Universität Karlsruhe (TH)
Institut für Telematik

TELEMATICS TECHNICAL REPORTS

Proceedings of the 2nd Inter-Domain Routing Workshop (IDRWS 2004) 1st - 2 nd of May, Amsterdam, The Netherlands

Lichtwald, Götz

May, 26th 2004

TM-2004-3

ISSN 1613-849X

http://doc.tm.uka.de/tr/

Institute of Telematics, University of Karlsruhe
Zirkel 2, D-76128 Karlsruhe, Germany

IDRWS 2004

Inter-Domain Routing Workshop 2004

Sponsor

Thanks ☺ !

Organization

- T. Griffin (Intel Research)
- D. Karrenberg (RIPE NCC)
- G. Lichtwald (ITM, University of Karlsruhe)
- O. Maennel (TH Munich)
- S. Mink (Schlund+Partner)
- H. Uijterwall (RIPE NCC)
- U. Walter (ITM, University of Karlsruhe)
- M. Williams (RIPE NCC)

What was the intension form this Workshop

- It was **not** “Yet another workshop about …”
- Instead we want to start something new
  we wanted to bring together
  - Academics
  - Operators
  - Vendors

  With the intension is to discuss every talk from
  - the Operators
  - the Vendors
  - the Academic

  point of view!

Program for 1st of May

	10:30 - 11:00	Welcome / Door open
Future of Routing	11:00 - 11:30	Randy Bush "*Happy Packets - Initial Results*"
	11:30 - 11:45	Simon Leinen "*Arguments for path selection by end-systems and outline of a pure source-routing approach*"
	11:45 - 12:00	Rüdiger Volk "*Considering application fit for standard requirements of iBGP*"
Lunch	12:00 - 13:00	---
Internet Routing Registry	13:00 - 13:30	Geoff Huston "*Allocations and Advertisements*"
	13:30 - 14:00	Larry Blunk "*Towards a Cohesive Internet Routing Registry System*"
	14:00 - 14:30	Georgos Siganos "*Nemecis: A tool to analyze the IRR registries*"
Coffee Break	14:30 - 15:00	---
Alternatives	15:00 - 15:15	kc Claffy (p.p. Dimitri Krioukov) "*Introduction to compact routing*"
	15:15 - 15:30	Christoph Reichert "*IP-Protection for Fast Inter-Domain Resilience*"
	15:30 - 16:00	Götz Lichtwald "*Stabilizing the BGP control plane*"
Coffee Break	16:00 - 16:30	---
Next Generation Networks	16:30 - 17:00	Karl Schrodi "*Inter-Domain Routing Issues in Next Generation Networks*"
	17:00 - 17:15	Thomas Engel "*Inter-domain Resilience for QoS Traffic*"
	17:15 - 17:30	Thomas Schwabe "*Independence of Inter-Domain QoS Signaling and Routing*"
Social	17:30	The social event will take place at Lieve

Program for 2nd of May

Operational Issues	9:30 - 9:45	Stefan Mink "*Detecting unwanted route readvertisements*"
	9:45 - 10:00	Volodymyr Yakovenko "*Some aspects of more specific prefixes routing*"
	10:00 - 10:30	Simon Leinen "*Living with partial routing*"
Coffee Break	10:30 - 11:00	---
Tools	11:00 - 11:30	Kihong Park "*Steps Toward Large-scale Meaningful BGP Simulation*"
	11:30 - 12:00	Olivier Marcé "*Embedded routing monitoring: prototype and results*"
	12:00 - 12:30	Maurizio Pizzonia "*Visual Analysis of Inter-Domain Routing Dynamics*"
Lunch	12:30 - 14:00	---
Operational Challenges	14:00 - 14:15	Bruno Quoitin "*Cooperative Incoming Traffic Engineering*"
	14:15 - 14:30	Steve Uhlig "*Towards a more systematic approach for interdomain traffic engineering*"
	14:30 - 14:45	Cristel Pelsser "*MPLS Traffic Engineering across AS boundaries*"
Coffee Break	11:45 - 15:15	---
Dangers / Outlook	15:15 - 15:45	Timothy Griffin "*BGP Wedgies --- Bad Routing Policy Interactions that Cannot be Debugged*"
Panel Discussion	15:45	Chair: T. Griffin "*Internet and IDR - ten years from now*"

Happy Packets: Some Initial Results

IDRWS/Amsterdam 2004.05.01

Randy Bush <randy@psg.com>
Tim Griffin <tim.griffin@intel.com>
Z. Morley Mao <zmao@eecs.umich.edu>
Eric Purpus <epurpus@cs.uoregon.edu>
Dan Stutsbach <dstutsba@cs.uoregon.edu>

<http://psg.com/~randy/040501.idrws.pdf>

Thanks to Our Sponsors

- NSF via award ANI-0221435
- The University of Oregon (Dan & Eric)
- The University of Michigan (Morley)
- Internet Initiative Japan (Randy)
- Intel Corporation (Tim)
- Verio and Sprint (bandwidth)
- Juniper & Cisco (routers)

Central Question

- What is the relationship between **control plane** instability and **data plane** instability?

- Related Questions:
  - Is the quantity of BGP updates good or bad?
  - Who wants to see zero BGP updates?

Internet Weather

We frequently hear comments such as

- Internet routing is fragile, collapsing, ...,
- BGP is broken or is not working well,
- Day X was a bad routing day on the internet,
- Change X to protocol Y will improve routing,
- Etc.

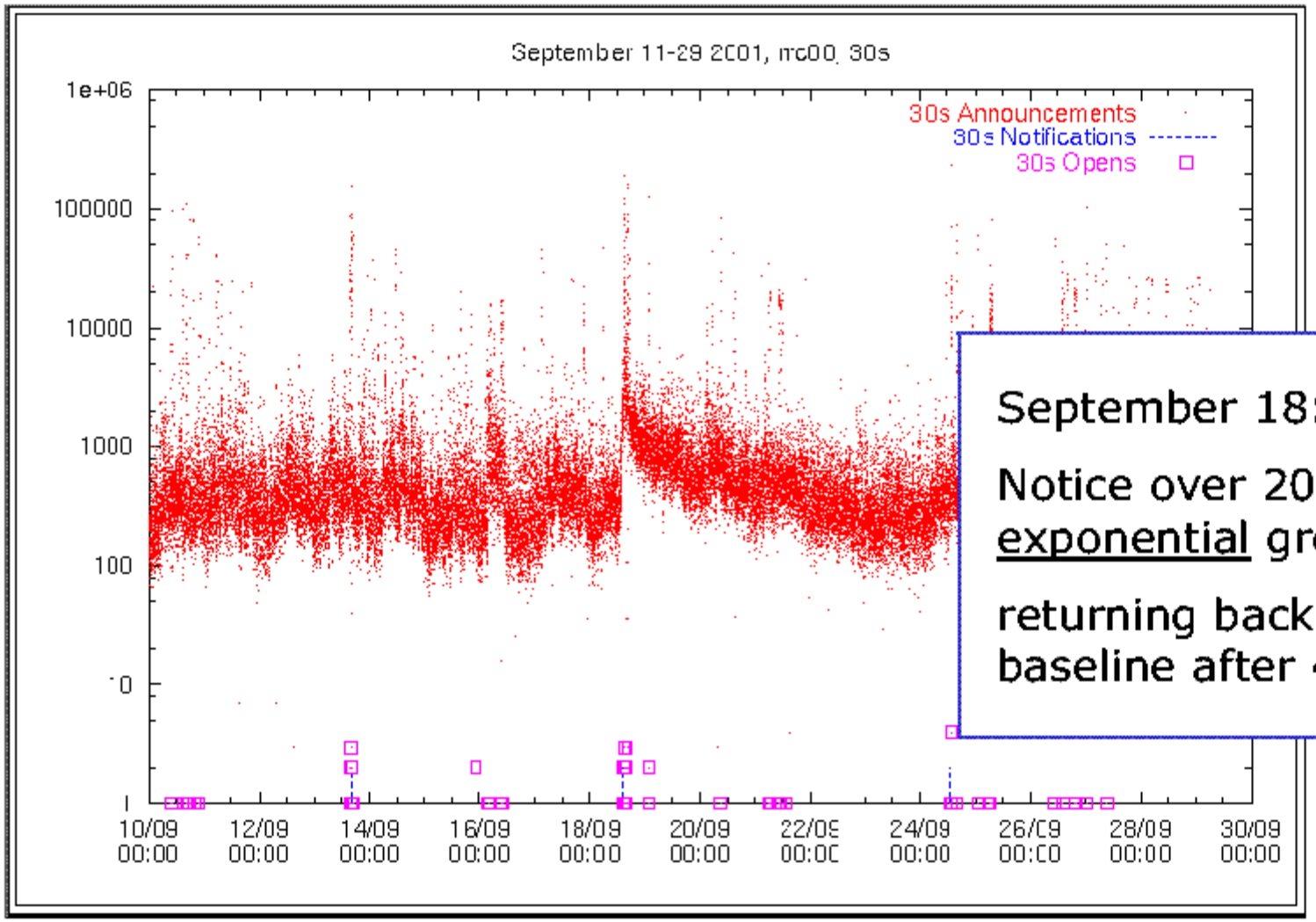
And we often measure routing dynamics and say that some measurement is better or worse than another

Internet [Routing] Instability

- We are told that a lot of BGP updates is equated with internet instability
- “There are too many BGP updates, so BGP must be broken.”

A view on content of the same messages

Number of prefix announcements in 30 sec intervals



September 18:

Notice over 20-fold exponential growth

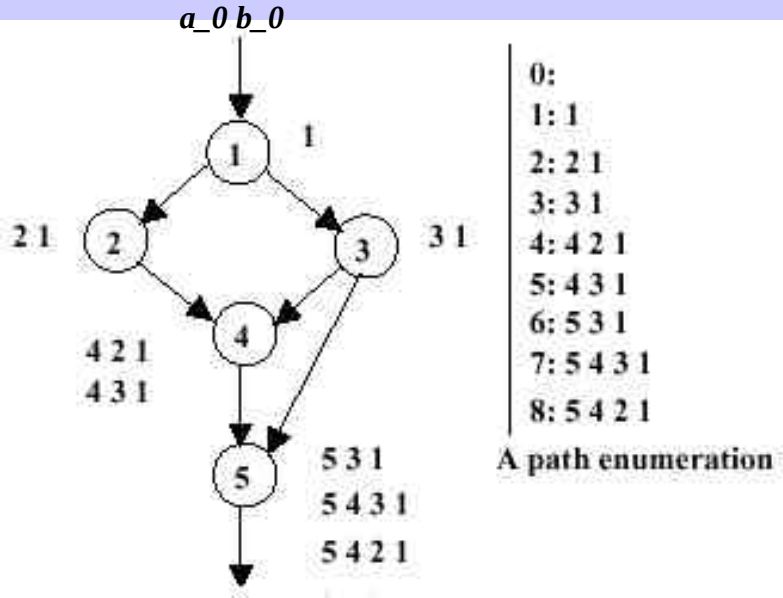
returning back to baseline after 4 days!

Big Events

- The Renesys folk and others are looking at big events
- We at looking at single announcements
- So our work does not contradict Renesys, but it does suggest we consider some of the assumptions (see Lan Wang et alia)
- And we are measuring data plane performance waiting for the next big event

Why I'm Going Crazy Trying to Interpret those BGP Updates?

It is easy to construct a 5 node BGP system where a simple Announce/Withdraw signal (a_0 b_0) at one node can produce any of these 52 output signals at another...



a_6b_6	$a_7a_6b_6$	$a_8a_6b_6$	$a_8a_6a_8b_8a_7b_7$
$a_6a_7b_7$	$a_7a_6a_7b_7$	$a_8a_6a_7b_7$	$a_8a_7a_6a_7a_8b_8$
$a_6a_8b_8$	$a_7a_6a_8b_8$	$a_8a_6a_8b_8$	$a_8a_7a_8a_6a_8b_8$
$a_6a_7a_8b_8$	$a_7a_8a_6b_6$	$a_8a_7a_6b_6$	$a_8a_7a_8b_8a_6b_6$
$a_6a_8a_7b_7$	$a_7b_7a_6b_6$	$a_8b_8a_6b_6$	$a_8b_8a_6b_6a_7b_7$
$a_6b_6a_7b_7$	$a_7a_6a_7a_8b_8$	$a_8a_6a_7a_8b_8$	$a_8b_8a_7a_6a_7b_7$
$a_6b_6a_8b_8$	$a_7a_6b_6a_8b_8$	$a_8a_6a_8a_7b_7$	$a_8b_8a_7b_7a_6b_6$
$a_6a_7b_7a_8b_8$	$a_7a_8a_6a_8b_8$	$a_8a_6b_6a_7b_7$	
$a_6a_8a_7a_8b_8$	$a_7a_8b_8a_6b_6$	$a_8a_7a_6a_7b_7$	
$a_6a_8b_8a_7b_7$	$a_7b_7a_6a_8b_8$	$a_8a_7a_6a_8b_8$	
$a_6b_6a_7a_8b_8$	$a_7b_7a_8a_6b_6$	$a_8a_7a_8a_6b_6$	
$a_6b_6a_8a_7b_7$	$a_7a_6a_7b_7a_8b_8$	$a_8a_7b_7a_6b_6$	
$a_6b_6a_7b_7a_8b_8$	$a_7b_7a_6b_6a_8b_8$	$a_8b_8a_6a_7b_7$	
$a_6b_6a_8a_7a_8b_8$	$a_7b_7a_8a_6a_8b_8$	$a_8b_8a_7a_6b_6$	
$a_6b_6a_8b_8a_7b_7$	$a_7b_7a_8b_8a_6b_6$	$a_8a_6a_8a_7a_8b_8$	

Routing Quality

- But what is *good* routing? How can we say one measurement shows routing is better than another unless we have metrics for routing quality?
- We often work on the assumption that number of prefixes, speed or completeness of convergence, etc. are measures of routing quality

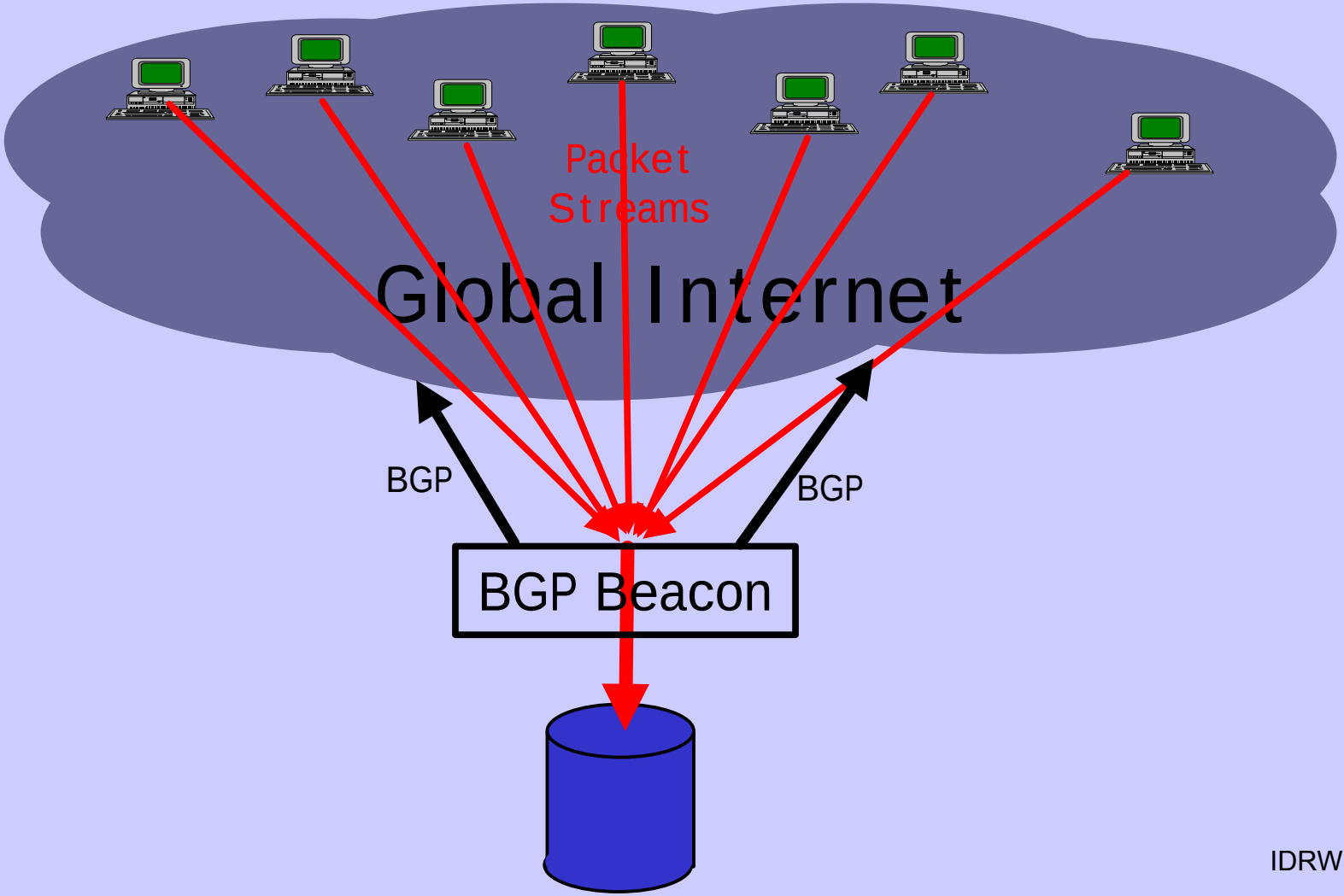
Happy Packets

- The measure which counts is whether the users' packets reach their destination
- If the users' packets are happy, the routing system, and other components, are doing their job
- We call these ***Happy Packets***
- There are well-known metrics for the data plane, Delay, Drop, Jitter, and Reordering
- So we set out to measure Control Plane quality by measuring the Data Plane

Router Scaling

- While data plane performance is the goal, we can't have routers falling over processing chatty BGP
- But, as long as network BGP growth increases load on the routers below Moore's law, it is not clear we are in danger

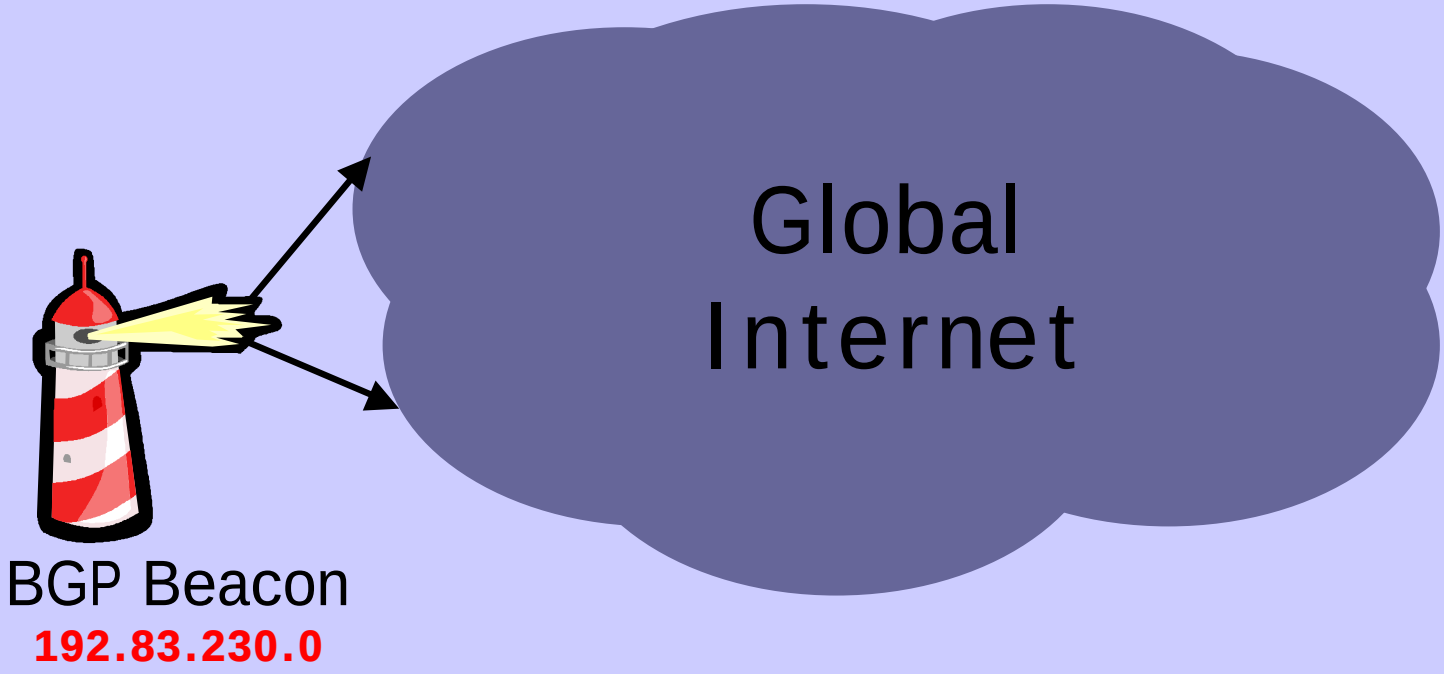
Experimental Setup



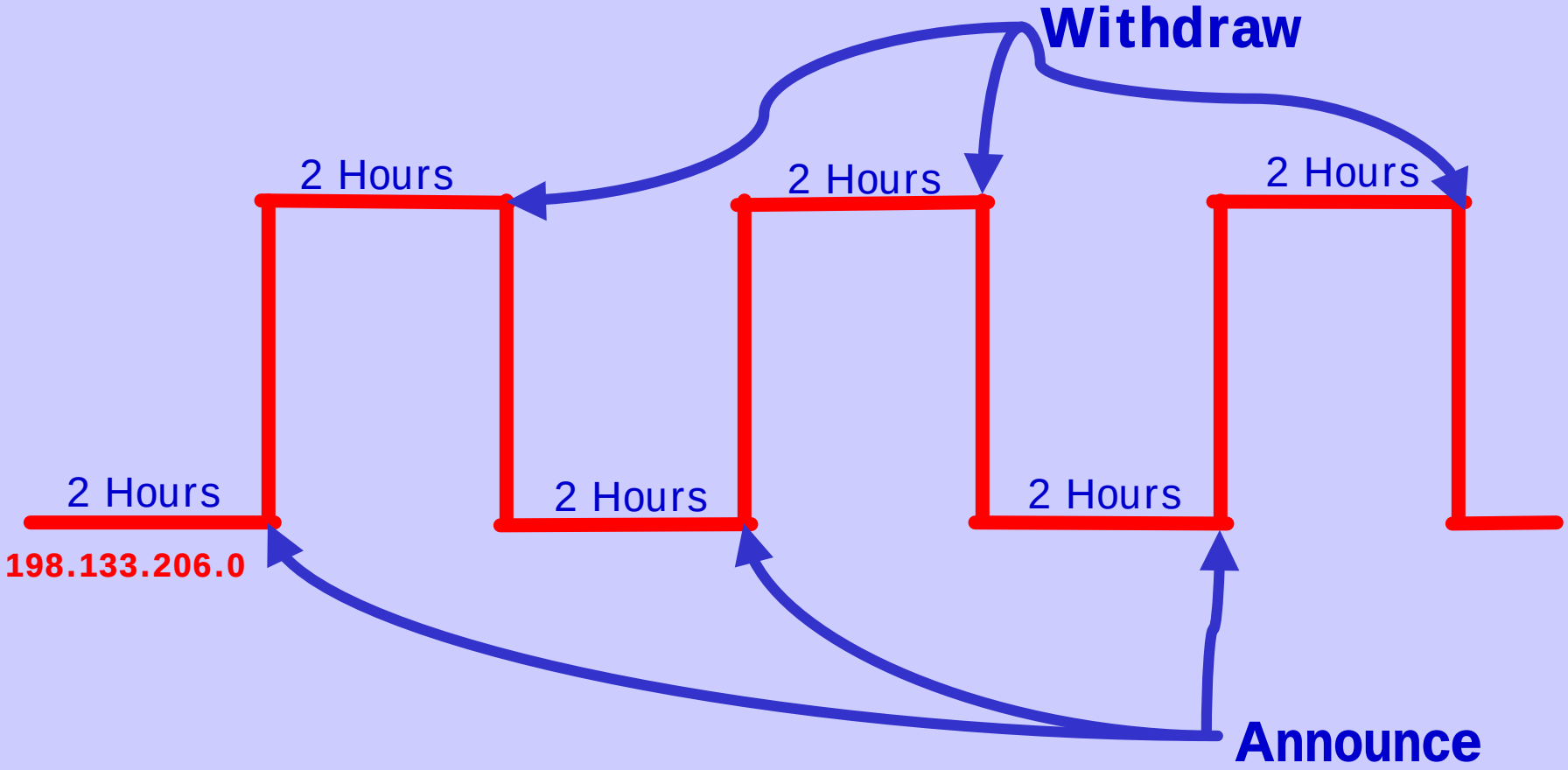
BGP Beacon

BGP Beacon:

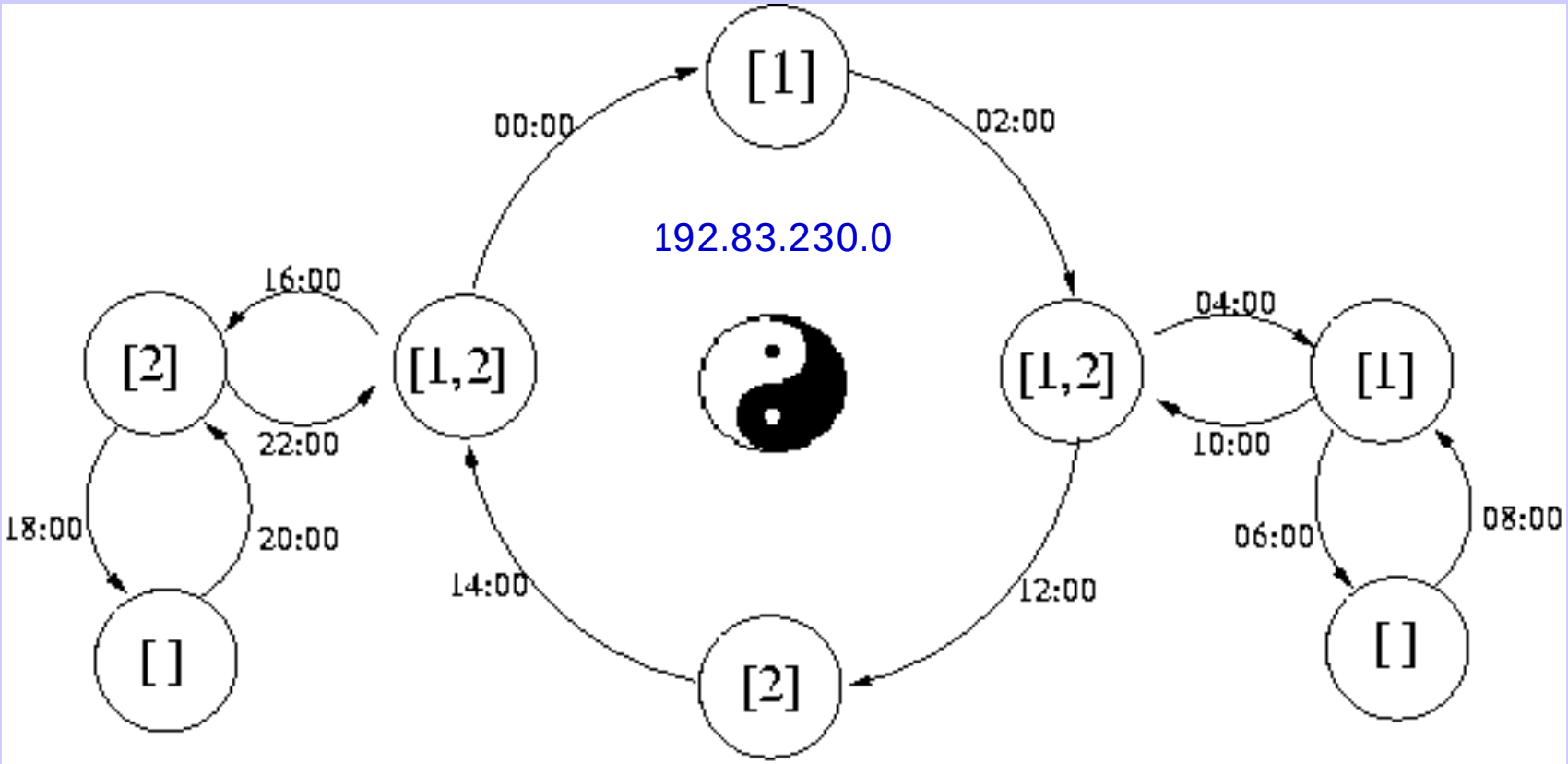
A prefix that is Announced and Withdrawn at **well-known** times



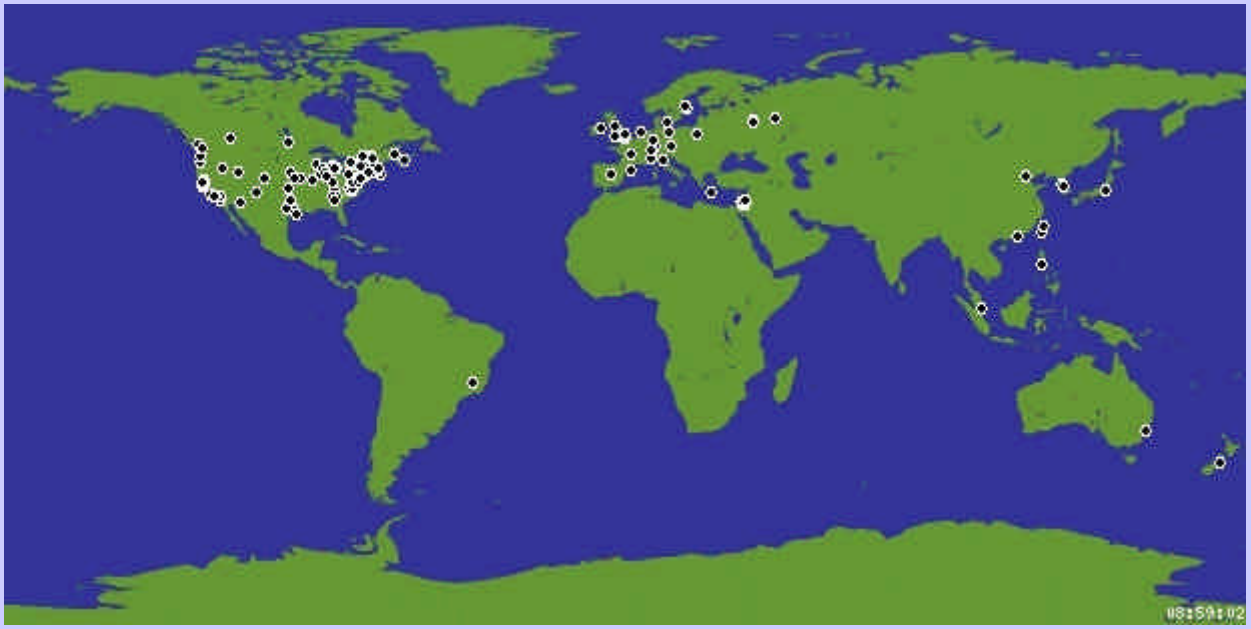
BGP Beacons
Announce & Withdraw



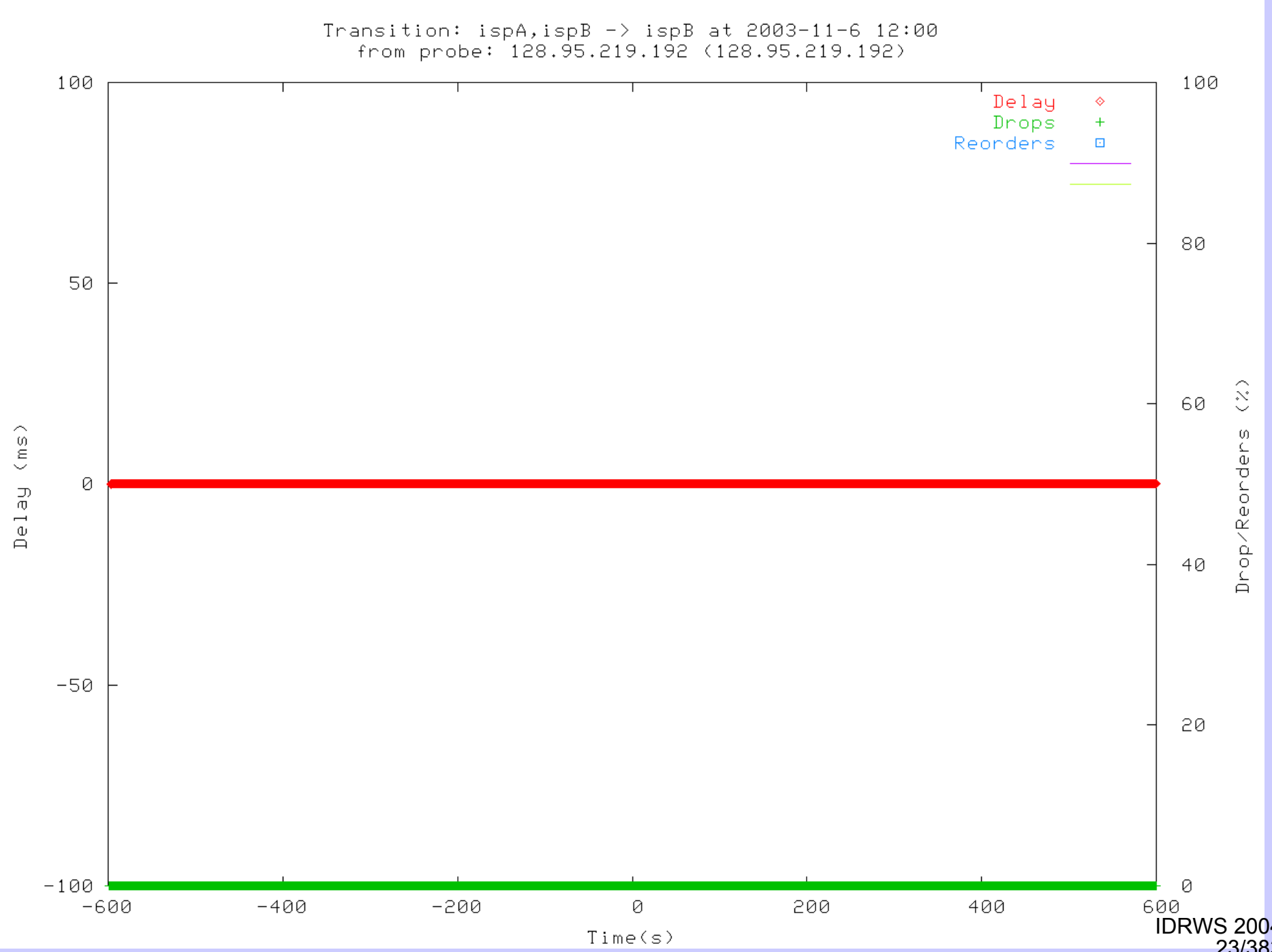
Multi-Homed Beacon

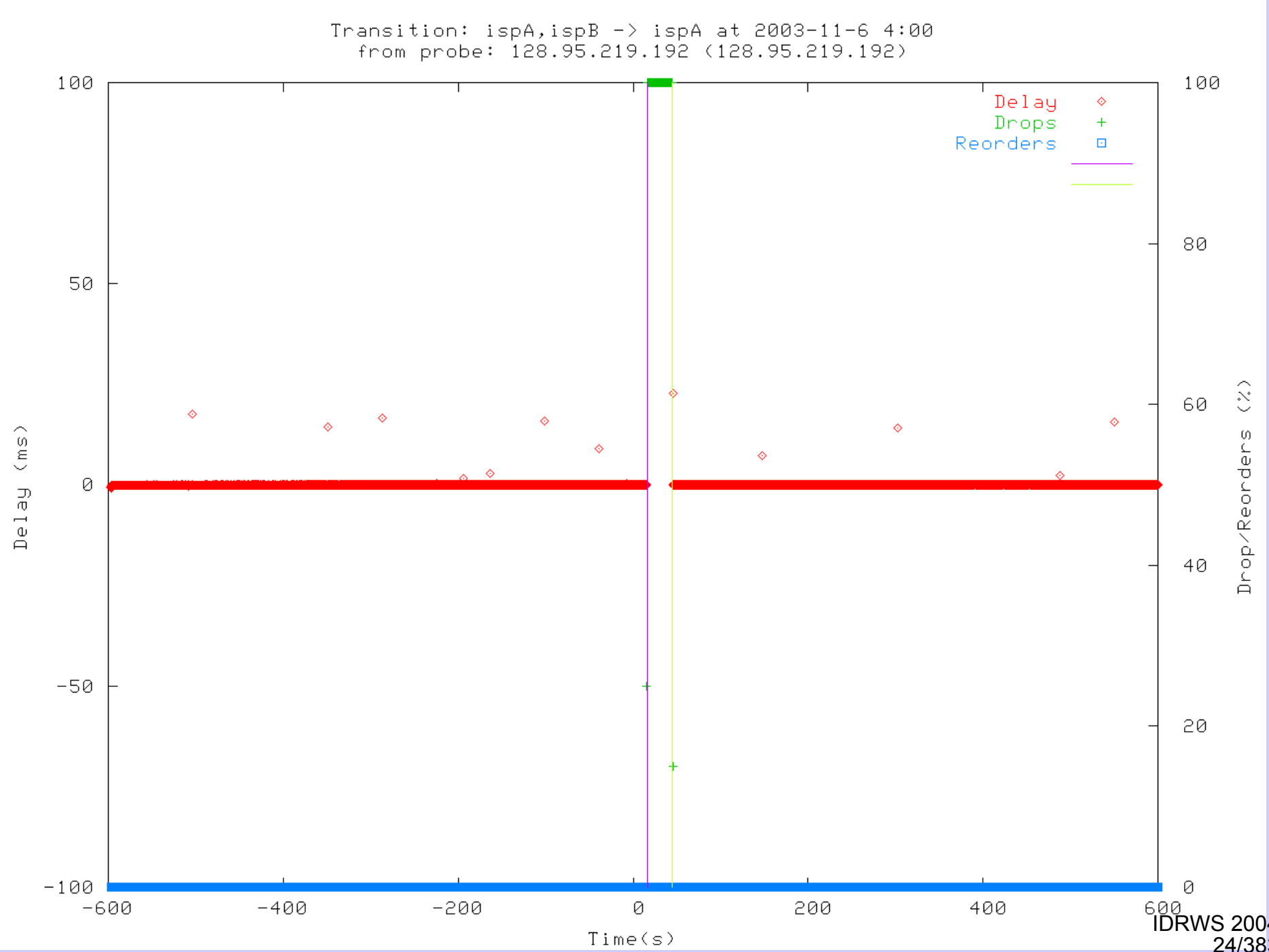


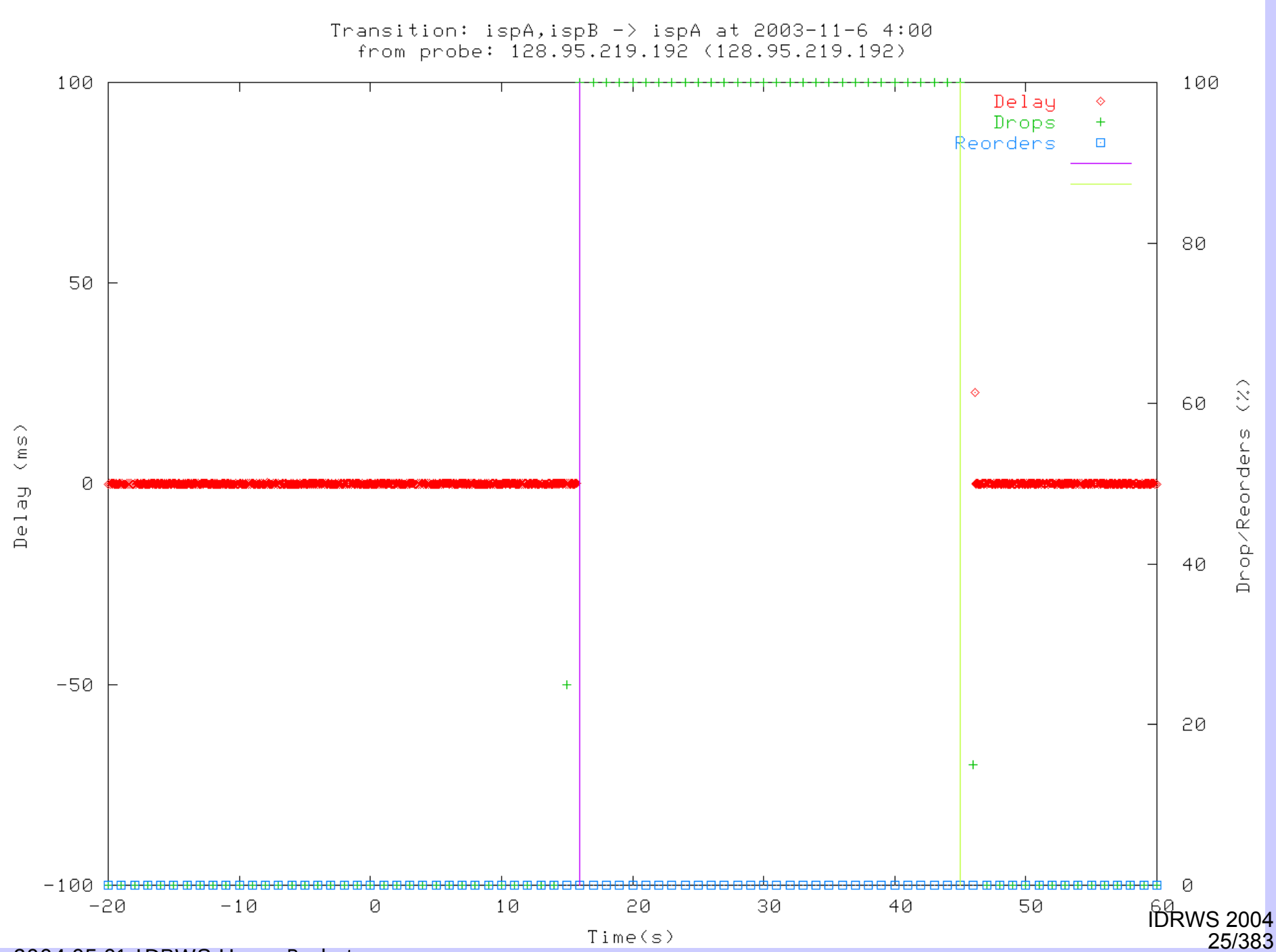
Packet Stream Sources on PlanetLab (and RON)

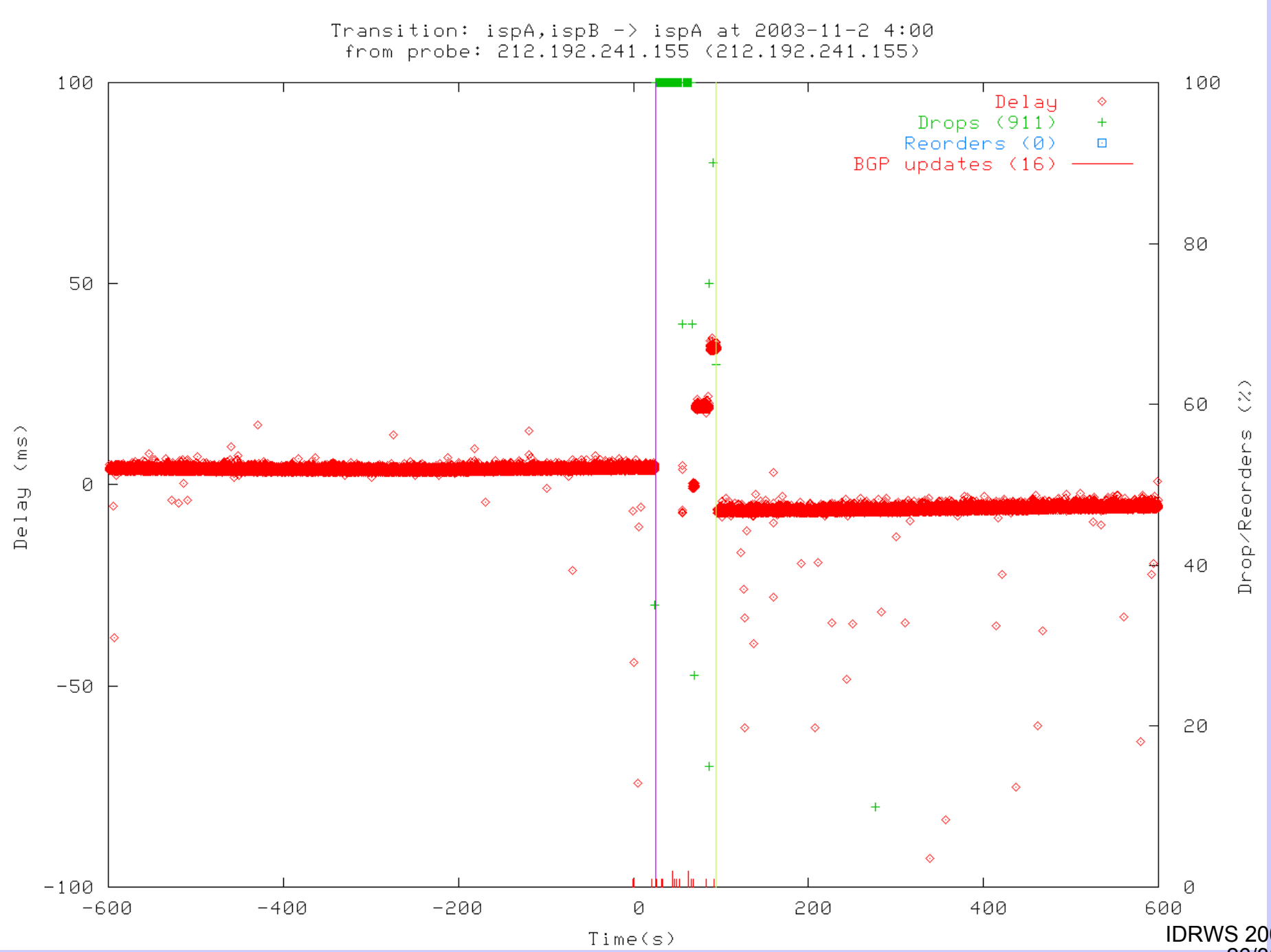


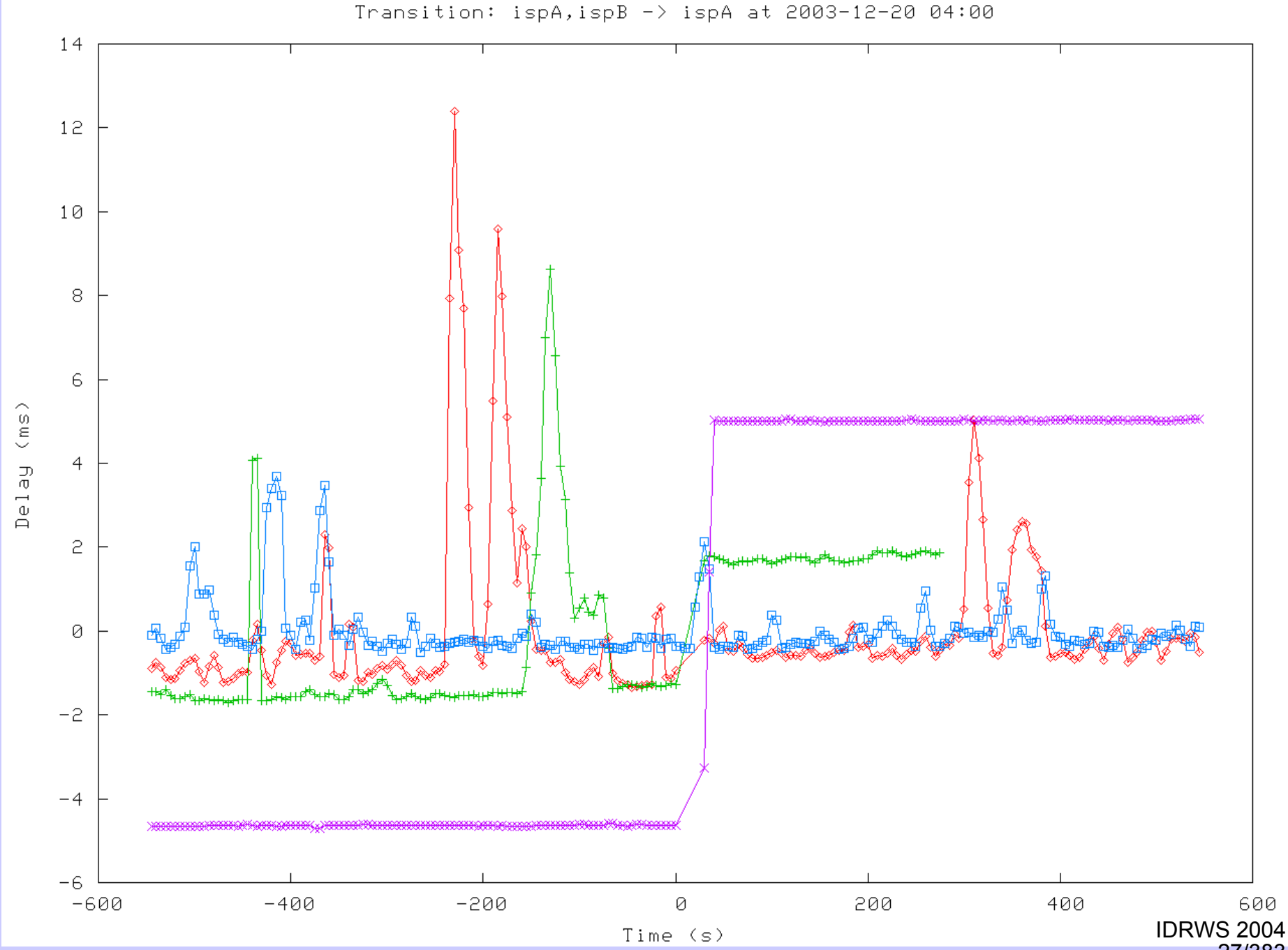
370 nodes at 155 sites
Biased toward R&E Networks
<http://planet-lab.org>

