Network Measurement

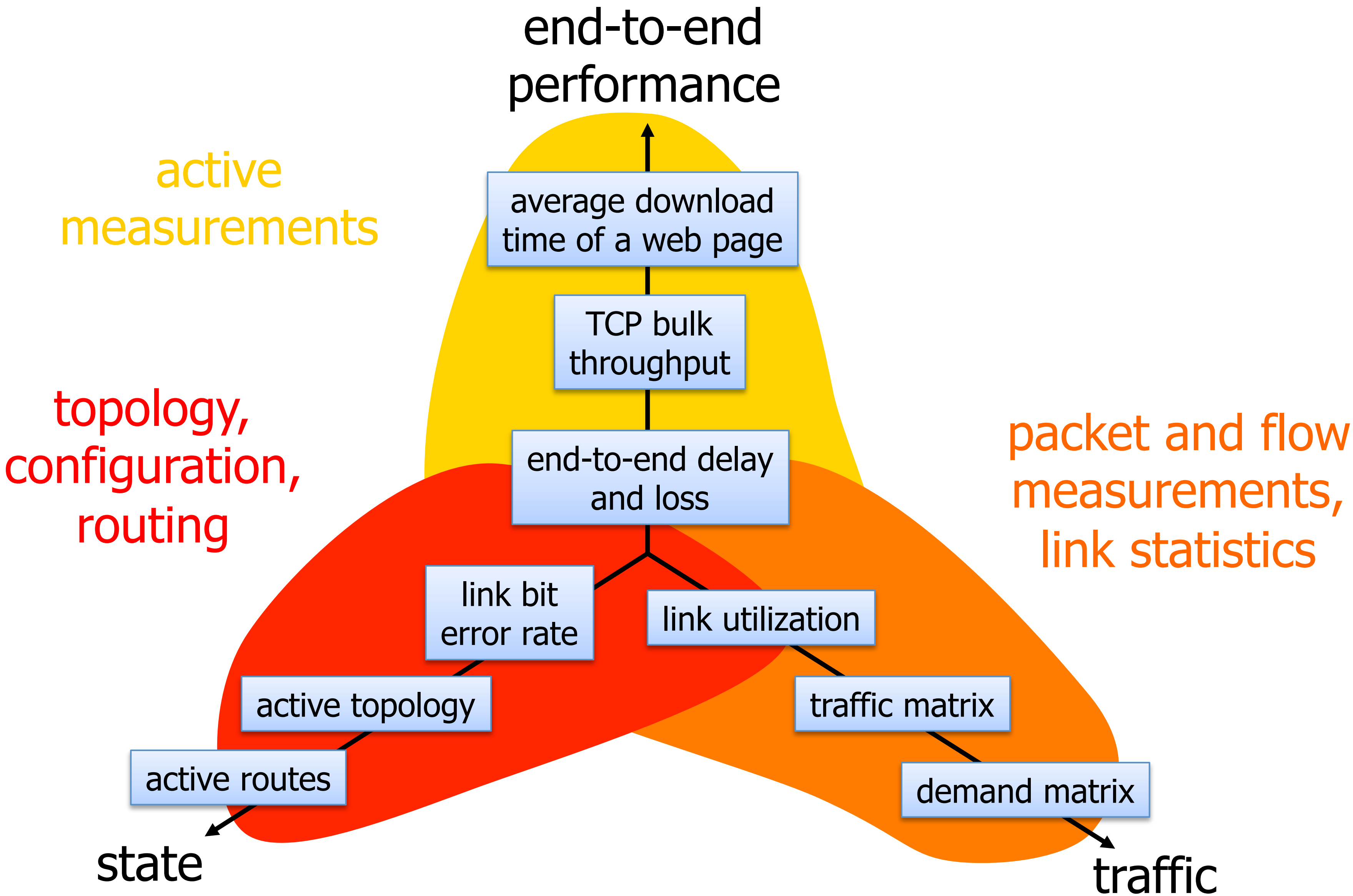
COS 461 Recitation

http://www.cs.princeton.edu/courses/archive/spr14/cos461/

Why Measure the Network?

- Scientific discovery
  - Characterizing traffic, topology, performance
  - Understanding protocol performance and dynamics
- Network operations
  - Billing customers
  - Detecting, diagnosing, and fixing problems
  - Planning outlay of new equipment

