Building traffic matrices to support peering decisions

Paolo Lucente

the pmacct project | AS286

<paolo at pmacct dot net>

http://www.pmacct.net/

EPF #5 meeting, Cannes, France – Sep 2010

Building traffic matrices to support peering decisions

Agenda

- **Introduction**
- The tool: pmacct
- Setting the pitch
- Case study: peering at AS286

Why speaking of traffic matrices?

- Are traffic matrices useful to a network operator in the first place? Yes …
  - Capacity planning (build capacity where needed)
  - Traffic Engineering (steer traffic where capacity is available)
  - Better understand traffic patterns (what to expect, without a crystal ball)
  - Support peering decisions (traffic insight, traffic engineering at the border, support what if scenarios)

What a traffic matrix to support peering decisions can do for you

- Analysis of existing peers and interconnects:
  - Support policy and routing changes
  - Fine-grained accounting of traffic volumes and ratios
  - Determine backbone costs associated to peering
  - Determine revenue leaks
- Planning of new peers and interconnects:
  - Who to peer next
  - Where to place next interconnect
  - Modeling and forecasting

A traffic matrix to support peering decisions in practice

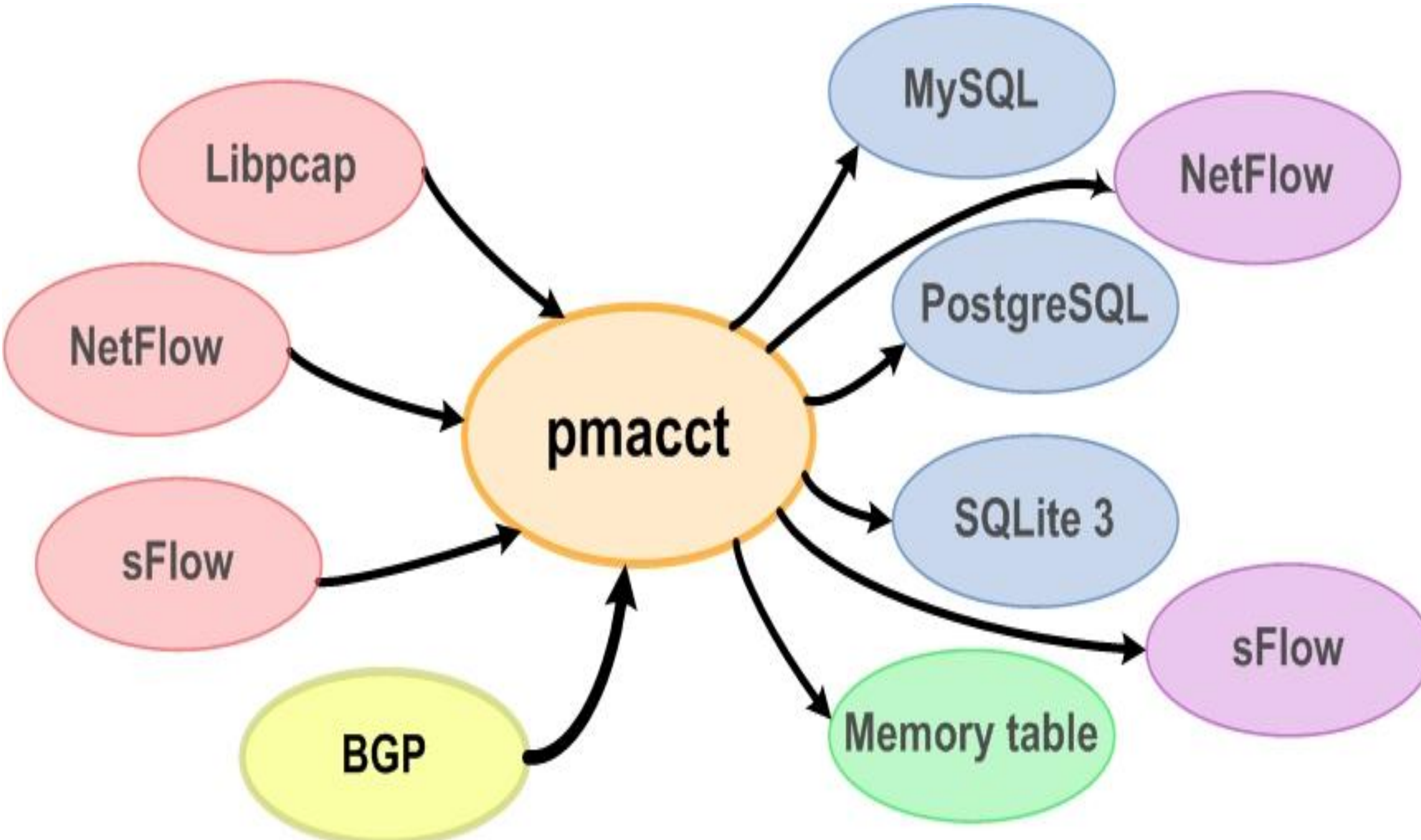
- What is needed:
  - BGP
  - Telemetry data: NetFlow, sFlow
  - Collector infrastructure: tool, system(s)
  - Storage: RDBMS, RRD or home-grown solution
  - Maintenance and post-processing scripts
- Risks:
  - 800 pound gorilla project