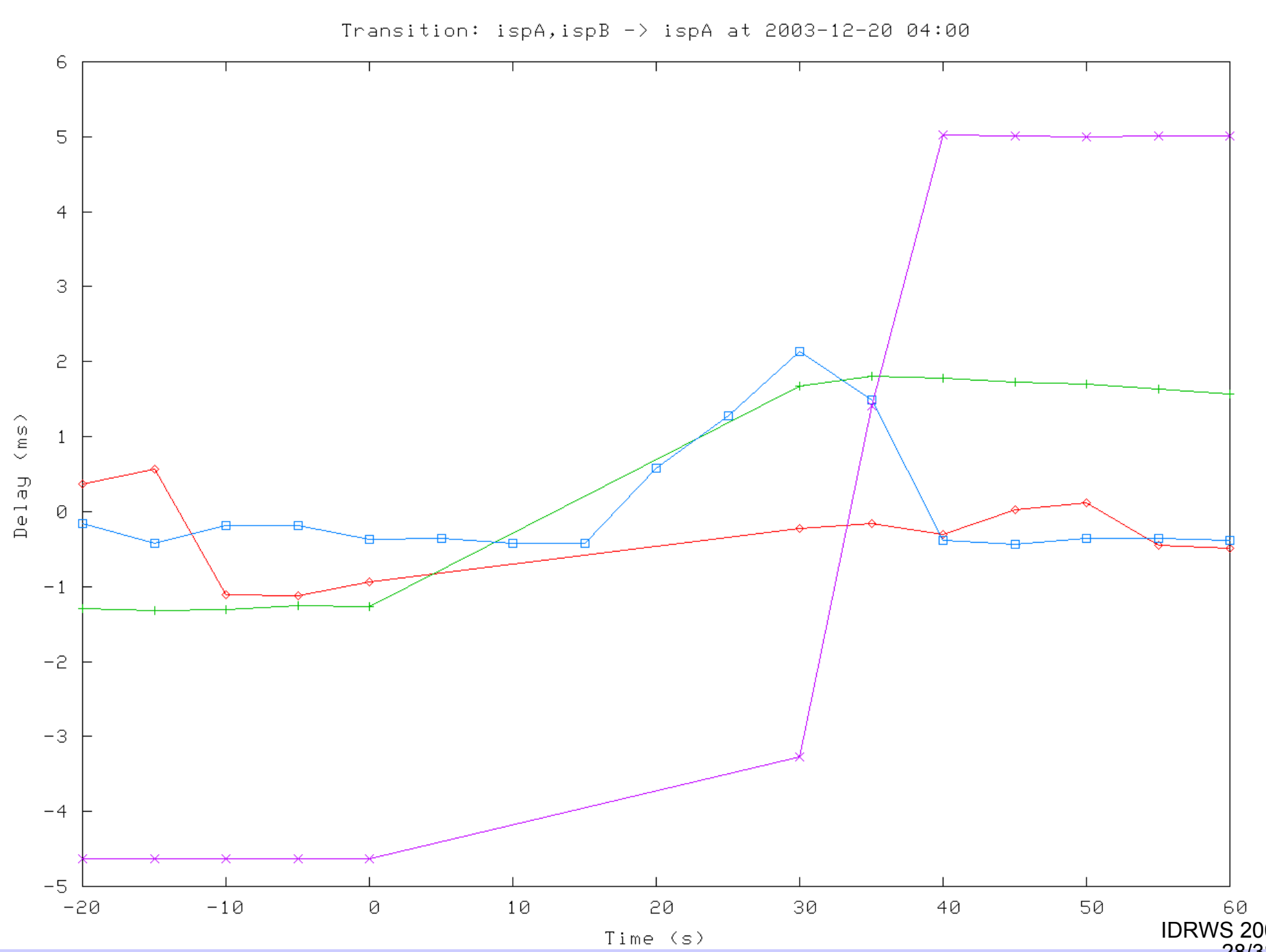

Transition: ispA,ispB -> ispB at 2003-11-6 12:00
from probe: 128.95.219.192 (128.95.219.192)
Delay
Drops
Reorders
100
50
0
-50
-100
Delay (ms)
100
80
60
40
20
0
Drop/Reorders (%)
-600
-400
-200
0
200
400
600
Time(s)

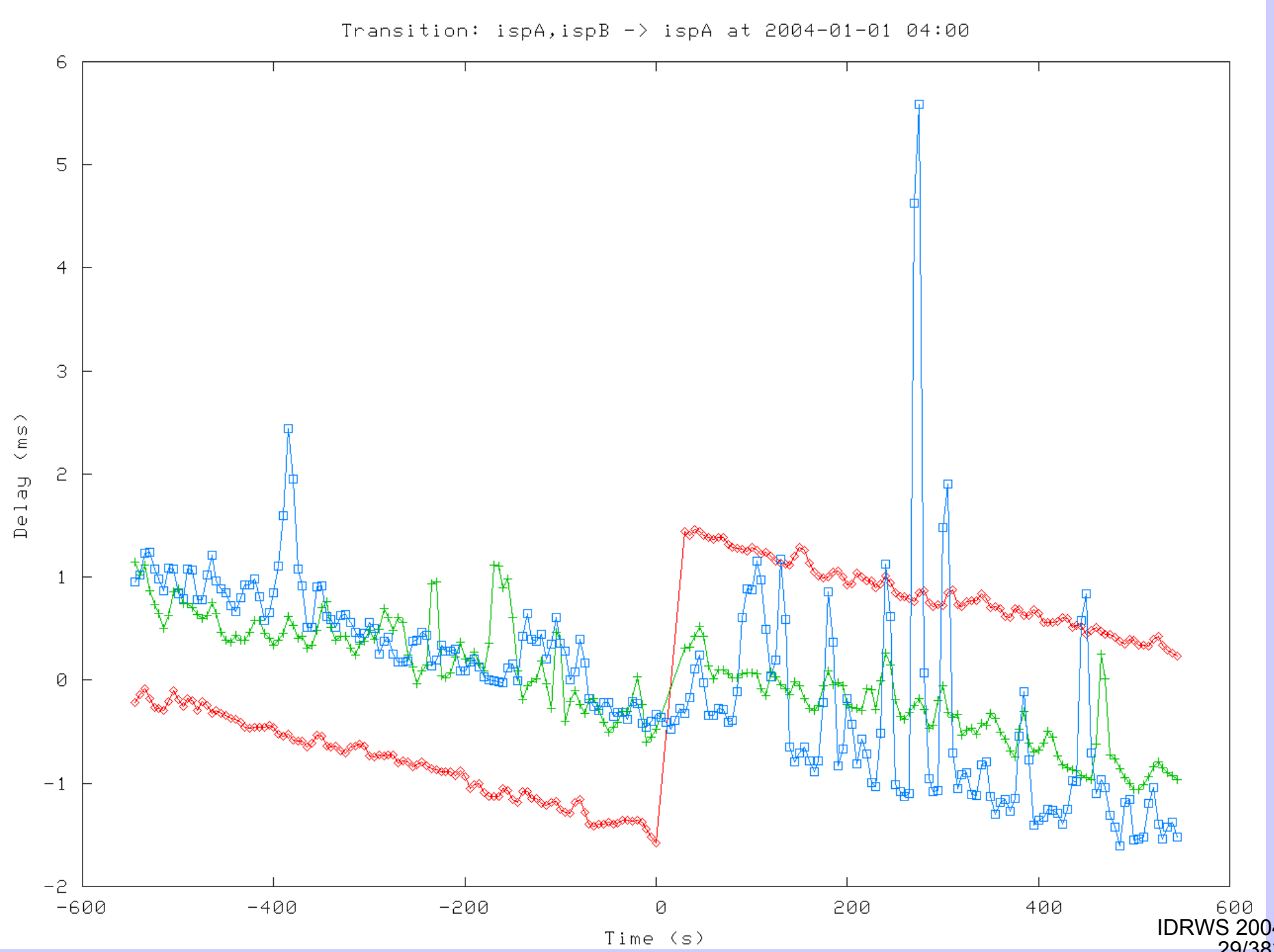

Transition: ispA,ispB -> ispA at 2003-11-6 4:00
from probe: 128.95.219.192 (128.95.219.192)
Delay
Drops
Reorders
Delay (ms)
Drop/Reorders (%)
Time(s)
100
50
0
-50
-100
80
60
40
20
-600
-400
-200
200
400
600

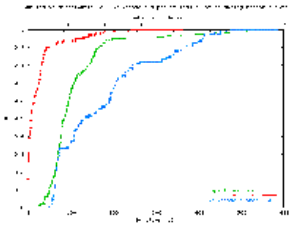
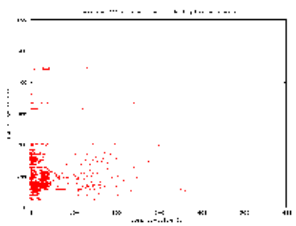
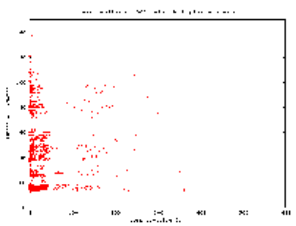
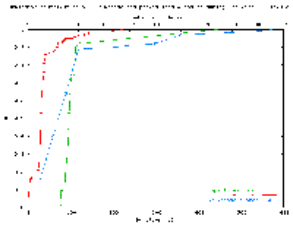
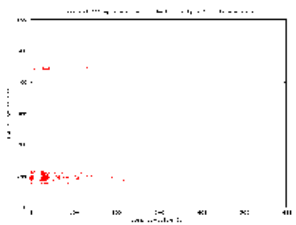
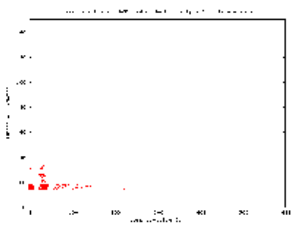
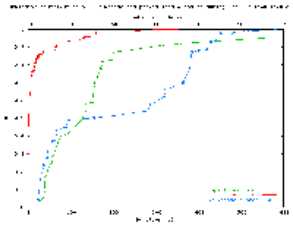
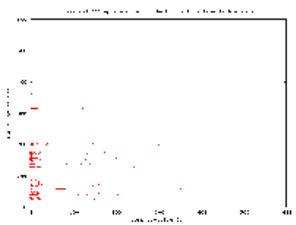
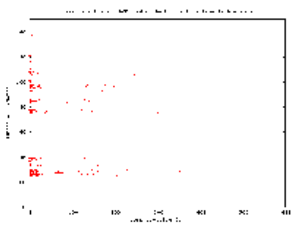
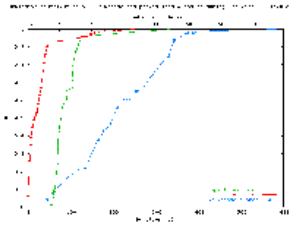
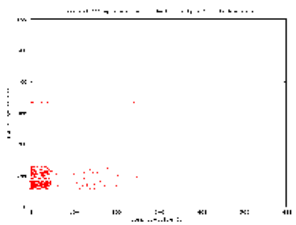
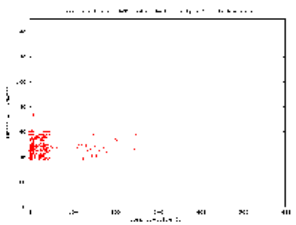
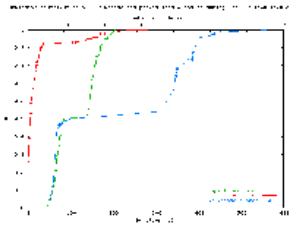
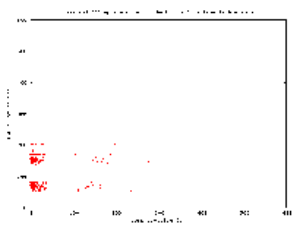
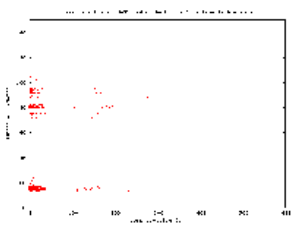

Transition: ispA,ispB -> ispA at 2003-11-6 4:00
from probe: 128.95.219.192 (128.95.219.192)
Delay
Drops
Reorders
100
50
0
-50
-100
Delay (ms)
100
80
60
40
20
0
Drop/Reorders (%)
-20
-10
0
10
20
30
40
50
60
Time(s)

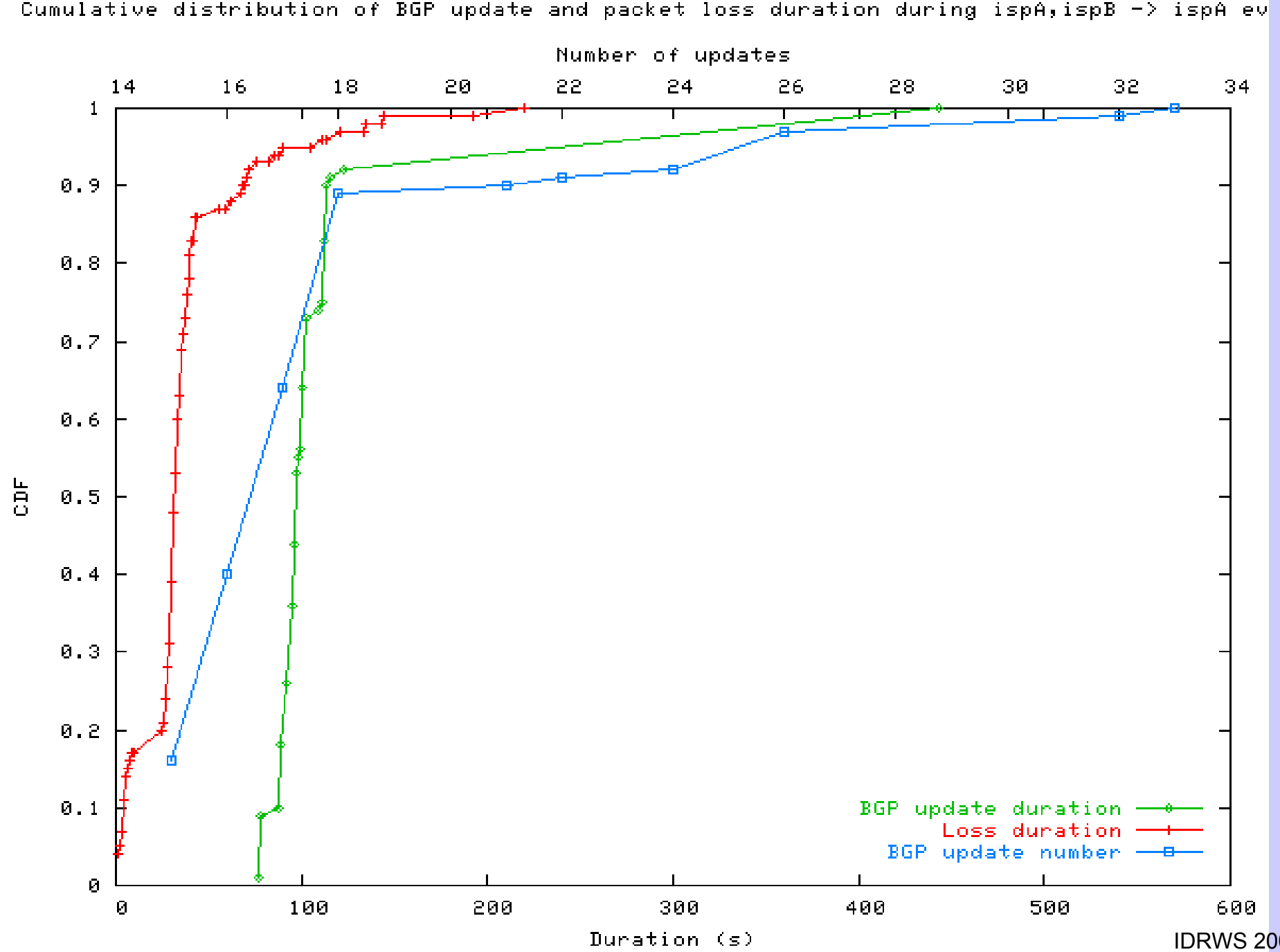

Transition: ispA,ispB -> ispA at 2003-11-2 4:00
from probe: 212.192.241.155 (212.192.241.155)
Delay
Drops (911)
Reorders (0)
BGP updates (16)
Delay (ms)
Drop/Reorders (%)
Time(s)
100
50
0
-50
-100
80
60
40
20
-600
-400
-200
200
400
600

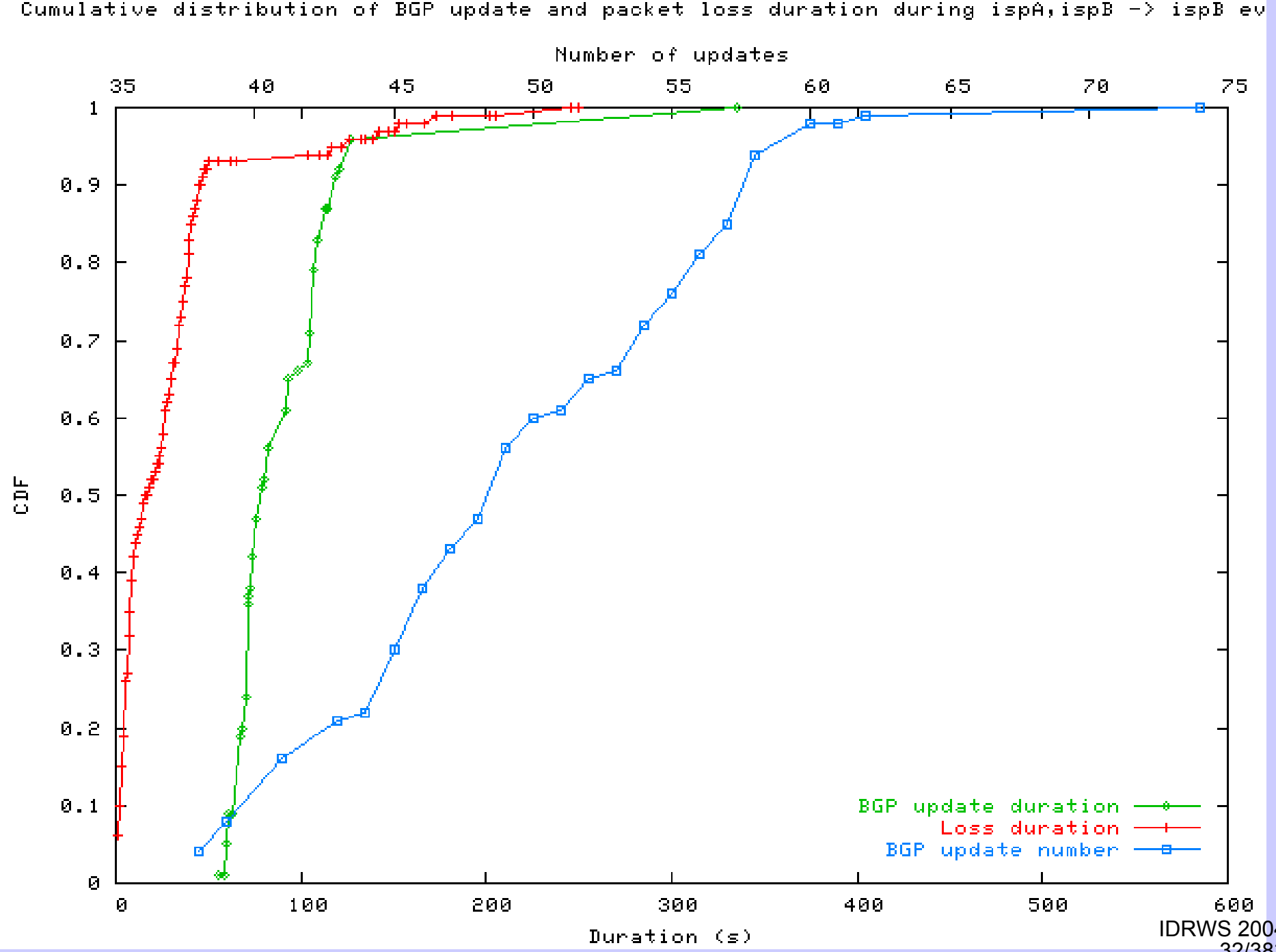

Transition: ispA,ispB -> ispA at 2003-12-20 04:00
Delay (ms)
Time (s)
14
12
10
8
6
4
2
0
-2
-4
-6
-600
-400
-200
0
200
400
600