Types of Measurement

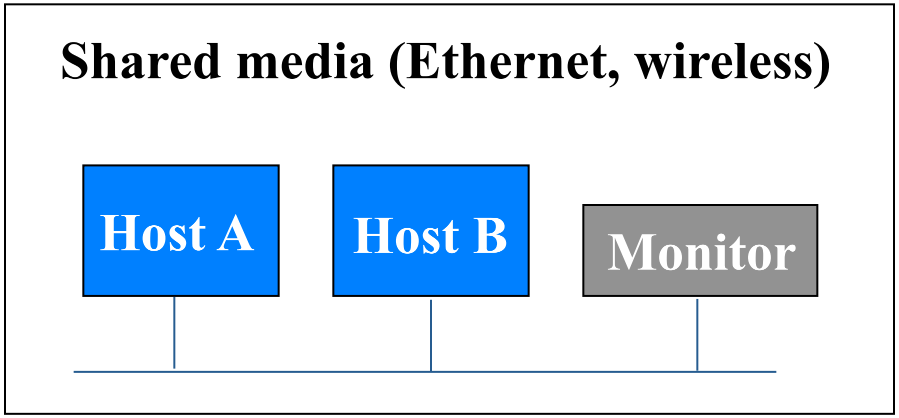


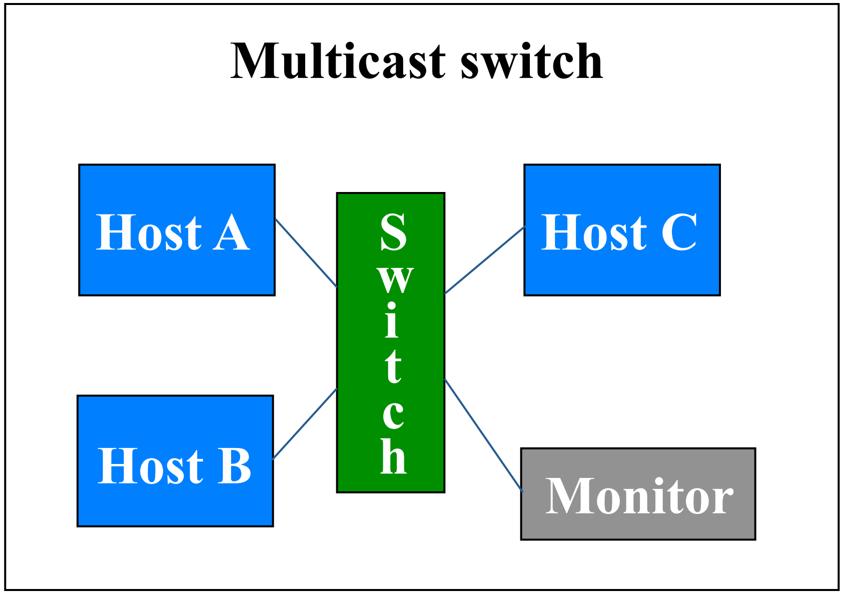
Traffic Measurement

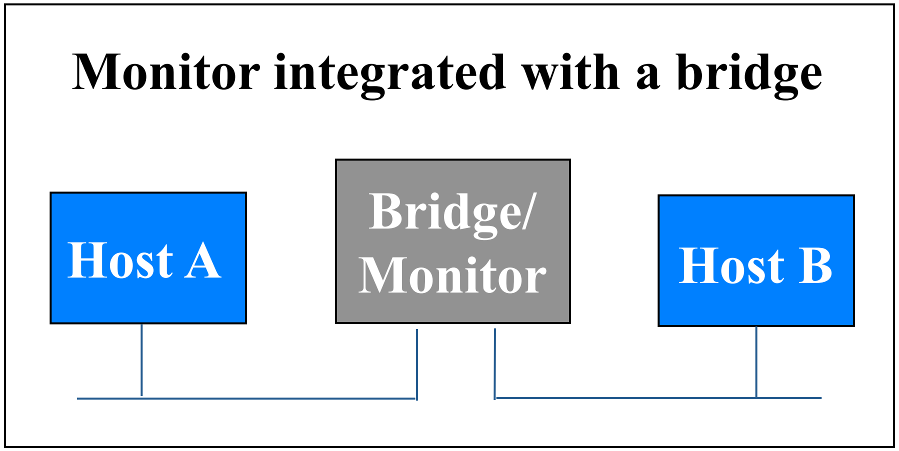
Packet Monitoring

- Definition
  - Passively collecting IP packets on one or more links
  - Recording IP, TCP/UDP, or application-layer traces
- Scope
  - Fine-grain information about user behavior
  - Passively monitoring the network infrastructure
  - Characterizing traffic and diagnosing problems