Building traffic matrices to support peering decisions

Agenda

- Introduction
- **The tool: pmacct**
- Setting the pitch
- Case study: peering at AS286

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



Introducing BGP natively into a NetFlow/sFlow collector

- pmacct introduced a Quagga-based BGP daemon
  - Implemented as a parallel thread within the collector
  - Doesn’t send UPDATEs and WITHDRAWs whatsoever
  - Behaves as a passive BGP neighbor
  - Maintains per-peer BGP RIBs
  - Supports 32-bit ASNs; IPv4 and IPv6 families
- Why BGP at the collector?
  - Telemetry reports on forwarding-plane
  - Telemetry should not move control-plane information over and over

Building traffic matrices to support peering decisions

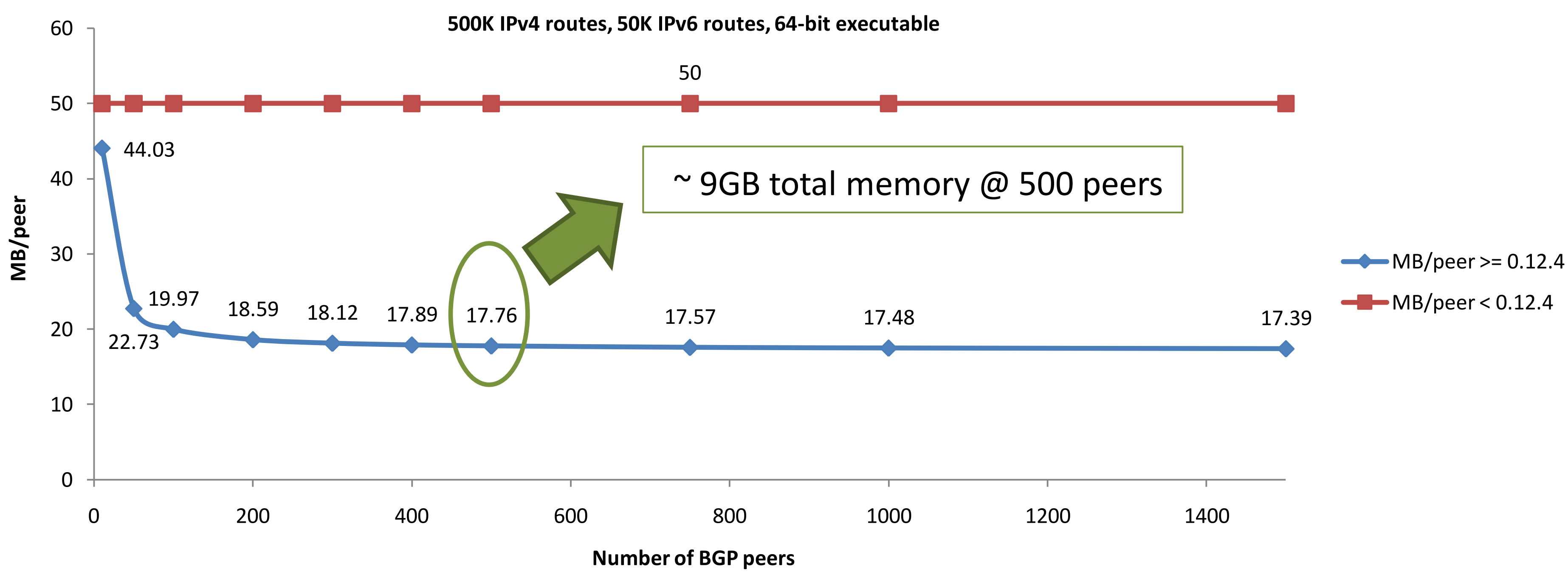
Agenda

- Introduction
- The tool: pmacct
- **Setting the pitch**
- Case study: peering at AS286

Getting BGP to the collector

– Let the collector BGP peer with all PE devices: facing peers, transit and customers.