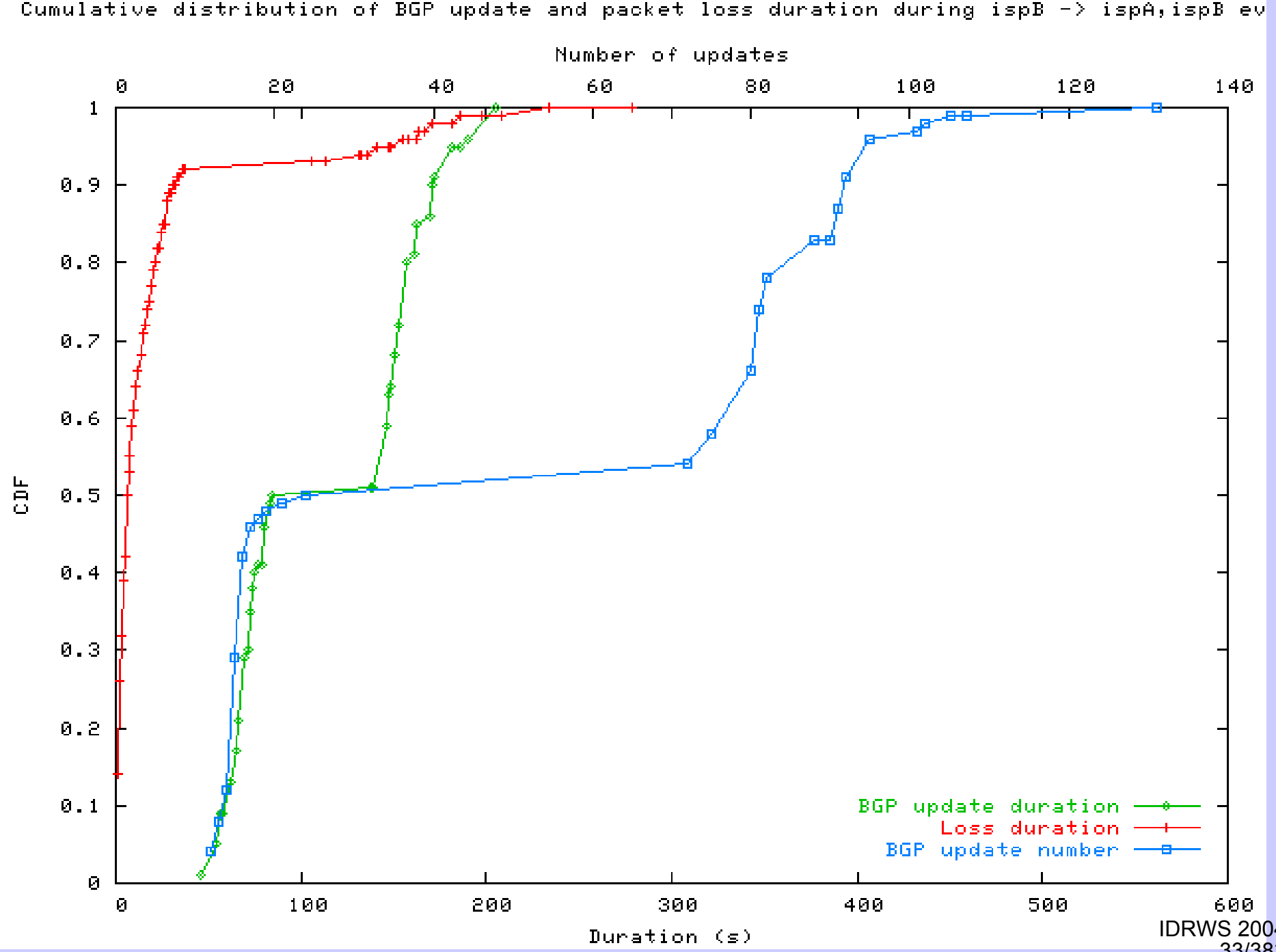

Transition: ispA,ispB -> ispA at 2003-12-20 04:00
Delay (ms)
Time (s)
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-20
-10
0
10
20
30
40
50
60

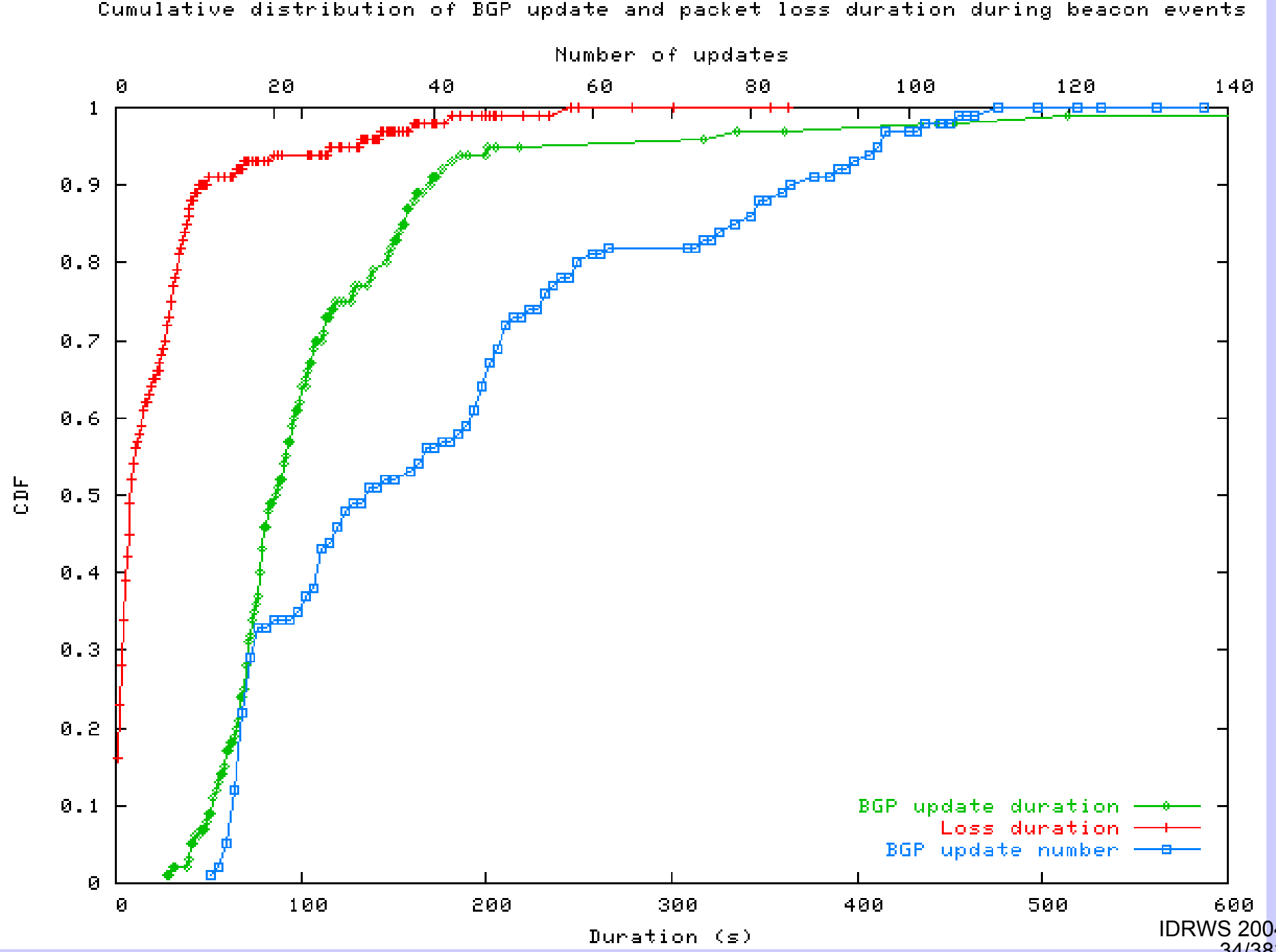

Transition: ispA,ispB -> ispA at 2004-01-01 04:00
Delay (ms)
Time (s)
6
5
4
3
2
1
0
-1
-2
-600
-400
-200
0
200
400
600

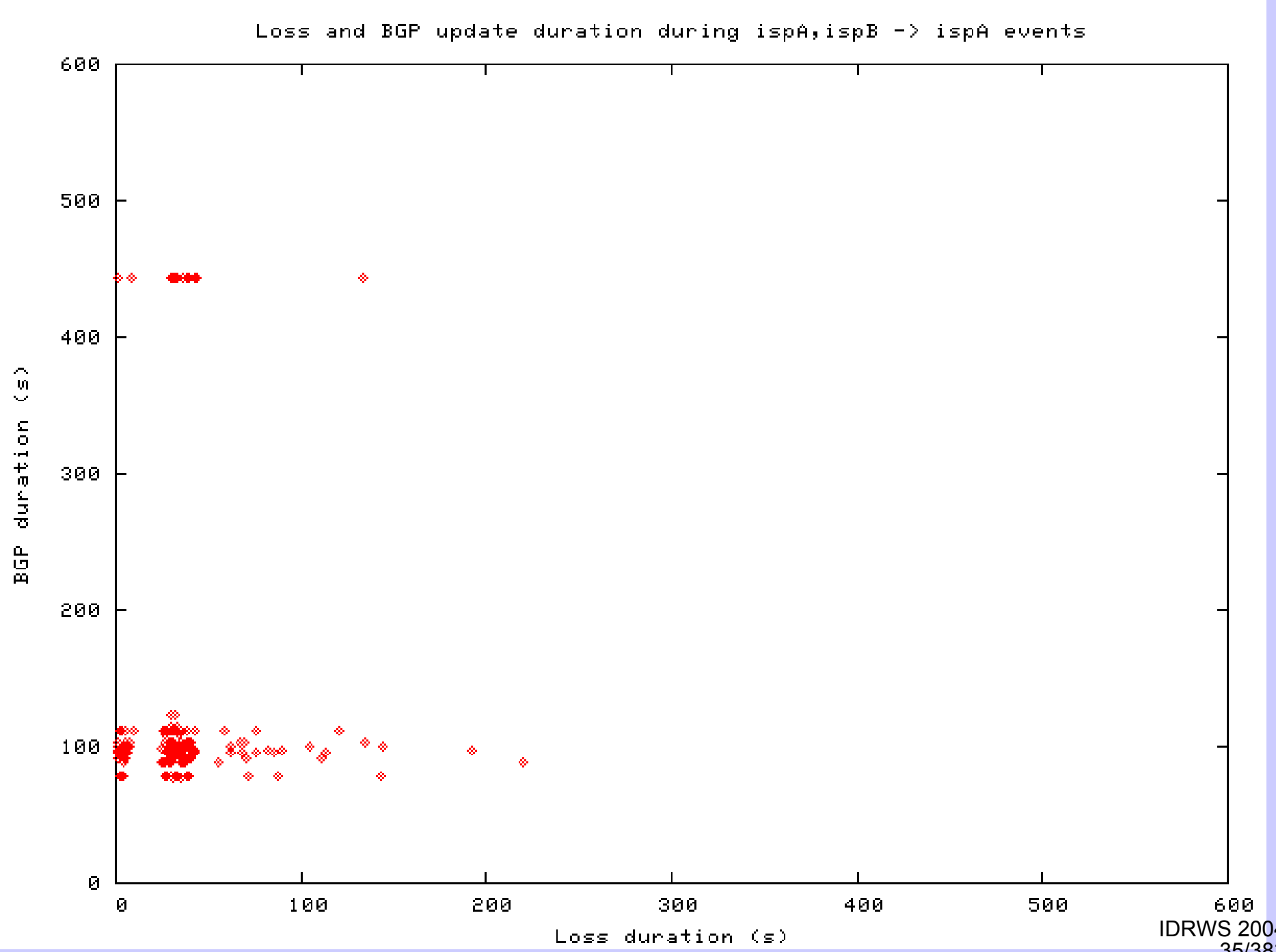
Event	CDF	Loss duration vs BGP duration	Loss duration vs BGP count
All	 [ps][eps]	 [ps][eps]	 [ps][eps]
ispA,ispB -> ispA	 [ps][eps]	 [ps][eps]	 [ps][eps]
ispA -> ispA,ispB	 [ps][eps]	 [ps][eps]	 [ps][eps]
ispA,ispB -> ispB	 [ps][eps]	 [ps][eps]	 [ps][eps]
ispB -> ispA,ispB	 [ps][eps]	 [ps][eps]	 [ps][eps]

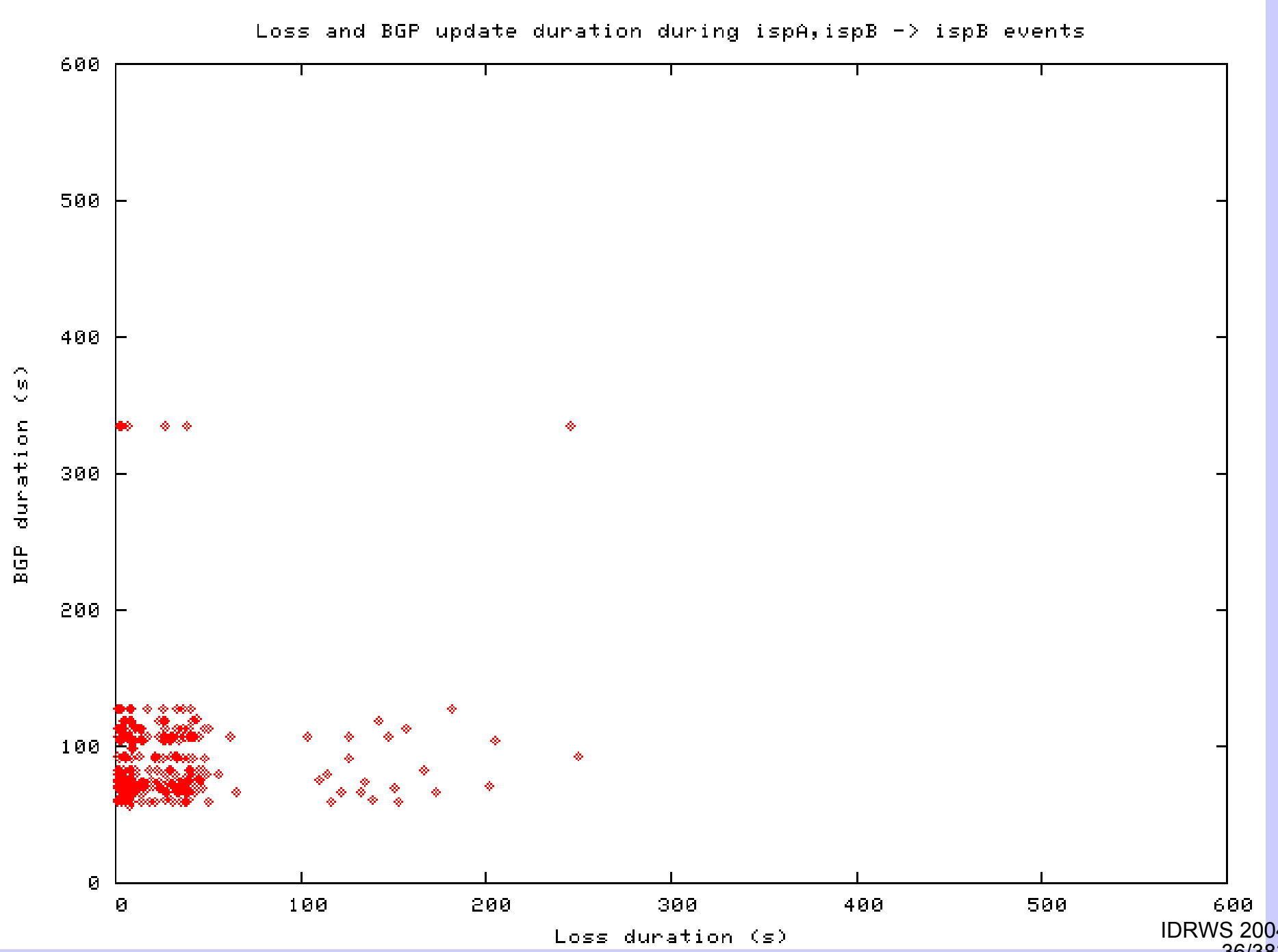

Cumulative distribution of BGP update and packet loss duration during ispA,ispB -> ispA ev
Number of updates
14
16
18
20
22
24
26
28
30
32
34
CDF
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
100
200
300
400
500
600
Duration (s)
BGP update duration
Loss duration
BGP update number

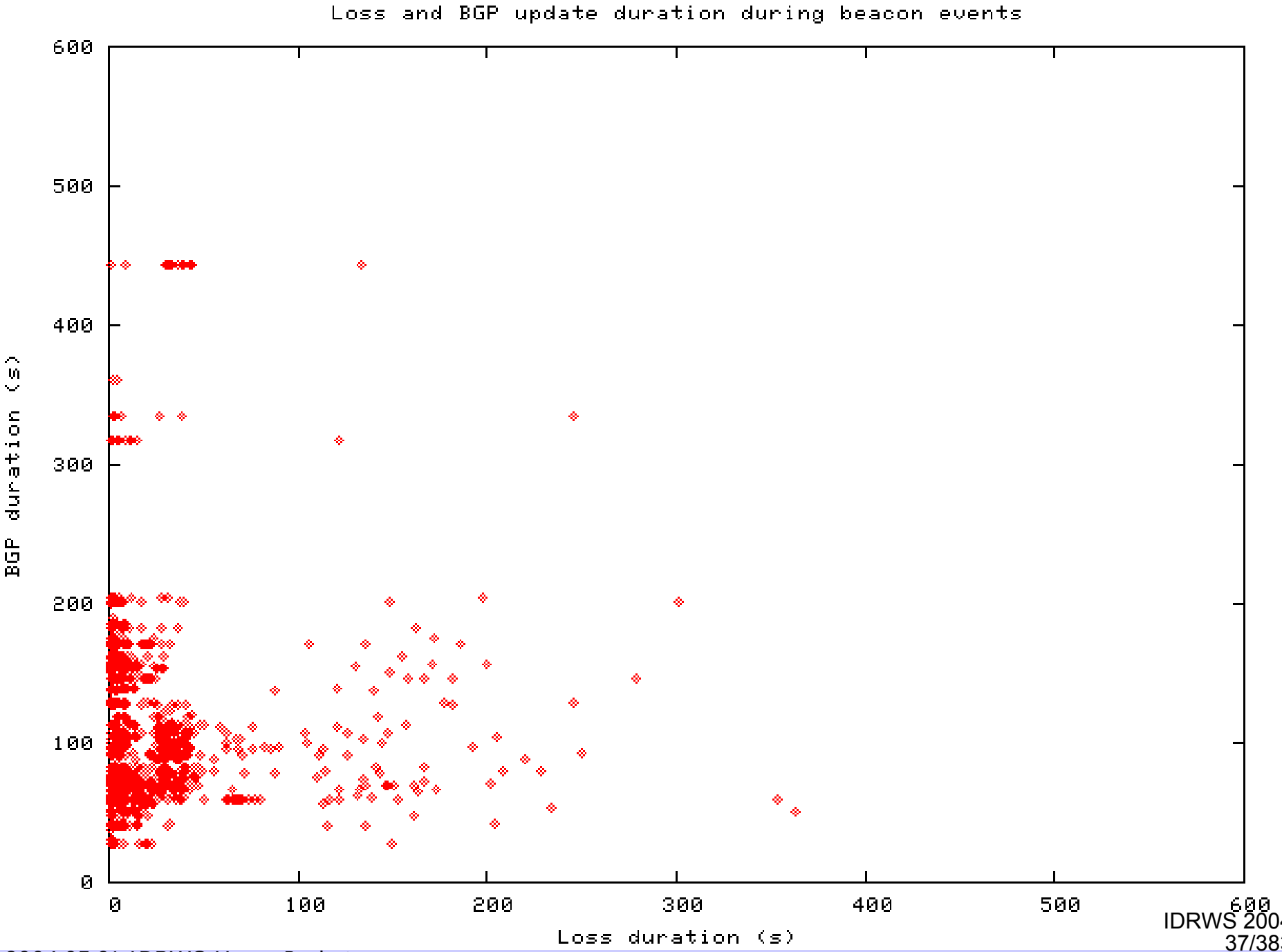

Cumulative distribution of BGP update and packet loss duration during ispA,ispB -> ispB ev
Number of updates
35
40
45
50
55
60
65
70
75
CDF
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
100
200
300
400
500
600
Duration (s)
BGP update duration
Loss duration
BGP update number

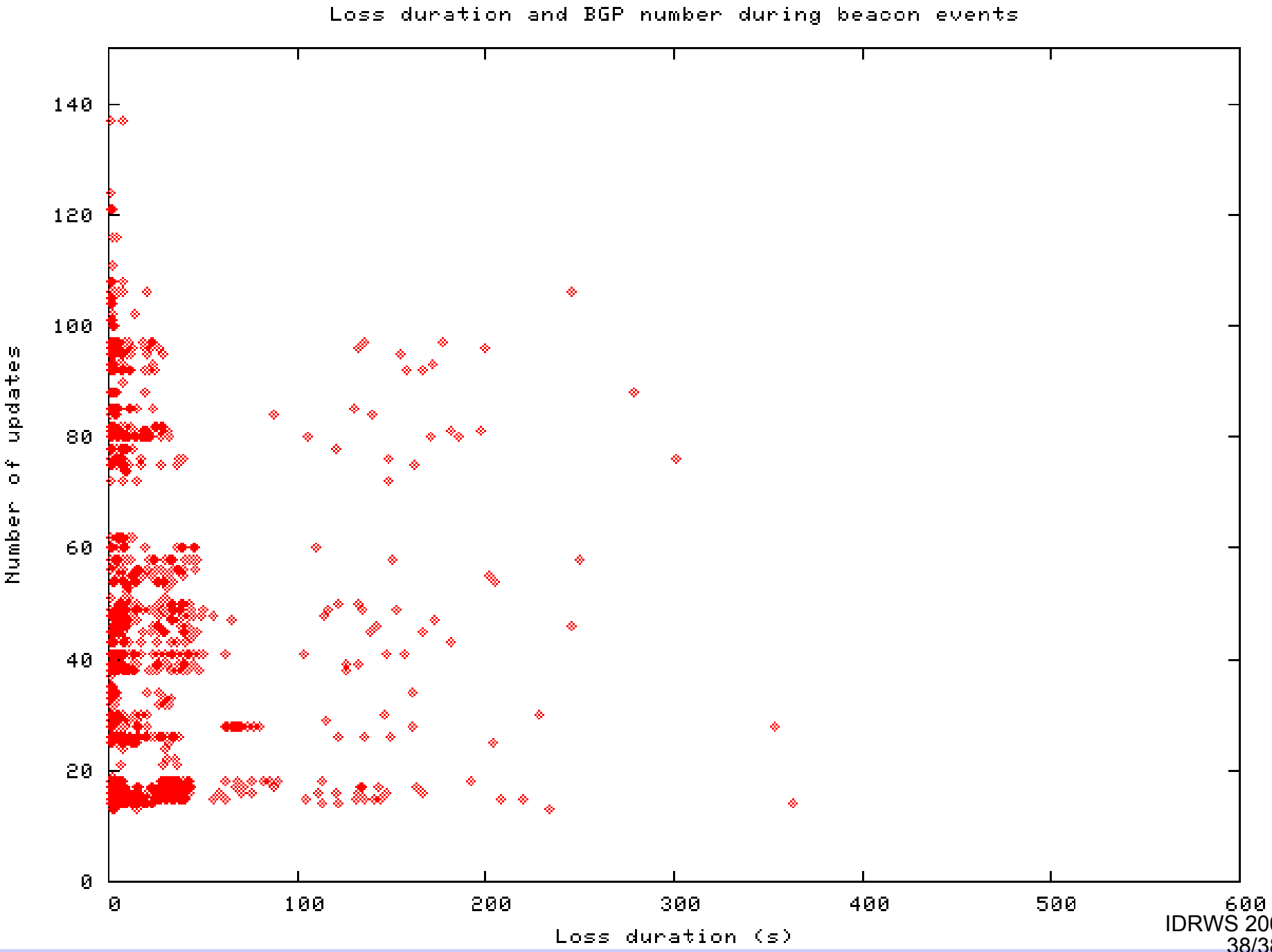

Cumulative distribution of BGP update and packet loss duration during ispB -> ispA,ispB ev
Number of updates
0
20
40
60
80
100
120
140
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
CDF
0
100
200
300
400
500
600
Duration (s)
BGP update duration
Loss duration
BGP update number

