Verification of Zebra as a BGP Measurement Instrument

Or

Can You Make Accurate Measurements With A Length of String

All Work and Experimentation done by
Hongwei Kong
Agilent Labs, China
hong-wei_kong@agilent.com

Presented by
Lance Tatman
Agilent Labs, US
lance_tatman@agilent.com

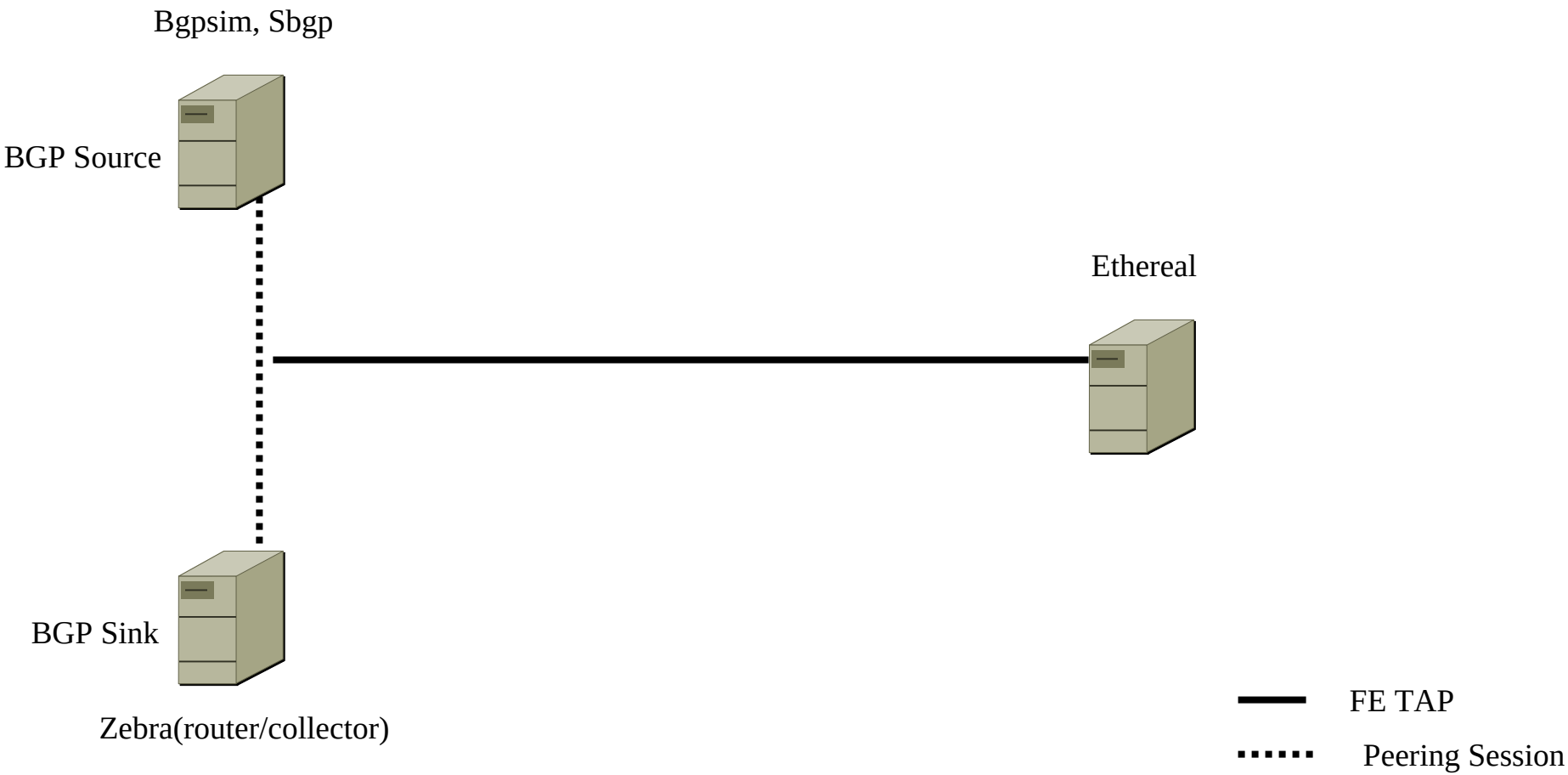
Should You Believe What You See

- Zebra is in use at RIPE and Oregon RouteViews as a BGP message recorder

- We, the research community, have been using BGP data recorded by Zebra for analysis of BGP behavior for several years now

- How good is the Zebra Data?