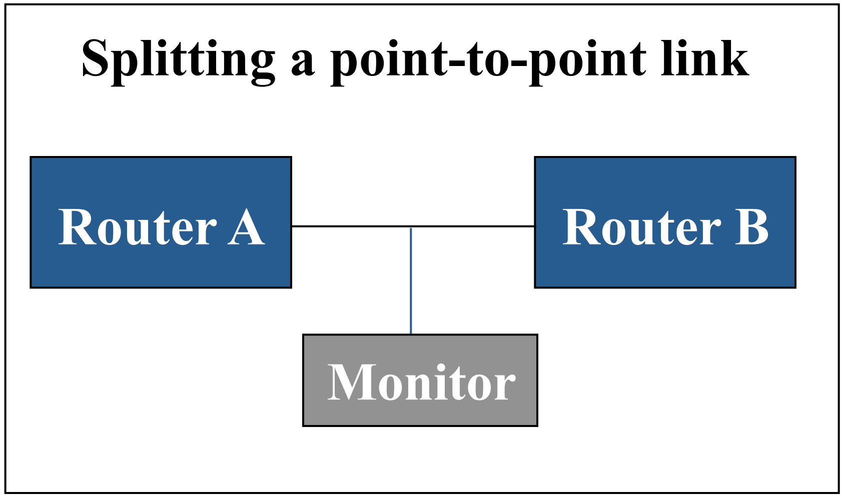
Monitoring a LAN Link

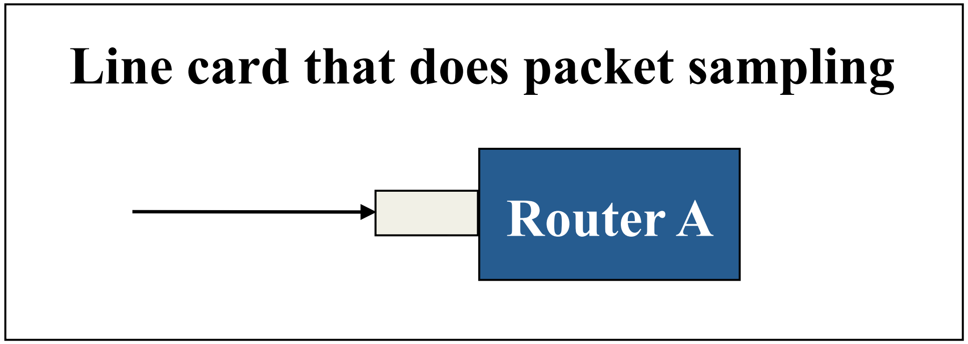






Monitoring a WAN Link





Selecting the Traffic

- Filter to focus on a subset of the packets
  - IP addresses/prefixes (e.g., to/from specific sites)
  - Protocol (e.g., TCP, UDP, or ICMP)
  - Port numbers (e.g., HTTP, DNS, BGP, Napster)

- Collect first n bytes of packet
  - Medium access control header (if present)
  - IP header (typically 20 bytes)
  - IP+UDP header (typically 28 bytes)
  - IP+TCP header (typically 40 bytes)
  - Application-layer message (entire packet)

What to measure to..

- Understand router workload model
  - Distribution of packet sizes
- Quantify web transfer sizes
  - Number of packets/bytes per connection
- Know which servers are popular & who their heavy clients are
  - Collect source/destination IP address (on port 80)
  - Collection application URLs (harder!)
- Know if a denial-of-service attack is underway
  - SYN flooding (spoofable)
  - Unusual # requests to particular (potentially expensive) page

Analysis of IP Header Traces

- Source/destination addresses
  - Identity of popular Web servers & heavy customers
- Distribution of packet delay through the router
  - Identification of typical delays and anomalies
- Distribution of packet sizes
  - Workload models for routers
- Burstiness of the traffic on the link over time
  - Provisioning rules for allocating link capacity
- Throughput between pairs of src/dest addresses
  - Detection and diagnosis of performance problems

TCP Header Analysis

- Source and destination port numbers
  - Popular applications; parallel connections
- Sequence/ACK numbers and packet timestamps
  - Out-of-order/lost packets; throughput and delay
- Number of packets/bytes per connection
  - Web transfer sizes; frequency of bulk transfers
- SYN flags from client machines
  - Unsuccessful requests; denial-of-service attacks
- FIN/RST flags from client machines
  - Frequency of Web transfers aborted by clients

Packet Contents

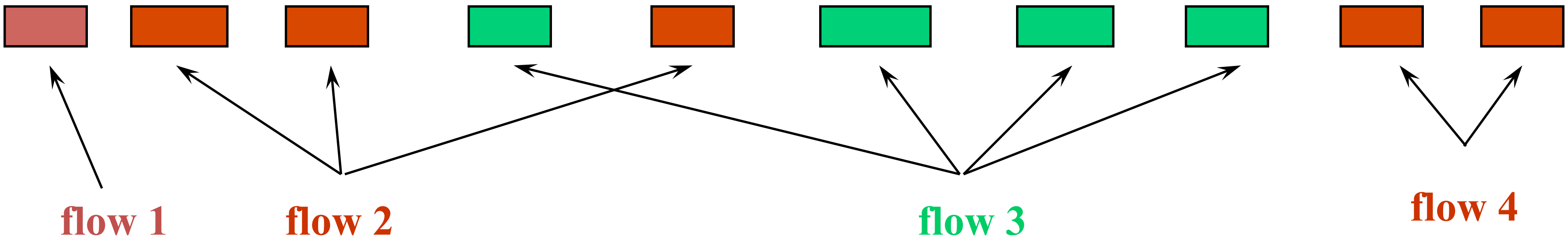
- Application-layer header
  - HTTP and RTSP request and response headers
  - FTP, NNTP, and SMTP commands and replies
  - DNS queries and responses; OSPF/BGP messages
- Application-layer body
  - HTTP resources (or checksums of the contents)
  - User keystrokes in Telnet/Rlogin sessions

