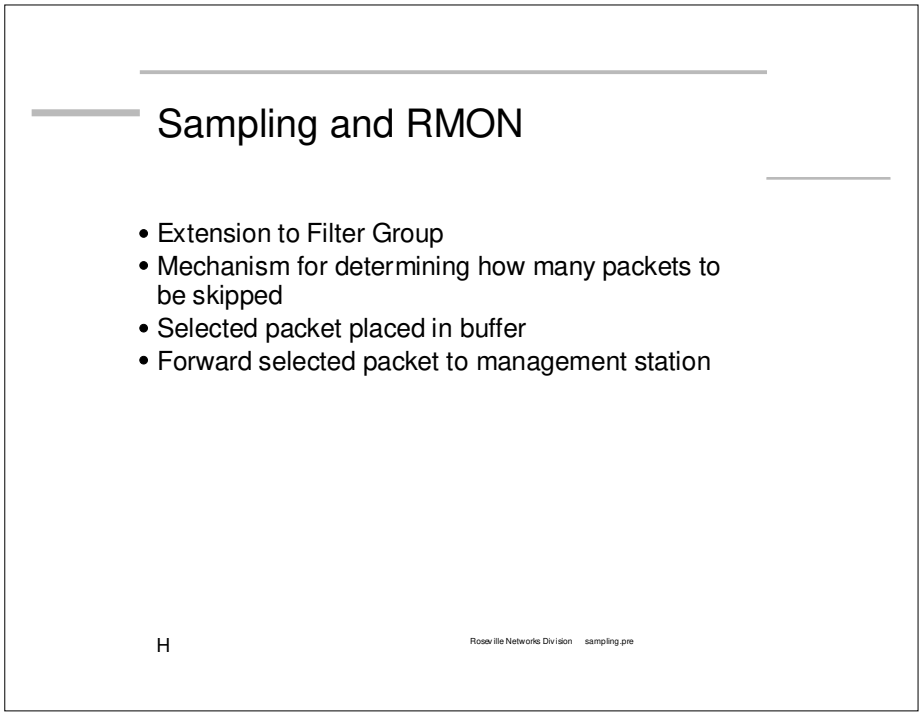
- Common sampling is based on time - leads to bias: consider 2 sources both producing same number of frames, one sending evenly over time, the other in intense bursts. Time based sampling will miss bursts and consistently underestimate traffic from bursty source
- Sampling based on frame count avaoinds this problem
- Simple scheme based on looking at every 400th packet (sample rate of 0.25%) will not give representative sample - may synchronise with periodic traffic patterns and introduce bias
- Introducing randomness will avoid synchronisation and bias
- Very unlikely that M produced exactly 400,000 frames. Can specify small range of values that are very likely, say 95% likely) to contain actual value - 95% confident that M actually sent between 375000 and 425000 frames. ie 400,000 frames +/- 6%
- Have shown that accuracy does not depend on number of frames on segment only number of samples taken. Useful for high speed media



- HP EASE demonstrates a practical implementation of sampling for traffic monitoring
- Sample probes connected all round network look at all frames
- Samples sent to central sample analyser as they are generated
- Each sample contains: where generated (probe id), total frame count at time sample was taken, header of sampled frame
- Basic segment statistics (frame/byte/broadcast/multicast/error counts) accurately maintained by probe are passed as part of sample counter values - don't need to estimete these.
- Information about errors that occur between samples is recorded and sent with next sample
- Sampling rate and destination to send samples configured using SNMP
- Samples forwarded to sample analyser using SNMP trap
- US patent granted May 24 1994 number 5 315 580



- Sample collection via polling requires probe to be able to buffer samples
- Determining number of buffers needed is difficult,need to predict how often to collect samples - buffer overflow will happen
- HP EASE samples time stamped when received by sample analyser (very close to time sample was taken) - centralised time synchronisation. If use polling need to time stamp sample at probe - distributed time synchronisation of probes and sample analyser: difficult and expensive
- In bursty periods, more frequent polling will be required - harder for probe to deal with. Buffer overflow will cause more errors
- Need to associate frame and byte counts seen when sample taken with each buffer, easy to parcel this up in single trap as opposed to multiple gets for each sample. Network overhead is less in forwarding single trap, compare to (get request+get response)*n. More efficient for sample analyser.



- Skip table for determining packets to be skipped - we did this orginally: need at least 128 entries downloaded by management station. This can be a pain.
- Originally thought that needed particular and controlled distribution of packets to be skipped
- Further investigation showed that random number generation for packets to be skipped was OK: specify mean and probe works out min and max for range based on its capabilities