Our Method to Test for Truth



First We Verify route_btoa

- Method: Send known BGP data across wire. Record-Decode-Verify
  - Tested on Linux and Solaris with different results
- **Here's Why**
- When multi-protocol NLRI reachable/unreachable attribute present for IPv6 prefixes route_btoa cannot decode correctly
  - Interesting these messages were only observed on rrc03 (AMS-IX).
  - route_btoa can support this but support tied to capabilities of the kernel during compilation. Checks for kernel IPv6 support.
- When multi-protocol NLRI reachable/unreachable attribute present for IPv4 multicast prefixes route_btoa cannot decode correctly
  - Interesting we didn't see any of these on any of the RIPE systems
  - Turns out route_btoa does support this, but it is tied to capabilities of the kernel during compilation. Checks for kernel multicast routing support

While Verifying route_btoa We Found A Couple of Odd Things With Zebra…First

- Some, but not all BGP "OPEN" messages are saved by Zebra in an alternative format, a format not recognized by route_btoa- reason is unknown
  - This does not occur on Zebra-to-Zebra sessions, but does occur on Zebra-to-bgpsim and Zebra-to-sbgp sessions. Observed in RIPE data.
  - AS and IP addresses, both source & destination are recorded as 0. Causes route_btoa to decode message as NULL.
- Zebra should save in correct format and/or route_btoa should support irregular dump headers

While Verifying route_btoa We Found A Couple of Odd Things With Zebra…Second

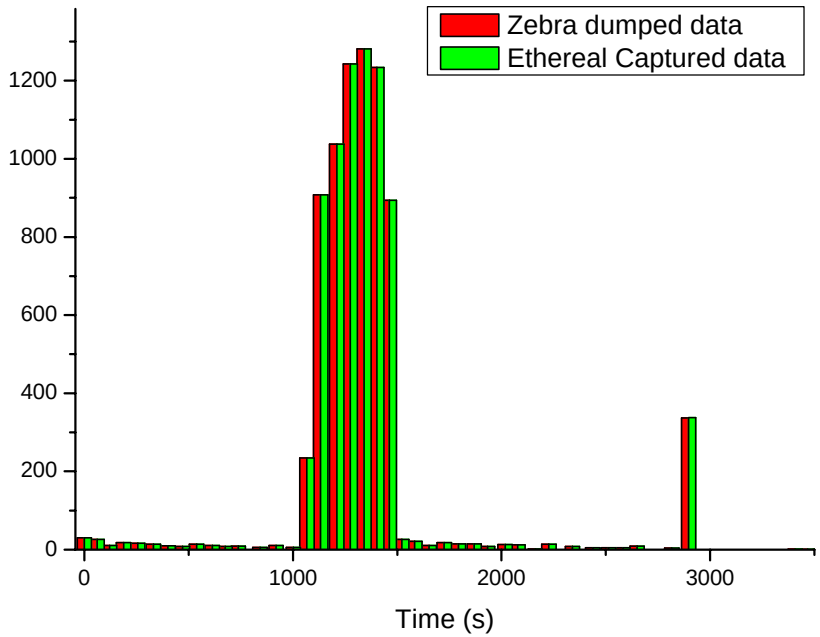
- **Very Large BGP Messages are Incompletely Saved by Zebra**
  - In fact, this happens with all messages we observed with a length field of 4096 bytes
- **Here is why**
  - Zebra dump module buffer size is bgp-max-packet-size(4096Bytes) + bgp-dump-header-size(12bytes)
  - Zebra dump module does not take into account bgp-dump-message-header
    - Includes things like: source & destination AS, Interface index, Address Family, IP addresses
  - Zebra Bug Fixed by adding 40 bytes to buffer

Significance of Zebra's Incomplete Saving of Messages with Header Field Length 4096

