

# **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

The image shows the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Internals, and Help. Below the menu is a toolbar with various icons. The filter bar at the top shows the filter expression `icmp.type == 0 || icmp.type == 8`. The packet list pane displays two packets:

No.	Time	Source	Destination	Protocol	Length	Info
618	123.23309406	192.168.178.45	216.58.220.99	ICMP	98	Echo (ping) request id=0x50a0, seq=0/0, ttl=64 (reply in 619)
619	123.24326806	216.58.220.99	192.168.178.45	ICMP	98	Echo (ping) reply id=0x50a0, seq=0/0, ttl=58 (request in 618)

The packet details pane shows the expanded view of packet 618:

- Frame 618: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
- Ethernet II, Src: HonHaiPr\_bf:cb:9f (ac:d1:b8:bf:cb:9f), Dst: Avn\_71:f9:98 (34:31:c4:71:f9:98)
- Internet Protocol Version 4, Src: 192.168.178.45 (192.168.178.45), Dst: 216.58.220.99 (216.58.220.99)
  - 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: 0x51e8 (20968)
  - Flags: 0x02 (Don't Fragment)
  - Fragment offset: 0
  - Time to live: 64
  - Protocol: ICMP (1)
  - Header checksum: 0xc14c [validation disabled]
  - Source: 192.168.178.45 (192.168.178.45)
  - Destination: 216.58.220.99 (216.58.220.99)
  - [Source GeoIP: Unknown]
  - [Destination GeoIP: Unknown]
- Internet Control Message Protocol
  - Type: 8 (Echo (ping) request)
  - Code: 0
  - Checksum: 0xa75f [correct]
  - Identifier (BE): 20640 (0x50a0)
  - Identifier (LE): 41040 (0xa050)
  - Sequence number (BE): 0 (0x0000)
  - Sequence number (LE): 0 (0x0000)
  - [\[Response frame: 619\]](#)
- Data (56 bytes)
  - Data: 00...
  - [Length: 56]

The bottom status bar shows: Internet Protocol Version 4 (ip).... Packets: 59195 · Displayed: 2 (0.0%) Profile: Default

# ICMP *echo-reply* in Wireshark

The image shows a Wireshark network packet capture. The top toolbar includes menus (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Internals, Help) and various icons for packet capture and analysis. Below the toolbar, a filter bar shows the active filter: `icmp.type == 0 || icmp.type == 8`. The packet list pane displays two packets:

No.	Time	Source	Destination	Protocol	Length	Info
618	123.23309406	192.168.178.45	216.58.220.99	ICMP	98	Echo (ping) request id=0x50a0, seq=0/0, ttl=64 (reply in 619)
619	123.24326806	216.58.220.99	192.168.178.45	ICMP	98	Echo (ping) reply id=0x50a0, seq=0/0, ttl=58 (request in 618)

The packet details pane for packet 619 is expanded, showing the following structure:

- Frame 619: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
- Ethernet II, Src: Avm\_71:f9:98 (34:31:c4:71:f9:98), Dst: HonHaiPr\_bf:cb:9f (ac:d1:b8:bf:cb:9f)
- Internet Protocol Version 4, Src: 216.58.220.99 (216.58.220.99), Dst: 192.168.178.45 (192.168.178.45)
  - 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): 20640 (0x50a0)
  - Identifier (LE): 41040 (0xa050)
  - Sequence number (BE): 0 (0x0000)
  - Sequence number (LE): 0 (0x0000)
  - [\[Request frame: 618\]](#)
  - [Response time: 10.174 ms]
- Data (56 bytes)
  - Data: 00...
  - [Length: 56]

The bottom status bar shows: Internet Protocol Version 4 (ip)... Packets: 77247 · Displayed: 2 (0.0%) Profile: Default

