Trajectory Sampling for Direct Traffic Observation

Nick Duffield Matt Grossglauser

AT&T Labs - Research

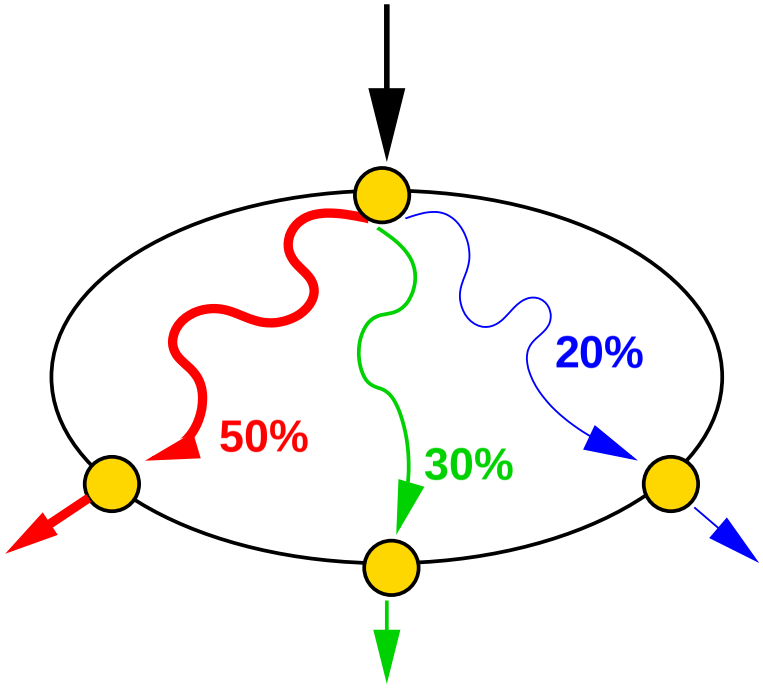
AT&T

Traffic Measurement

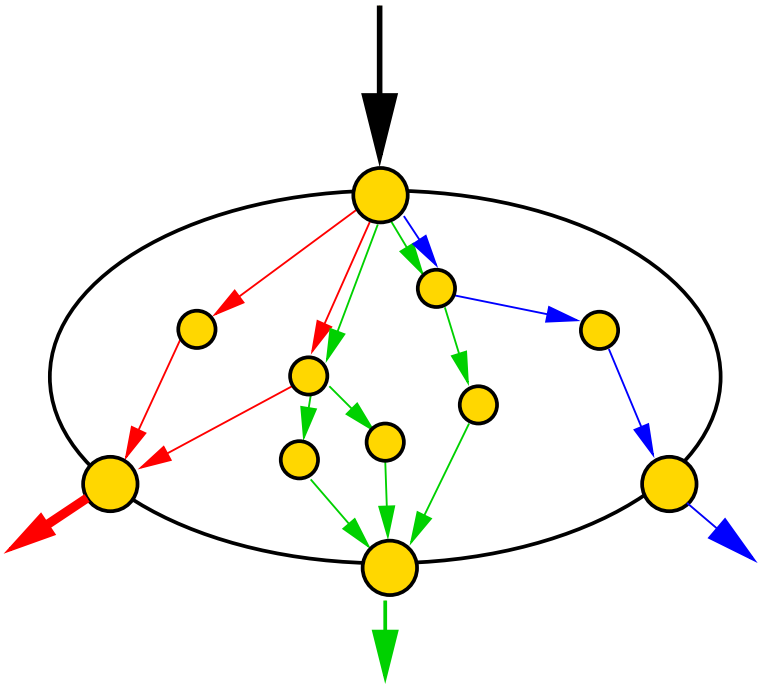
- Control loops on several time-scales
- Goals and approaches of traffic measurement:
  - User perspective:
    - End-to-end path characterization: ping, traceroute, packet pair, ...
  - Traffic load into network domain:
    - Aggregate flow measurement: cisco netflow
  - Network resources:
    - Isolated: aggregate link measurments (SNMP)
    - Resource allocation: incoming traffic $\Rightarrow$set of resources
- Traffic engineering relies on *traffic* $\Rightarrow$*resource* mapping

Traffic Matrix vs. Path Matrix

- Traffic matrix = load
  - # Bytes from ingress i to egress j
- Path matrix: joint load and resource allocation
  - Spatial flow of traffic through domain
  - # bytes *for all paths* p from ingress i to egress j

