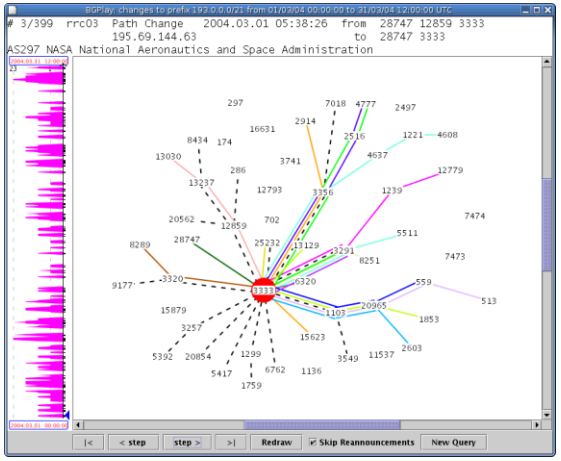
Exploiting Router Programmability to Ease Routing and Traffic Analysis

Maurizio Pizzonia
pizzonia@dia.uniroma3.it
Roma Tre University – Dip. Inf. Aut.

RIPE 61 - Rome - November 17th, 2010

brief introduction of our research group

- Roma Tre University – Comp Sc. and Aut. Dept.
- research group in networking and visualization
  - mostly routing and BGP
- famous project
  - bgplay.routeviews.org
- collaborate with RIPE NCC since 2003



router programmability opportunities

- perform some elaboration directly on the router
  - avoid another box and reduce opex
  - event handling on the same box
  - dynamic configuration change
- script based
  - cisco EEM, Junscript, etc.
- more performing approaches
  - cisco AXP, Juniper SDK

uniroma3 and router programming

- Juniper collaboration for performing research on Junos SDK
- starting April 2010
- support from CASPUR

projects

- traffic: exploit Junos SDK for **traffic matrix computation**
  - ongoing project
  - objective: two new commands
    - set traffic-matrix on
    - show traffic-matrix-row
- routing: **«beyond the best»**
  - analyze BGP messages sent by peerer for “BGP SLA” verification

1st project: traffic matrix

- concerning traffic to be routed..
  - what is the «demand» of the external customers/providers/peers etc on our network?
- needed for
  - capacity planning
  - traffic engineering
  - what if analysis
  - etc.

to / from	Milano	Firenze	Roma	Napoli	Catania
Milano	-	?	?	?	?
Firenze	?	-	?	?	?
Roma	?	?	-	?	?
Napoli	?	?	?	-	?
Catania	?	?	?	?	-

state of the art

- large state of the art
- first works are for plain telephone networks (1937!)
- NANOG 43: Best Practices for Determining Traffic Matrices – Tutorial, Blili et al.
- RIPE 61: Best Practices on Network Planning, Filsfils et al.
  - yesterday in the plenary!

- here only a quick review

state of the art: mathematical methods

- Gunnar et al. Traffic matrix estimation on a large IP backbone: a comparison on real data

	Europe	America
Worst-case bound prior	0.10	0.39
Simple gravity prior	0.26	0.78
Entropy w. gravity prior	0.11	0.22
Bayes w. gravity prior	0.08	0.25
Bayes w. WCB prior	0.07	0.23
Fanout	0.22	0.40
Vardi	0.47	0.98

- problems
  - needs tuning
  - mean errors reported up to 98% even when best tuned
  - estimation can be good in the best cases
    - but it depends on the topology
  - very hard to tune without a real traffic matrix!