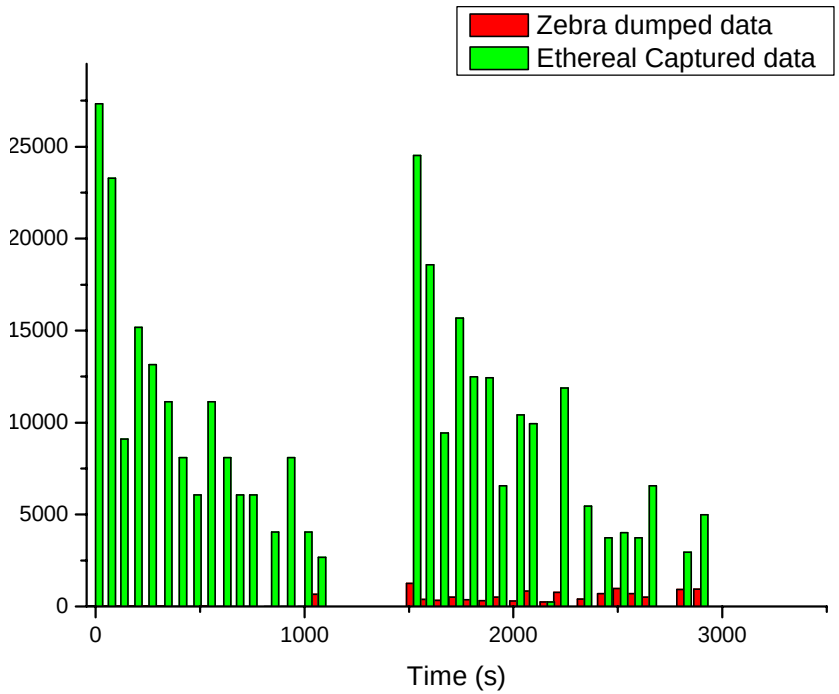
File	# of Prefixes With Zebra bug		# of Prefixes Without Zebra Bug		% of Prefixes not Decoded by route_btoa		
	#A	#W	#A	#W	A	W	Total
Routeview-20021219.2022	220543	3370	225174	119284	2%	97.2%	35%
Routeview-20021219.2037	125850	1821	128343	117669	2%	98.5%	48.1%
Routeview-20021219.2052	259489	5806	265288	121775	2%	95.2%	31.5%
Routeview-20021219.2107	129341	1396	131777	19045	2%	92.7%	13.3%
RRC03-20030106.0930	107730	3278	107730	87850	0%	96.3%	43.2%
RRC03-20030131.0230	134127	6180	134127	112542	0%	94.5%	43.1%

Loss of Prefixes Due to Incompletely Captured BGP Messages:

Count of Messages Recorded



Count of Prefixes Recorded

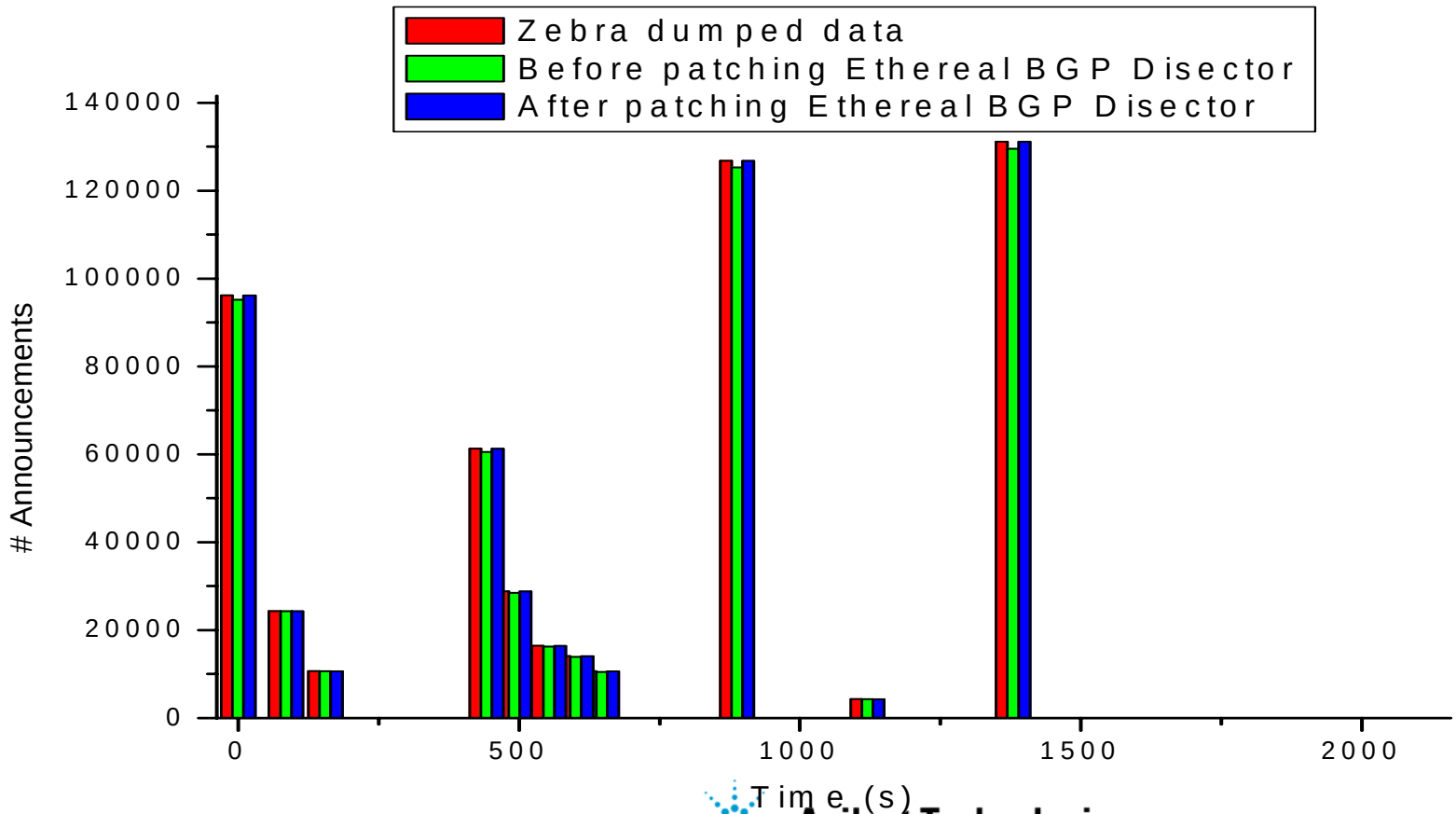