– Determine memory footprint (below in MB/peer)



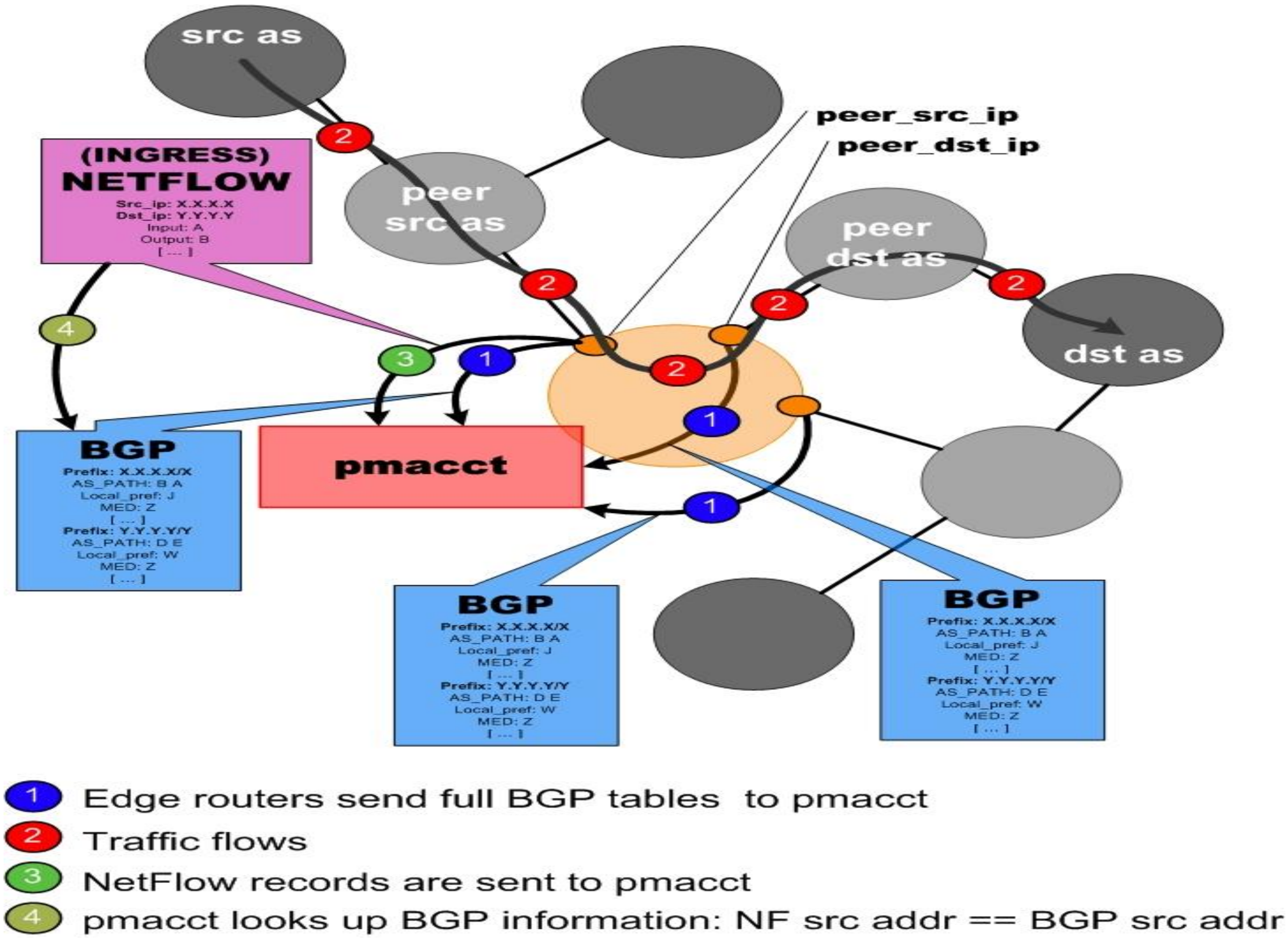
Getting BGP to the collector (cont.d)