state of the art: the vendor perspective

- make a full mash of $O(n^2)$ tunnels RSVP-TE and read the counters!
  - many operators are not happy about it!
  - operational cost, problems with load sharing (ECMP)
- other option: MPLS FEC counters
  - topology issues when LSR and LER are not distinct
  - only internal TM

state of the art: the networker perspective

- count bytes destined to each bgp next-hop!
  - no way:
    forwarding engine does not know bgp next-hops
- do that outside the router
  - netflow (v5, v9, sampling)
  - sampling tradeoff: **router load vs. precision**
  - may under-estimate traffic (up to 50% less)

objective

- each router independetly computes one row of the (external) traffic matrix
- easy activation
  - ideally: set traffic-matrix on
- easy retrieval
  - ideally: show traffic-matrix-row
  - SNMP

junos and traffic counting

- on JunOS you can define arbitrary, per destination, counters (packets and bytes)
  - to apply to an interface
- can be automated by SDK programming
  - SDK support goes beyond plain change of configuration (special rules can be defined)

configuring counting

- **[edit firewall family** *inet* **filter** *filter-name* **]**

```
term term-name {
    from {
        destination-address {
            prefix1
            prefix2
            ……
        }
    }
    then {
        count counter-name      # accept and count
    }
}
```

count per bgp next-hops

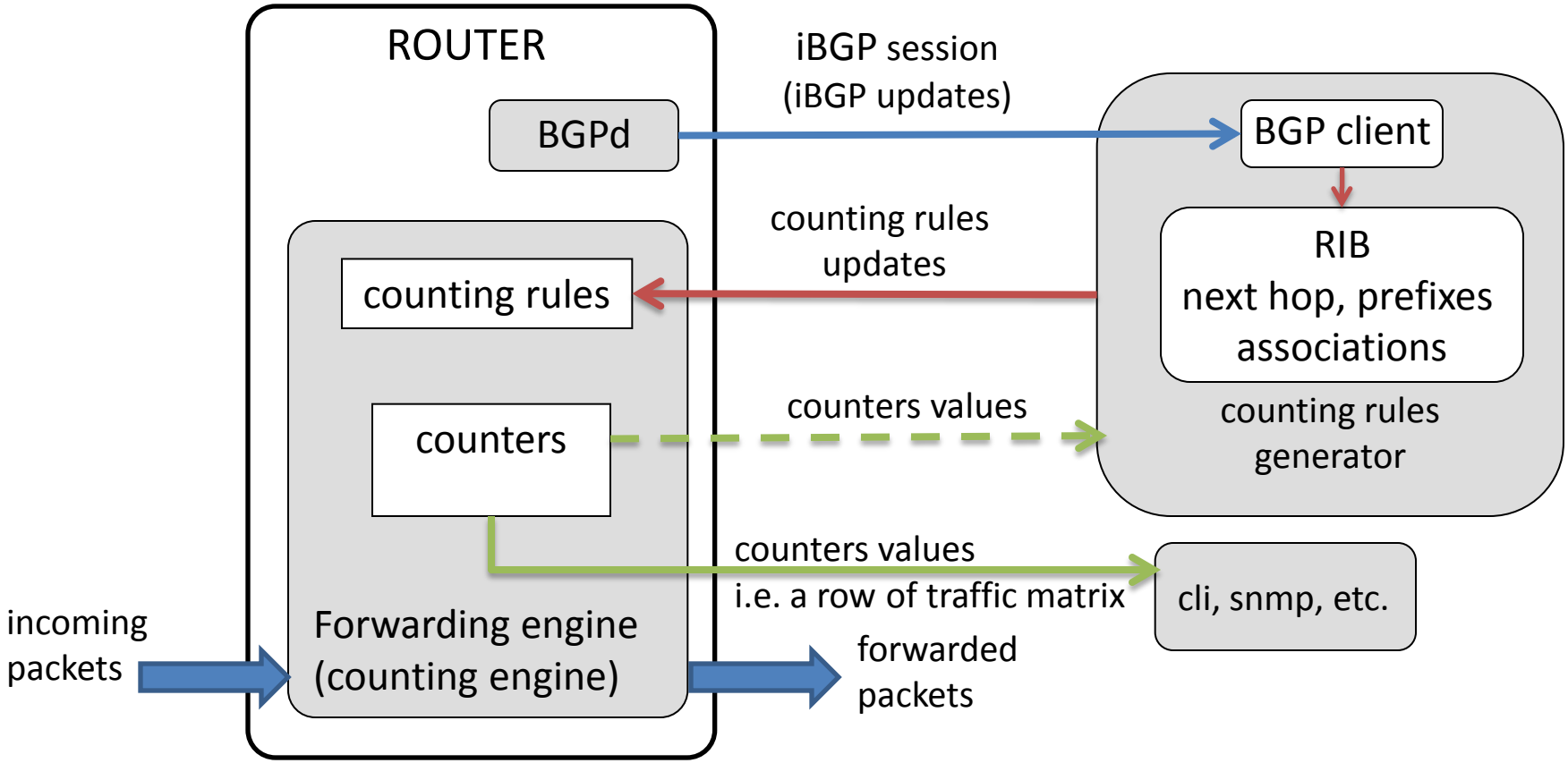
- one term/counter for each bgp next-hop
- only hundreds of terms and counters
  - each counter is a cell of the traffic matrix
  - each term may have >100k destination to match
  - destinations sum up to about 300k
- optimizations
  - skewed distribution of prefixes over nexthops (default, aggregation)
  - Draves et al. Constructing optimal IP routing tables INFOCOM ’99
  - we have 370 counters for the GARR routing table
    - thanks to GARR for providing us their RIB