Application-Layer Analysis

- URLs from HTTP request messages
  - Popular resources/sites; 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, error frequency, query latency
- Contents of Telnet/Rlogin sessions
  - Intrusion detection (break-ins, stepping stones)

Flow Measurement (e.g., NetFlow)

IP Flows



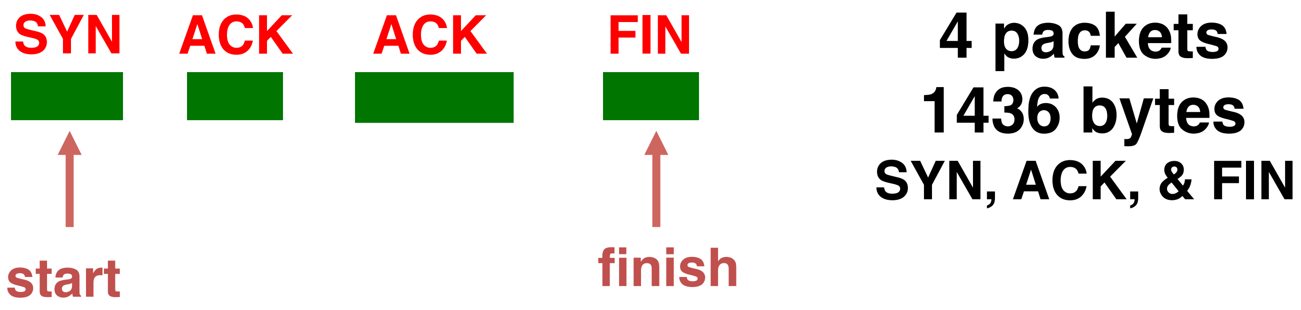
- Set of packets that "belong together"
  - Source/destination IP addresses and port numbers
  - Same protocol, ToS bits, …
  - Same input/output interfaces at a router (if known)
- Packets that are "close" together in time
  - Maximum spacing between packets (e.g. 30 sec)
  - E.g.: flows 2 and 4 are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

Flow Abstraction

- Not exactly the same as a "session"
  - Sequence of related packets may be multiple flows
  - Related packets may not follow the same links
  - "Session" is hard to measure from inside network
- Motivation for this abstraction
  - As close to a "session" as possible from outside
  - Router optimization for forwarding/access-control
  - … might as well throw in a few counters

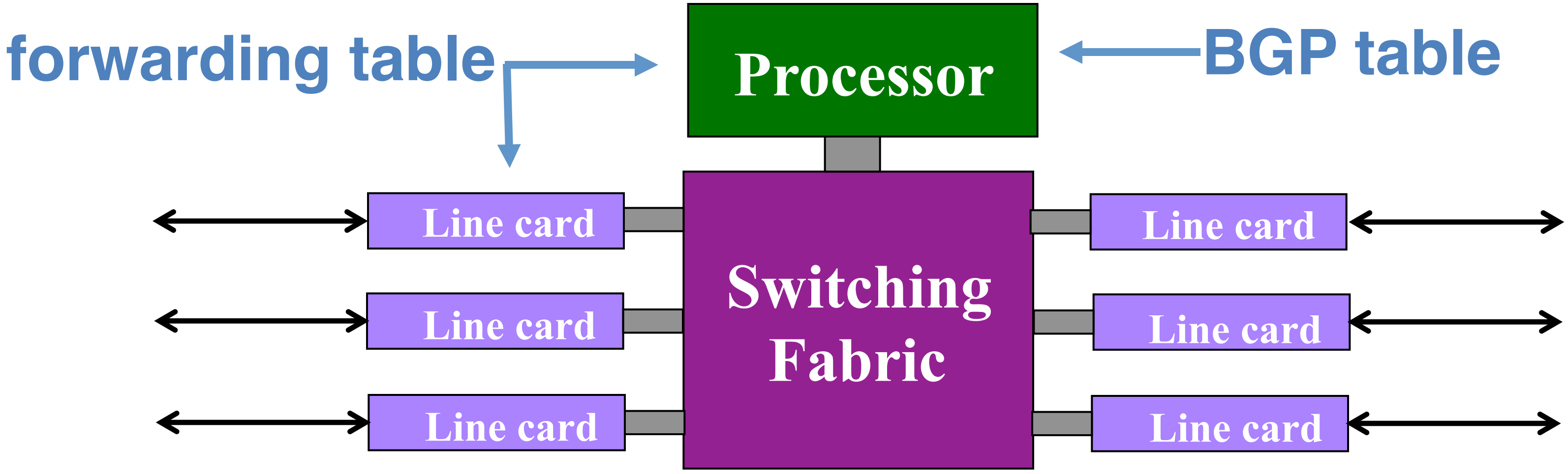
Traffic Statistics (e.g., Netflow)