- Set the collector as iBGP peer at the PE devices:
  - Configure it as a RR client for best results
  - Collector acts as iBGP peer across (sub-)ASes
- BGP next-hop has to represent the remote edge of the network model:
  - Typical scenario for MPLS networks
  - Can be followed up to cover specific scenarios like:
    - BGP confederations
      - Optionally polish the AS-Path up from sub-ASNs
    - default gateway defined due to partial or default-only routing tables

Getting telemetry to the collector

- Export ingress-only measurements at all PE devices: facing peers, transit and customers.
  - Traffic is routed to destination, so plenty of information on where it's going to
  - It's crucial instead to get as much as possible about where traffic is coming from
- Leverage data reduction techniques at the PE:
  - Sampling
  - Aggregation (but be sure to carry IP prefixes!)

Telemetry data/BGP correlation



Storing data persistently

- Data need to be aggregated both in spatial and temporal dimensions before being written down:
  - Optimal usage of system resources
  - Avoids expensive consolidation of micro-flows
  - Suitable for project-driven data-sets
- Open-source RDBMS appear a natural choice
  - Able to handle large data-sets
  - Flexible and standardized query language
  - Solid and evolving storage and indexing engines
  - Scalable: clustering, spatial and temporal partitioning

Storing data persisently (cont.d)

Tag: agent_id
BGP Fields: as_src … med
Counters: packets, bytes
Time: stamp_inserted, stamp_updated

```
create table acct_bgp (
    agent_id INT(4) UNSIGNED NOT NULL,
    as_src INT(4) UNSIGNED NOT NULL,
    as_dst INT(4) UNSIGNED NOT NULL,
    peer_as_src INT(4) UNSIGNED NOT NULL,
    peer_as_dst INT(4) UNSIGNED NOT NULL,
    peer_ip_src CHAR(15) NOT NULL,
    peer_ip_dst CHAR(15) NOT NULL,
    comms CHAR(24) NOT NULL,
    as_path CHAR(21) NOT NULL,
    local_pref INT(4) UNSIGNED NOT NULL,
    med INT(4) UNSIGNED NOT NULL,
    packets INT UNSIGNED NOT NULL,
    bytes BIGINT UNSIGNED NOT NULL,
    stamp_inserted DATETIME NOT NULL,
    stamp_updated DATETIME,
    PRIMARY KEY (…)
);
```

```
shell> cat pretag.map
id=100  peer_src_as=<customer>
id=80   peer_src_as=<peer>
id=50   peer_src_as=<IP transit>
[ … ]
```

```
shell> cat peers.map
id=65534 ip=X in=A
id=65533 ip=Y in=B src_mac=J
id=65532 ip=Z in=C bgp_nexthop=W
[ … ]
```

– In any schema (a subset of) BGP primitives can be freely mixed with (a subset of) L1-L7 primitives

Post-processing and reporting

– Traffic delivered to a BGP peer, per location:

```
mysql> SELECT  peer_as_dst, peer_ip_dst, SUM(bytes), stamp_inserted
       FROM acct_bgp
       WHERE peer_as_dst = <peer | customer | IP transit> AND
             stamp_inserted = < today | last hour | last 5 mins >
       GROUP BY peer_as_dst, peer_ip_dst;
```

– Aggregate AS PATHs to the second hop:

```
mysql> SELECT SUBSTRING_INDEX(as_path, '.', 2) AS as_path, bytes
       FROM acct_bgp
       WHERE local_pref = < IP transit pref> AND
             stamp_inserted = < today | yesterday | last week >
       GROUP BY SUBSTRING_INDEX(as_path, '.', 2)
       ORDER BY SUM(bytes);
```

– Focus peak hour (say, 8pm) data:

```
mysql> SELECT … FROM … WHERE stamp_inserted LIKE '2010-02-% 20:00:00'
       …
```

Post-processing and reporting (cont.d)

– Traffic breakdown, ie. top N grouping BGP peers of the same kind (ie. peers, customers, transit):

```
mysql> SELECT … FROM … WHERE …
       local_pref = <<peer | customer | IP transit> pref>
       …
```

– Download traffic matrix (or a subset of it) to 3rd party backbone planning/traffic engineering application (ie. Cariden, Wandl, etc.):

```
mysql> SELECT peer_ip_src, peer_ip_dst, bytes, stamp_inserted
       FROM acct_bgp
       WHERE [ peer_ip_src = <location A> AND
             peer_ip_dst = <location Z> AND … ]
             stamp_inserted = < today | last hour | last 5 mins >
       GROUP BY peer_ip_src, peer_ip_dst;
```