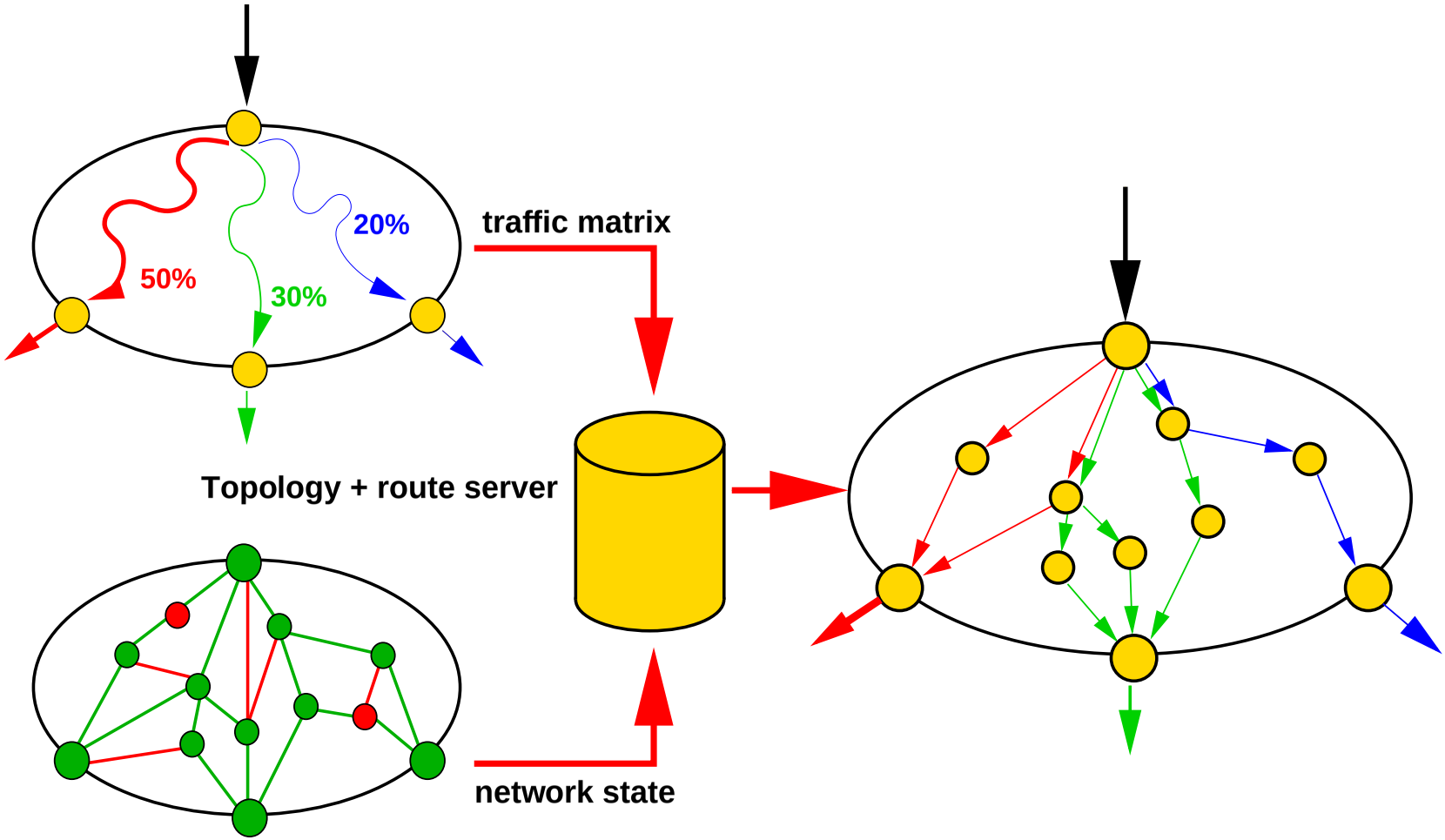


Traffic matrix



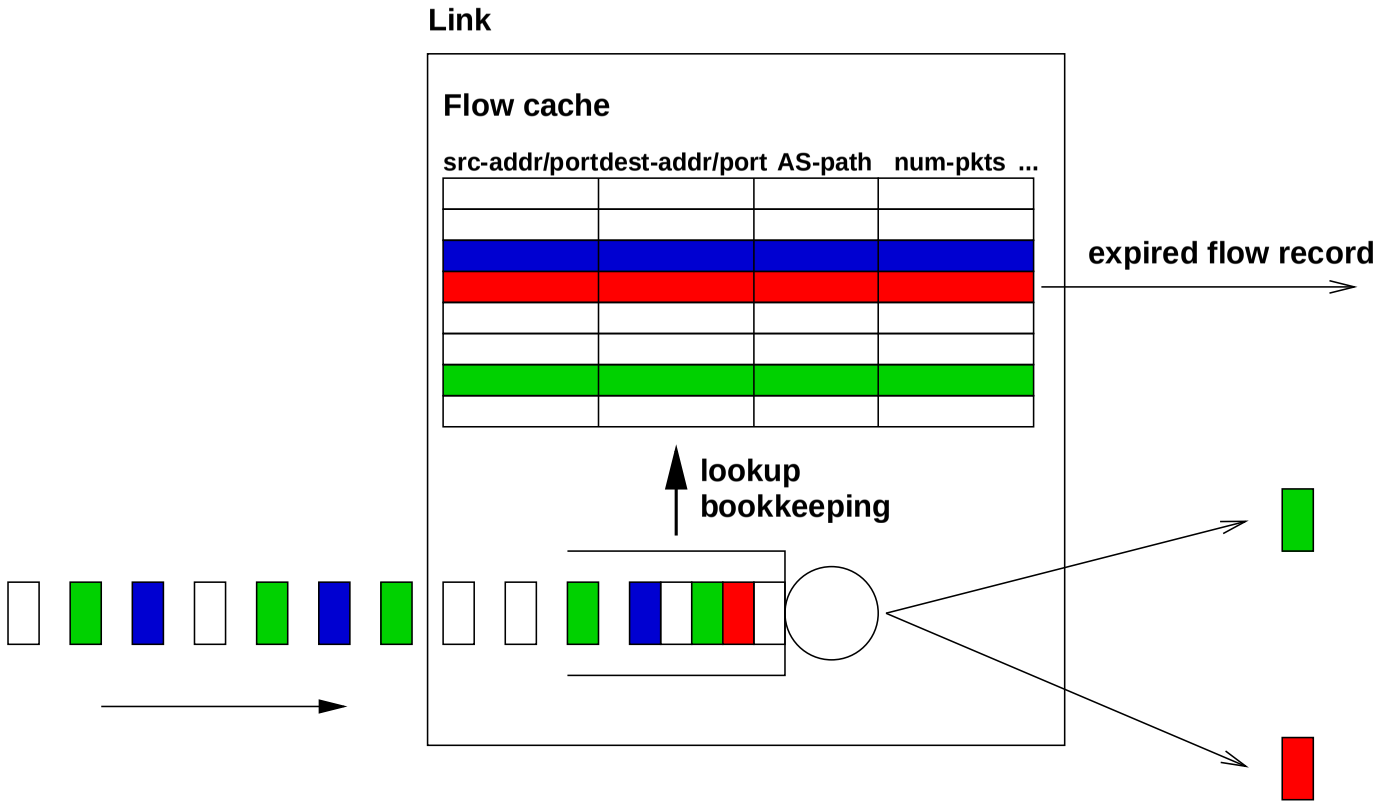
Path matrix

From Flow Measurements to the Path Matrix



Introduction

Flow Aggregation (cisco netflow)



- Disadvantages:
  - Measurement traffic hard to control + massive
  - Cache size ⇒forced expiry can lead to bias
  - Inherent reporting delay ⇒long-lived flows

AT&T

Network State Uncertainty Complicates Path Matrix Inference

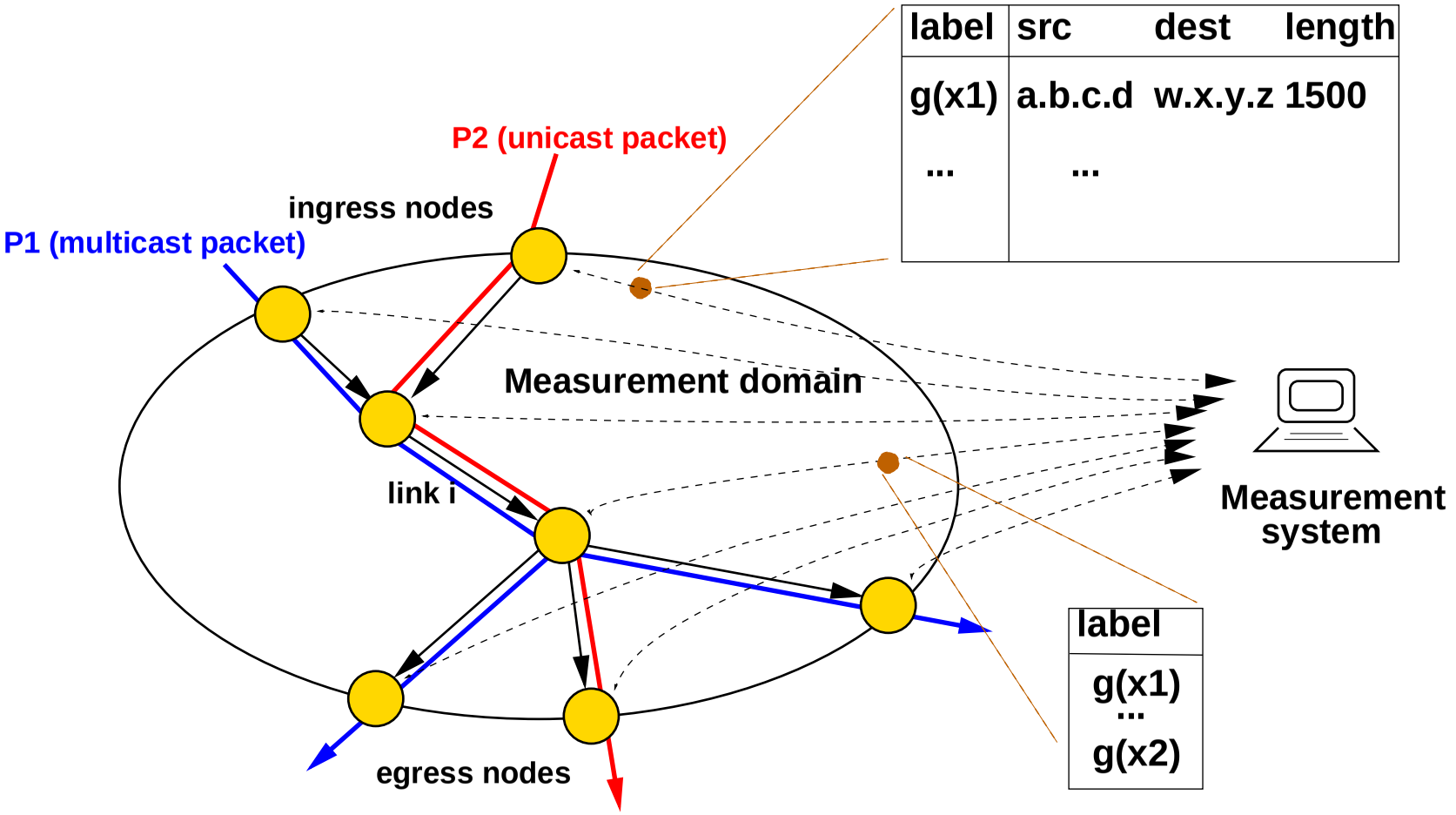
- Complex routing
  - Dynamic routing and QoS routing
- Vendor-specific implementations
  - OSPF shortest path splitting
- Deliberate randomness
  - RED, routing updates
- Inter-domain dynamics
  - BGP advertisements
- Higher-level configuration management
  - Policy-based networking
- Faults

