Pingbeat:
Y’know, for pings!

Joshua Rich
Technical Support Engineer
February 2016
Ping theory

1. Source creates an ICMP *echo-request* and sends this to the target.
   a. Contains an *identifier* and *sequence number* to keep track of this specific ping request
2. Source records the timestamp of when the *echo-request* was sent.
3. Target receives the source *echo-request* and creates their own ICMP *echo-reply*, sending this back to the source.
   a. Contains the *identifier* and *sequence number* in addition to a timestamp of when the message was sent back.
4. Source receives the *echo-reply* and calculates Round-Trip Time (RTT) based on recorded timestamps.

What happens if the target doesn’t respond?

- Requests are retried after a configured timeout period.
- After configured number of retries, source gives up and records packet loss.
ICMP echo-request in Wireshark
ICMP *echo-reply* in Wireshark

Frame 619: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0


Version: 4
Header Length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
Total Length: 84
Identification: 0x0000 (0)
Flags: 0x00
Fragment offset: 0
Time to live: 58
Protocol: ICMP (1)
Header checksum: 0x5935 [validation disabled]
Source: 216.58.220.99 (216.58.220.99)
Destination: 192.168.178.45 (192.168.178.45)
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]

Internet Control Message Protocol
Type: 0 (Echo (ping) reply)
Code: 0
Checksum: 0xaf5f [correct]
Identifier (BE): 28646 (0x50a8)
Identifier (LE): 41046 (0xa056)
Sequence number (BE): 0 (0x0000)
Sequence number (LE): 0 (0x0000)
[Request Frame: 618]
[Response time: 10.174 ms]

Data (56 bytes)
Data: 0000000000000000000000000000000000000000000000000000000000000000...
[Length: 56]
**Inspiration**

SmokePing - the venerable goto network monitor in NOCs...

---

**O+P AG Servers**

- **Octopus**
  - Median RTT from janes: 14.3 ms av md
  - Median RTT from mipsrvOL: 14.2 ms av md
  - Median RTT from otoro: -nan s av md

- **Router UPC Aarweg**
  - Median RTT from janes: 15.1 ms av md
  - Median RTT from mipsrvOL: 14.7 ms av md
  - Median RTT from otoro: -nan s av md

- **Router Bächleinstr**
  - Median RTT from janes: 7.5 ms av md
  - Median RTT from mipsrvOL: 9.3 ms av md
  - Median RTT from octopus: 19.8 ms av md

---

Copyright Elastic 2015 Copying, publishing and/or distributing without written permission is strictly prohibited
A lightweight network monitoring probe

(based on)
What are beats?

- Beats are lightweight shippers that capture all sorts of operational data from your servers and ship it to Logstash and Elasticsearch.
- They use a common open-source platform, libbeat, that makes it easy to write your own beat.
- Written in Go.
- Designed to be small on memory and CPU.
Check out a demo at demo.elastic.co!
What is Pingbeat?

- Complement the packetbeat high-level application protocol analysis with more low-level network protocol metrics.
- Single binary, single YAML config file needed (easy to deploy).
- Supports any outputs that libbeats supports.
  - e.g., Elasticsearch, Logstash, Redis and file outputs.
- No need to install Logstash for each probe! Just one binary and one config file!
- Small memory footprint (10-15 MB RSS) and fast ping response.
- Ideally install in many places in your network to get a world-view of latency across the network.

Available now!

https://github.com/joshuar/pingbeat

go get github.com/joshuar/pingbeat
Pingbeat Dashboard

Pingbeat - Line Graph

Pingbeat - Table

Pingbeat - Percentiles

Pingbeat - Top 10 Lowest RTT

Pingbeat - Top 10 Highest RTT
Filtering to my LAN and ISP hosts...
Filtering to everything else...
The Netflix effect...
If you like Pingbeat, you may also like:

- **Unifedbeat**
  - https://github.com/cleesmith/unifiedbeat
  - *Index into ElasticSearch the alert records from network intrusion detection software.*

- **Nagioscheckbeat**
  - https://github.com/PhaedrusTheGreek/nagioscheckbeat
  - *Index Nagios checks into Elasticsearch*

- **Factbeat**
  - https://github.com/jarpy/factbeat
  - *Ship Facter facts to Elasticsearch*

- **Hsbeat**
  - https://github.com/YaSuenag/hsbeat
  - *Index JVM stats/metrics to Elasticsearch*
Tutorial: The Power of Open Data with ELK

Thursday 10:40am, D2.211