Briefly on scalability

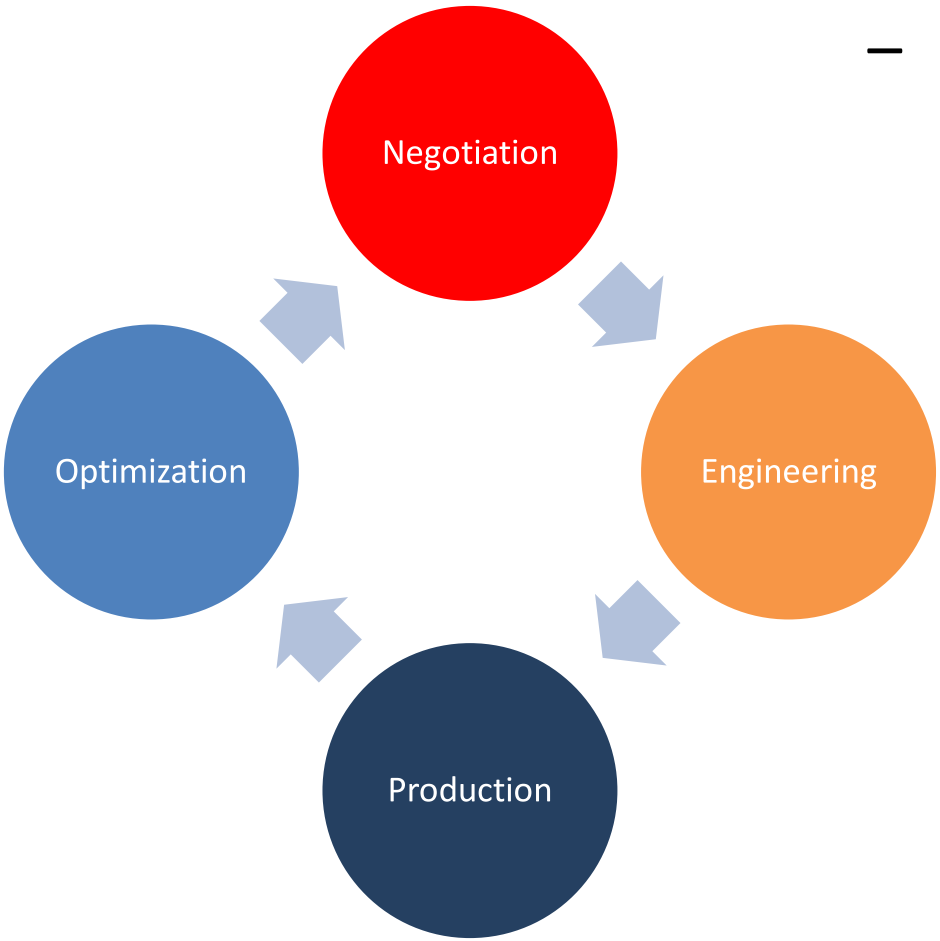
- A single collector might not fit it all:
  - Memory: can’t store all BGP full routing tables
  - CPU: can’t cope with the pace of telemetry export
  - Divide-et-impera approach is valid:
    - Assign PEs (both telemetry and BGP) to collectors
    - Assign collectors to RDBMSs; or cluster the RDBMS.
- The matrix can get big, but can be reduced:
  - Keep smaller routers out of the equation
  - Filter out specific services/customers on dense routers
  - Focus on relevant traffic direction (ie. upstream if CDN, downstream if ISP)
  - Sample or put thresholds on traffic relevance