- Packet header info
  - Source and destination addresses and port #s
  - Other IP & TCP/UDP header fields (protocol, ToS)
- Aggregate traffic information
  - Start and finish time (time of first & last packet)
  - Total # of bytes and number of packets in the flow
  - TCP flags (e.g., logical OR over sequence of packets)

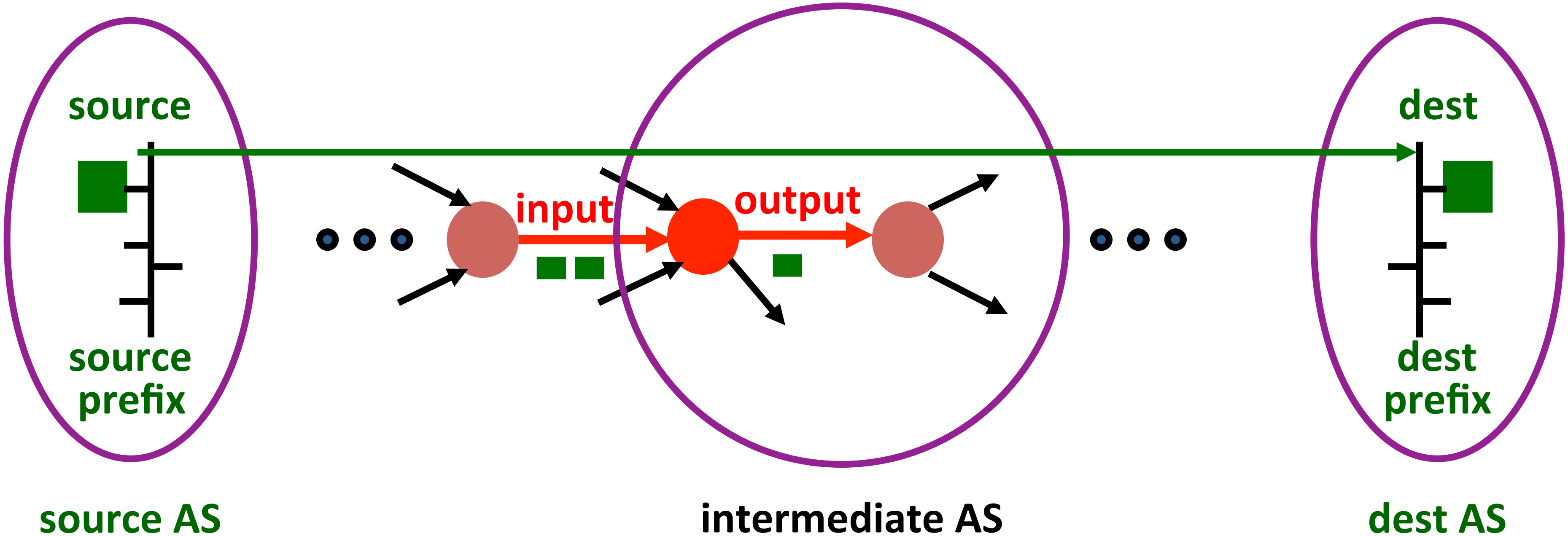


Recording Routing Information

- Input and output interfaces
  - Input interface is where packets entered the router
  - Output interface is "next hop" in forwarding table
- Source and destination IP prefix (mask length)
  - Longest prefix match on src and dest IP addresses



Measuring Traffic as it Flows By



Source and destination: IP header

Source and dest prefix: forwarding table or BGP table

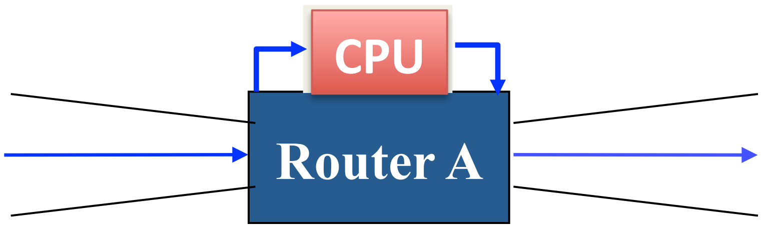
Source and destination AS: BGP table

Packet vs. Flow Measurement

- Basic statistics (available from both techniques)
  - Traffic mix by IP addresses, port numbers, protocol
  - Average packet size
- Traffic over time
  - Both: traffic volumes on medium-to-large time scale
  - Packet: burstiness of the traffic on a small time scale
- Statistics per TCP connection
  - Both: volume of traffic transferred over the link
  - Packet: frequency of lost or out-of-order packets