## SmokePing Targets:

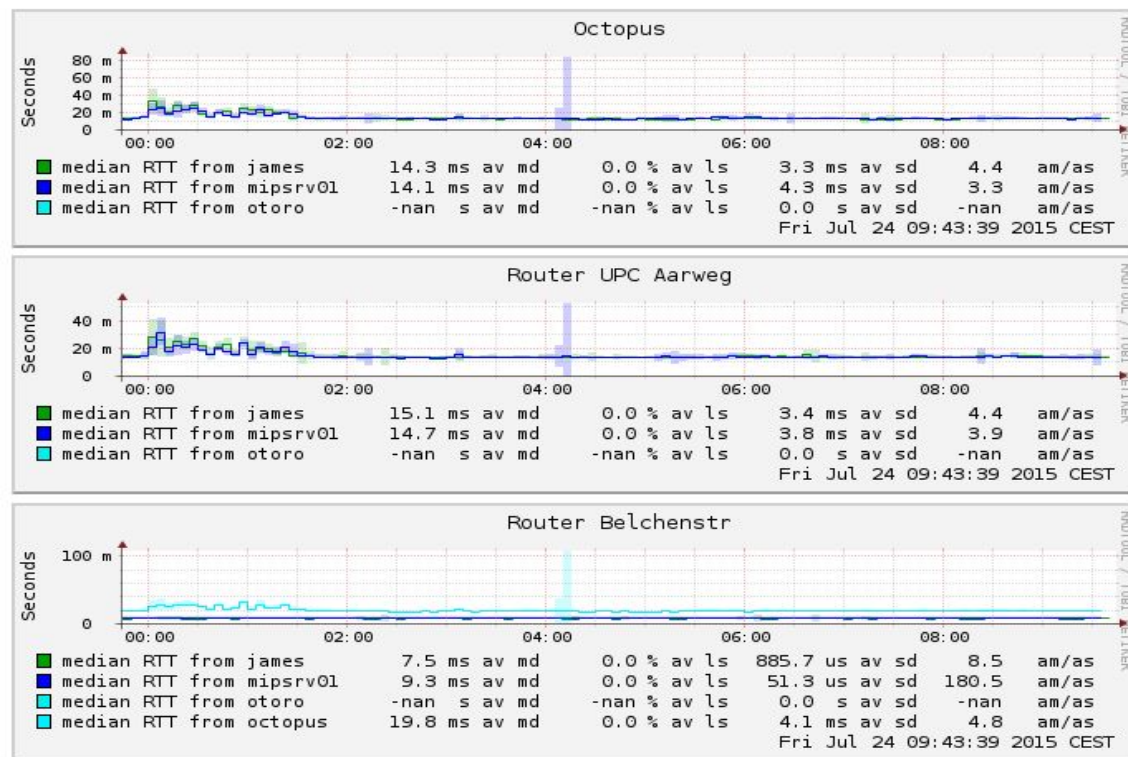
Hierarchy:

Default Hierarchy ▾

Filter:

- Charts
- O+P Managed
  - BCP
  - SwissContent
  - Horyzon
  - Floerli ccom
  - RingStreet
  - ESPROS Boxes
  - Ivic
  - ITIS Boxes
  - PNN
  - MOBIFLICK
  - MIP
  - O+P Internal
    - Octopus
    - Router UPC Aarweg
    - Router Belchenstr
    - James
    - Zimbox
    - Rigi
    - Parker
  - O+P Home
  - World
  - Root Name Servers
  - Multi Target Graphs
  - VoipGateway