AT&T

Trajectory Sampling for Direct Traffic Observation

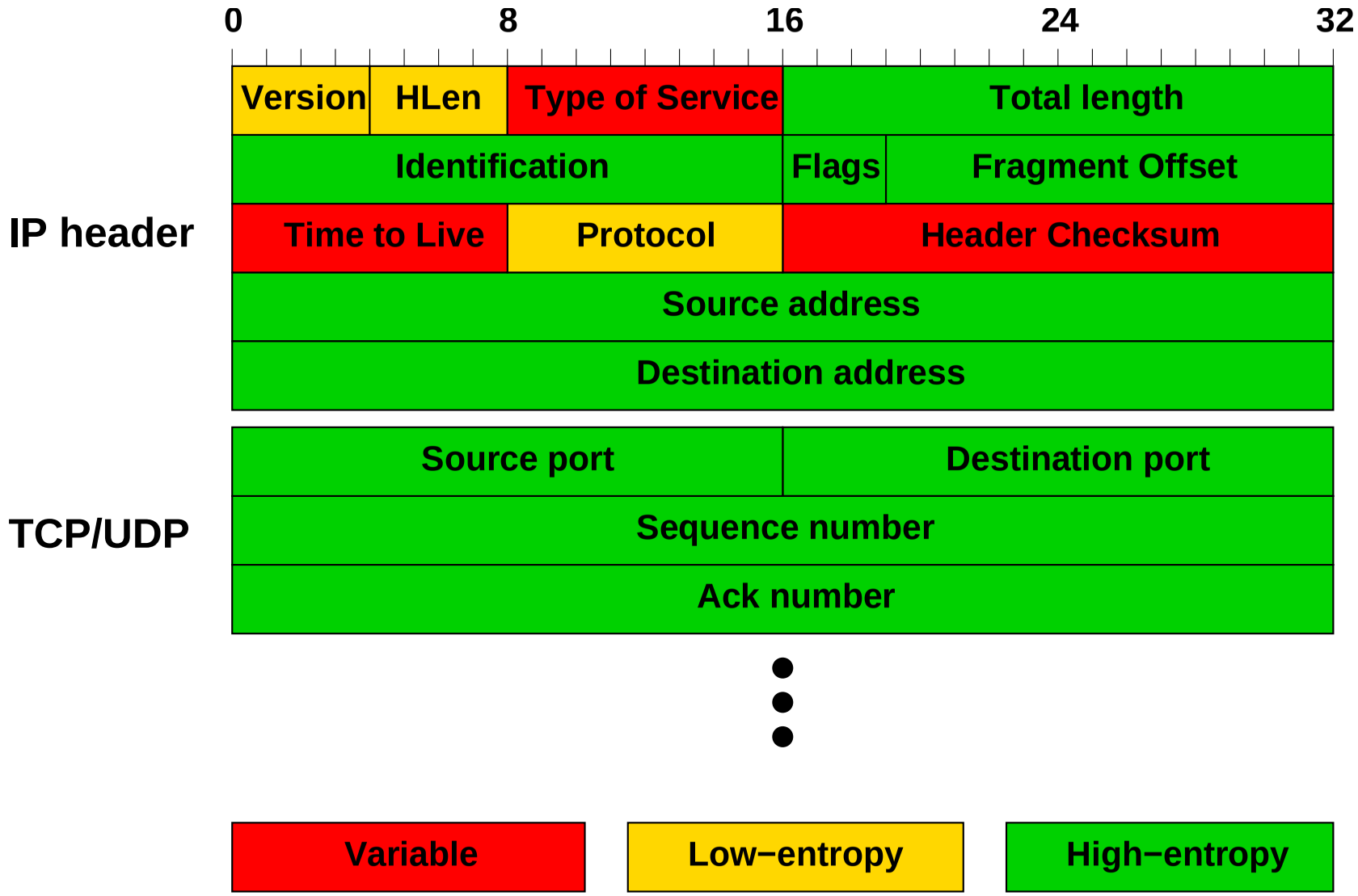
- Goal: direct observation of traffic
  - No model and estimation of network state
- Basic idea of trajectory sampling:
  - Sample packets at each link
  - Sampling decision based on hash over *packet content*
  - Consistent sampling $\Rightarrow$packet trajectories
  - Trajectories only require *packet labels*
  - Use second hash for labels
- Exploit packet entropy to obtain statistically relevant subset of packets based on deterministic function

Trajectory Sampling



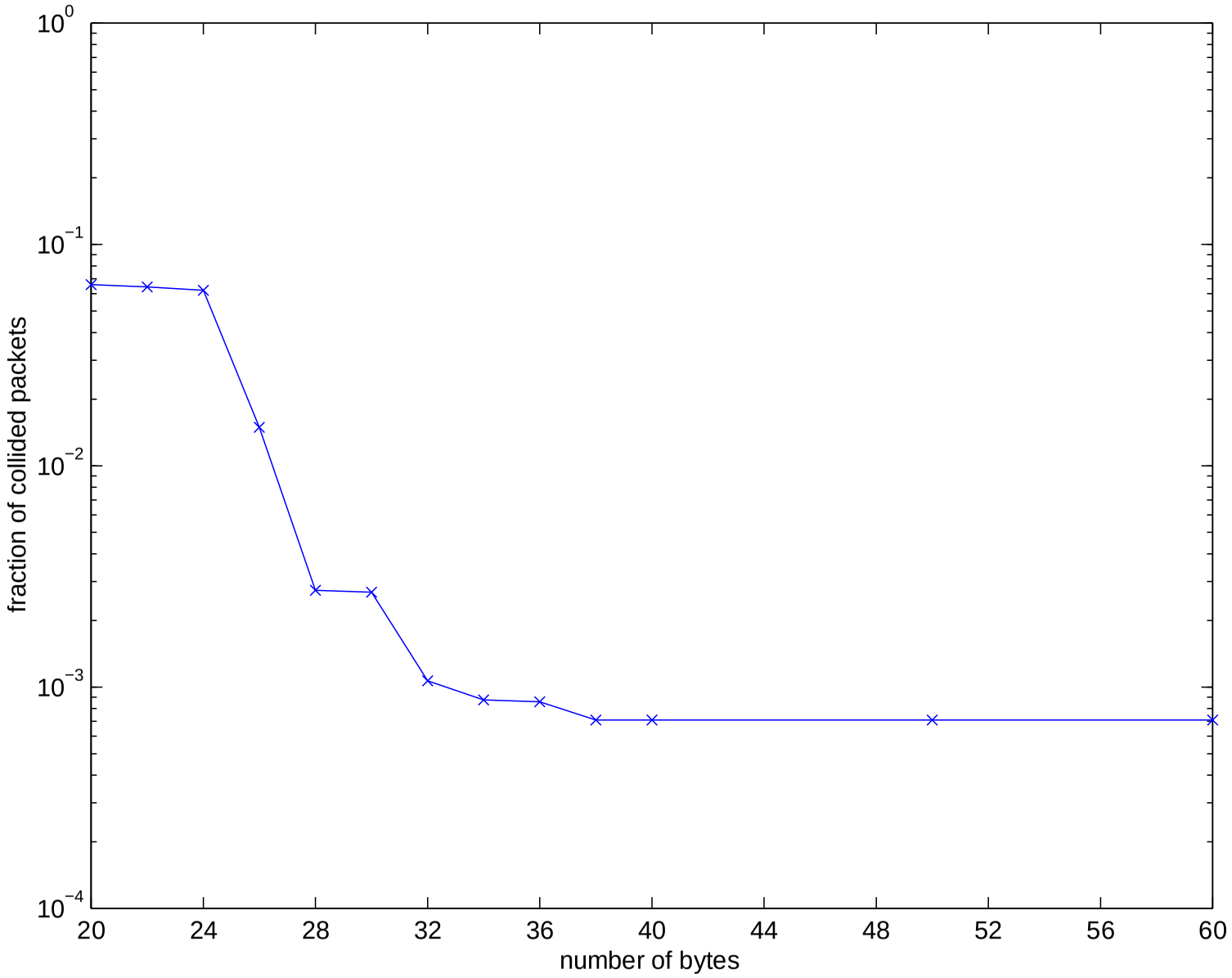
- Collect fields of interest only once (ingress)
- Multicast requires no special treatment