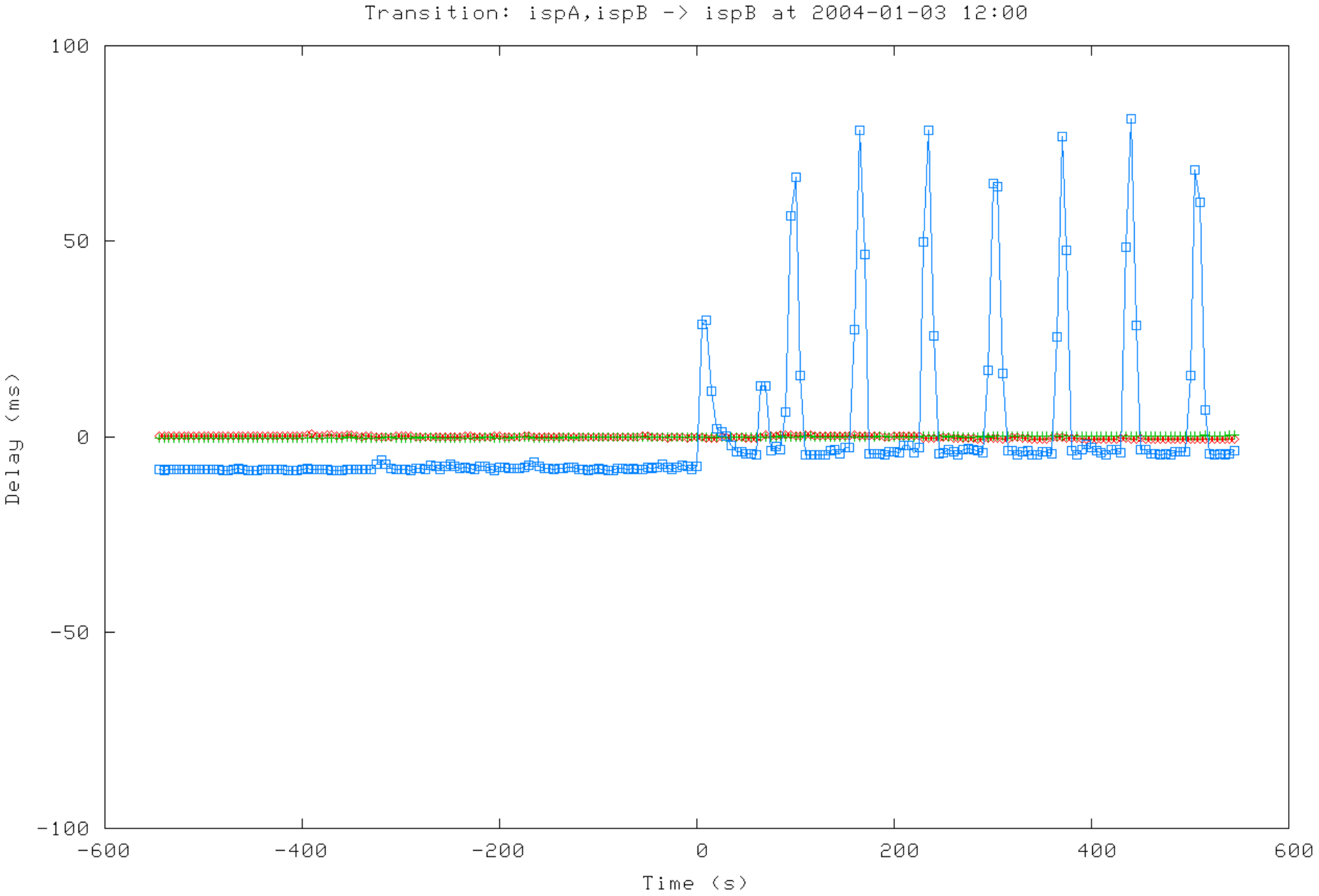

Cumulative distribution of BGP update and packet loss duration during beacon events
Number of updates
0
20
40
60
80
100
120
140
CDF
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
100
200
300
400
500
600
Duration (s)
BGP update duration
Loss duration
BGP update number

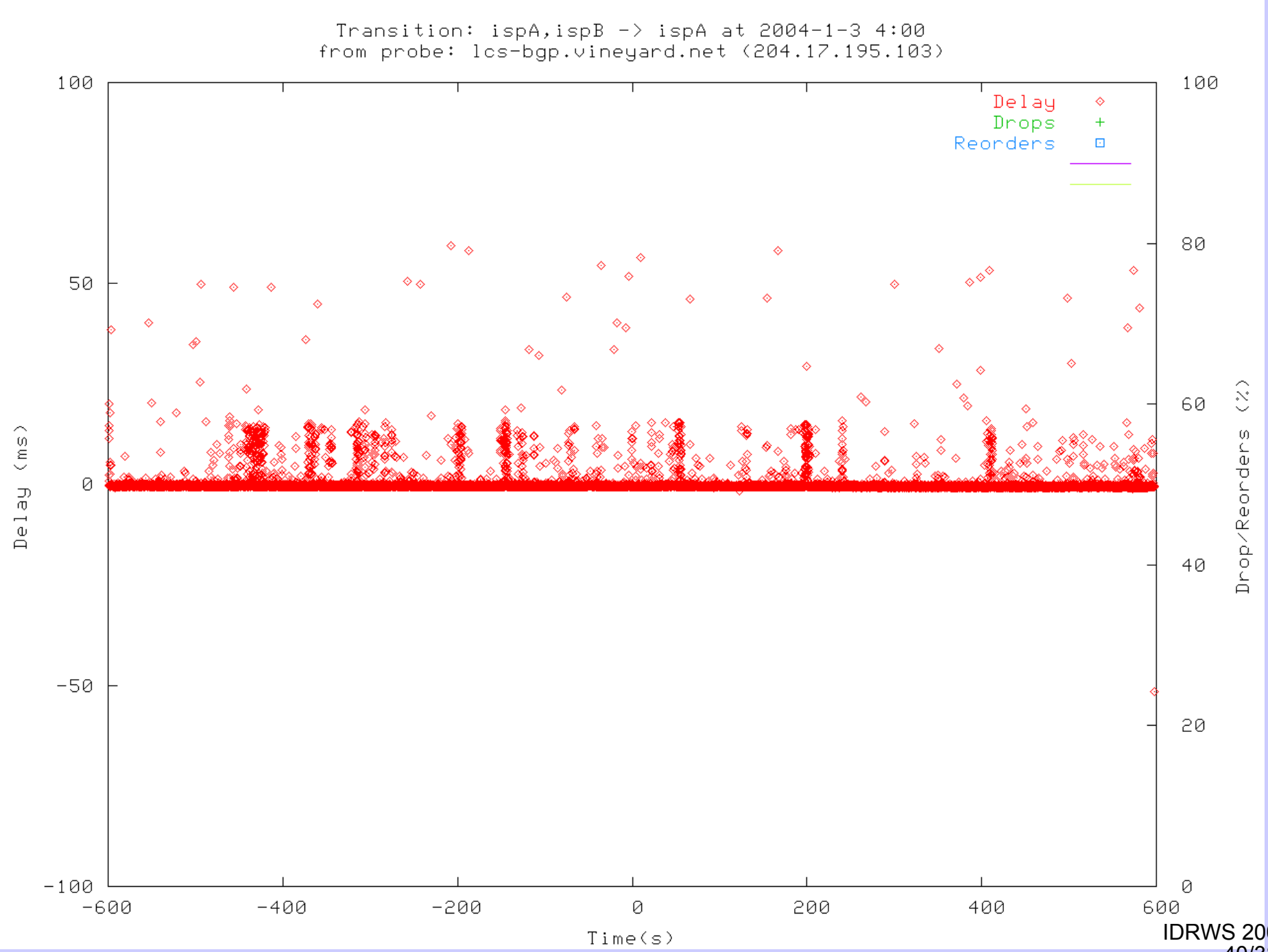

Loss and BGP update duration during ispA,ispB -> ispA events
BGP duration (s)
Loss duration (s)
0
100
200
300
400
500
600

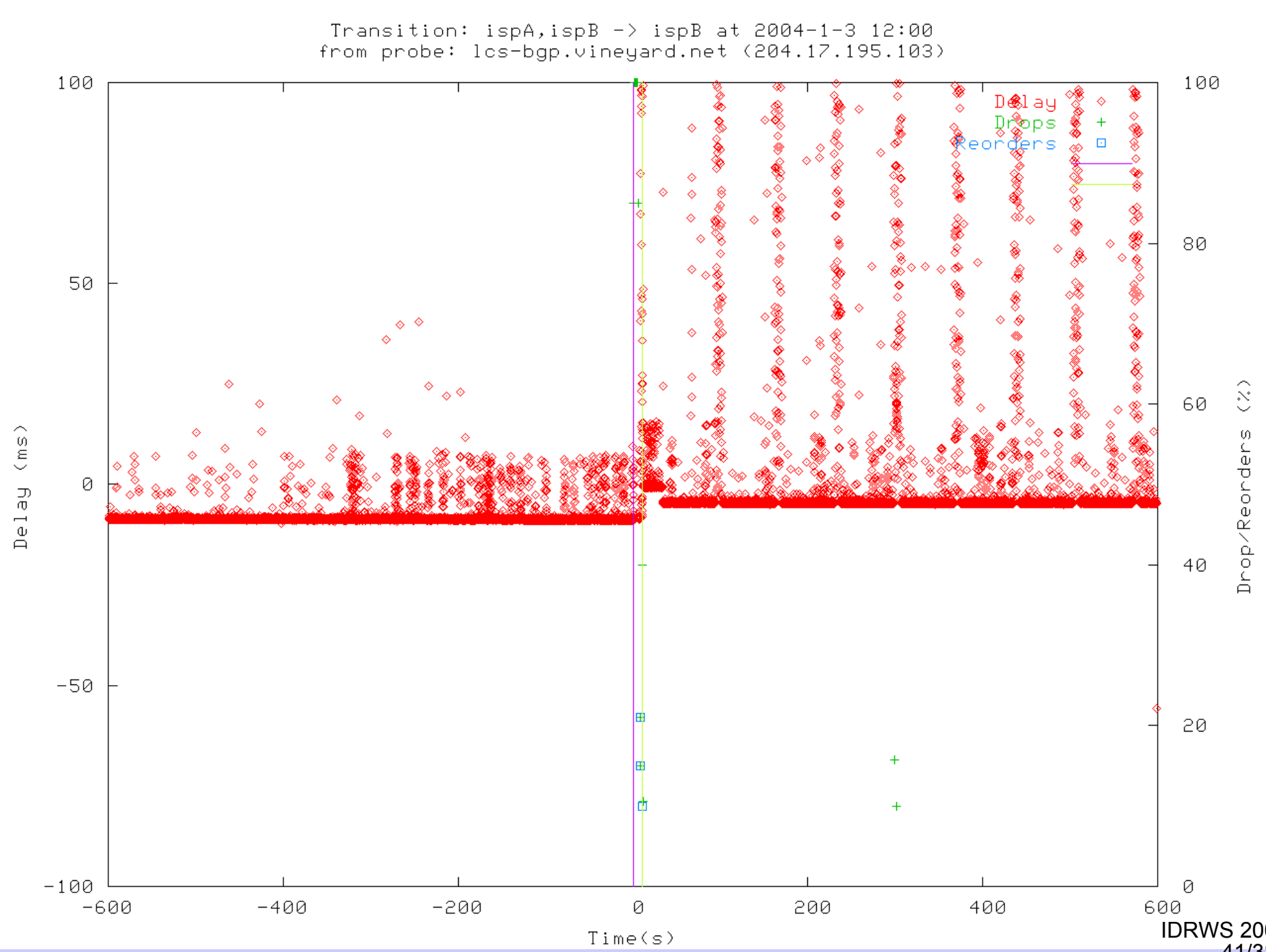

Loss and BGP update duration during ispA,ispB -> ispB events
BGP duration (s)
Loss duration (s)
0
100
200
300
400
500
600

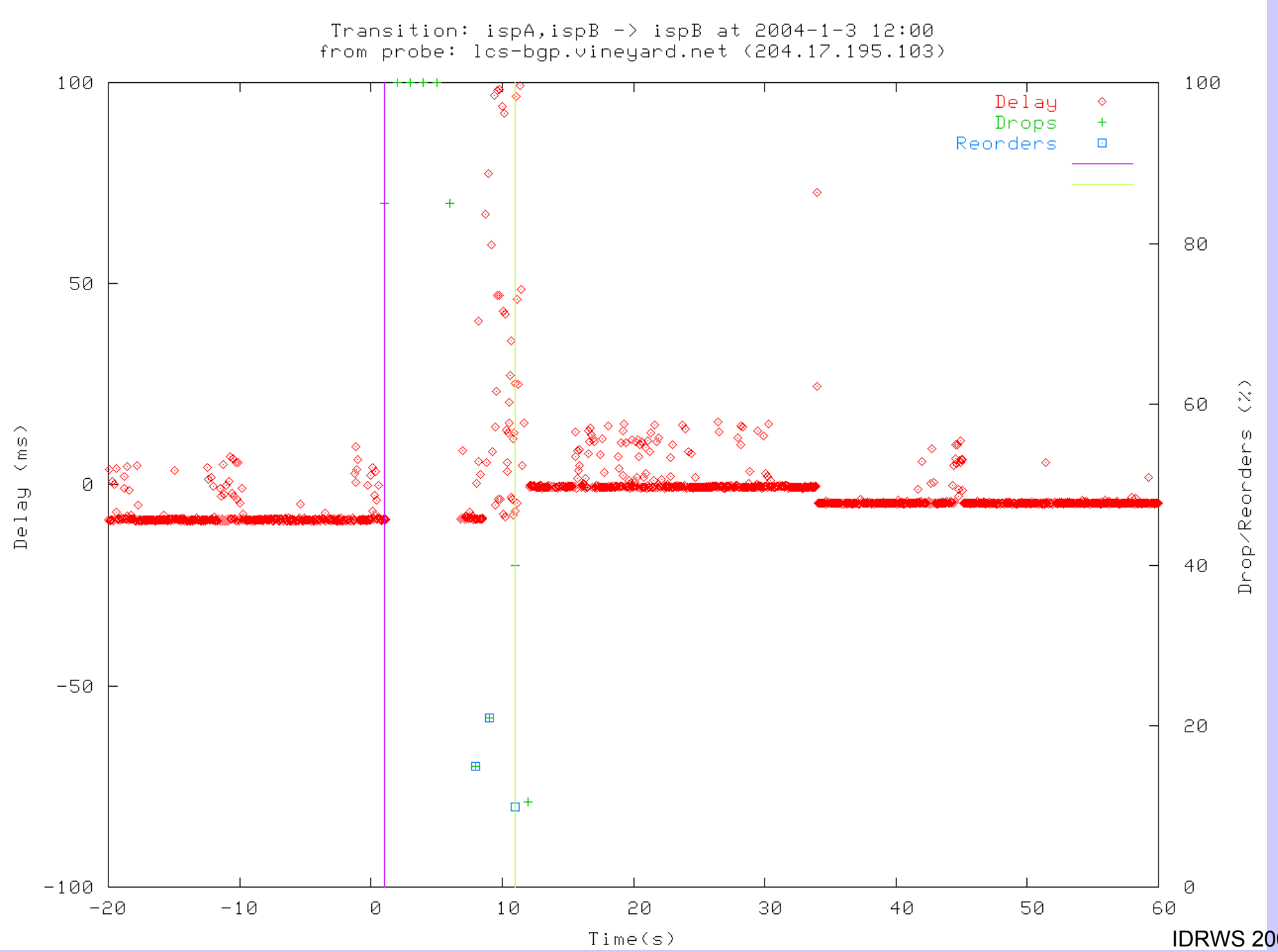

Loss and BGP update duration during beacon events
BGP duration (s)
Loss duration (s)
0
100
200
300
400
500
600


Loss duration and BGP number during beacon events
Number of updates
0
20
40
60
80
100
120
140
0
100
200
300
400
500
600
Loss duration (s)


Transition: ispA,ispB -> ispB at 2004-01-03 12:00
100
50
0
-50
-100
Delay (ms)
-600
-400
-200
0
200
400
600
Time (s)
206.197.119.144 (206.197.119.144)
mit-bgp-test.playground.net (131.149.6.42)
lcs-bgp.vineyard.net (204.17.195.103)


Transition: ispA,ispB -> ispA at 2004-1-3 4:00
from probe: lcs-bgp.vineyard.net (204.17.195.103)
Delay
Drops
Reorders
100
50
0
-50
-100
Delay (ms)
100
80
60
40
20
0
Drop/Reorders (%)
-600
-400
-200
0
200
400
600
Time(s)


Transition: ispA,ispB -> ispB at 2004-1-3 12:00
from probe: lcs-bgp.vineyard.net (204.17.195.103)
Delay
Drops
Reorders
Delay (ms)
Drop/Reorders (%)
Time(s)
100
50
0
-50
-100
80
60
40
20
-600
-400
-200
200
400
600


Transition: ispA,ispB -> ispB at 2004-1-3 12:00
from probe: lcs-bgp.vineyard.net (204.17.195.103)
Delay
Drops
Reorders
Delay (ms)
Drop/Reorders (%)
Time(s)
100
50
0
-50
-100
80
60
40
20
-20
-10
10
20
30
40
50
60

References

- Tim Griffin, “What is the sound of one route flapping” Dartmouth talk slides, June 2002
- "Global Routing Instabilities During Code Red II and Nimda Worm Propagation“ Jim Cowie and Andy Ogielski, Renesys. NANOG 23, October 2001
- C. Labovitz, G. R. Malan, F. Jahanian, “Internet Routing Instability”, TON 1998
- C. Labovitz, R. Malan, F. Jahanian, “Origins of Internet Routing Instability”, Infocom 1999

References (2)

- C. Labovitz, A. Ahuja, F. Jahanian, “Experimental Study of Internet Stability and Wide-Area Network Failures”, FTCS 1999
- C. Labovitz, A. Ahuja, A. Bose, F. Jahanian, “Delayed Internet Routing Convergence” Sigcomm 2000
- C. Labovtiz, A. Ahuja, R. Wattenhofer, S. Venkatachary, “The Impact of Internet Policy and Topology on Delayed Routing Convergence”, Infocom 2001

Alternate Universe: An Internet Where Routers Don’t Route

IDRWS 2004

Simon Leinen, SWITCH <simon@switch.ch>

Remember source routing?

The Internet used to have it as an option:

Loose and Strict Source Routing, but

- many networks block it "for security reasons"
- most routers don’t support it efficiently
- limited usability in IPv4 (better in IPv6)
- lack of mechanisms to find alternate paths

So source routing is never used today

- Except for debugging and measurement

Why hosts should choose paths

- Application of an end-to-end argument
- Different applications prefer different paths, e.g.
  - shortest (delay)
  - widest
  - cheapest
  - multiple paths
    - for resilience
- Hard for network to guess the right path(s)
  - Attempts at giving network more information
    - RSVP *++
    - Diffserv +
    - MPLS *
  - *) require path setup +) end-systems involved
- Hosts could just try paths and compare
  - or be arbitrarily smart about selection/combination

What if...

The Internet had ONLY explicit routing

- but no path setup in routers,
- i.e. each packet includes full source route

There were a "path service" (or multiple)

- A host would ask it for paths to a destination
- Paths returned would be decorated with metrics
  - delay, bottleneck capacity
  - cost
  - lifetime

Wouldn’t headers be awfully long?

But you can use compression

- e.g. only encode next-hop indexes
  - If the average router has 10 neighbors, then the average hop could be encoded in four bits, so a 32-hop source route could be as short as an IPv6 address...
- interesting variant: invertible source routes
  - If hop labels are negotiated between adjacent routers, source routes can be used bidirectionally

Hosts need topology knowledge?

Only to get to the path service initially...

Then they could cache paths to hosts they send to.

What about servers with millions of clients?

- With invertible routes, burden is distributed

Is this "Path service" even feasible?

- It would have to scale
- It would have to be fast
  - at least for the first path
- It would have to be very robust

- It could be hierarchical, like DNS
- It could be based on DHTs, cf. peer-to-peer nets
- It could be based on map distribution

But what if something breaks!?

- The source can try another path
  - (or be using it in parallel already)
- Or a router could do local repair
  - (and send ICMPs)

Hasn’t this been trashed long ago?

Well,

- "Source Routing for Campus-wide Internet Transport" (Saltzer, Reed, Clark 1980)
- NIMROD
- UUCP

But now might be a good time to look at it again

- We start to understand Internet-scale networks
- Current routing architecture is showing its limits
- Inspiration from peer-to-peer networks

Internet Nirvana?

The combination of

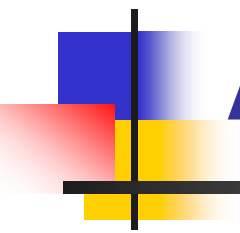
- End-host selectable paths
- Well-performing path service(s)
- Transparent Pricing (shudder) of Links

could result in an Internet with

- Smarter upper layers
  - that can use multiple paths creatively
- Better incentive structure
- Rich topology

Thanks!

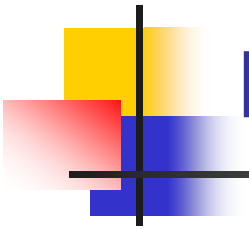
Questions?