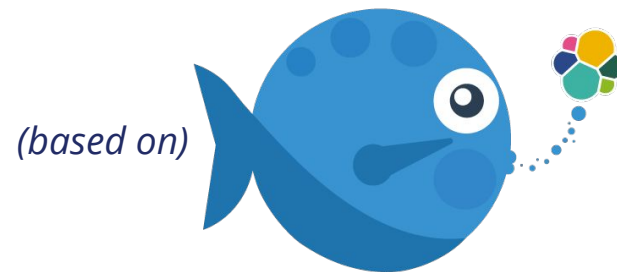
## O+P AG Servers



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

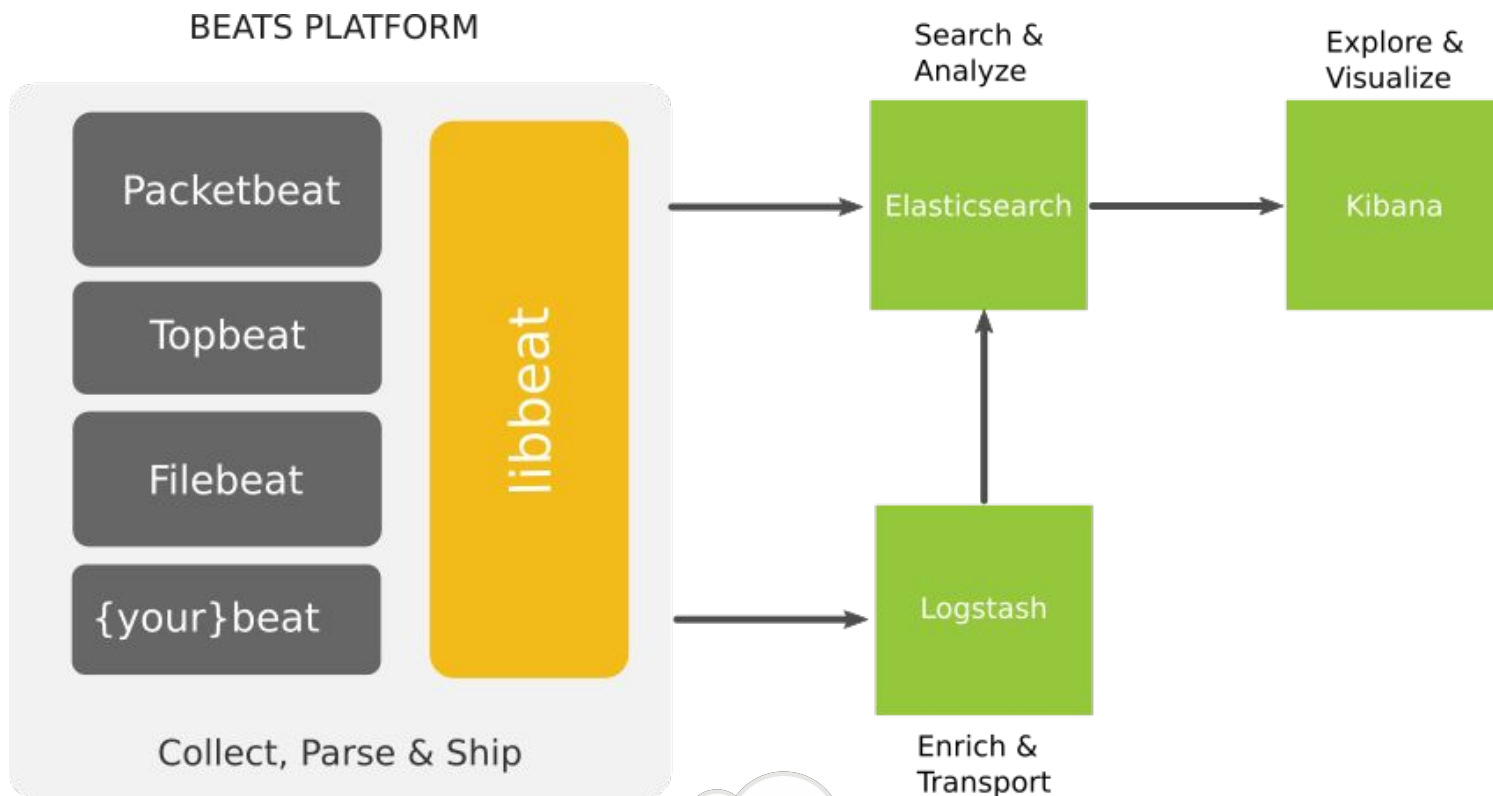