Fields Included in Hashes



AT&T

Collisions: Identical Packets are Rare



AT&T

Sampling and Identification Hashes

- x: subset of packet bits, represented as binary number
- Sampling hash: $h : x \rightarrow \{0, \ldots, A-1\}$
  - $h(x) = x \mod A$
  - sample packet if $h(x) \leq r$
  - r/A: thinning factor, control fraction sampled
- Identification hash: $g : x \rightarrow \{0, \ldots, M-1\}$
  - $g(x) = x \mod M$
- Conditions for $h(.)$ and $g(.)$:
  - A, M: appropriately chosen prime numbers
  - $h(x)$ and $g(x)$ should be "independent" $\Rightarrow A \neq M$
  - Avoid predictable correlation: $q^k \neq \pm a \mod M$ for small a, k (radix q) [Knuth]

AT&T

Quality of Pseudo-Random Sampling

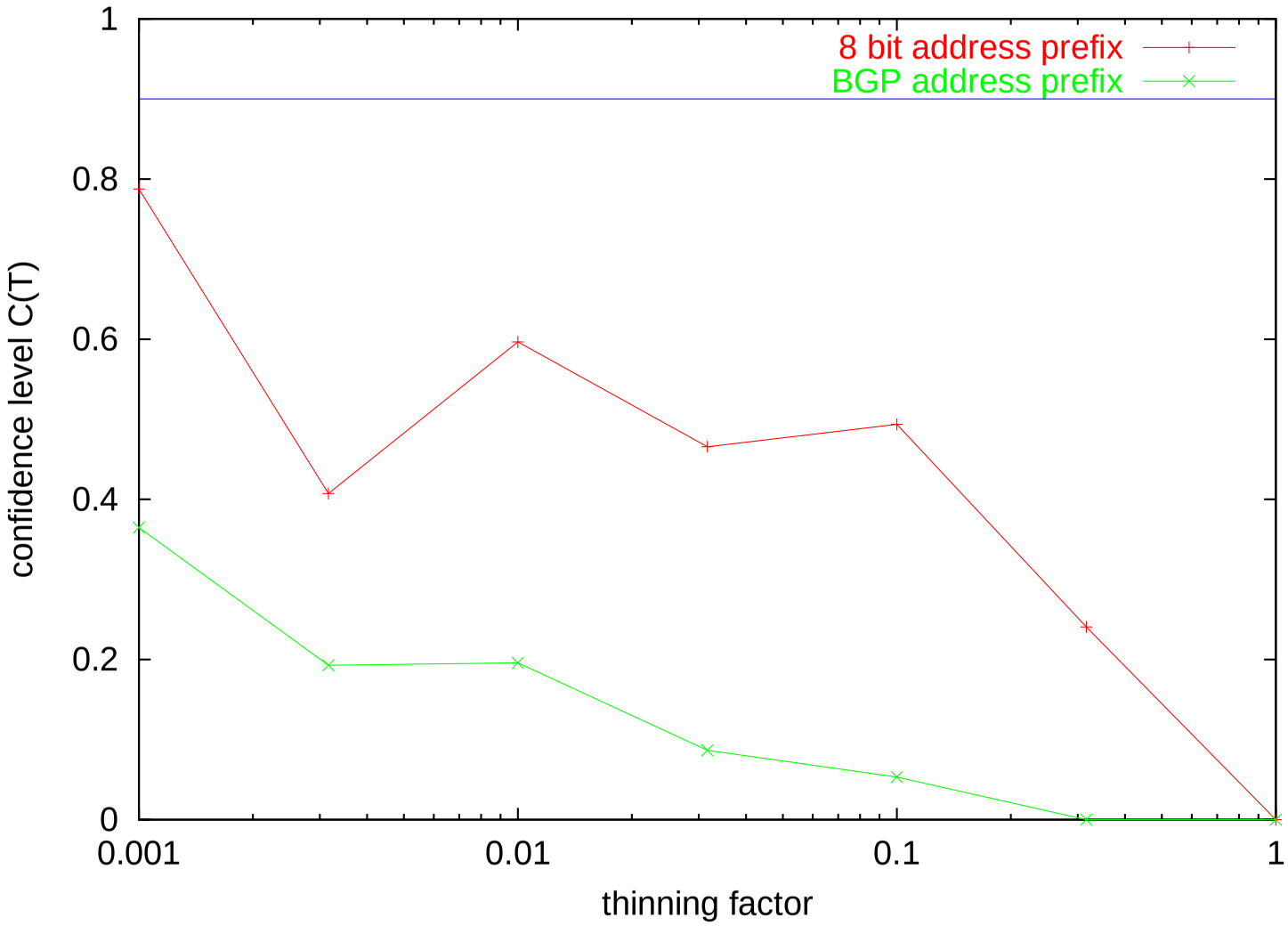
- We want to infer traffic metrics from sampled subset
  - For example: what fraction of traffic of customer x on link y?
- Question: is sampled subset statistically representative?
  - Distribution over full set = distribution over sample set?
- Experiment: check independence of sampling decision and packet fields
  - χ^2-test over source addresses
  - Hypothesis: sampled distribution consistent with full distribution
  - Contingency table:

m_{01}	m_{02}	$\ldots$	m_{0I}	m_0
m_{11}	m_{12}	$\ldots$	m_{1I}	m_1
n_1	n_2	$\ldots$	n_I	n

  - χ^2 statistic: $T = \sum_{i=0}^{1} \sum_{j=1}^{I} \frac{(m_{ij} - \overline{m}_{ij})^2}{\overline{m}_{ij}}$
  - Confidence level $C(T)$ for hypothesis, where C is cdf of χ^2 with $I - 1$ degrees of freedom