Allocations vs Announcements

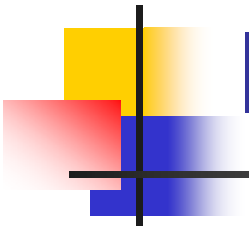
A comparison of RIR IPv4 Allocation Records with Global Routing Announcements

Geoff Huston
May 2004
(Activity supported by APNIC)



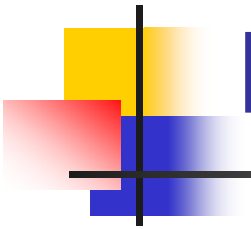
BGP Prefix Length Filters

- Some years back a number of ISPs introduced prefix length filters on the routes they accepted from their peers
- This practice was taken up by others and is now widespread across the Internet
- The filters are typically based on observations of minimum allocation sizes of RIR allocations within /8 address blocks



Implications

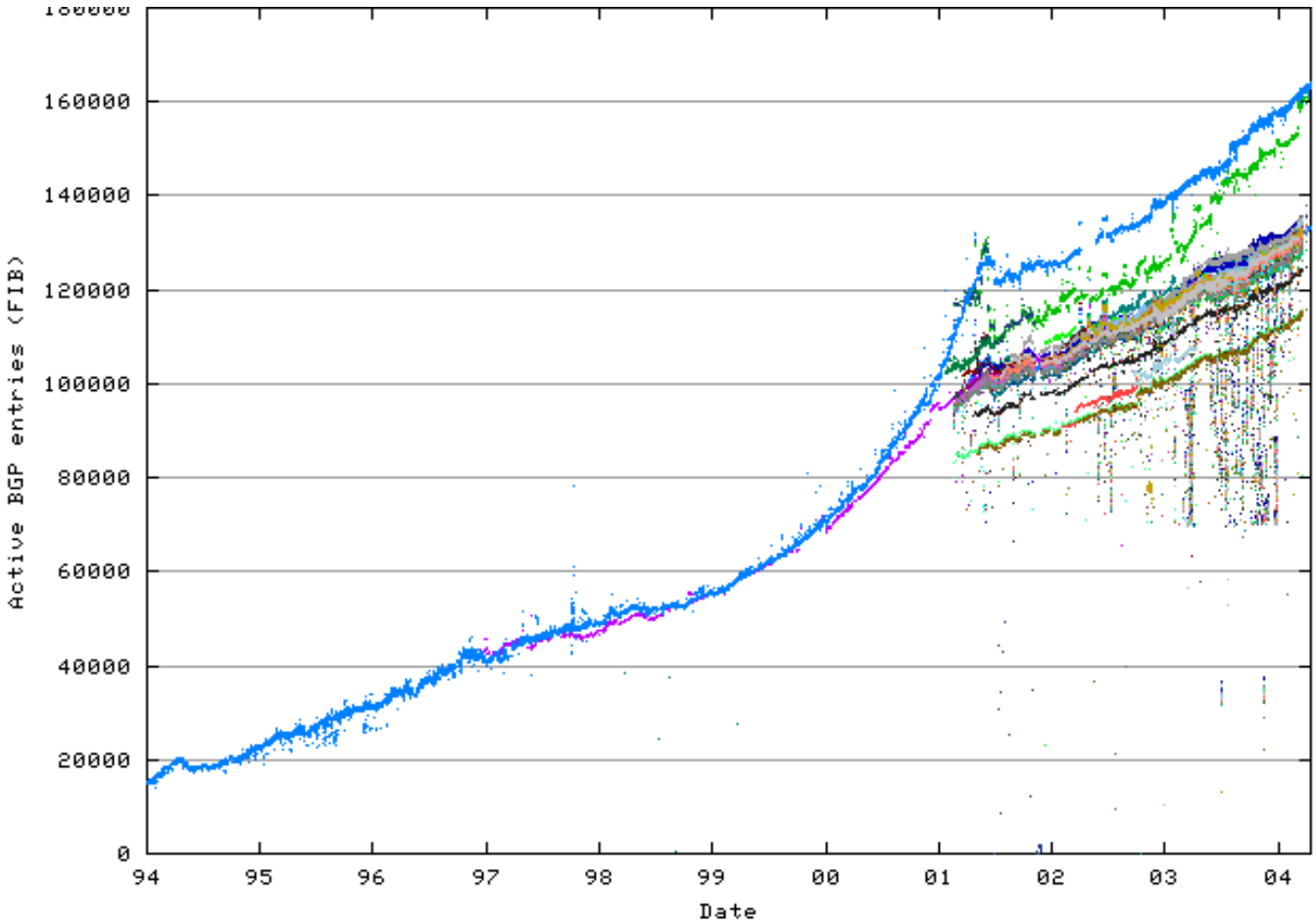
- The generic assumption behind the use of these filters is that:
  - ISPs should globally advertise the RIR allocated address block as a single aggregate
  - If more specific fragments of an RIR allocation are advertised for local resilience and traffic engineering reasons, these fragmentary advertisements should be scoped such that they do not spread globally



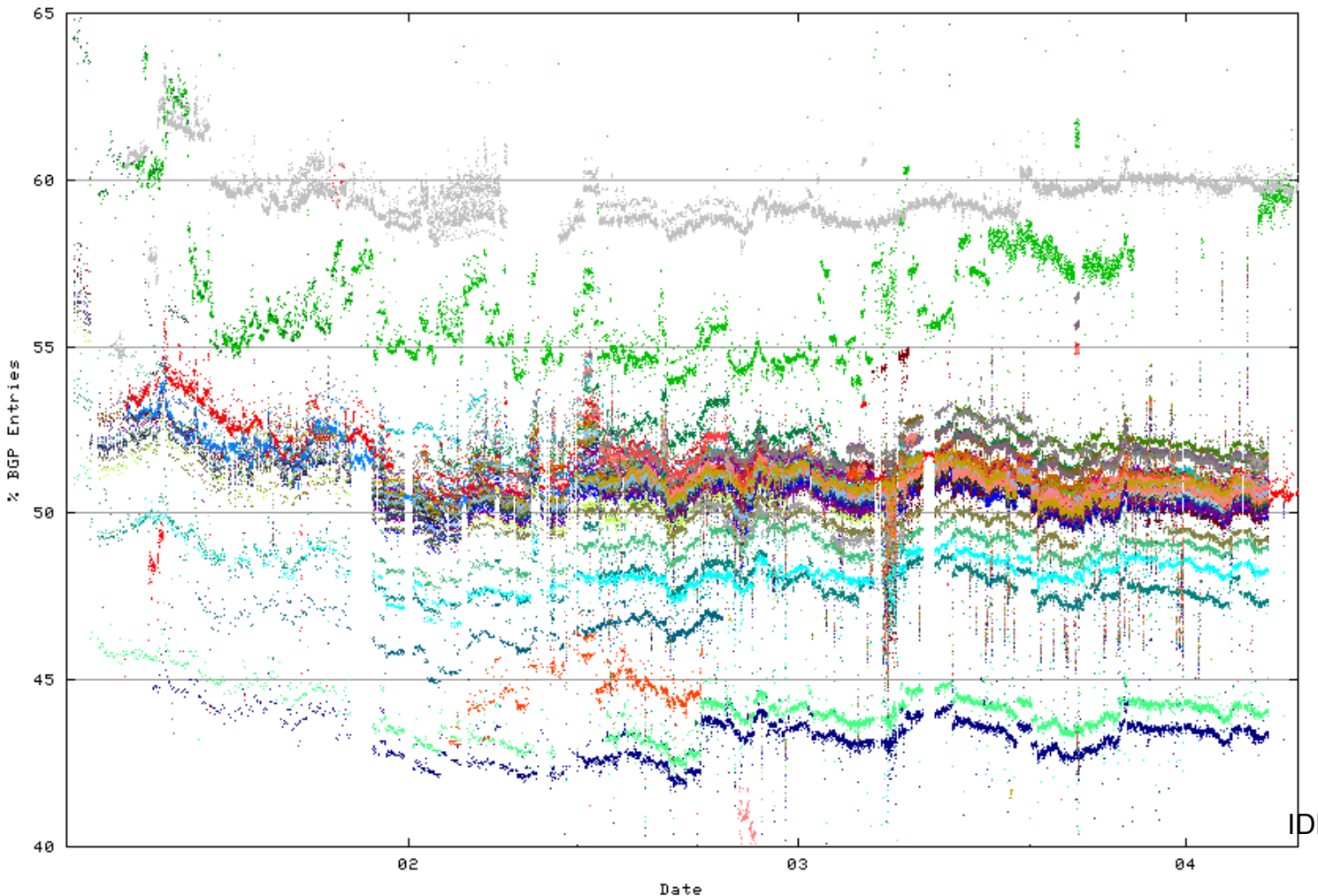
How big is the problem?

- Does prefix filtering help?
- More generally, how “big” are the more specific advertisements in the BGP table?
  - What is the percentage of more specific fragmentary advertisements?
  - How much address space do these more specifics cover?
  - Do they add new routing information?

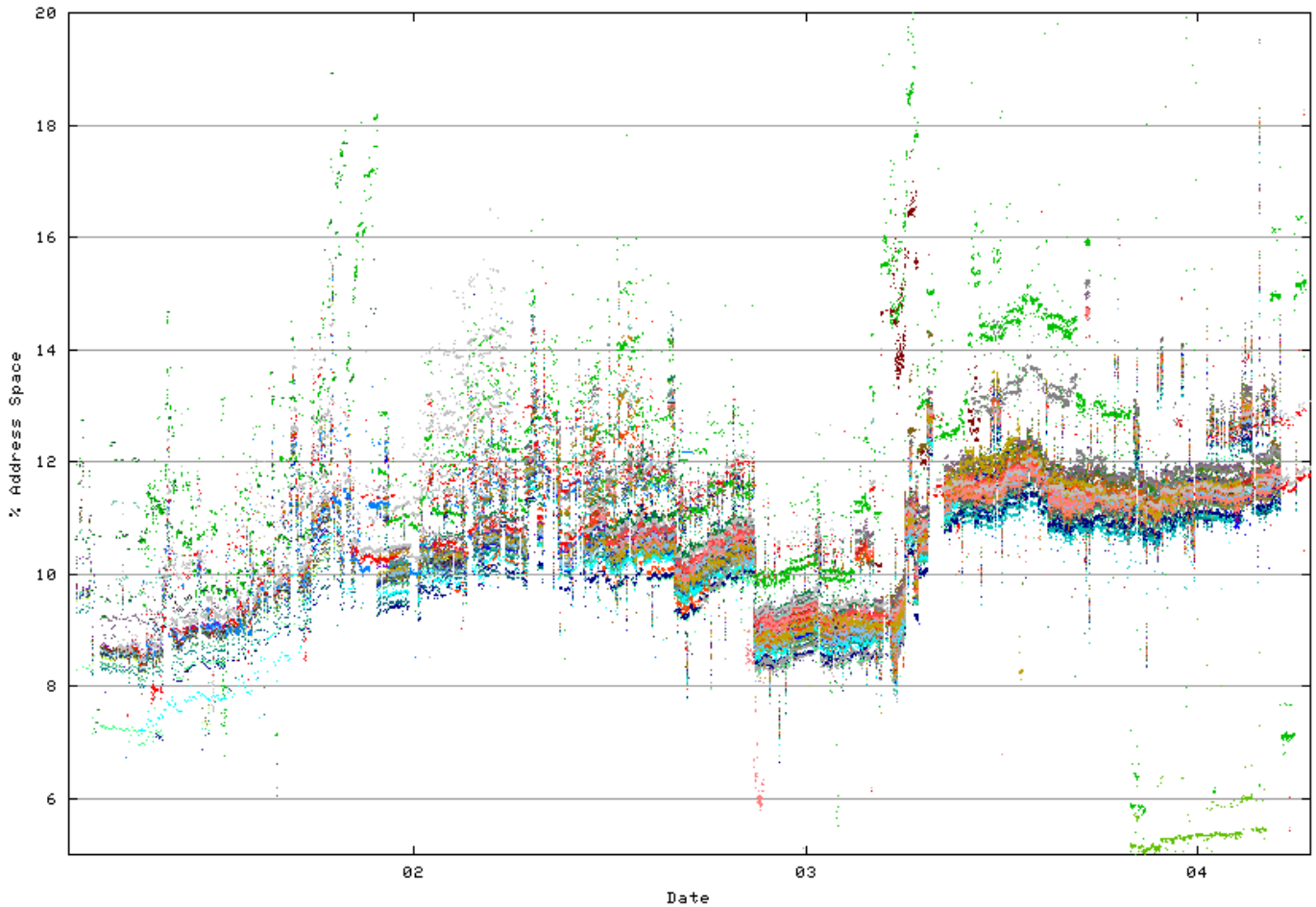
BGP Routing Table history

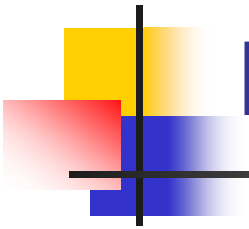


More Specific Advertisements



Address Span of Specifics

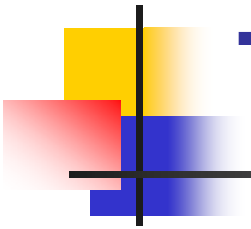




More Specifics

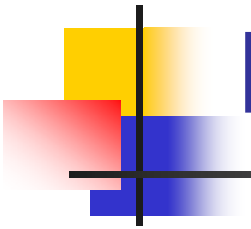
- Appear to be the ‘noise’ of the BGP table. They account for:
  - 55% of the routing entries,
  - 12% of the advertised address space
  - appear to offer no new route paths

- Is the use of more specifics an artefact of inappropriate address assignment policies?



The Question

- How accurate is this assumption that RIR allocations and advertisements are aligned?
- Has this changed in recent times?



Methodology

- Compare the prefixes listed in the RIR delegated files (a log of allocations) with the prefixes contained in a dump of the BGP routing table

Recent RIR and BGP Data

- **4364** RIR IPv4 allocations
  (1 Jan 2003 - 15 April 2004)
- **907** allocations are NOT announced as yet
- **3457** allocations are announced
- **10874** routing advertisements are used to span these 3457 allocations

- Each RIR allocation generates an average of **3.1** routing advertisements

2003/2004 Data (cont)

3457 RIR allocations are advertised

Of these:….

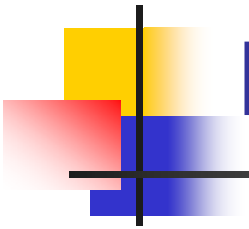
2776 Advertisements precisely match the RIR Allocation

8027 Advertisements are more specifics of 1163 RIR allocations

- **66%** of RIR allocations are directly advertised as routing advertisements without more specifics
- **34%** of RIR allocations generate more specific advertisements
- Where more specifics are advertised there are **6.9** more specific advertisements for each RIR allocation

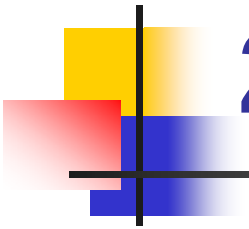
Prefix Length Distribution

Allocation		Advertisements															
Size	**Total**	**Total**	**More Specifics**	**/11**	**/12**	**/13**	**/14**	**/15**	**/16**	**/17**	**/18**	**/19**	**/20**	**/21**	**/22**	**/23**	**/24**
/11	**6**	**102**	**98**	**4**					**77**	**5**				**1**	**4**	**3**	**8**
/12	**16**	**729**	**723**		**6**			**4**	**50**	**22**	**66**	**81**	**54**	**60**	**97**	**95**	**194**
/13	35	450	431			19	12	7	50	22	52	59	74	47	54	18	36
/14	51	565	530				32	5	28	15	35	137	109	11	17	49	124
/15	65	713	666					45	21	17	43	55	72	72	84	57	245
/16	204	865	691						171	24	56	74	92	138	64	39	204
/17	157	677	562							112	39	55	82	88	84	44	170
/18	299	1052	836								214	86	85	77	91	63	434
/19	687	1985	1447									531	145	145	156	94	907
/20	1022	2504	1715										739	139	183	152	1241
/21	70	112	50											62	2	2	46
/22	215	332	165												167	22	143
/23	256	313	109													204	109
/24	471	471	0														470
Total	3554	10039	7202	0	0	19	44	57	270	190	439	997	1398	779	902	744	4129



Limiting the sample to 2004

- Is this level of fragmentation of RIR Allocated address blocks getting better or worse in recent times?
- One way to look at this is to use a smaller data pool of very recent data and compare it with the larger pool already presented



2004 Data

- **1232** RIR IPv4 allocations (up to 15 Apr)
- **462** allocations are NOT announced as yet
- **770** allocations are announced
- **1469** routing advertisements are used to span these 770 allocations
  - Each RIR allocation generates an average of **1.9** routing advertisements

2004 Data (cont)

752 RIR allocations are advertised

Of these:…

629 Advertisements precisely match the RIR Allocation

827 Advertisements are more specifics of 197 RIR allocations

- **74%** of RIR allocations are directly advertised as routing advertisements without more specifics
- **26%** of RIR allocations generate more specific advertisements
- Where more specifics are advertised there are **4.2** more specific advertisements for each RIR allocation

2004 Data – Prefix length Distribution

Allocation		Advertisements														
Size	Total	Total	More Specifics	/12	/13	/14	/15	/16	/17	/18	/19	/20	/21	/22	/23	/24
/12	4	31	30	1			3	10	3			4	4	3		3
/13	8	30	23		7	1		6	7	4	3	1		1		
/14	10	37	31			6	2	6	3	3	4	12			4	1
/15	10	43	35				8	4	8		17					2
/16	96	232	146					84	4	14	15	29	56	4		24
/17	30	47	23						24	5	4	8	4	1		1
/18	49	119	82							37	12	14	7	8	2	39
/19	125	225	126								97	24	24	31	9	38
/20	228	468	281									178	18	42	17	204
/21	27	35	10										25		2	8
/22	44	58	25											33	5	20
/23	48	52	12												40	12
/24	89	89														89
Total	768	1466	824		7	7	13	110	49	63	152	270	138	123	79	441

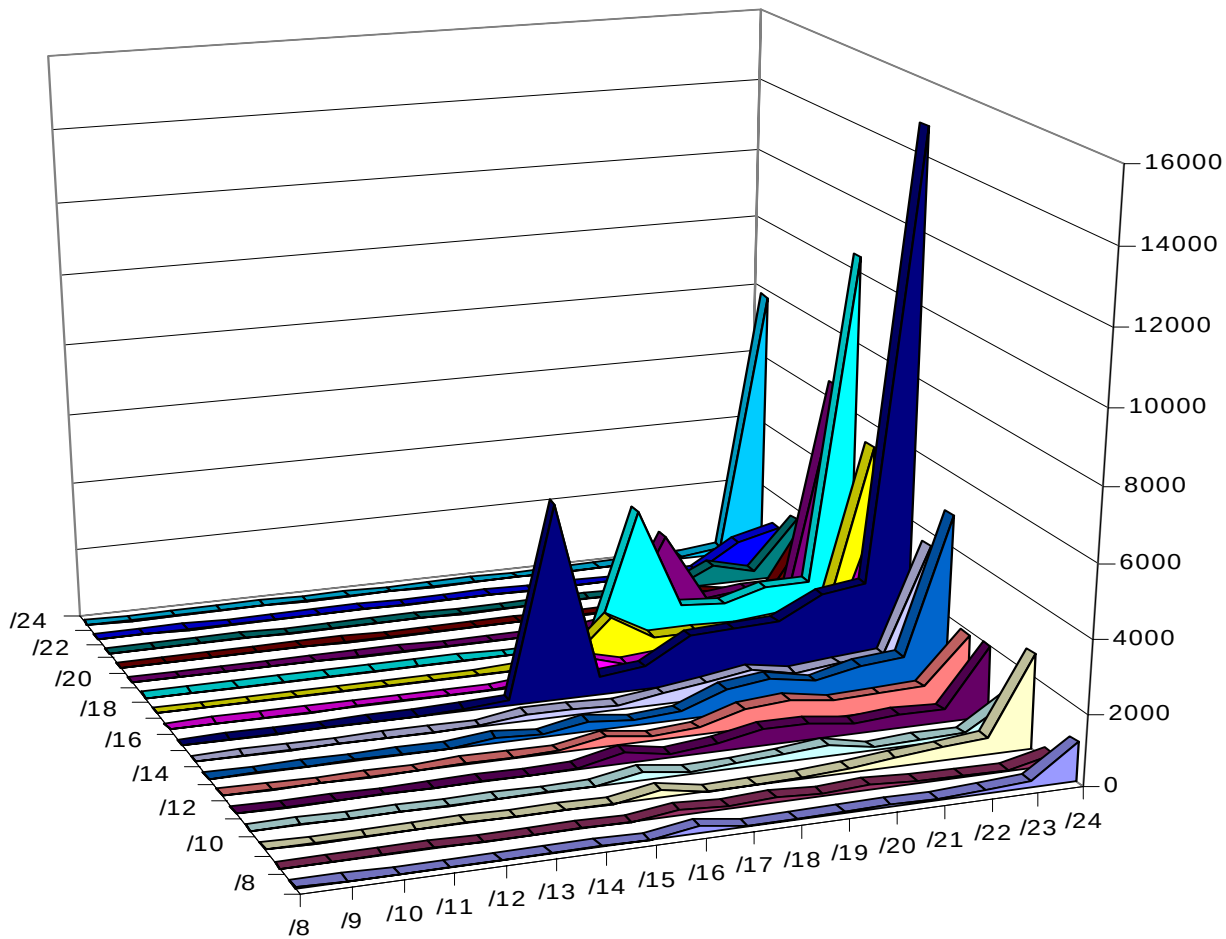
Trends of Fragmentation of Allocations

- The following graphs look at the entire data set of all RIR allocations and compare these to the current state of the routing table. The dates used in the analysis are the dates of the RIR allocation.

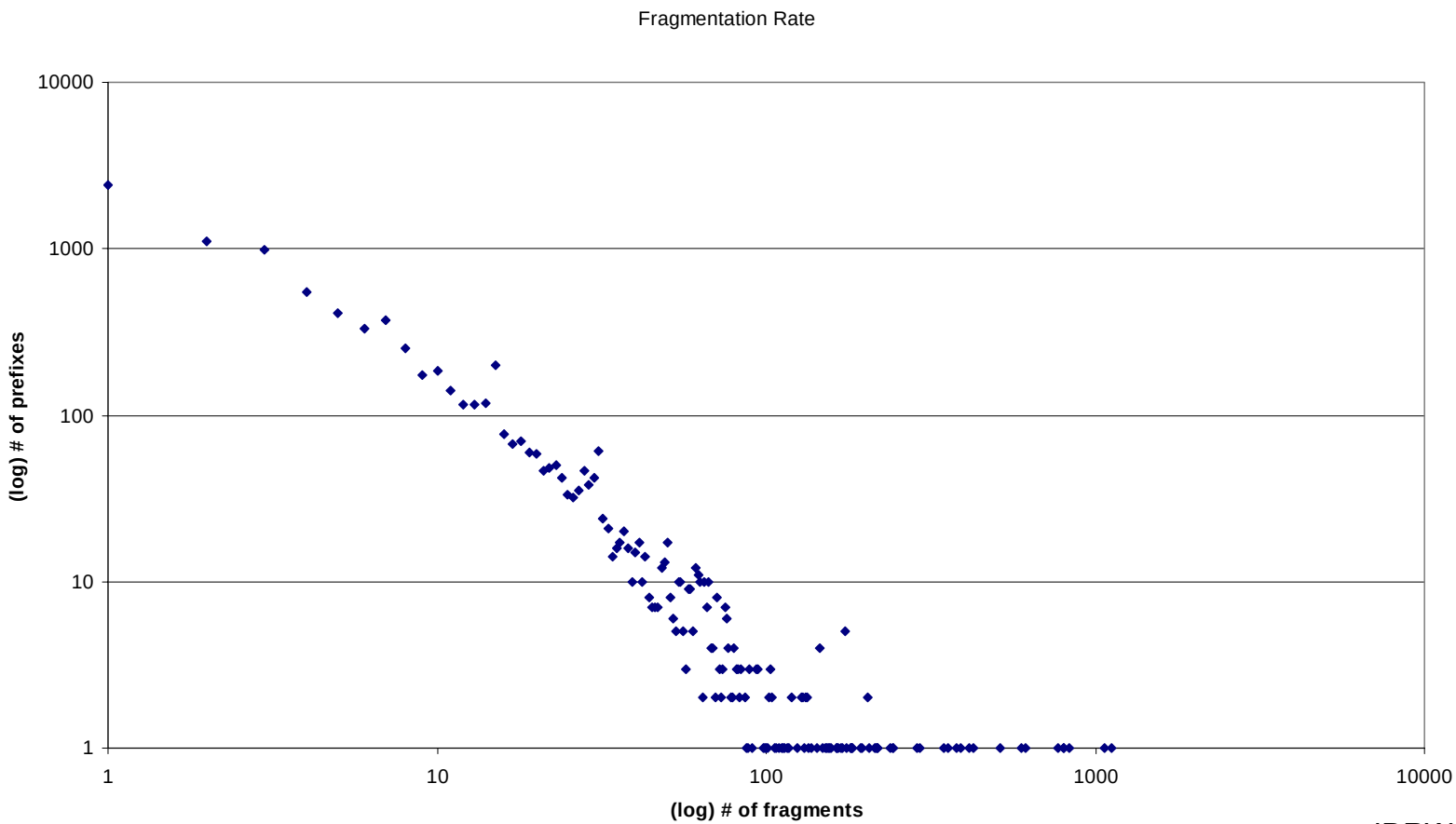
Prefix Length Distribution

Allocation Size	Total	Not Advertised	Advertisements Total	More Specifics	/8	/9	/10	/11	/12	/13	/14	/15	/16	/17	/18	/19	/20	/21	/22	/23	/24	/25	/26	/27	/28	/29	/30	/31	/32
/8	**44**	**13**	**1864**	**1845**	19	4	1	1	7	5	10	10	206	15	24	48	62	49	120	195	1088								
/9	**4**		**1064**	**1064**				1		3		7	132	67	133	80	154	84	62	33	308								
/10	**16**	**2**	**4136**	**4133**			3	2	9	7	6	7	203	6	11	56	124	240	353	476	2632		1						
/11	**33**	**4**	**2202**	**2193**				9	6	3	6	10	248	57	121	192	292	72	129	147	910								
/12	**89**	**14**	**4656**	**4637**					19	13	10	31	323	106	222	466	450	288	379	364	1985								
/13	172	17	**5512**	5460					3	49	39	44	290	119	202	591	676	489	536	576	1897		1						
/14	340	19	**9783**	9629	1				2	6	145	57	266	136	226	707	848	624	893	997	4875								
/15	431	33	**7136**	6927						2	9	198	182	123	283	463	647	412	532	648	3637								
/16	9481	2805	**30361**	24634			2	2	12	16	56	131	5508	516	629	1351	1439	1464	2125	2305	14805								
/17	1227	116	**8261**	7525						1	1	2	87	645	289	423	528	530	957	689	4102		6			1			
/18	2077	257	**9395**	8142								1	9	44	1199	505	515	478	634	666	5343								1
/19	5813	797	**18236**	14354							2	3	3	10	87	3777	855	774	1136	1150	10430			2			4		3
/20	4879	991	**11022**	8328							1		2	1	4	176	2510	542	641	701	6441				1				2
/21	1783	702	**2745**	2397								1	1			4	5	337	181	196	2020								
/22	2425	1011	**2590**	2004										1	1	2	2	2	578	278	1726								
/23	2665	1262	**1875**	1093													1	1	5	775	1093								
/24	27392	19233	**8205**										7	1	3	9	18	43	95	241	7788								
/25	42	39	**3**																	1	2								
/26	29	27	**2**																		2								
/27	21	20	**1**													1													
/28	11	10	**1**															1											
/29	5	5																											
Total	58915	27377	115128	90493	20	4	6	15	58	105	285	502	7467	1847	3434	8851	9126	6430	9356	10438	71084	0	8	2	1	1	4	0	6