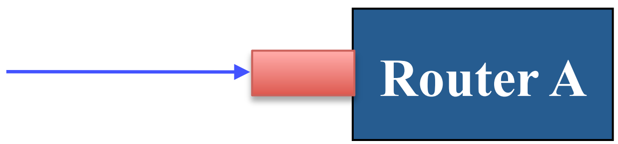
Collecting Flow Measurements

Route CPU that generates flow records



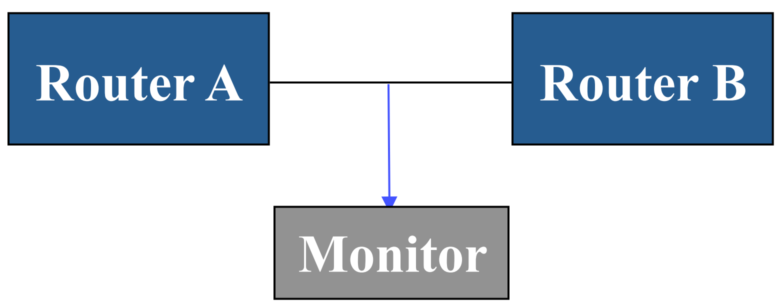
…may degrade forwarding performance

Line card that generates flow records



…more efficient to support measurement in each line card

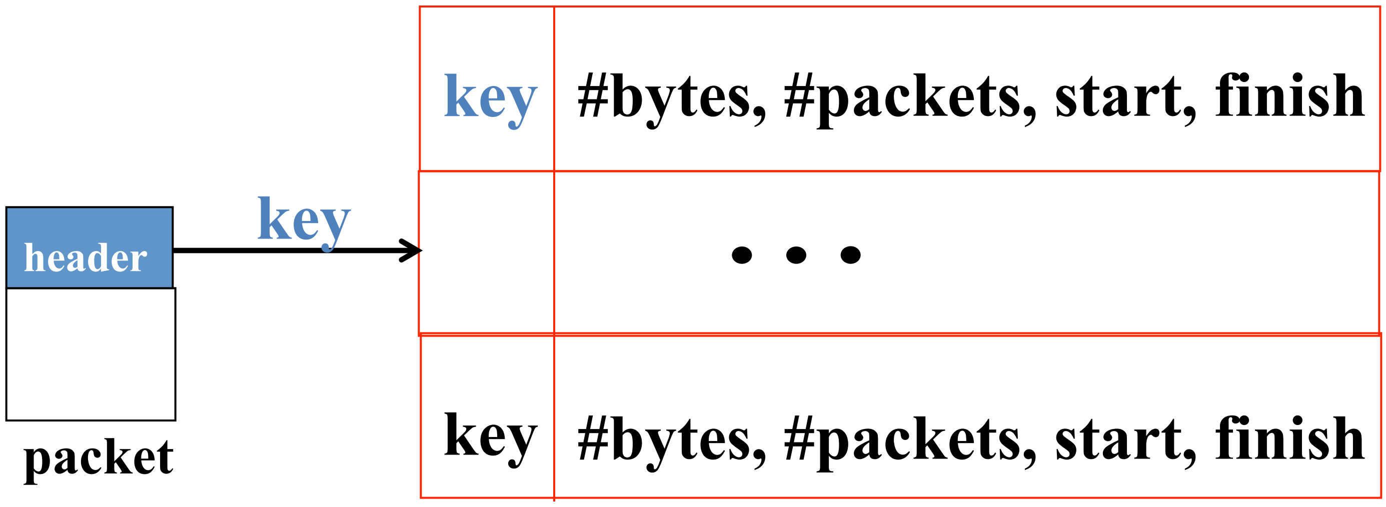
Packet monitor that generates flow records



…third party

Mechanics: Flow Cache

- Maintain a cache of active flows
  - Storage of byte/packet counts, timestamps, etc.
- Compute a key per incoming packet
  - Concatenation of source, destination, port #s, etc.
- Index into the flow cache based on the key
  - Creation or updating of an entry in the flow cache



Mechanics: Evicting Cache Entries

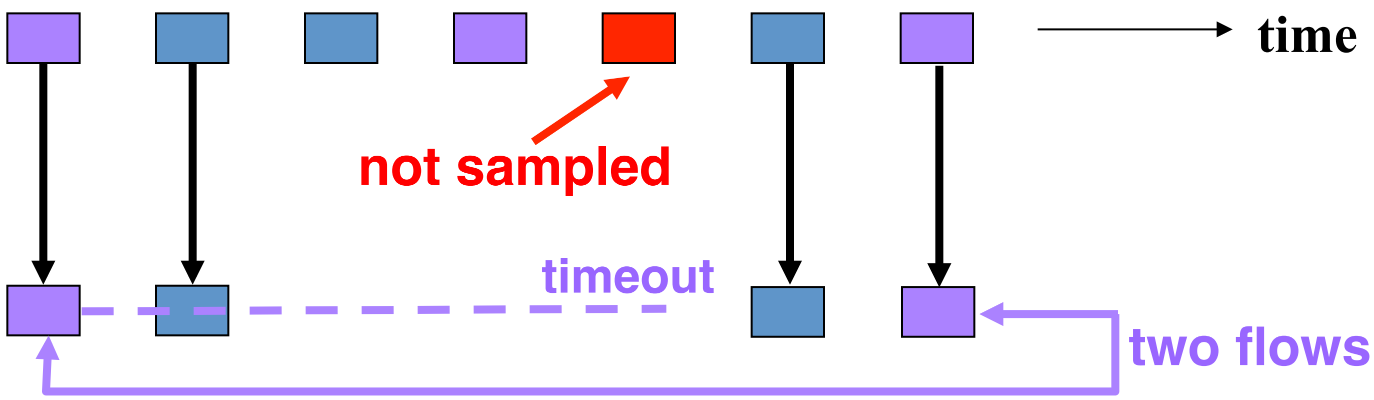
- Flow timeout
  - Remove flows not receiving a packet recently
  - Periodic sequencing to time out flows
  - New packet triggers the creation of a new flow
- Cache replacement
  - Remove flow(s) when the flow cache is full
  - Evict existing flow(s) upon creating a cache entry
  - Apply eviction policy (LRU, random flow, etc.)
- Long-lived flows
  - Remove flow(s) persisting a long time (e.g., 30 min)