χ^2-Test for Independence of Sampling Decision & Addresses

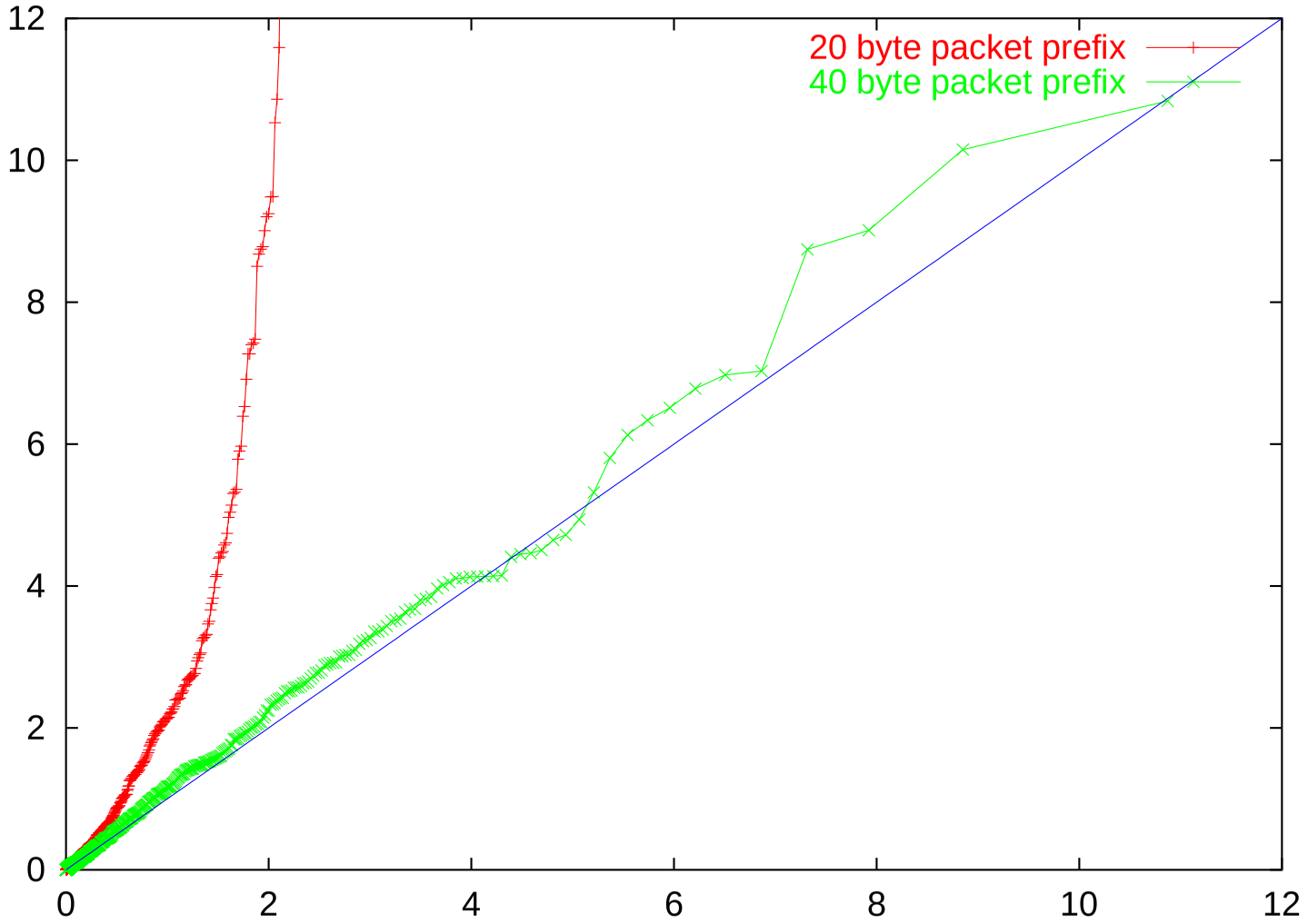
- If $C(T) < 1 -$ significance level $\Rightarrow$accept hypothesis



