Network traffic telemetry
(NetFlow, IPFIX, sFlow)

Paolo Lucente

pmacct

MENOG 13 meeting, Kuwait City – Sep 2013

Network traffic telemetry
(NetFlow, IPFIX, sFlow)

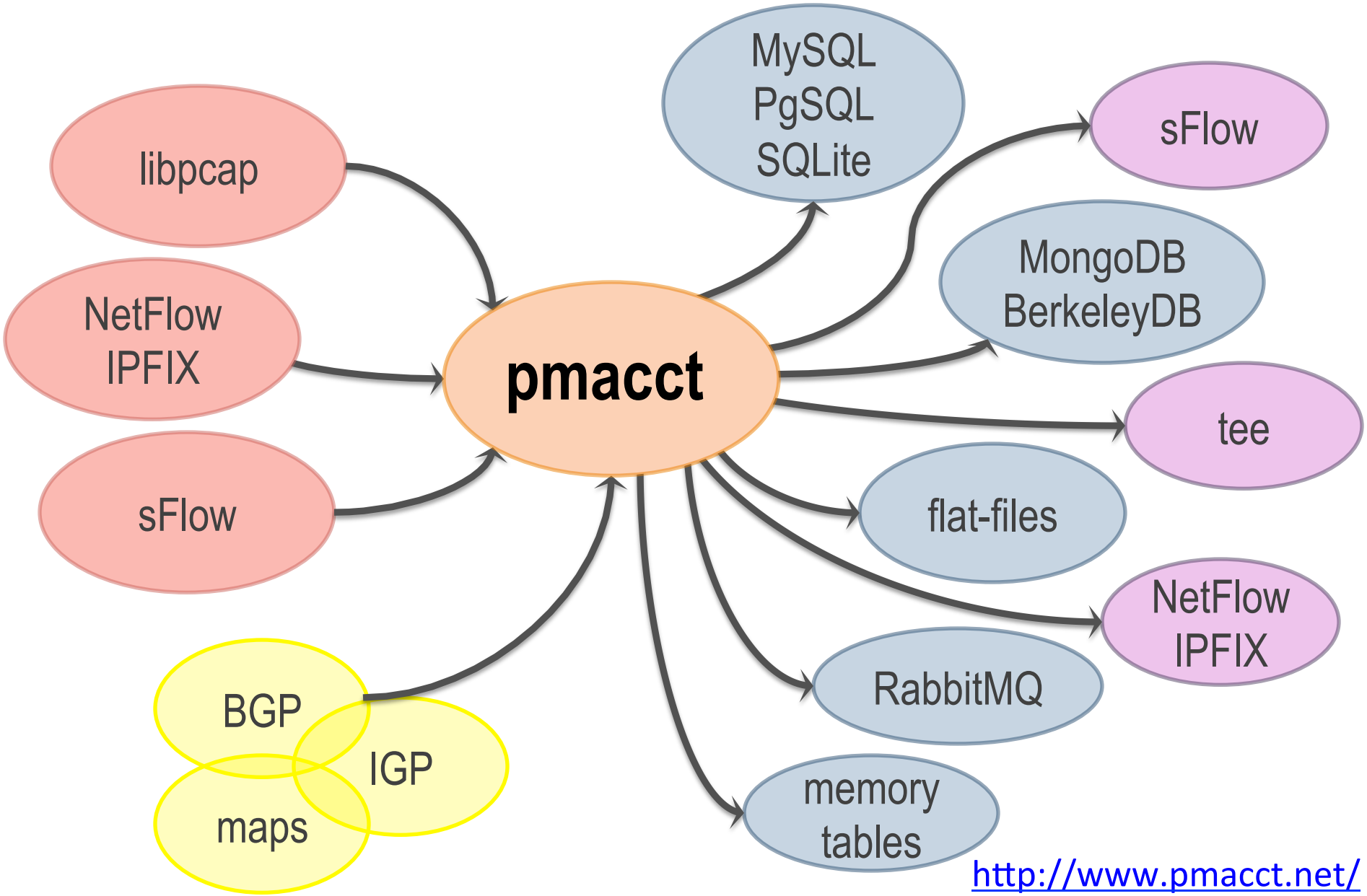
Agenda

- **whoami: Paolo & pmacct**
- Ramblings: Square 0 to reporting

whoami: Paolo

- Been originally working for operators for a while
- Been working for vendors for a little while after that
- Been involved with IP accounting for a while
  - Hence I stumbled upon NetFlow in the 90's ☺
- Within operators, network traffic telemetry is beneficial in several contexts, ie.:
  - Traffic engineering
  - Capacity planning
  - Peering
  - …
  - and also *(ie. not only)* security

pmacct is open-source, free, GPL'ed software



http://www.pmacct.net/

Network traffic telemetry
(NetFlow, IPFIX, sFlow)

Agenda

- whoami: Paolo & pmacct
- **Ramblings: Square 0 to reporting**

Square 0

- NetFlow is not the only existing export protocol
- Consistent trend is to build extensible protocol formats by means of templates (NetFlow v9, IPFIX) or modules (sFlow)
- Export protocols, versions and templates (or modules) available are solely mandated by the underlying network hardware, ie.:
  - Cisco? NetFlow!
  - Juniper? NetFlow and IPFIX on M/T/MX; sFlow on EX
  - Brocade? sFlow!