# Pingbeat

*A lightweight network monitoring probe*

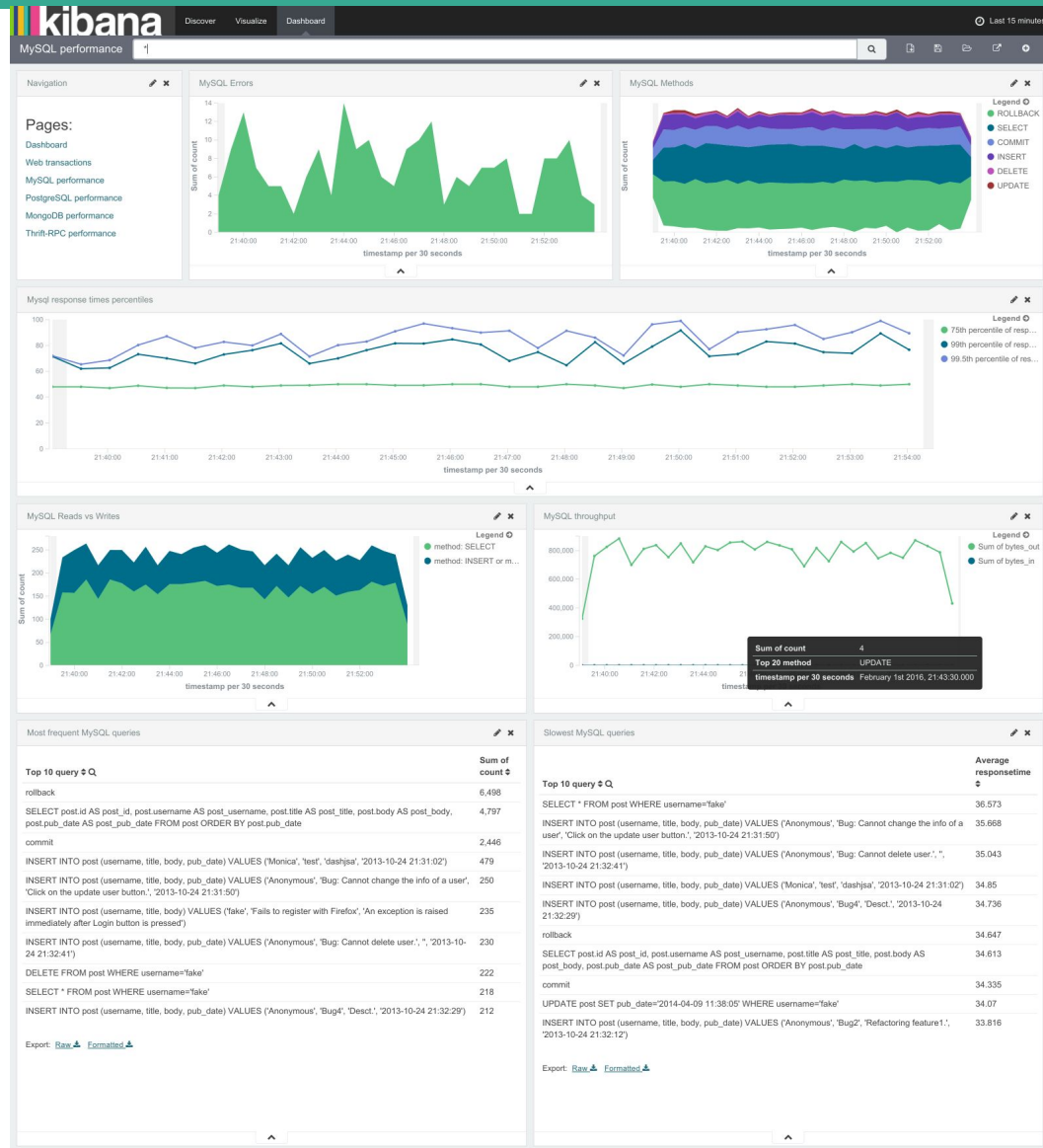


# What are beats?

- 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.