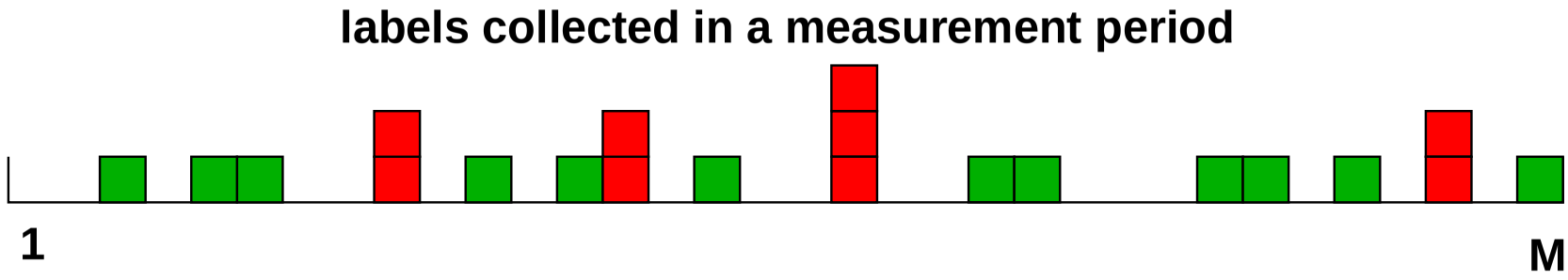
AT&T

Bitwise Address Distribution

- Independence of sampling and lth packet bit



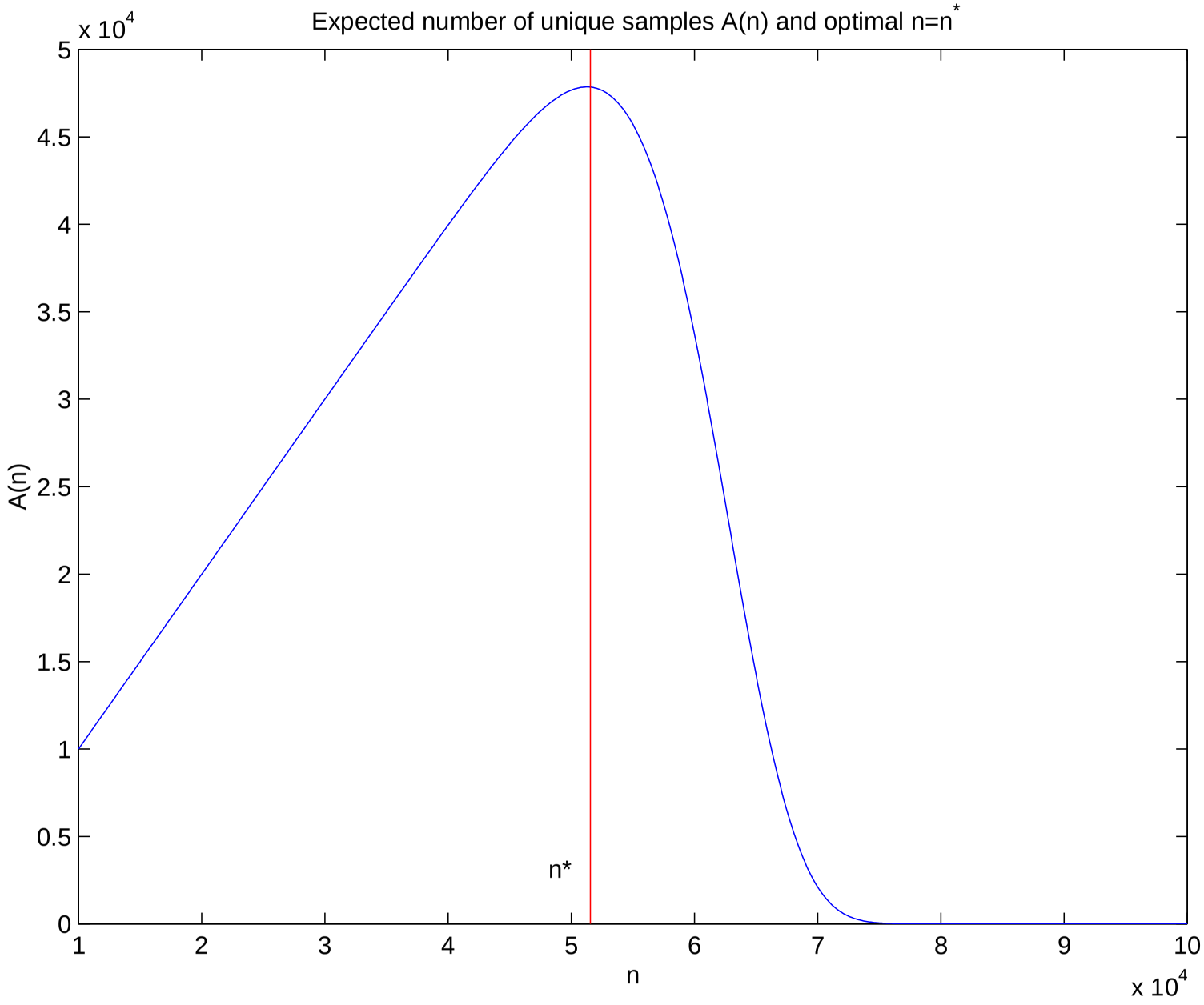
Optimal Sampling



- Fix amount of measurement traffic c per period
- Tradeoff: collisions vs. label size
- Problem:
  - n: number of samples in sampling period
  - M: alphabet size, $m = \log_2 M$ [bits/label]
  - $n \cdot m$: total amount of measurement traffic [bits]
  - Goal: maximize number of *unique* labels subject to $n \cdot m \leq c$.
  - Optimal alphabet size: $M^* = c\log(2)$
    Optimal number of samples: $n^* = \frac{M^*}{\log(M^*)}$

Example: $c = 10^6$ bit $\Rightarrow m^* = 19.4$ bit/label
$n^* = 5.15 \cdot 10^4$ samples

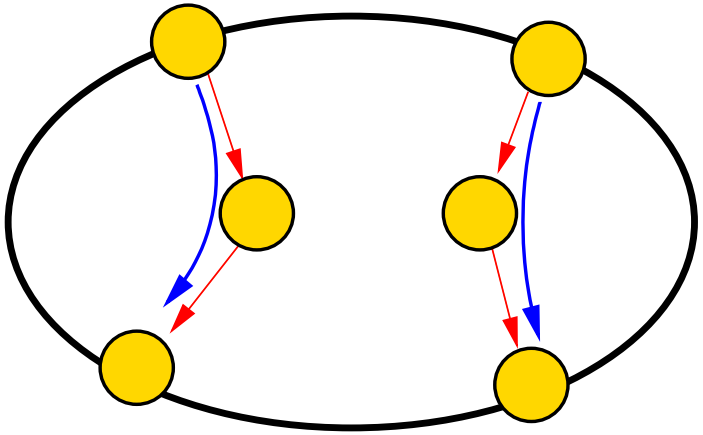
Number of Unique Samples ($c = 10^6$ [bits])



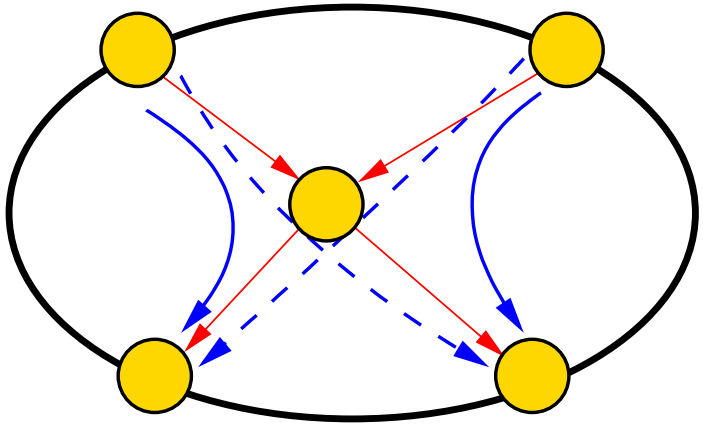
AT&T

Trajectory Disambiguation

- Label subgraph: set of links where label appeared
- Some collisions can be disambiguated

