Network Measurement: Measuring the Path

Joy Zhang

Measuring the Path

Available Bandwidth/Bottleneck
- BFind,Pathchar,Cartouche
- Pathneck

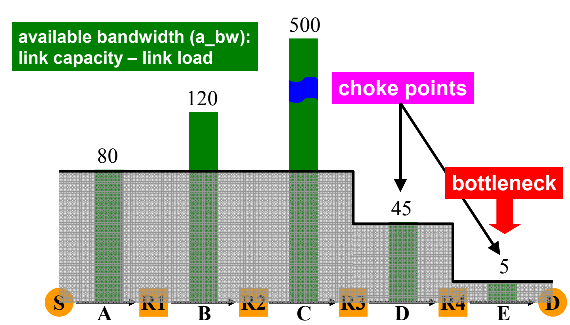
Link Capacity:
- Pathchar
- CapProbe

Loss/Delay/Re-ording
- Tulip

Pathneck

Pathneck is active probing tool, measuring bottleneck bandwidth:
- Low overhead (i.e., in order of 10s-100s KB)
- Fast (i.e., in order of seconds)
- Single-end control
- High accuracy

Bottleneck & Available Bandwidth



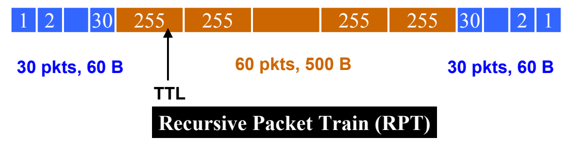
Available Bandwidth Estimation

Packet train probing

- train_rate > a_bw ➔ train_length increases
- train_rate ≤ a_bw ➔ train_length keeps same

Locating bottlenecks needs the packet train length info from each link

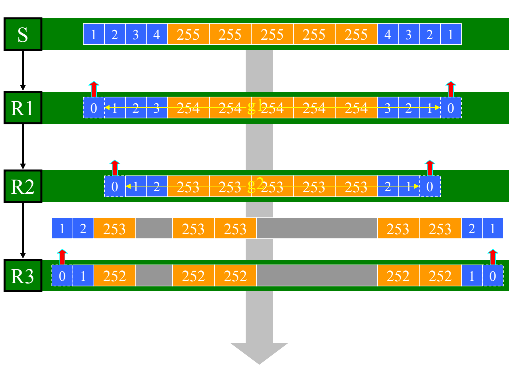
Probing Packet Train in Pathneck



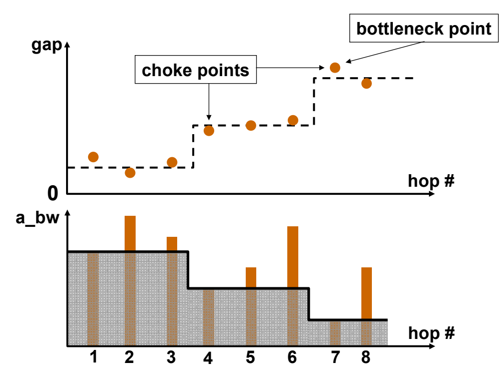
- Load packets are used to measure available bandwidth
- Measurement packets are used to obtain location information

Transmission of RPT

gap values are the raw measurement



Choke Point Detection



Configuration Parameters

Confidence Threshold (conf)

- Set the minimum step change in the step function
- To filter out the gap measurement noise
- Default: conf ≥ 10% available bandwidth change

Detection Rate (d_rate)

- N probings for each destination
- A hop must appear as a choke point for at least M times (d_rate ≥ M/N)
- To select the most frequent choke point
- Default: d_rate ≥ 5/10 = 50%

Patheneck: the Algorithm

1. Probe the same destination 10 times

2. conf ≥ 10% filtering
   - For each probing, only pick the choke points which satisfy conf ≥ 10% threshold

3. d_rate ≥ 50% filtering
   - A hop must appear as a choke point in at least 5 times to be selected

4. The last choke point is the bottleneck

Accuracy Evaluation

Location measurement accuracy

- Abilene experiments
- Testbed experiments on Emulab (U. of Utah)
  - Construct different types of bottleneck scenarios using real traffic trace

Bandwidth estimation accuracy

- Internet experiments on RON (MIT)
  - Compare with IGI/PTR/Pathload

Accuracy Evaluation Results

Location measurement accuracy (on Emulab)

- 100% accuracy for capacity determined bottlenecks
- 90% accuracy for load determined bottlenecks, mainly due to the dynamics of competing load
- At most 30% error with reverse path congestion

Bandwidth estimation accuracy (on RON)

- Pathneck returns upper bound for the bottleneck available bandwidth
- On RON: consistent with available bandwidth estimation tools

Properties

✓ **Low overhead**
- 33.6KB each probing

✓ **Fast**
- 5 seconds for each probing
- (1-2 seconds if RTT is known)

✓ **Single end control**

✓ **Over 70% of accuracy**

Limitations

✗ **Can not measure the last hop**
- ✓ Fixed recently (use ICMP ECHO packets for the last hop)

✗ **ICMP packet generation time and reverse path congestion can introduce measurement error**
- They directly change the gap values
- Considered as measurement noise

✗ **Packet loss and route change will disable the measurements**
- Multiple probings can help

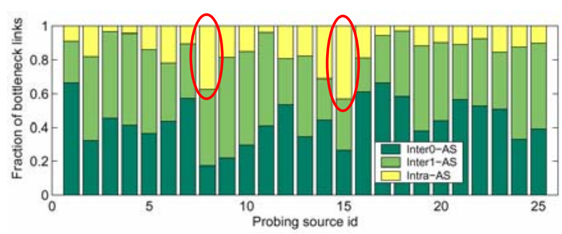
✗ **Can not pass firewalls**
- Similar to most other tools