handling RIB changes

- RIB changes can be obtained by iBGP rr-client
- the “receiver”
  - generates update for counting rules
    - changes to matching rules (destinations) of terms
  - configures on-the-fly filter/counters into the router
- counters should be regularly sampled for accounting
  - can be easily done within the router

handling RIB changes

- the system architecture



criticalities

- "dirty" configuration?
  - NO, Junos SDK allows hidden filters, regular configuration is unchanged
- RIB changes very quickly, commit at every BGP update?
  - new hardware support special filters optimized for fast update
  - on our **old** hardware it takes...
    - about 8 seconds to process 17 «rule updates», we batch every 10 seconds and optimize: 17 is the biggest batch
    - about 40 seconds to load the whole RIB (at boot)
    - hw M7i with old CFEB

criticalities

- does the throughput suffers any penalty?
  - preliminary tests: no packet loss
- what about precision?
  - preliminary tests: 3 seconds hole
    - changing a rule R, R does not count for 3 seconds
  - we are very curious about behaviour of new hw
- about the tests: cannot state the final word
  - Smartbit 600 used (up to 4 GB traffic)
  - but only two FastEthernet on our hardware :(
  - **looking for a well equipped lab interested in testing it (call open to all of you!)**

2nd project: “beyond the best”

- current BGP monitoring/collecting methods…
  - collects only best routes (e.g. quagga)
  - collects updates
    - BMP
    - mirror+BGPdecoding (see Vissicchio et al. INM/WREN 2010)
- for certain applications you may avoid collection
  - …and avoid collector
- BGP SLAs
  - e.g. for upstream: reach ASXXX **directly** for **99% of time**
  - have you ever thought about it? Cloud computing customers would be happy to have a targeted SLA

beyond the best and router programming

- objective: analyze BGP updates as they arrive on the wire
  - TCP reconstruction
  - BGP decoding (use open implementation)
- express SLA with a proper language in router configuration
- report SLA violations

beyond the best: present and future

- present
  - preliminary test with external system on a cisco 7201
  - targeted to update collection
  - criticism: traffic mirror can hurt
    - not really true, see our paper at INM/WREN 2010
  - need external collecting system
- future
  - BGP update analysis performed within the router
    - Junos SDK
  - no mirror, no external system
  - SLA violations on syslog, summary by email

conclusions

- router programmability provides new opportunities
  - for monitoring tools
  - for new custom services
  - …without additional box and with high performance
- risks (of the new technology) can be mitigated by collaborations with university
  - Juniper supports this collaboration model
    - see my presentation at TERENA 2010

questions?