Measurement Overhead

- Per-packet overhead
  - Computing the key and indexing flow cache
  - More work when the average packet size is small
  - May not be able to keep up with the link speed
- Per-flow overhead
  - Creation and eviction of entry in the flow cache
  - Volume of measurement data (# of flow records)
  - Larger # of flows when # packets per flow is small
  - May overwhelm system collecting/analyzing data

Sampling: Packet Sampling

- Packet sampling before flow creation
  - 1-out-of-m sampling of individual packets
  - Creation of flow records over the sampled packets
- Reducing overhead
  - Avoid per-packet overhead on 1 – (1/m) packets
  - Avoid creating records for many small flows



BGP Monitoring

Motivation for BGP Monitoring

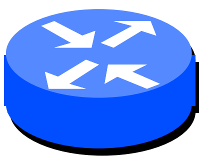
- Visibility into external destinations
  - What neighboring ASes are telling you
  - How you are reaching external destinations
- Detecting anomalies
  - Increases in number of destination prefixes
  - Lost reachability or instability of some destinations
- Input to traffic-engineering tools
  - Knowing the current routes in the network
- Workload for testing routers
  - Realistic message traces to play back to routers

BGP Monitoring: A Wish List

- Ideally: know what the router knows
  - All externally-learned routes
  - Before applying policy and selecting best route
- How to achieve this
  - Special monitoring session on routers that tells everything they have learned
  - Packet monitoring on all links with BGP sessions
- If you can't do that, you could always do…
  - Periodic dumps of routing tables
  - BGP session to learn best route from router