1. Bottleneck Distribution

Common Assumption: bottlenecks are most likely to appear on the peering and access links, i.e., on Inter-AS links

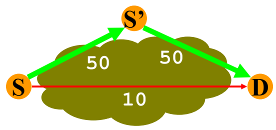
Identifying Inter/Intra-AS links
- Only use AS# is not enough (Mao et al [SIGCOMM03])
- We define Intra-AS links as links at least one hop away from links where AS# changes
- Two types of Inter-AS links: Inter0-AS & Inter1-AS links
- We identify a subset of the real intra-AS links

1. Bottleneck Distribution (cont.)



Up to 40% of bottleneck links are Intra-AS
- Consistent with earlier results [Akella et al IMC03]

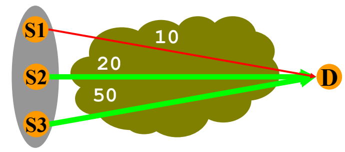
3. Avoidance — Overlay Routing



Useful metric: the estimated bandwidth on S-S'-D is larger than those on S-D

53% of 63,440 overlay attempts are useful

3. Avoidance — Multihoming



Method

- **Use multiple sources in the same region to simulate multihoming**
- **Useful metric: if the bandwidth on the worst path can be improved by at least 50% by all other sources**

78% of 42,285 multihoming attempts are useful

Tulip

Pinpoint location of network faults
- **Packet loss**
- **Reordering**
- **Significant queuing**

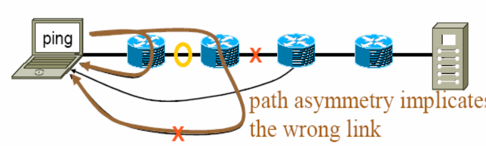
Real time diagnosis of faults

From the end user's perspective
- **More powerful than traceroute and ping**

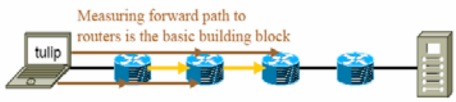
No special privileges locally or on routers

Limitation with existing diagnosis tools

- SNMP stats are limited in scope
  - can be used only inside your domain
  - granularity issues
- *ping and traceroute*-like tools don't deal with path asymmetry



Overview of tulip



- Localizes reordering, queuing and loss (so far)
  - single-ended: works from a host to an arbitrary IP address
- Infers link properties by subtracting path properties
  - path to router should be a prefix of the end-to-end forward path

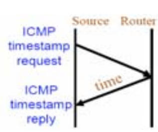
Tools used by tulip

OOB Probes
- TTL and payload configurable
- Approximates application packets

ICMP timestamp request
- Get a router's receive and send timestamps
- Optional feature according to RFC

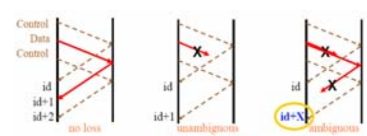
IP-ID
- Identify order in which packets were sent
- ID's may not be sequential or even monotonically increasing

Queuing on the forward path



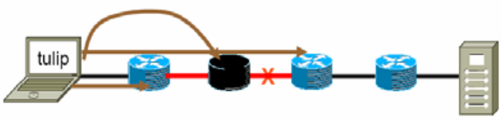
- Send ICMP timestamp requests at a constant rate
- Find the rate at which they arrive at the router
- Deviations are due to queuing; use the median

Loss on the forward path



- Checks losses in the forward direction
- Concludes loss only if second probe invokes id+1
- Check packet may check for ICMP rate-limiting
- Multiple loss scenarios not considered
- Breaks if data packet is fragmented
- Breaks for multiple active probes

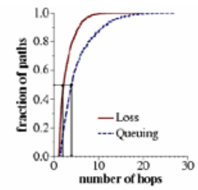
Diagnosis granularity of tulip

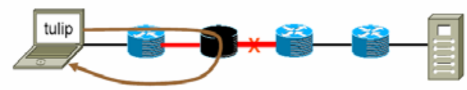


- Granularity: uncertainty in the location of the fault
  - when a router does not support the required features
  - when probes take a non-prefix path to a router

Diagnosis granularity of tulip (2)

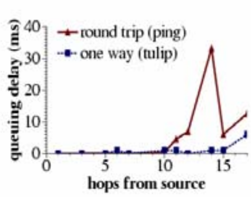
- Median is 2 hops for loss and 4 hops for queuing
  - ICMP timestamp probes do not have the prefix path property





- Round trip probing can further improve diagnosis granularity

Consistency along the path (queuing)



median queuing delay to intermediate routers in an example path

- Tulip's one-way measurements are consistent
- Round trip measurements are polluted by reverse path conflation

Better timing information

- Problems:
  - timing information is separate from flow counters
  - ICMP timestamps require directly addressing the router
    - routing issues reduces their value
- Simple fix: timestamp TTL-expired messages
  - backwards compatible, incrementally deployable
    - use 32 unused bits in the TTL-expired messages

Better counter support

- Problem:
  - IP-ID is a shared counter
    - what if all of you start using tulip?
    - the architecture suggests per-flow counters
- Simple fix: maintain N (constant) counters
  - hash source address and probe IP-ID to pick the counter
  - backwards compatible, incrementally deployable (today, N=1)

Conclusions

- **Localizes first faulty node to within ~3 hops**
- **Detects some packet reordering**
- **Faster than ping and traceroute**
- **How useful are the results given the accuracy and granularity?**
- **Conclude that loss and queuing occur close to the destination**
- **Suggest that routers implement ICMP timestamps and per flow counters**