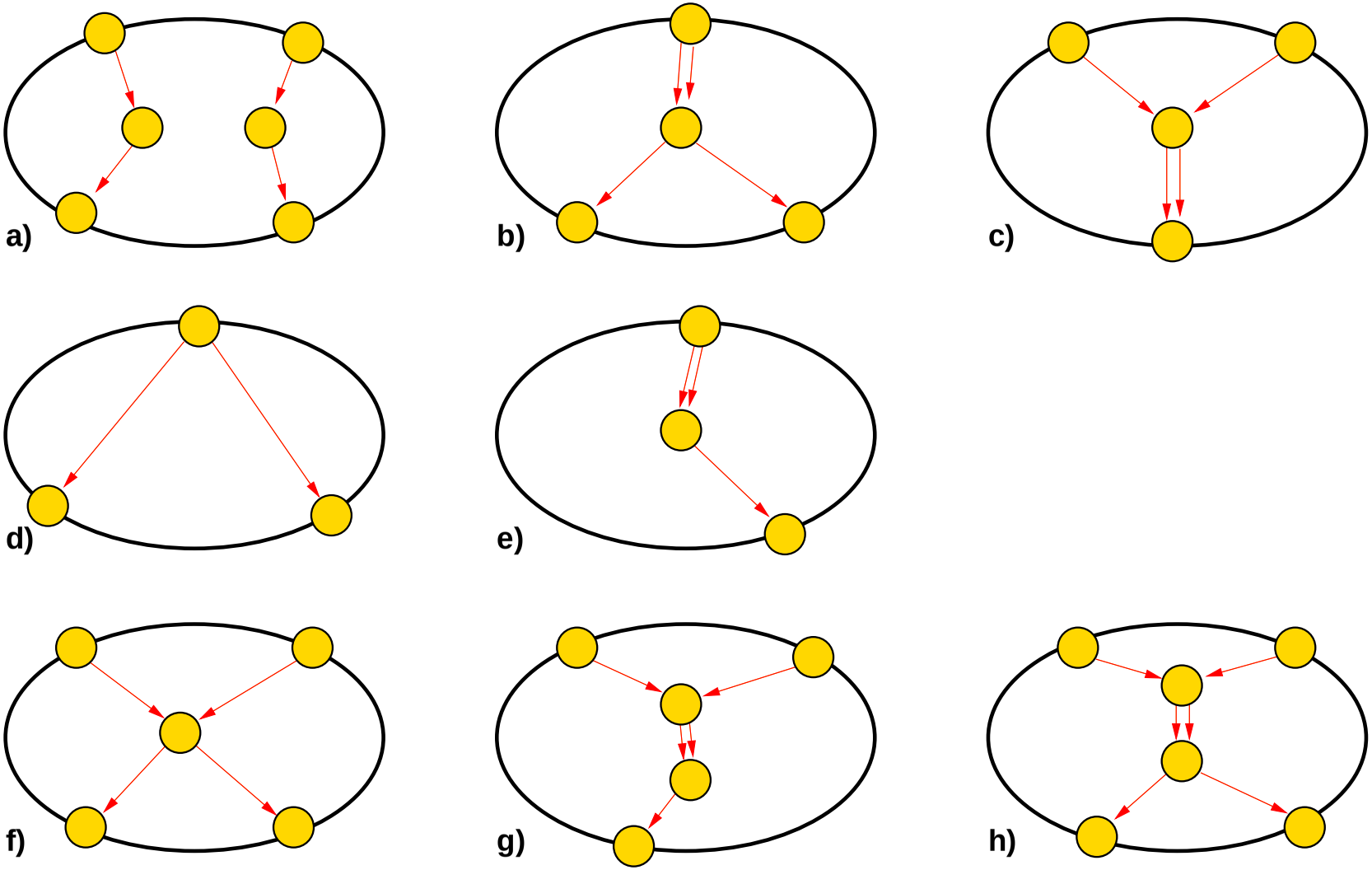


unambiguous



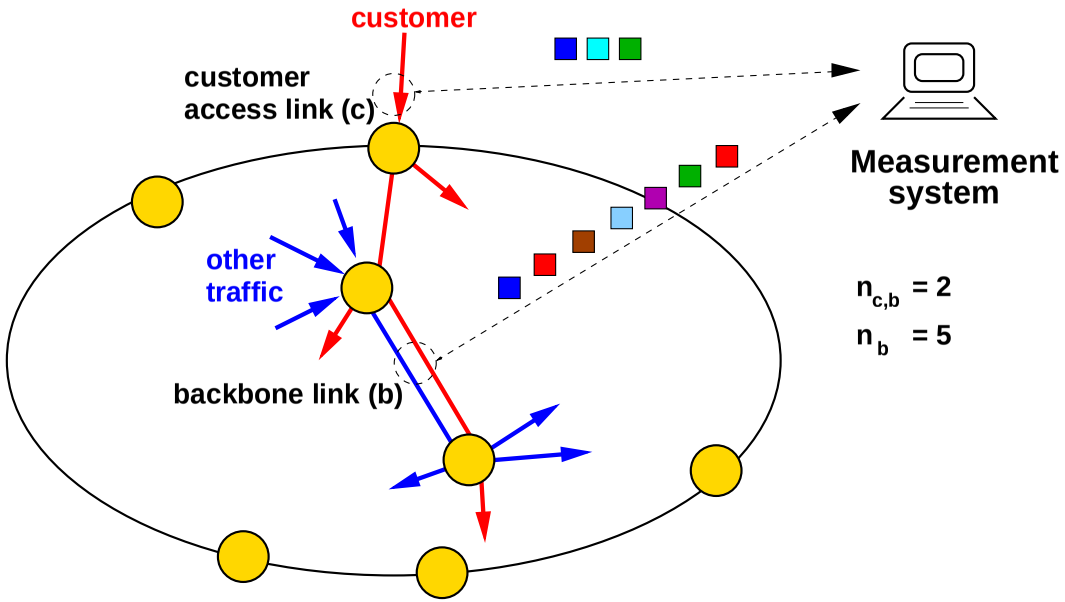
ambiguous

Examples of Trajectory Disambiguation



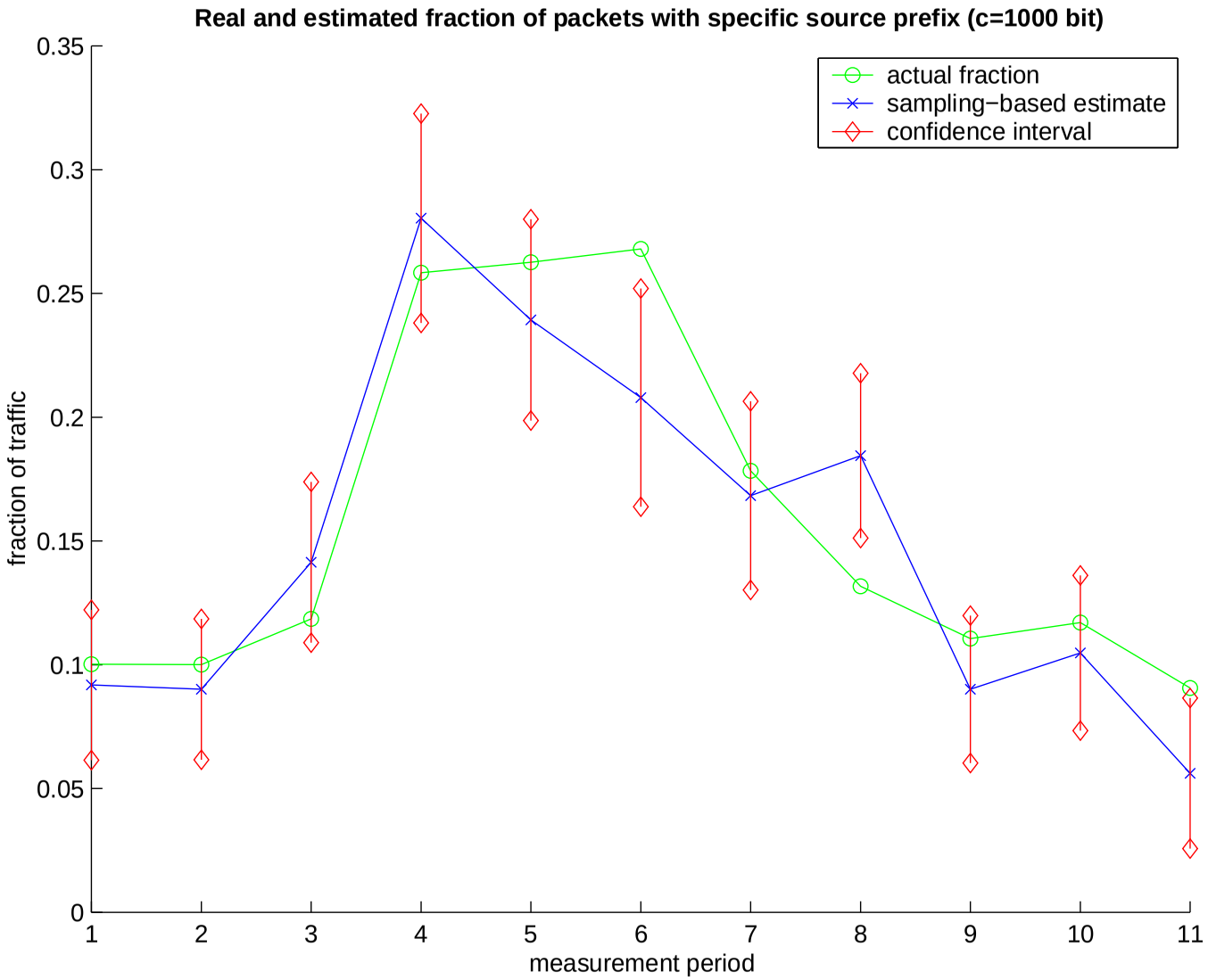
Inference Experiment

- Experiment: inference from trajectory samples
  - Estimate fraction of traffic from customer
  - Customer traffic: small source address subset