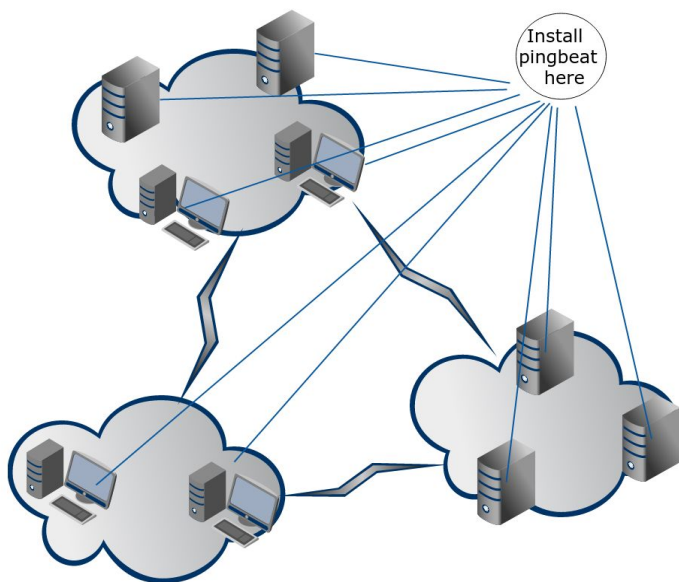


# Check out a demo at [demo.elastic.co!](http://demo.elastic.co!)



# 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.

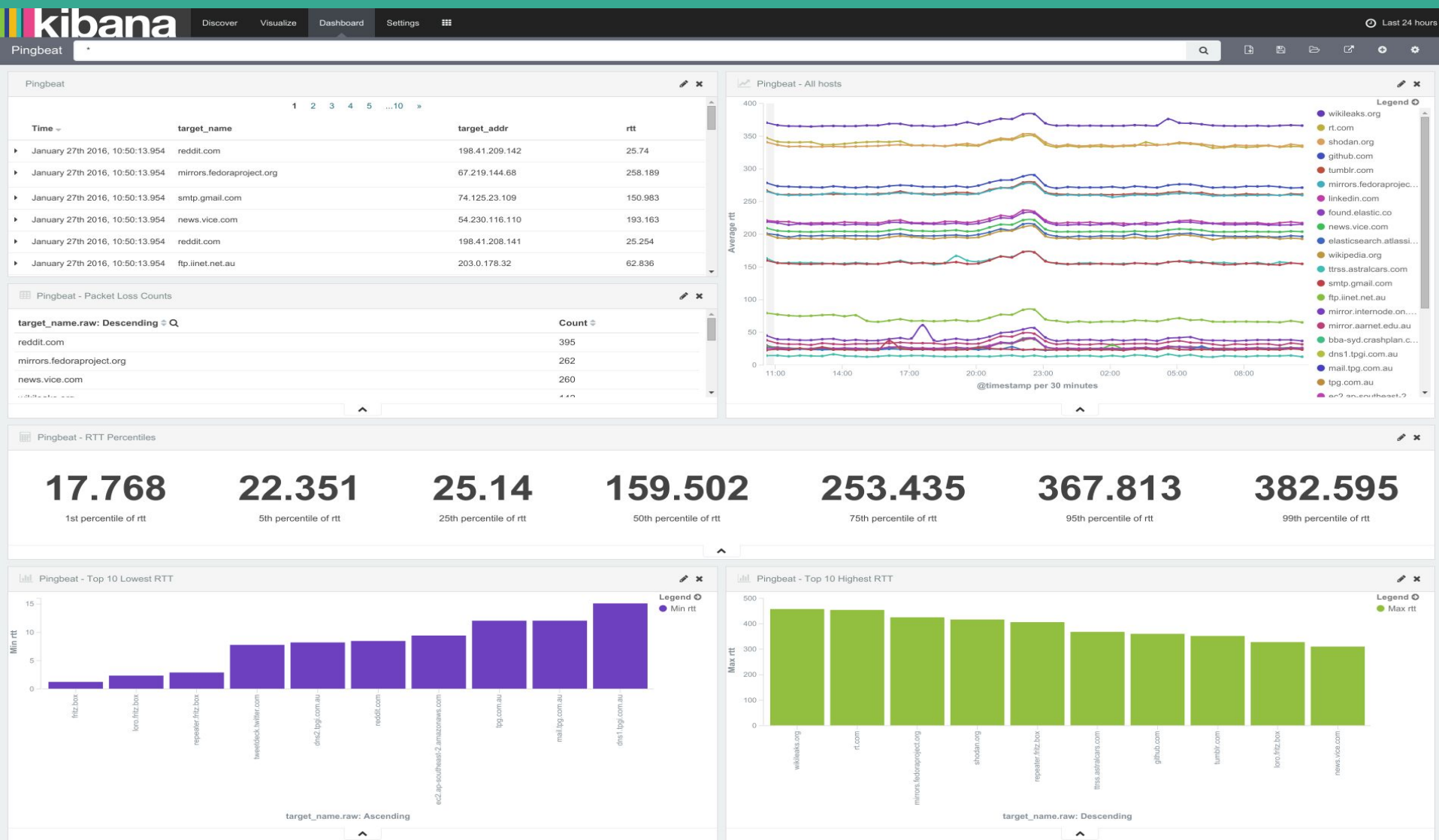