Who Watches the Watchers

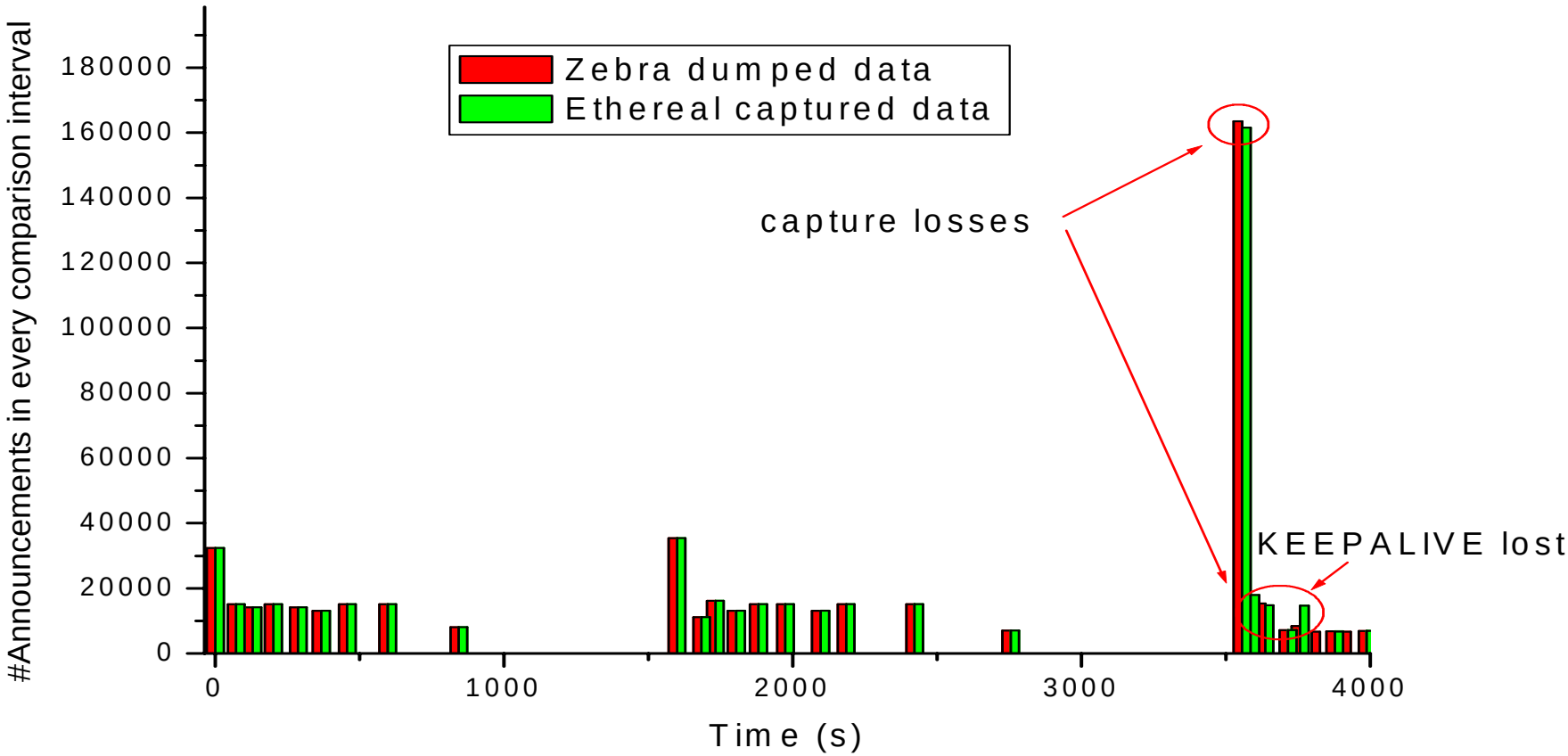
Finding a Bug in the Ethereal BGP Dissector