Prefix Distribution



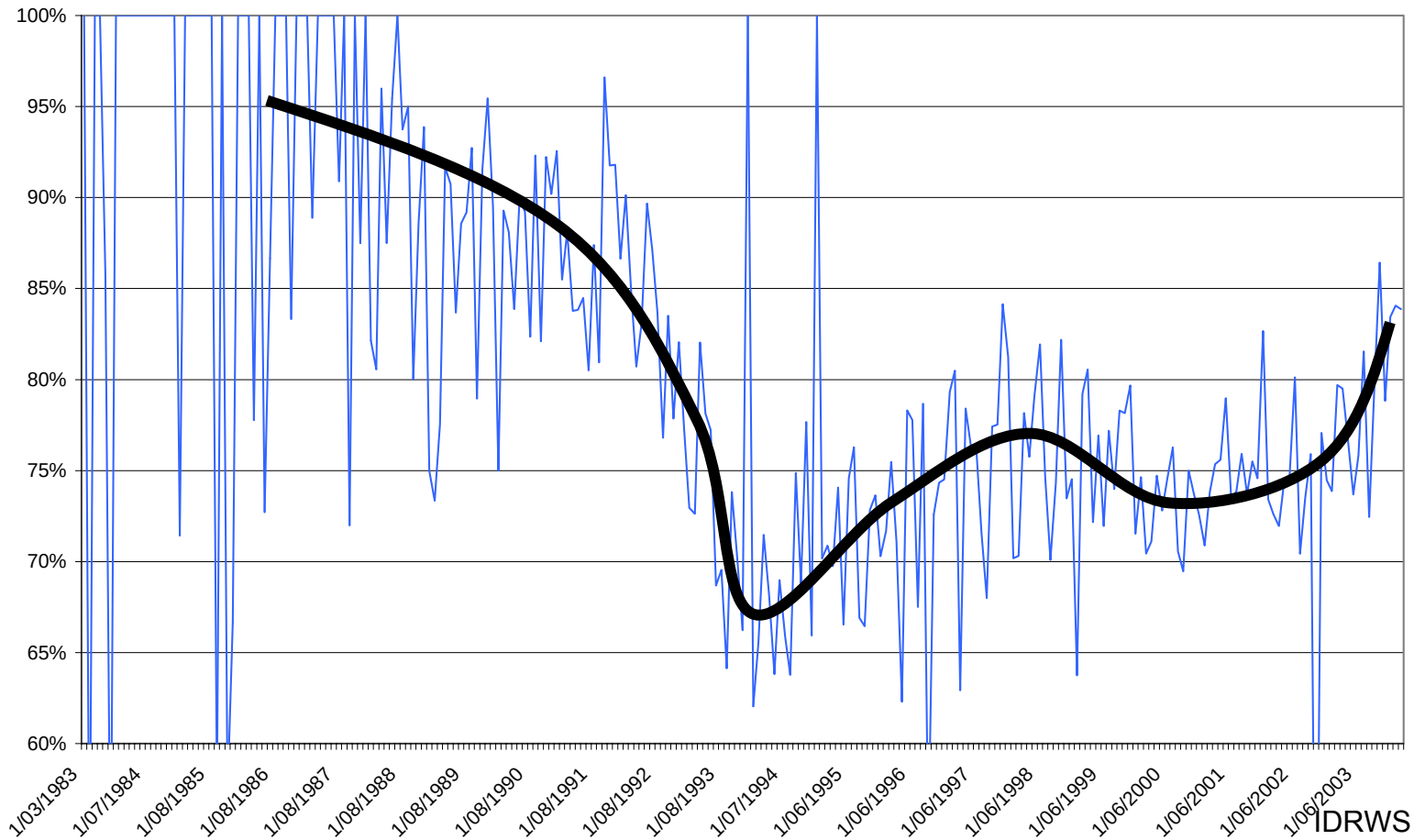
Fragmentation Distribution



Allocations Advertised ‘as is’

- This graph plots the proportion of address allocations that are advertised as allocated. The lower the proportion the greater the amount of allocations that are advertised only as fragments. The higher the number the better (in terms of reduction in advertisement fragmentation)
- This has been improving since August 2000

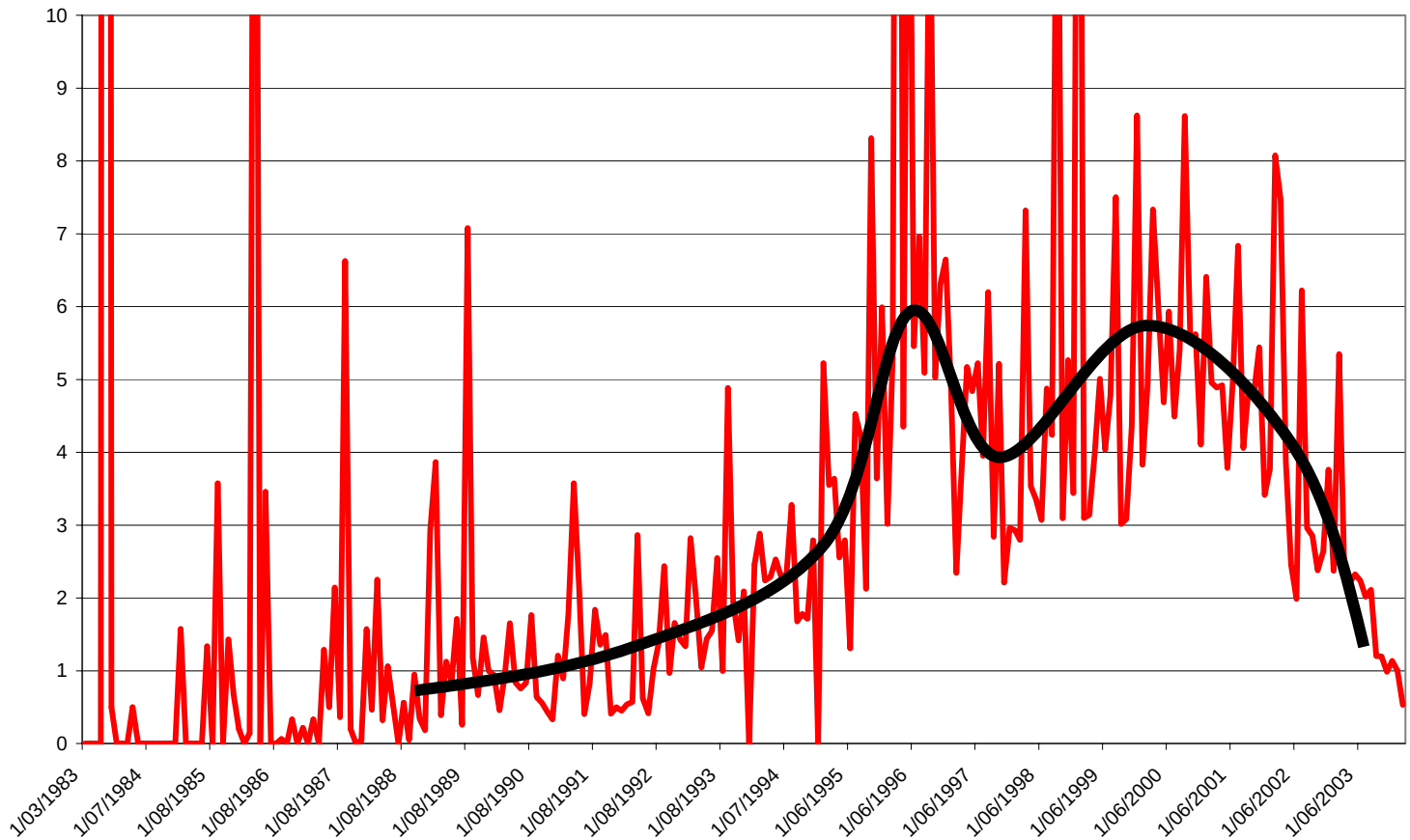
Allocations Advertised ‘as is’



Number of Fragmentary Advertisements as a proportion of Allocations

- This compares the number of fragmentary advertisements to the number of RIR allocations. The lower the number, the better
- The proportion of fragmentation of allocated blocks has been dropping since August 2000

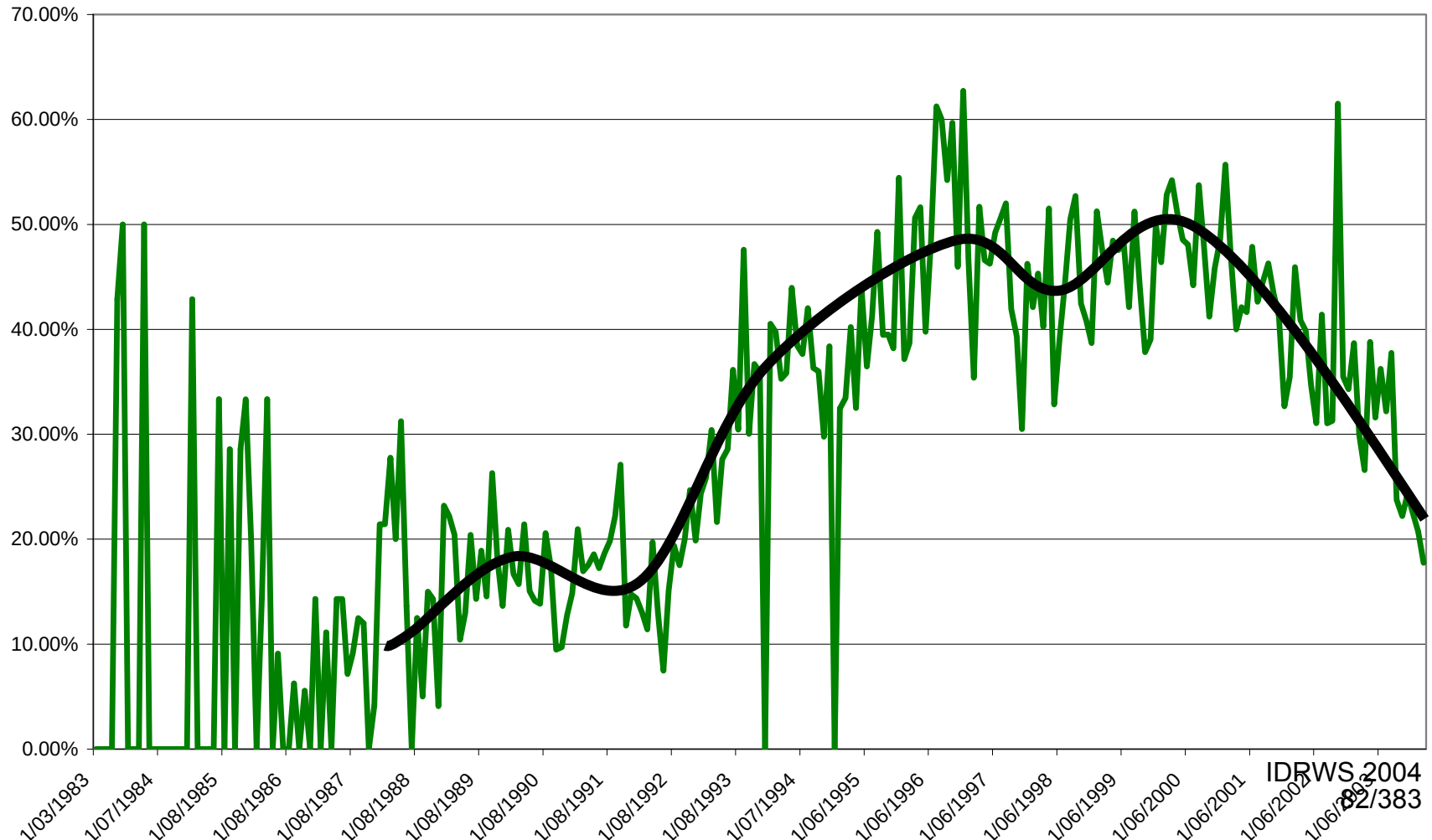
Number of Fragmentary Advertisements as a proportion of Allocations



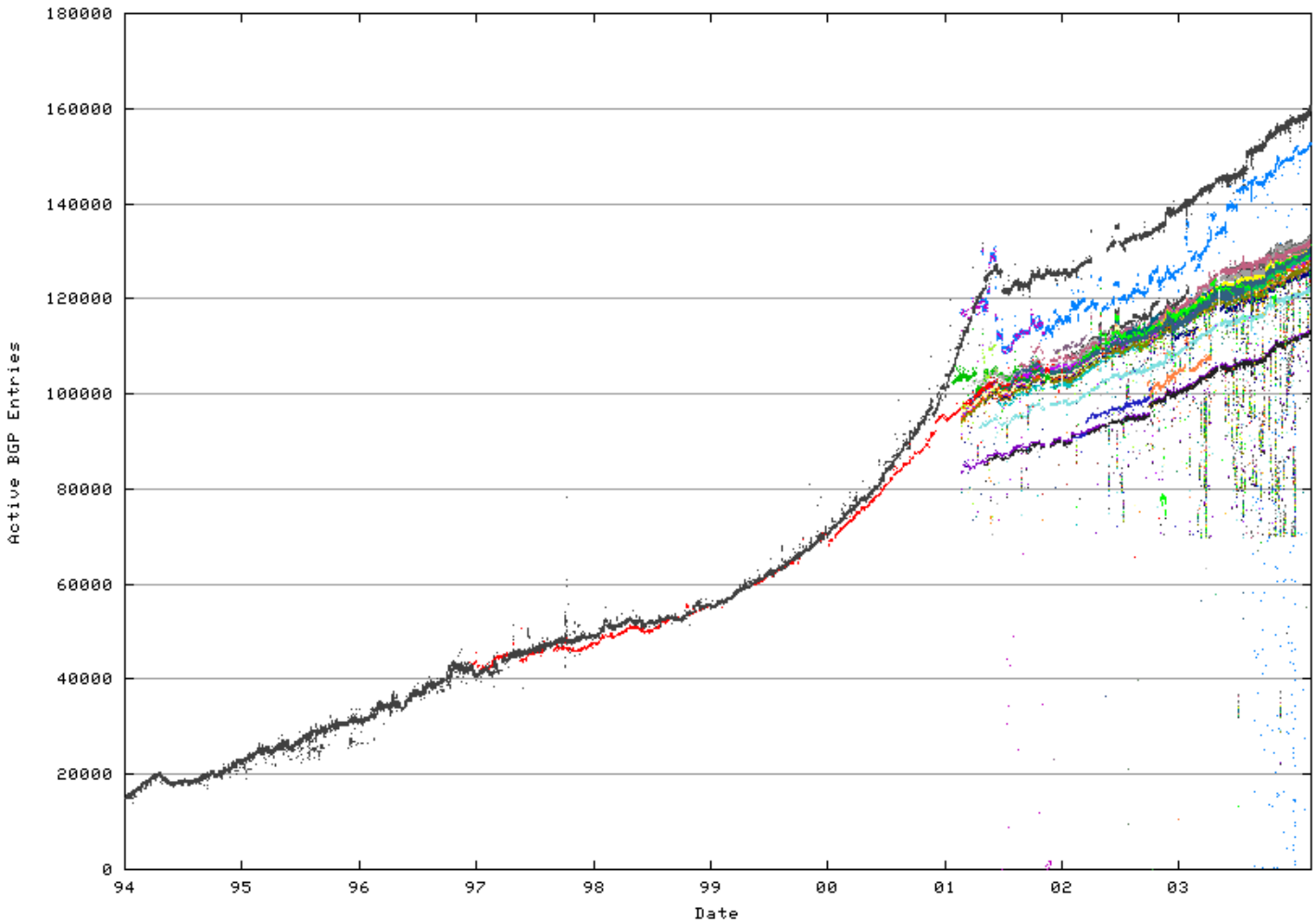
Proportion of Allocations that are advertised in Fragments

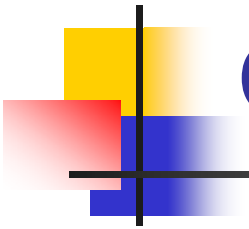
- This compares the number of allocations against the number of allocations that are advertised in one or more fragments. The lower the number the smaller the amount of fragmentation of allocations
- Again there is a noticeable decline since August 2000

Proportion of Allocations that are advertised in Fragments



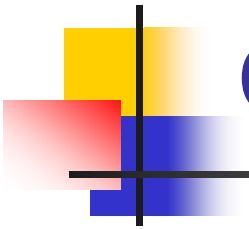
Just a reminder – BGP Routing Table Growth





Observations

- It appears that the major contributor to the growth of the routing table is the amount of advertisement fragmentation that occurs in allocated address space.
- This form of advertisement fragmentation peaked from 1997 – 2000
- The levels of advertisement fragmentation have been improving since late 2000.



Observations

- Taking an allocated block and advertising more specific /24 address prefixes is the predominate form of advertising a split allocation block in fragments
  - Many of these more specifics appear to be local (i.e. could be masked with NOEXPORT)
- One fifth of allocations are fragmented in this fashion, and, on average there are 6.6 additional advertisements of fragments of the address block
- /21, /22, /23 allocations have proportionately less advertised fragmentation than larger prefix sizes
- Levels of fragmentation of advertisements have been improving since late 2000, corresponding with a return to linear growth of the BGP routing table size.

Towards a Cohesive Internet Routing Registry System

Inter-domain Routing Workshop - Amsterdam
May 1, 2004
Larry Blunk, ljb@merit.edu. Merit Network, Inc.

Overview

- State of the Internet Routing Registry System
- Review of existing standards work
- Authority issues
- Authentication issues
- Other security concerns
- Replication and availibility
- Data correctness
- Extensibility
- Review
- Future of the IRR System
- References

State of the IRR System

- The IRR System is currently a very loosely defined concept
  - Based upon the RPSL (RFC 2622) standard
  - Merit hosts www.irr.net and mirrors ~50 other registries
    - No formal requirements or authority for mirrors
    - Confusion between RADB and IRR System
  - RIPE NCC also mirrors a number of registries
- Registries consist largely of smaller ISP's and networks
  - Some large ISP's present - Verio, Level3, and Savvis
  - Two open independent registries - RADB and ALTDB
- 3 RIR's run routing registries – APNIC, RIPE, and ARIN
  - ARIN's is open and not integrated with address registry
  - LACNIC has limited “RR-like” functionality (non-RPSL)

Review of standards work

- RPSL (RFC 2622) was published in 1999
  - Follow-on documents
    - RFC 2650 Using RPSL in Practice
    - RFC 2725 Routing Policy System Security
    - RFC 2769 Routing Policy System Replication
  - RPSLng – IPv6 and Multicast extensions – currently I-D
- CRISP Working Group concerns cross registry protocol issues
  - RFC 3707 defines a set of requirements for CRISP
  - Current focus is on domain and address registries
  - Specifications based around IRIS XML schema framework
  - What are the CRISP considerations for routing registries?

Authority issues

- RFC 2725 provides the current framework for RPSL authority
  - Authorization based on AS and IP prefix allocations
  - Currently supported by RIPE and APNIC registries
- Issues when going outside the integrated RIR/RR registries
  - An ISP wants to use their own registry
  - Third-party registries (RADB and ALTDB)
  - Cross registry issues (i.e., Prefix allocation by one RIR, and AS by another)

Authority issues (cont'd)

- Some pieces are puzzle may be already addressed
  - “::” for external references in RFC 2725
  - “delegated:” attr. and “repository:” object in RFC 2769
  - Should these be pulled together in a new document?
- Are there incremental approaches to improving authority?
  - Use of “integrity:” attribute from RFC 2769

Authentication issues

- Initial RPSL spec included poor authentication mechanisms
  - NONE and MAIL-FROM clearly bad choices
  - CRYPT-PW hashes subject to dictionary attacks
  - PGP is strong, but can be difficult for new users
- Several attempts to address deficiencies
  - Dropping NONE and MAIL-FROM support
  - Stronger password hashes (RIPE supports MD5 hashes)
    - Note: stronger hashes still subject to cracking
  - RADB no longer reveals pw hashes on queries/mirroring
  - RIPE deploying X.509 certificate based authentication
- Should authentication requirements be more formalized?
  - Should they be enforced (for participation in IRR system)?

Other security issues

- Security of the registry repositories
  - Is this a concern or can we assume they are safe?
  - Could archive PGP and X.509 signatures w/updates
    - Would allow remote verification of adds/removals
  - Should there be a “signature” attribute within objects
- Security of queries and mirror operations
  - Should registries sign replies to queries?
  - RFC 2769 defines a “repository-cert” for securing mirroring transactions
- What should be the considerations for future Inter-domain routing security enhancements (i.e. S-BGP and soBGP)?
  - Are there issues here routing registries could address?

Replication and availibility

- Currently, replication is handled by a simple near realtime mirroring protocol
  - Protocol is not particularly robust and poorly documented
  - RFC 2769 defines a more robust and secure protocol
    - Fairly complex and has yet to be implemented
    - Could other general replication schemes suffice?
- What availibility requirements should be considered?
  - Multiple mirrors?
  - Anycasting?
- Registries not currently documented in easily machine queried format
  - Could use “repository:” object to list mirrored registries

Data correctness

- Data correctness has long been an issue of concern with IRR's
  - Stale data that is not updated or removed from registries
  - Registration of “route:” objects merely to record allocated prefixes rather than actual announced routes
  - Registering more specific components of a prefix in case they “might” be announced at a future time.
- Some efforts have been made to analyze consistency
  - RIPE NCC RR Consistency Check project (RRCC)
  - Merit RADB “radb-reports”
  - Nemecis project
- Can the tools be better coordinated and easier to use?
- Are more active measures needed (flagging stale data)?

Extensibility

- RPSL recently updated with IPv6 and Multicast support
- Introduced further complexity into an already complex specification
- Has RPSL had its day?
- CRISP Working Group could provide opportunity to start fresh and support better extensibility.
- Should there be a transition or hybrid (XML+RPSL) model?

Review

- The current IRR System lacks a coherent model
- How should the authority model work?
  - Review models presented in RFC 2725 and RFC 2769
  - Where do local ISP and third party RR's fit in?
  - Should the RIR's delegate to external registries?
- Where can security be improved?
- How do we maintain data consistency?
- Is there sufficient reliability and redundancy?
- Where does the CRISP work fit in?
- What are the considerations for future inter-domain routing protocol security enhancements?

Future of the IRR System

- Propose creating IRR System requirements document
  - Could possibly work within IETF GROW working group
  - Should address requirements without necessarily getting into data representation (RPSL or IRIS) issues
  - Need to involve stakeholders (ISP's, end-user's, RIR's)
- Look at CRISP work as requirements are defined
- Consider an IRR Consortium or Association
  - Would set policies and formal requirements
  - Address security and accessibility

References

- RFC 2622 - http://www.ietf.org/rfc/rfc2622.txt
- RFC 2650 - http://www.ietf.org/rfc/rfc2650.txt
- RFC 2725 - http://www.ietf.org/rfc/rfc2725.txt
- RFC 2769 - http://www.ietf.org/rfc/rfc2769.txt
- RPSLng - http://www.ietf.org/internet-drafts/draft-blunk-rpslng-04.txt
- CRISP WG - http://www.ietf.org/html.charters/crisp-charter.html
- RFC 3707 - http://www.ietf.org/rfc/rfc3707.txt
- RRCC - http://www.ripe.net/rrcc/
- Nemecis - http://www.cs.ucr.edu/~siganos/papers/Nemecis.pdf

Nemecis: A tool to analyze the Internet Routing Registries

Georgos Siganos and Michalis Faloutsos
Dept. of CSE, U.C. Riverside
{siganos,michalis}@cs.ucr.edu

Problem

- We need cooperation between Autonomous Systems.
  - Internet Routing Registries (IRR) is an attempt
  - IRR: text based repository of BGP related policy
- Problem: IRR have not reached their full potential
  - has not been explored
  - its accuracy has not been quantified
  - is very complicated described in RPSL

Contribution: NEMECIS

- We provide a framework for BGP policy analysis
- We quantify the accuracy of the IRR
  - Check the policies for correctness / freshness
  - This was a long term goal of RIPE
- We develop a tool to analyze IRR data
  - We use a relational database to store the policies
  - Web based front-end for the database

The Rest of this talk

- Key concepts of IRR and NEMECIS
- Validation of our approach in practice
- How can we improve IRR?
- Conclusions

How is policy described in RPSL?