**Available now!**

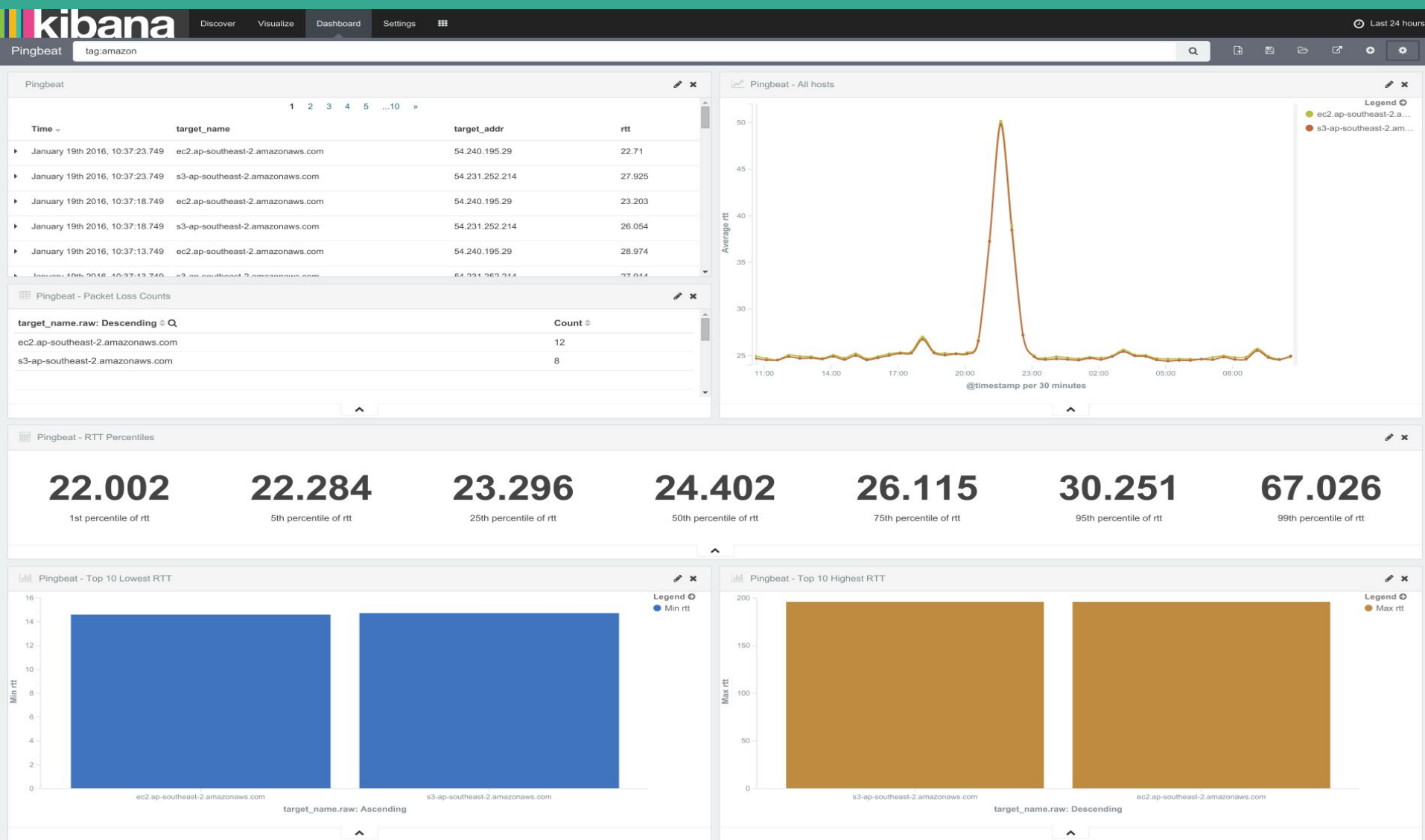
<https://github.com/joshuar/pingbeat>

**go get [github.com/joshuar/pingbeat](https://github.com/joshuar/pingbeat)**