- Using fixed version of Zebra we compared on-wire observations with Zebra dumps using a known data stream
- Zebra matched the known stream but data obtained from Ethereal using the Ethereal BGP Dissector contained fewer Announcements than expected
- **Here's Why**
  - If a BGP message header spans two TCP segments then it is not recognized by BGP Dissector and is not decoded
- Bug reported and fixed in version 0.9.12 of Ethereal

Ethereal BGP Dissector Bug for Cross TCP Segment BGP Messages:



Overcoming Limitations of libpcap



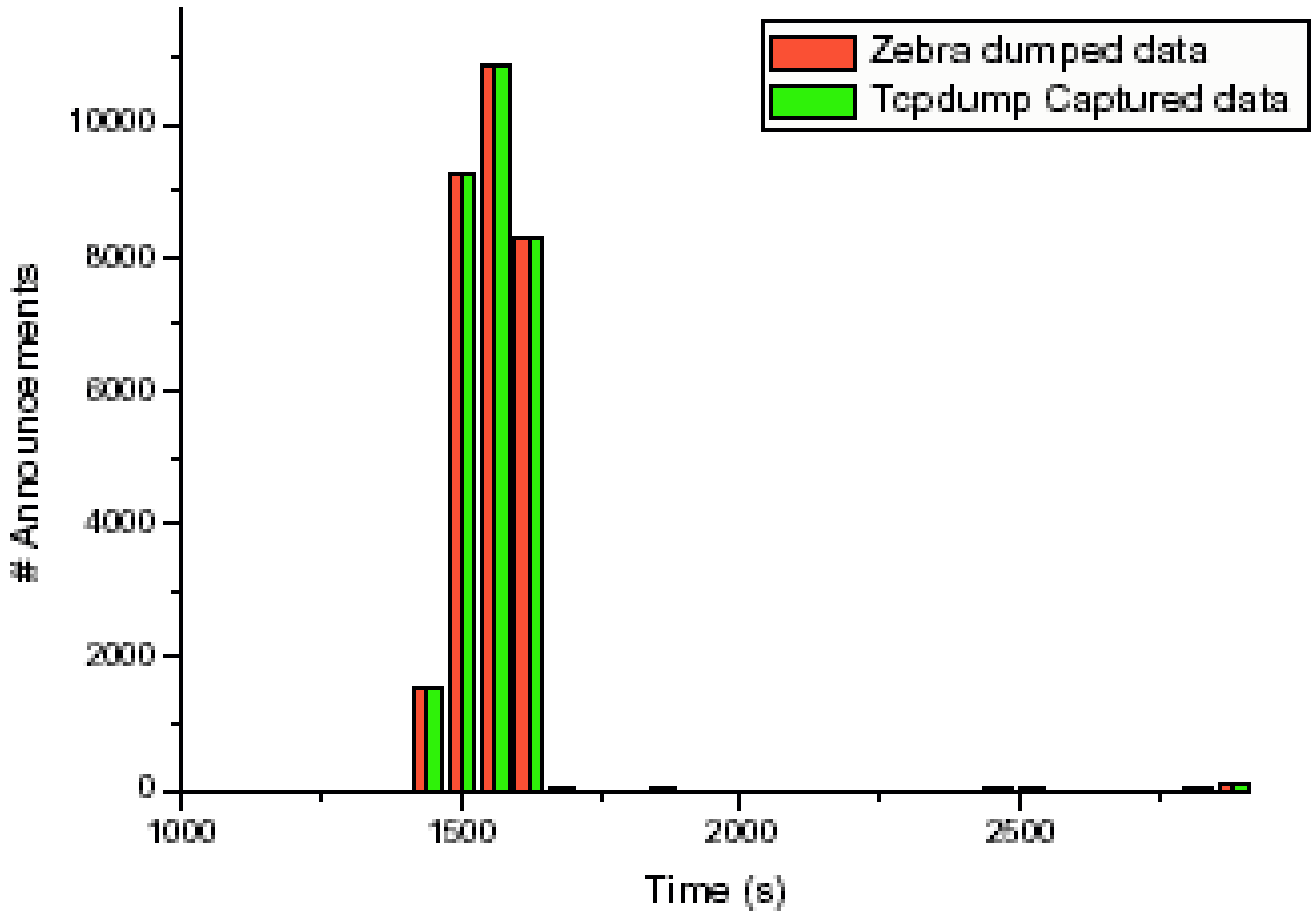
Overcoming Limitations of libpcap

- We observed losses in libpcap under heavy load
- **Here's Why**
  - Queue overflows in libpcap ver.0.7.2
  - Libpcap 0.8.030314
    - allows network adapter to directly capture to system memory
    - Implements large ring queue in system memory
- Rebuilt Ethereal with libpcap 0.8.030314
  - All loss was eliminated

More Problems With BGP Dissector

- **Next we introduced TCP segment losses using NIST Net**
  - Using BGP Dissector to reconstruct the session we found "extra" BGP messages.
  - Problem was reported to Ethereal developers
  - As of Ethereal ver 0.9.12 problem is still not fixed
- **Consequences**
  - Pay attention particularly when evaluating multi-hop BGP sessions reconstructed using BGP Dissector

Finally They Match-Most Bugs Fixed, Others Avoided



Other Zebra Issues Of Concern to Researchers

- **Timestamps don't reflect on-the-wire times**
  - Caused us to need to use keep-alives as synchronization markers
- **Missed keep-alives**
  - Causes session to break and retransmit of full table
- **Records only inbound BGP messages**
  - Miss outbound NOTIFICATION messages
- **Sends NOTIFICATION messages which break session**
- **10+ Second recording dead time after session reset**
- **Amount/complexity of code is overkill- only need a recorder**

Summary

- **Verified the the behaviors of the tools used to process Zebra BGP data files.**
  - revised these tools and solved the problems found
- **Explored the consistency of Zebra BGP data collections**
  - Found bugs in Zebra
- **Verified Zebra BGP data collecting module**
  - Without BGP session break, Zebra collects BGP data consistently
  - During session break, Zebra BGP data may not be consistent with on-wire captured data
  - Zebra can delay sending KEEPALIVE messages to the peer when there is heavy BGP traffic and result in session break and corrupted data.
  - Zebra Data capturing is delayed when there is heavy BGP traffic

More Information

- The full report is available on the RIPE RIS analysis page
  - http://www.ripe.net/ripencc/pub-services/np/ris/analysis.html

- Hong-Wei Kong
  - hong-wei_kong@agilent.com

- We are developing a BGP recording instrument and would like your suggestions on features and requirements