Square 0.5

- Another consistent trend: telemetry protocols being generalized:
  - Traditionally NetFlow/IPFIX and sFlow would report on network traffic
  - NetFlow now reports on firewall and NAT events
  - sFlow now reports on VM system resources
- Additional pressure put on the collector
  - Traditionally has to handle K or M flows/s
  - Now if a CGN blade resets, it could be tens or hundreds M events reported in one go.

Square 1

- Essentially, top-down approach
- Do:
  - Start with a vision
  - Start with goals to achieve
- Don't:
  - Collect just for the sake of
  - Collect because one day raw data will be useful ..
- *Goals drive to requirements*

Requirements (1/2)

- Given goals, network topology, hardware, etc.
  - Define an export model:
    - what devices (routers, switches, ..) need to export
    - which interfaces need to report
    - which direction, inbound or outbound or both?
    - whether to sample or not and if yes how much
- *Reality checks: don' t just assume hardware can follow the ideal export model; double-check vendor capabilities*

Requirements (2/2)

- Is the network traffic telemetry going to do the job alone or not? For example:
  - Correlation with BGP benefits peering and IP transit analysis (profitability, costs, violations, etc.)
  - Correlation with IGP (plus estimation methods) benefits traffic engineering and capacity planning
- Is there any margin for use of data reduction techniques?
  - Micro-flow information and as much as possible 1:1 sampling rate benefits security applications and R&D
- *Requirements drive to choice of tooling and storage (ie. flat-files, RRD, RDBMS, NoSQL, etc.)*

Collector implementation - gotchas

- Organizational
  - 800 pound gorilla project spanning across different departments (including IT and Security where existing)
- Technical
  - Exporters loose data along the way (hw/sw limits)
  - Collectors loose data along the way (horizontal scalability must be possible)
  - Over-engineered protocol features (ie. NetFlow v9, IPFIX sampling) drive to own creative implementations
- *Verify expectations and perform consistency checks against other sources, ie. interface counters*

Collector implementation - scalability

- Divide-et-impera approach, ie.:
  - Assign probes to collectors
  - Assign collectors to storages, ie. database partitions
- Work on data reduction (if allowed by the goal):
  - Increase sampling rate
  - Aggregation at the exporter or (better) at the collector
  - Fit data into larger time-bins
- Remove from the equation interfaces which are not strictly essential to the task, ie. low-speed
- *Do take into account on larger deployments that a single collector might not fit all the bill due to, typically, CPU or memory constraints*

Storage method selection

- Plenty on the offer: flat-files, RDBMS, RRD, noSQL DB, message brokers, memory tables, etc.:
  - I do not come here with general statements (in fact: the strategy of pmacct, for example, is to support them all)
  - In fully integrated products (ie. backend + frontend) this is transparent to end-users
  - Otherwise, if open access to data is provided/needed (which btw some fully integrated products also offer):
    - Depends, ie. whether a well-known query language like SQL is a pro; an indexed storage is a pro; an "have it all" to the micro-flow level kind of storage is a pro; etc.
    - Often it's matter of personal preferences, ie. what one feels most comfortable to query once in production

Maintenance (1/2)

- (Script to) verify the right set of data is hitting the collector:
  - Ingress + egress directions can lead to duplicates
  - Backbone + edge interfaces can lead to duplicates
  - Network infrastructure evolves: mistakes and forgetting is around the corner
- (Script to) track collector hardware resources:
  - Supports the divide-et-impera approach
  - It's good rule to do it anyway

Maintenance (2/2)

- (Script to) keep data-set size under control:
  - Valid for both memory and disk-based storage methods
  - Drop older data is less complex than consolidating; if going for inexpensive storage, ie. disk, consider collecting same data at different time resolutions, with different expiration periods
  - Aggregate same set of data on-the-fly in multiple ways in parallel, project-oriented!, is less complex than storing micro-flows then sub-aggregating to build specific reports

Reporting (1/2)

- Reporting via email or web is popular
- Key is interaction between backend and frontend:
  - If open access is granted to data-set, great stuff!
  - Otherwise, make sure new reports can be (easily) generated as requirements emerge
  - Reporting can influence organization of the backend, ie. RDBMS re-indexing
- *Flexibility must be possible (if Sales, Purchase, Product Management, etc. find it useful – be prepared to handle their creativity!)*

Reporting (2/2)

- Holy grail: a catch-all frontend encompassing a mix of native and scriptable functionalities (, reports, etc.)
- IMHO, value is really in having data available:
  - I got recently reported: "*I have put together a frontend capturing most of the data visualization requirements of my company in 48 hours with Twitter Bootstrap*".
  - And honestly it was a nice one.

Network traffic telemetry
(NetFlow, IPFIX, sFlow)

Thanks for your attention!

Questions? Now or later ..

Paolo Lucente

pmacct

<paolo at pmacct dot net>

Keep in touch via LinkedIn

MENOG 13 meeting, Kuwait City – Sep 2013