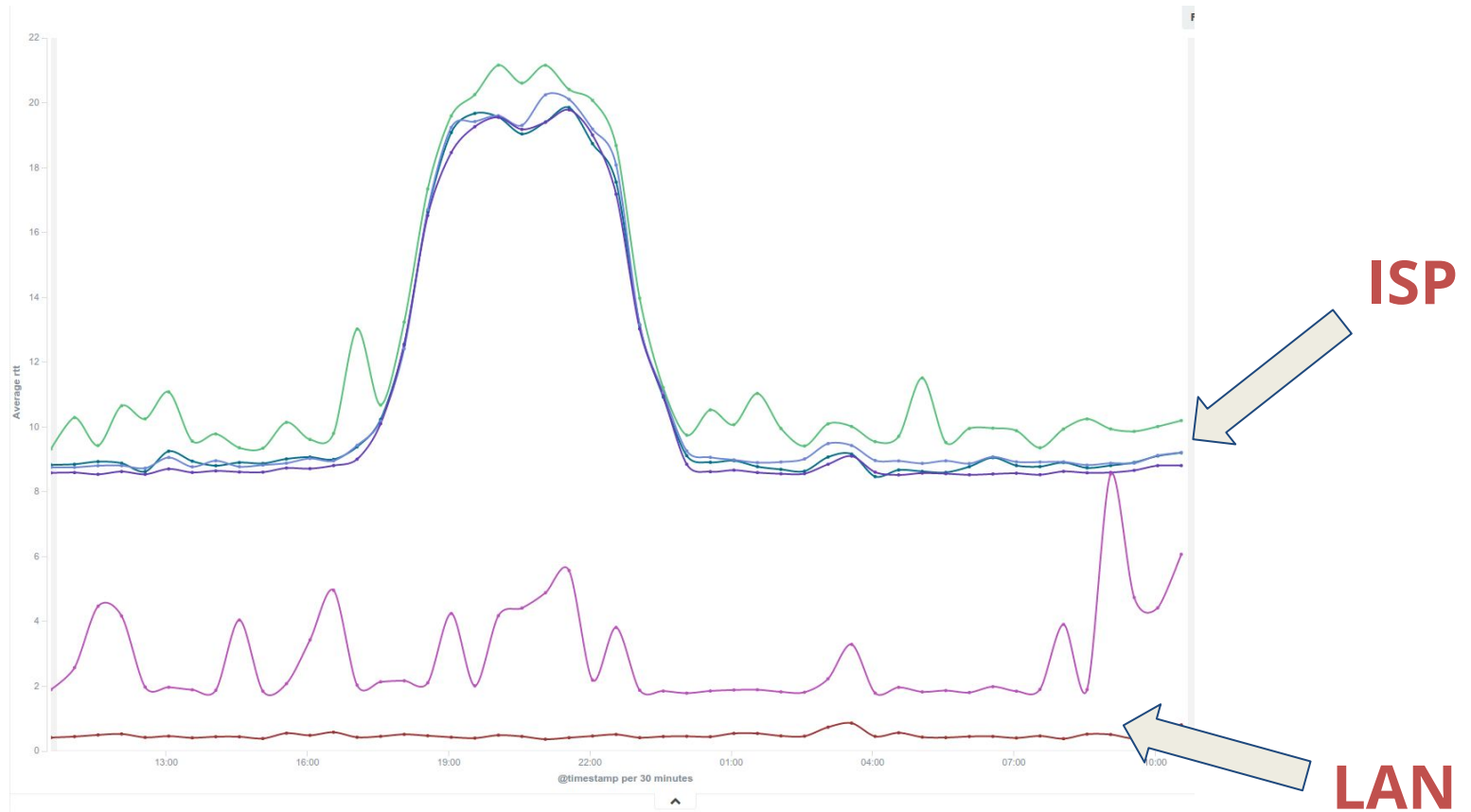
# Pingbeat Dashboard



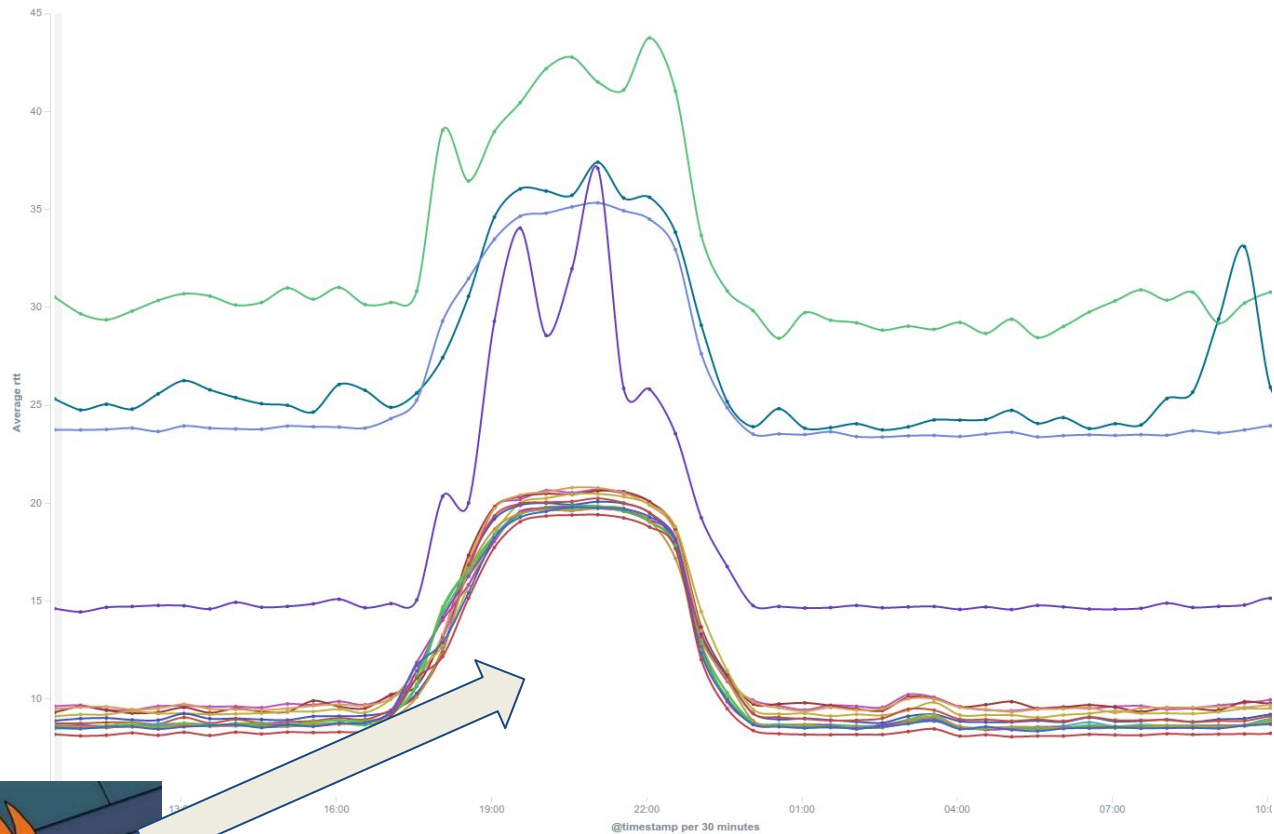
# Pingbeat Dashboard



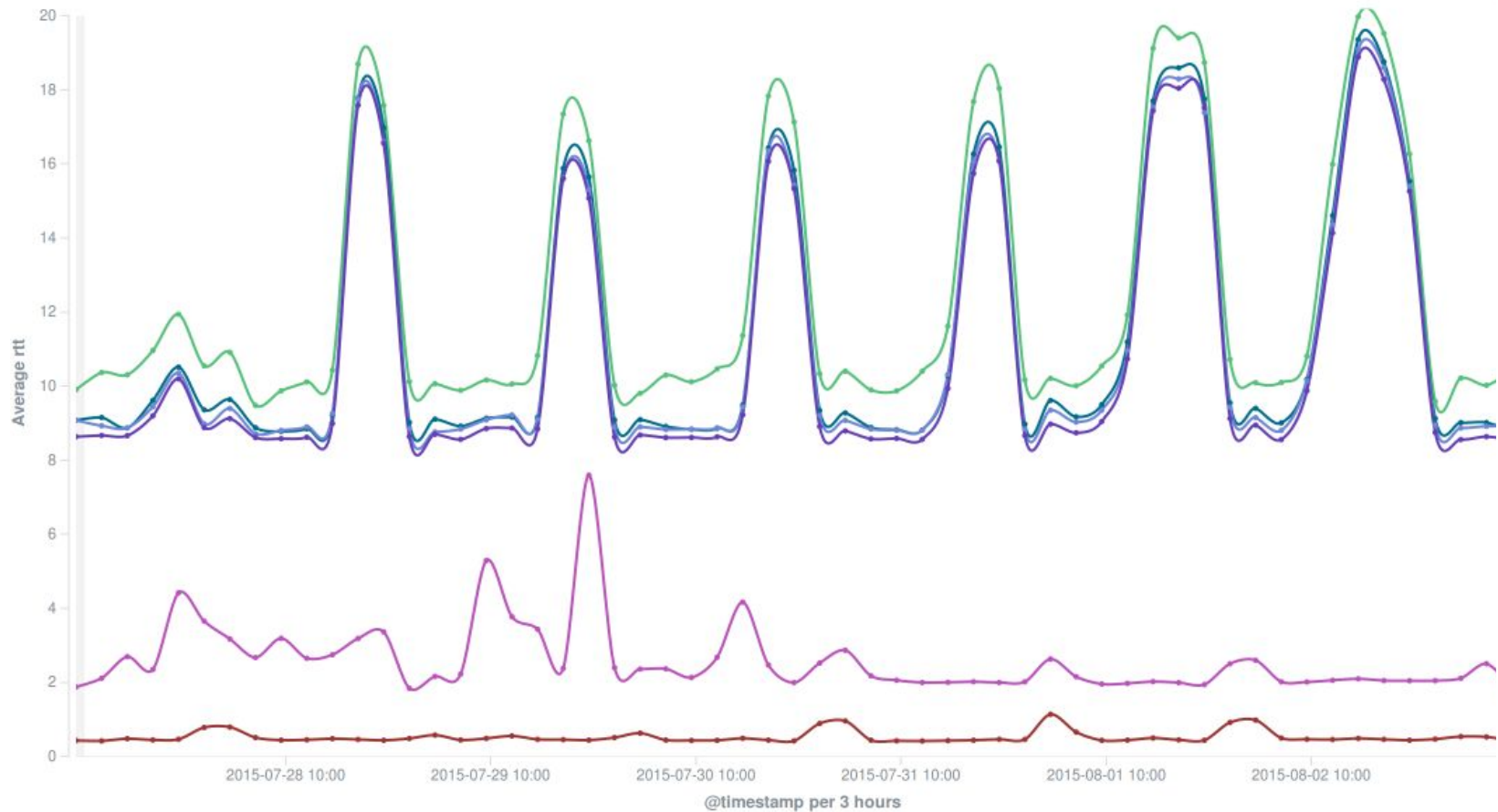
# Filtering to my LAN and ISP hosts...



# 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*

Want to learn more?

## **Tutorial: The Power of Open Data with ELK**

Thursday 10:40am, D2.211