Using Routers to Monitor BGP

Talk to operational routers using SNMP or telnet at command line

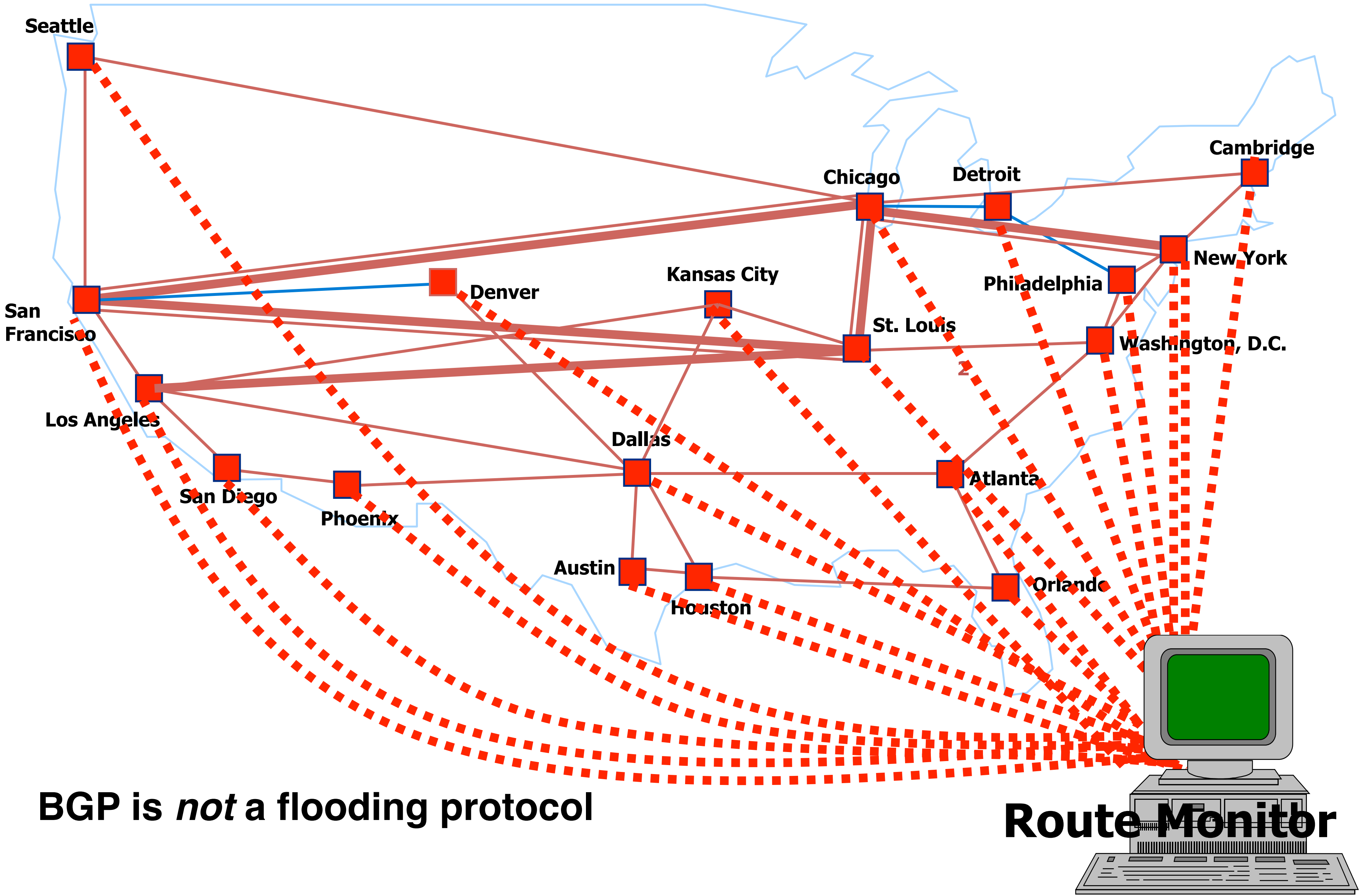


Establish a “passive” BGP session from a workstation running BGP software

- (-) BGP table dumps are expensive
- (+) Table dumps show all alternate routes
- (-) Update dynamics lost
- (-) Restricted to interfaces provided by vendors

- (+) BGP table dumps do not burden operational routers
- (-) Receives only best route from BGP neighbor
- (+) Update dynamics captured
- (+) Not restricted to interfaces provided by vendors

Collect BGP Data From Many Routers



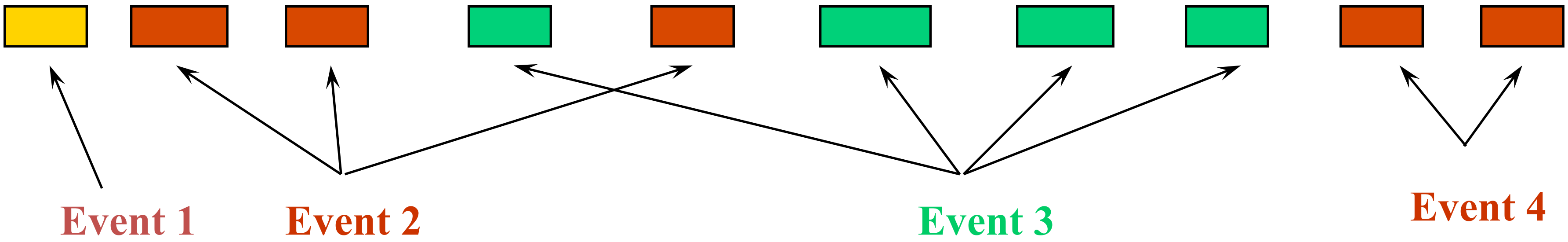
BGP is *not* a flooding protocol

BGP Table ("show ip bgp" at RouteViews)

	Network	Next Hop	Metric	LocPrf	Weight	Path
*	3.0.0.0	205.215.45.50			0	4006 701 80 i
*		167.142.3.6			0	5056 701 80 i
*		157.22.9.7			0	715 1 701 80 i
*		195.219.96.239			0	8297 6453 701 80 i
*		195.211.29.254			0	5409 6667 6427 3356 701 80 i
*>		12.127.0.249			0	7018 701 80 i
*		213.200.87.254	929		0	3257 701 80 i
*	9.184.112.0/20	205.215.45.50			0	4006 6461 3786 i
*		195.66.225.254			0	5459 6461 3786 i
*>		203.62.248.4			0	1221 3786 i
*		167.142.3.6			0	5056 6461 6461 3786 i
*		195.219.96.239			0	8297 6461 3786 i
*		195.211.29.254			0	5409 6461 3786 i

AS 80 is General Electric, AS 701 is UUNET, AS 7018 is AT&T
AS 3786 is DACOM (Korea), AS 1221 is Telstra

BGP Events



- Group of BGP updates that "belong together"
  - Same IP prefix, originating AS, or AS_PATH
- Updates that are "close" together in time
  - Maximum spacing between packets (e.g. 30 sec)
  - E.g.: events 2 and 4 are separated in time

Conclusions

- Measurement is crucial to network operations
  - Measure, model, control
  - Detect, diagnose, fix
- Network measurement is challenging
  - Large volume of measurement data
  - Multi-dimensional data
- Great way to understand the Internet
  - Popular applications, traffic characteristics
  - Internet topology, routing dynamics