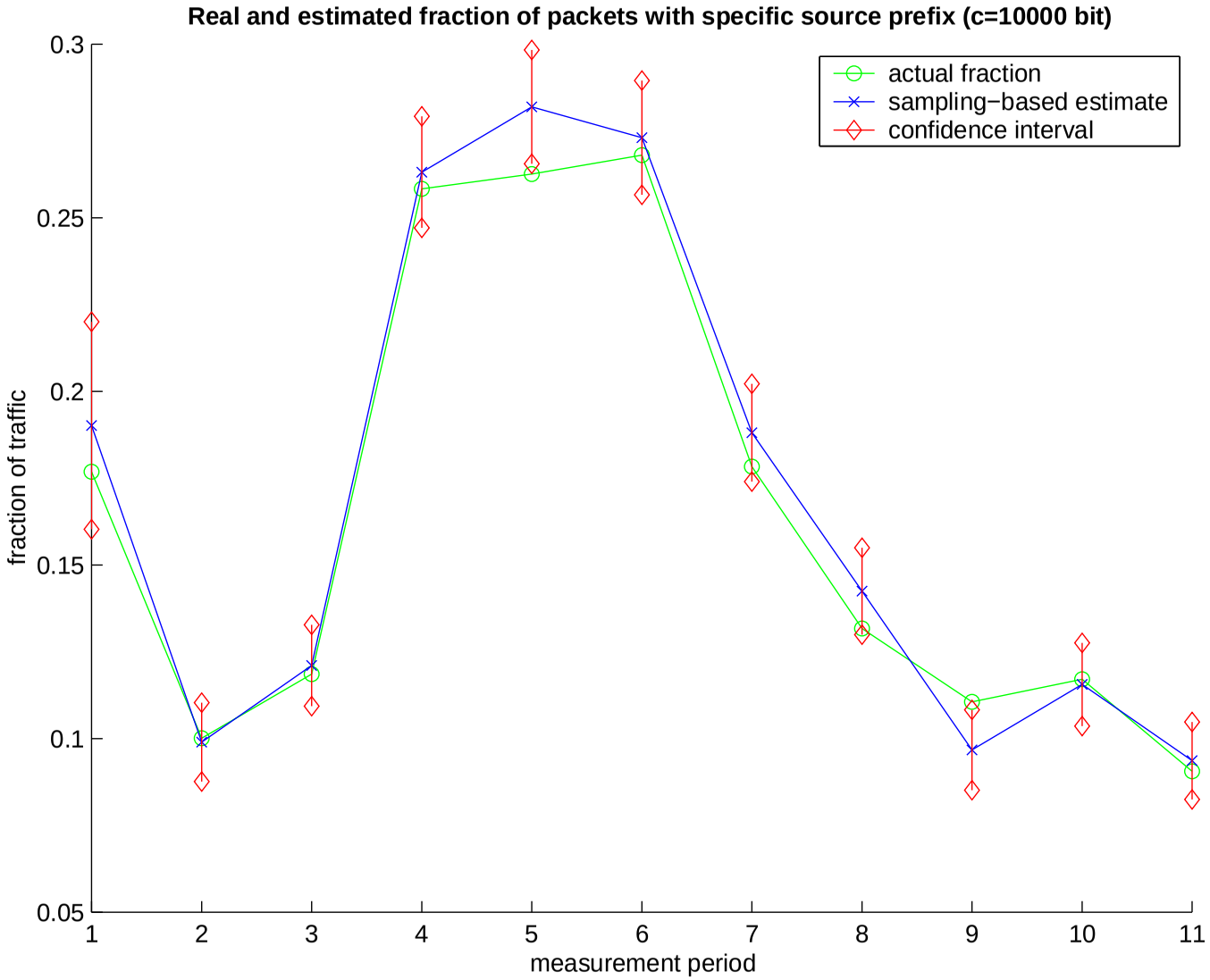


- Fraction of customer traffic on backbone: μ
  Estimator: $\hat{\mu} = n_{c,b}/n_b$
  $n_{c,b}$: # unique labels common on both links
  n_b: # unique labels on backbone link
- Ingress link and source address correlated

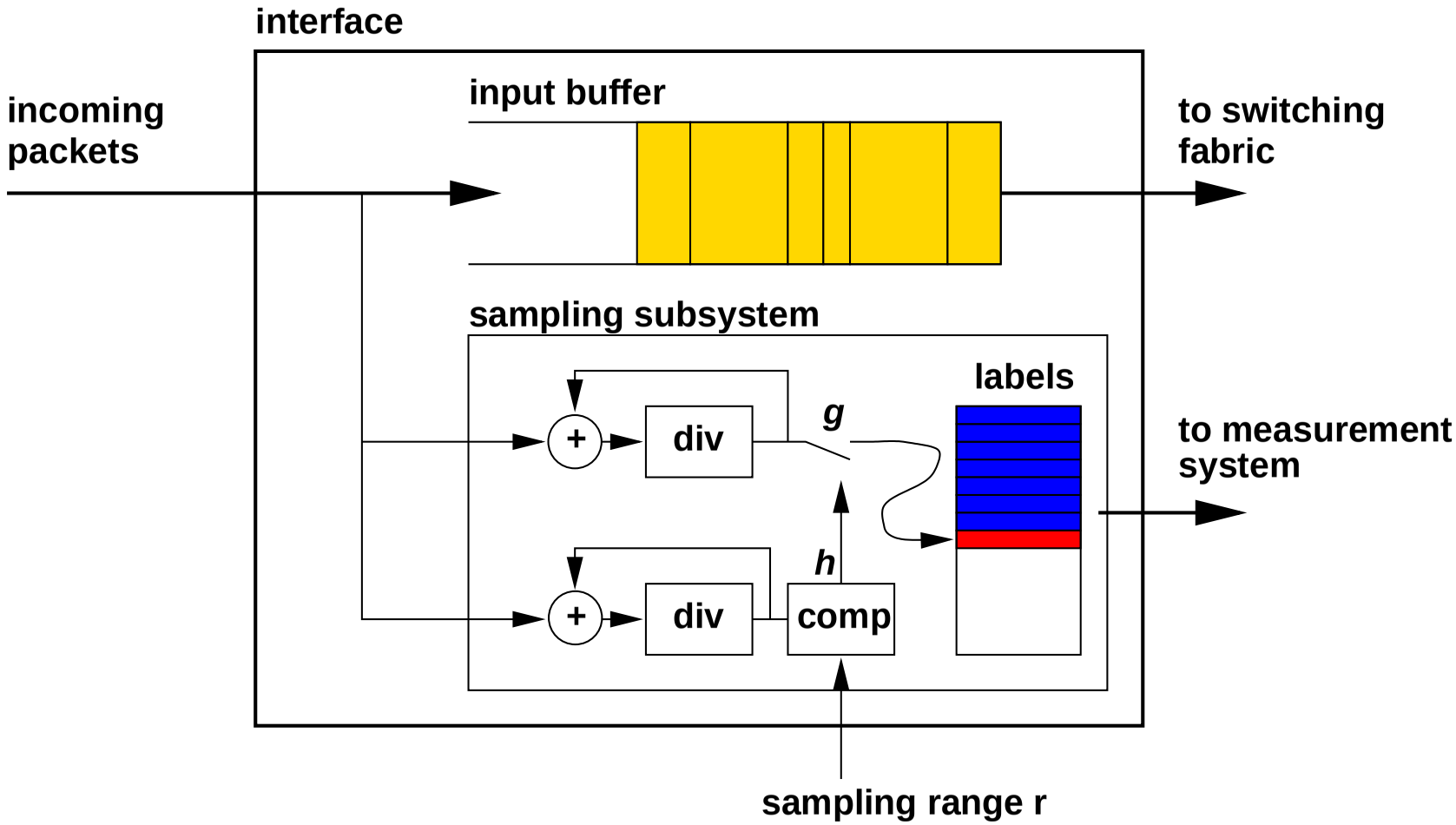
Estimated Customer Traffic ($c = 10^3$ [bits/epoch])



Estimated Customer Traffic ($c = 10^4$ [bits/epoch])



Sampling Device Implementation



AT&T

Sampling Device Implementation

- Interface vs. processing speed:
  - OC-192: 10Gbps
  - DSP (TI): $600 \cdot 10^6$ MACs (32bit) $\Rightarrow$20 Gbps
    I/O 300MHz $\times$ 256 bit $\Rightarrow$ 70 Gbps
  - Moore's Law vs. interface speed growth
- Tunneling (MPLS): some additional logic to look "behind" tunnel header

AT&T

Summary

- Advantages:
  - Provides path matrix = spatial flow of traffic
  - Direct method $\Rightarrow$requires no network model + state
  - No router state, cheap to implement
  - Multicast can be handled (trajectory = tree)
  - Measurement traffic can be controlled
  - Measurement delay negligible
- Disadvantages:
  - Requires support from all (most) interfaces
- Future work:
  - Efficient storage of trajectory samples
  - Build + evaluate estimators
  - Define management interface