Building traffic matrices to support peering decisions

Agenda

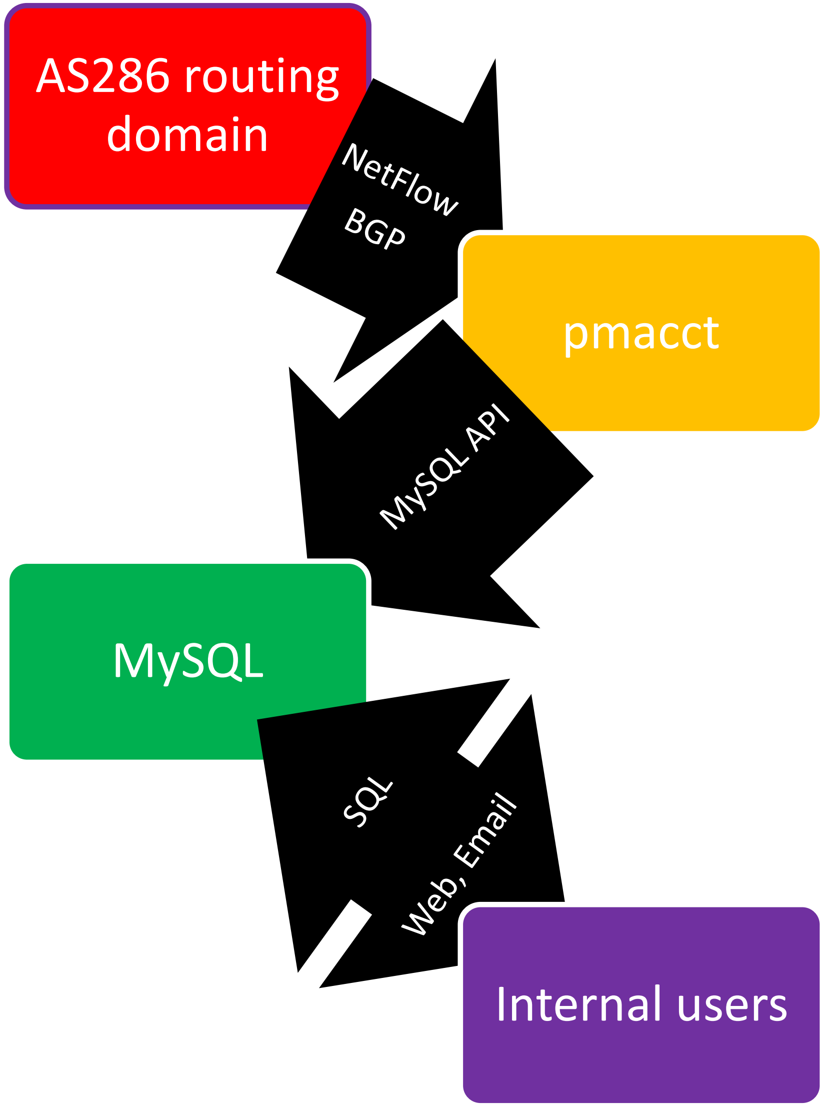
- Introduction
- The tool: pmacct
- Setting the pitch
- **Case study: peering at AS286**

Case-study: peering at AS286



- Peering as a cycle
- NetFlow + BGP traffic matrix steers peering optimization:
  - Identify new and “old” peers
  - Traffic analysis: backbone costs, 95th percentiles, ratios
  - Analysis of interconnection density and traffic dispersion
  - Forecasting and trending
  - Ad-hoc queries from Design & Engineering and indeed … the IPT Product Manager

Case-study: peering at AS286



- 250+ Gbps routing-domain
- 100+ high-end routers around the globe:
  - Export sampled NetFlow
  - Advertise full routing table
  - Mix of Juniper and Cisco
- Collector environment:
  - Runs on 2 Solaris/SPARC zones
  - pmacct: dual-core, 4GB RAM
  - MySQL: quad-core, 24GB RAM, 500 GB disk
- Data retention period: 6 months

Case-study: peering at AS286

- AS286 backbone routers are first configured from templates:
  - NetFlow + BGP collector IP address defined over there
  - Enabler for auto-discovery of new devices
- Edge interfaces are provisioned following service delivery manuals:
  - Relevant manuals and TSDs include NetFlow activation
  - Periodic checks NetFlow is active where it should
- Maps, ie. source peer-AS, are re-built periodically

Further information

- http://www.pmacct.net/lucente_pmacct_uknof14.pdf
  - AS-PATH radius, Communities filter, asymmetric routing
  - Entities on the provider IP address space
  - Auto-discovery and automation
- http://wiki.pmacct.net/OfficialExamples
  - Quick-start guide to setup a NetFlow/sFlow+BGP collector instance
- http://wiki.pmacct.net/ImplementationNotes
  - Implementation notes (RDBMS, maintenance, etc.)

Building traffic matrices to support peering decisions

Thanks for your attention!

Questions?

Paolo Lucente

the pmacct project | AS286

<paolo at pmacct dot net>

http://www.pmacct.net/