- Policy description resembles BGP filtering
  - Routes: from AS1 accept 138.23.0.0/16
  - Regular expressions on the AS Path: from AS1 accept <^AS1+ AS2*>
  - Communities: from AS1 accept community(xxx:yyy)
- RPSL provides high level structures to group routes
  - AS numbers (AS1): all routes the AS registers
  - AS-SET: AS numbers and other AS-SETS
  - ROUTE-SET: routes and other ROUTE_SETS

Example of An RPSL Description

```
route:          138.23.0.0/16
origin:         AS4
```

Registered/Maintained by AS4

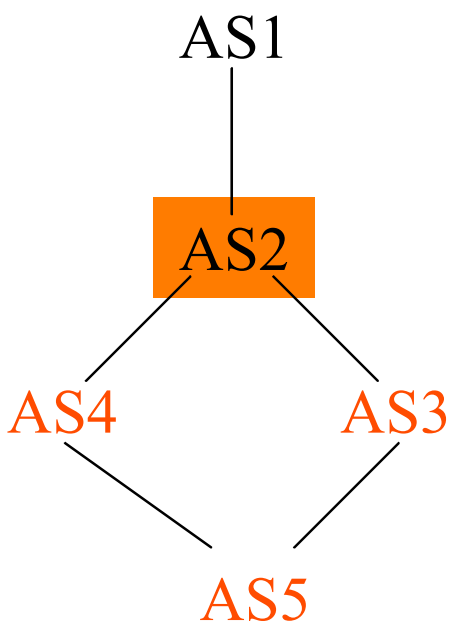
```
as-set:         AS3-ISP
members:        AS3, AS5
```

Registered/Maintained by AS3

```
as-set:         AS2-ISP
members:        AS2, AS3-ISP, AS4
```

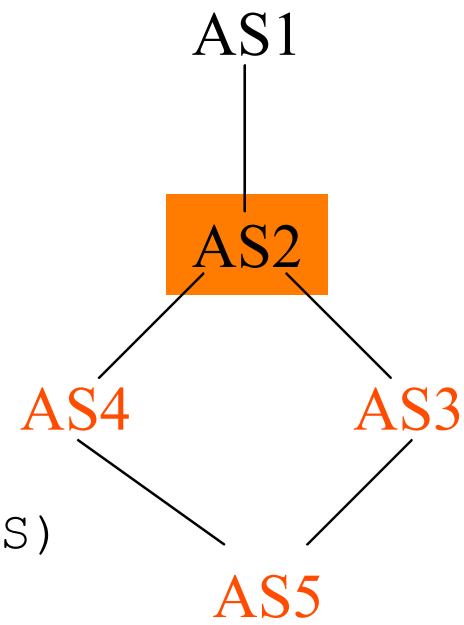
```
aut-num:        AS2
import:         from AS1 accept ANY
import:         from AS3 accept AS3-ISP
import:         from AS4 accept <^AS4+AS5*$>
export:         to AS3 announce ANY
export:         to AS4 announce ANY
export:         to AS1 announce AS2-ISP
```

Registered/Maintained by AS2



... things can become scary...

```
aut-num:      AS2
import: {
            from AS-ANY action pref=10;
                 accept community.contains(1:1);
            from AS-ANY action pref=0; accept ANY;
          } refine {
           from AS2:AS-CUSTOMERS
           accept (PeerAS OR AS2:AS-CUSTOMER:PeerAS)
           AND <^PeerAS+ AS2:AS-CUSTOMER:PeerAS*$>
          }
```



- Policies can be thousands of lines long
- Sets can contain tens of thousand of members
- Inconsistent / out of date policy

Nemecis: Three main phases

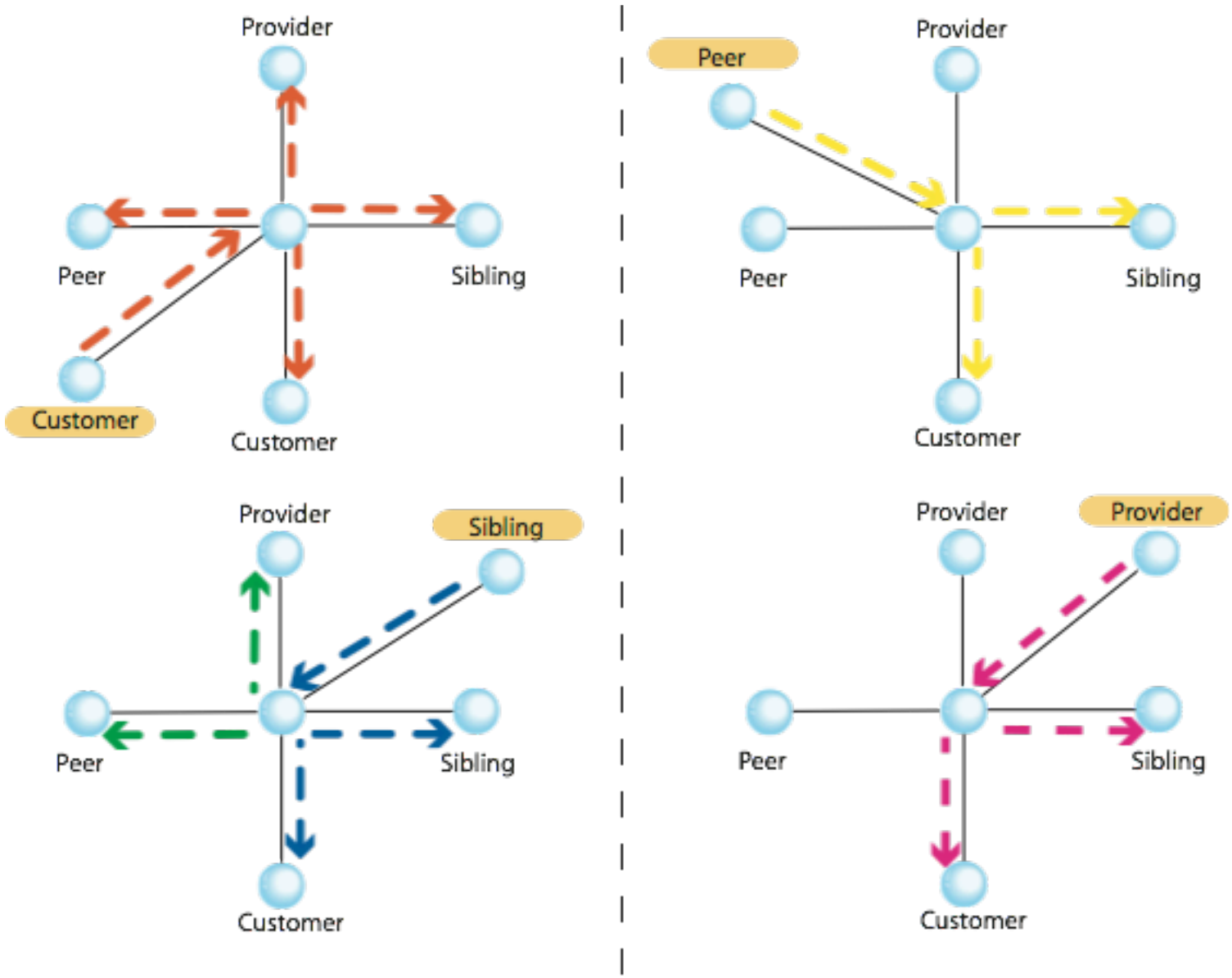
- Create the database:
  - Parse RPSL policy text, put data in tables
- Correlate import and export policies
  - Export = what I create + what I import
  - For each export find where it comes from
  - Find at the level of a link: what I do with incoming data
- Infer business relations: from link-level model
  - Examine export policies of two neighbors
  - If not enough or incomplete, use import policies
- Deal with incomplete and inaccurate data

Link Level Export Matrix

Links	1	...	i	j
1	x			
...		x		
i			x	export
j			export	x

- Relation Matrix of an AS: which link I export to which other AS
- The matrix should be symmetric

Compute Business Relations

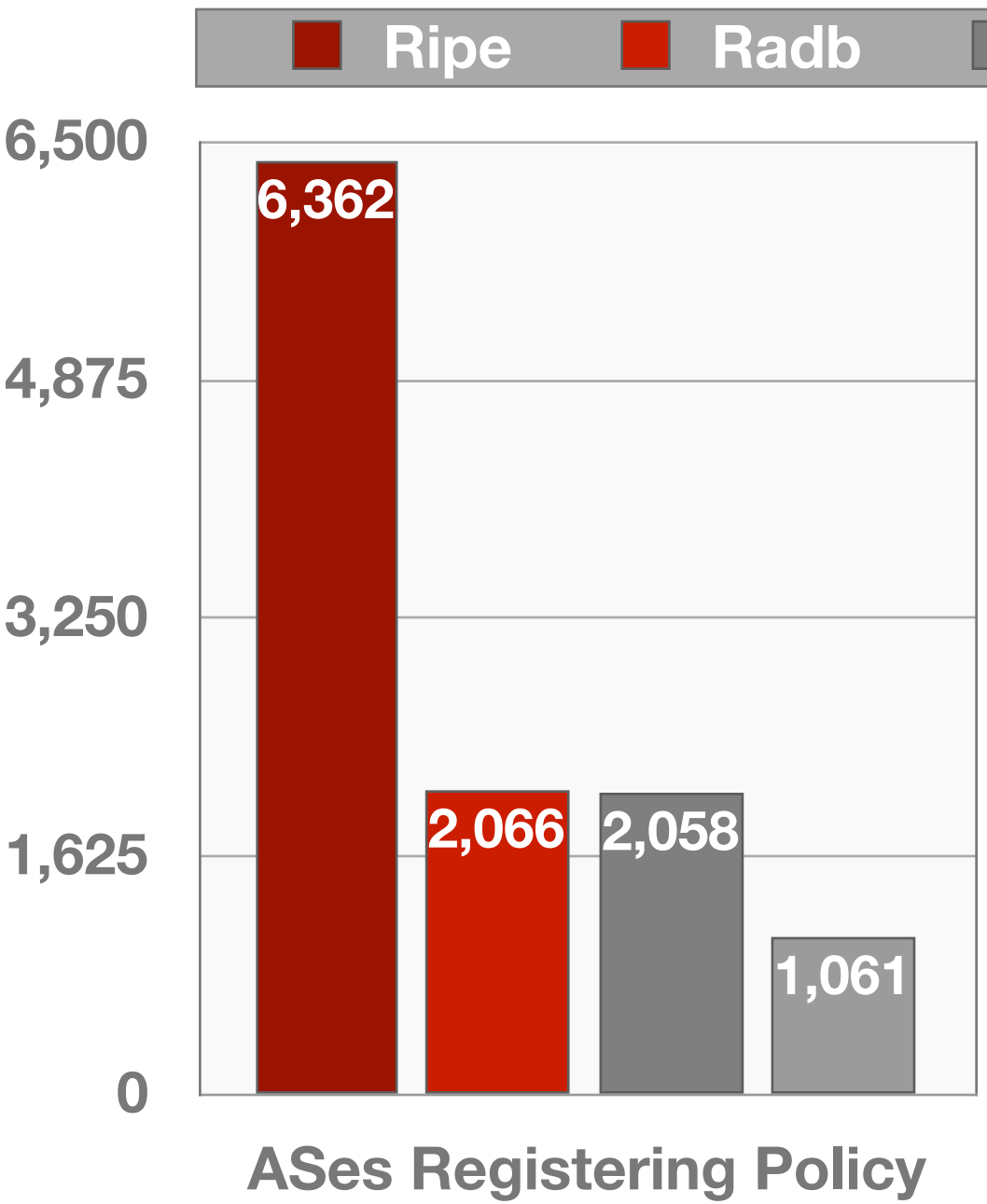


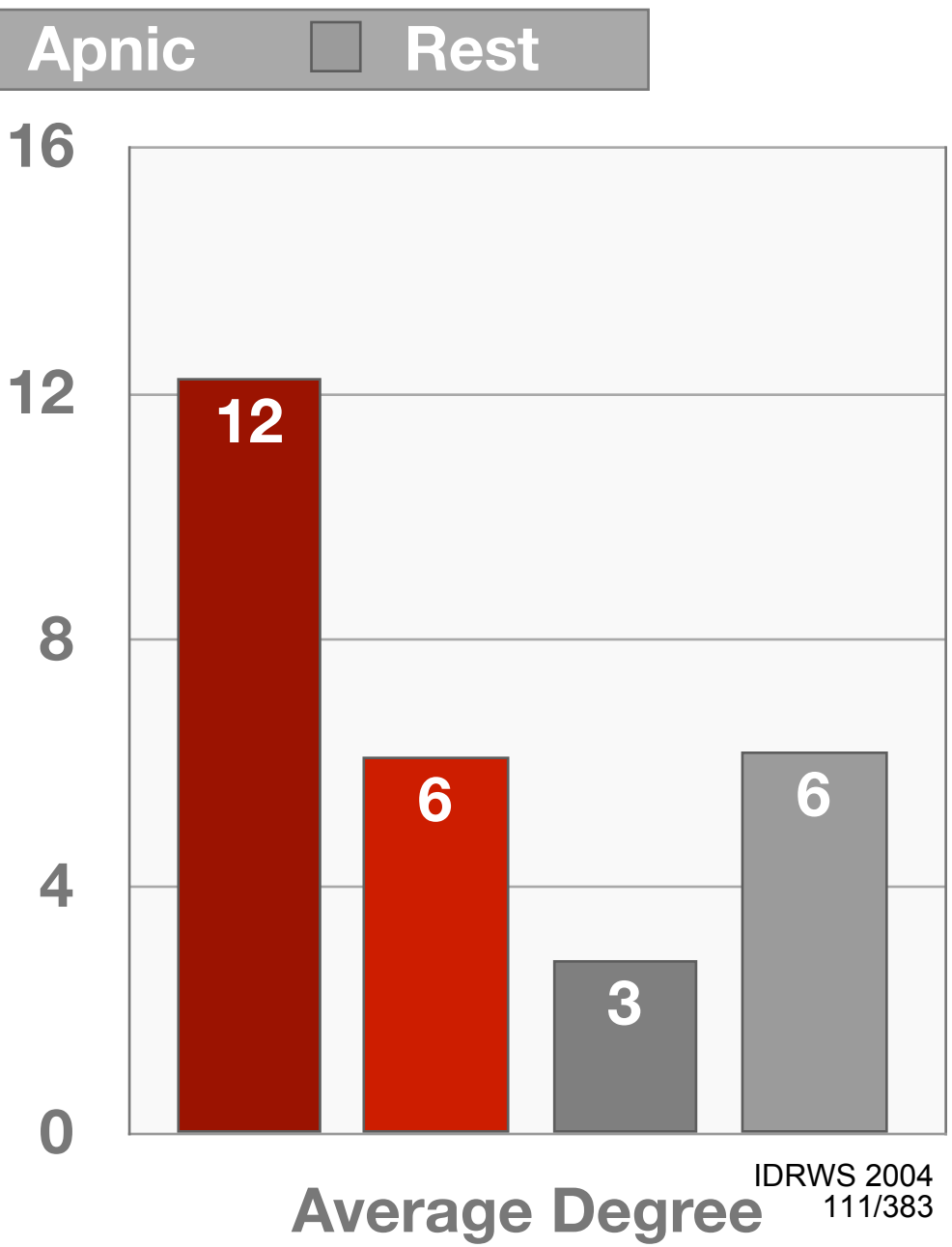
- Business relations can be grouped by the export filters

The Rest of this talk

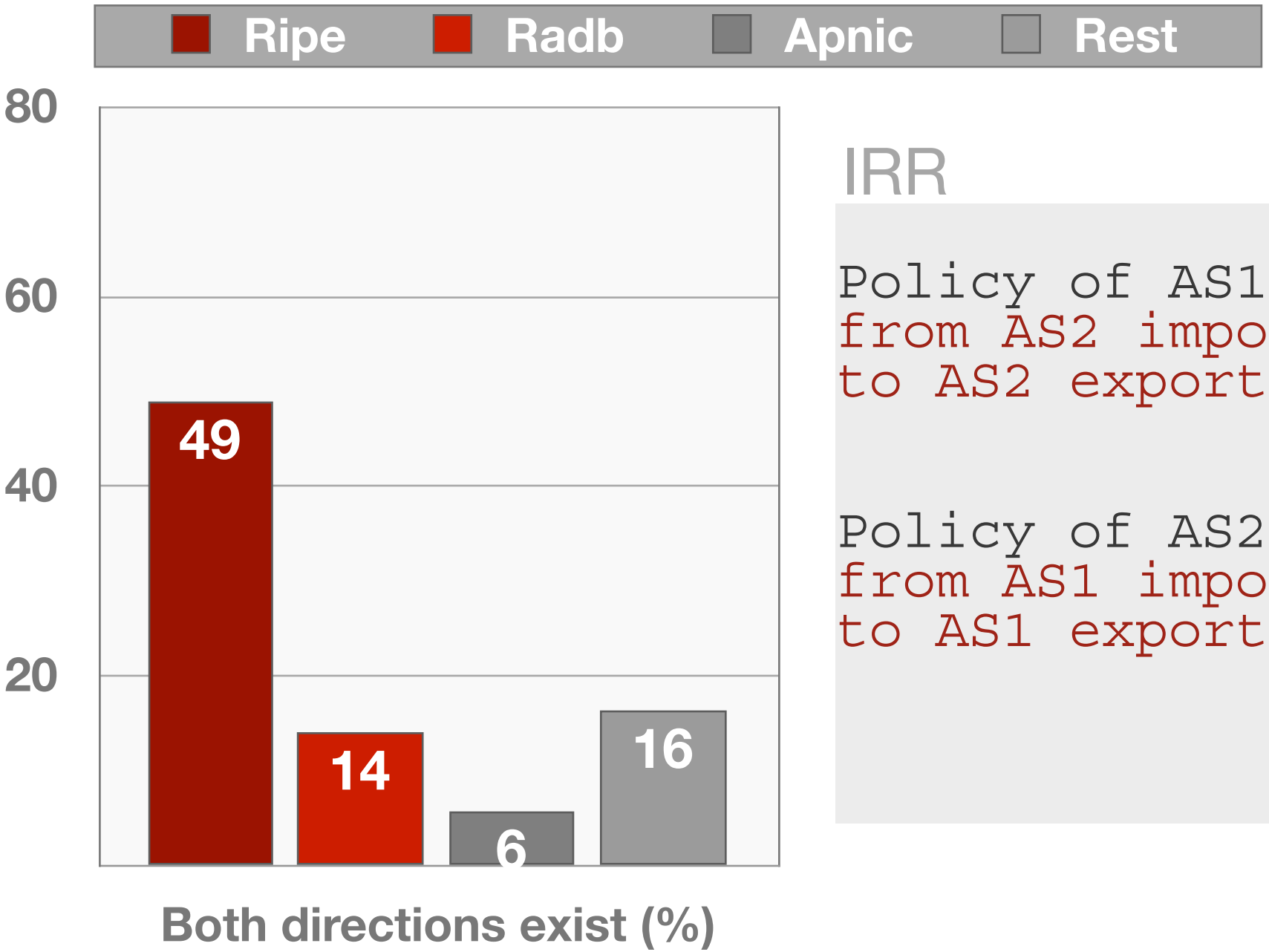
- Key concepts of IRR and NEMECIS
- Validation of our approach in practice
- How can we improve IRR?
- Conclusions

How many ASes register their policy?





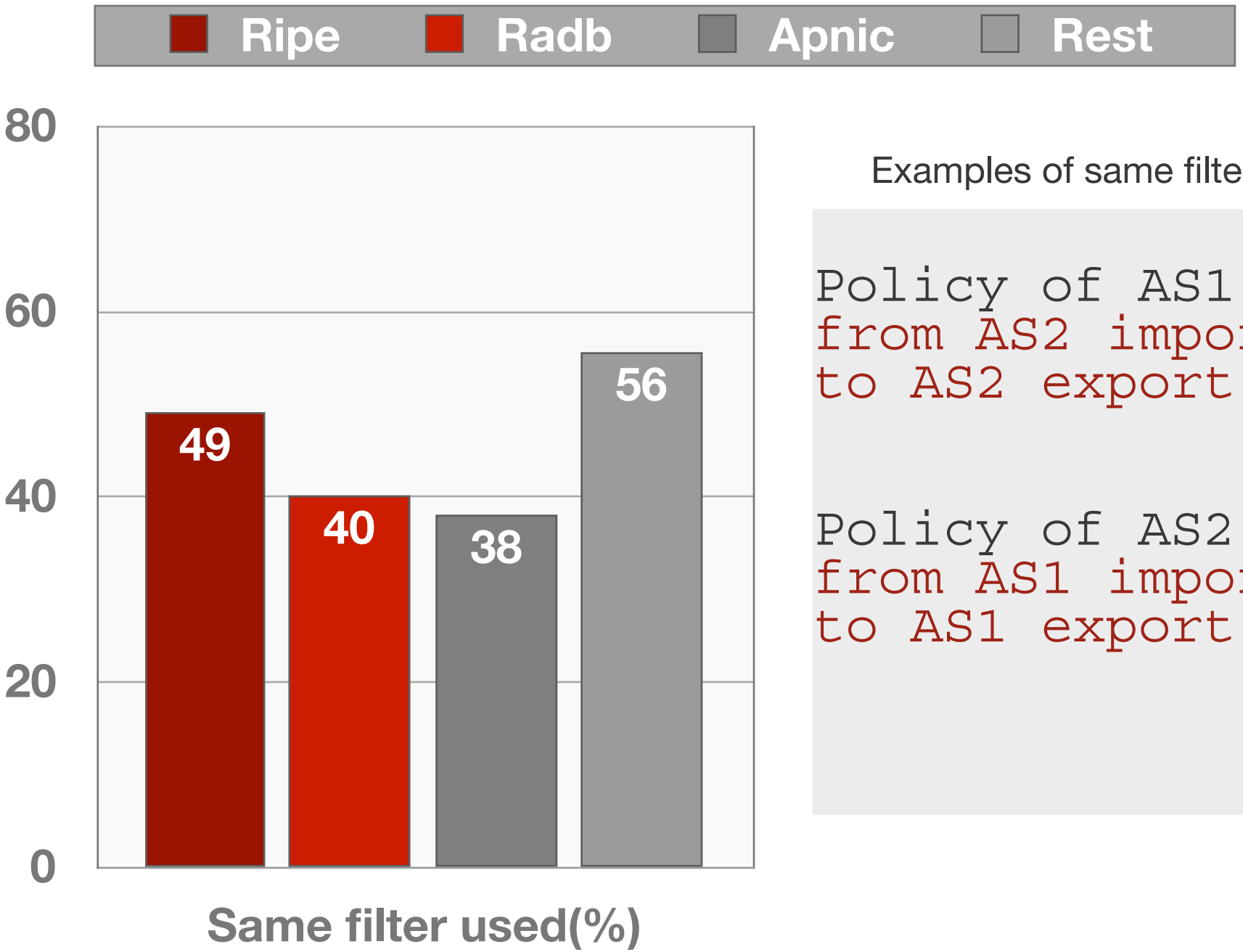
Do both peers register “each other”?



IRR

```
Policy of AS1:
from AS2 import B
to AS2 export A


Policy of AS2:
from AS1 import A
to AS1 export B
```

Do they use the same filter?



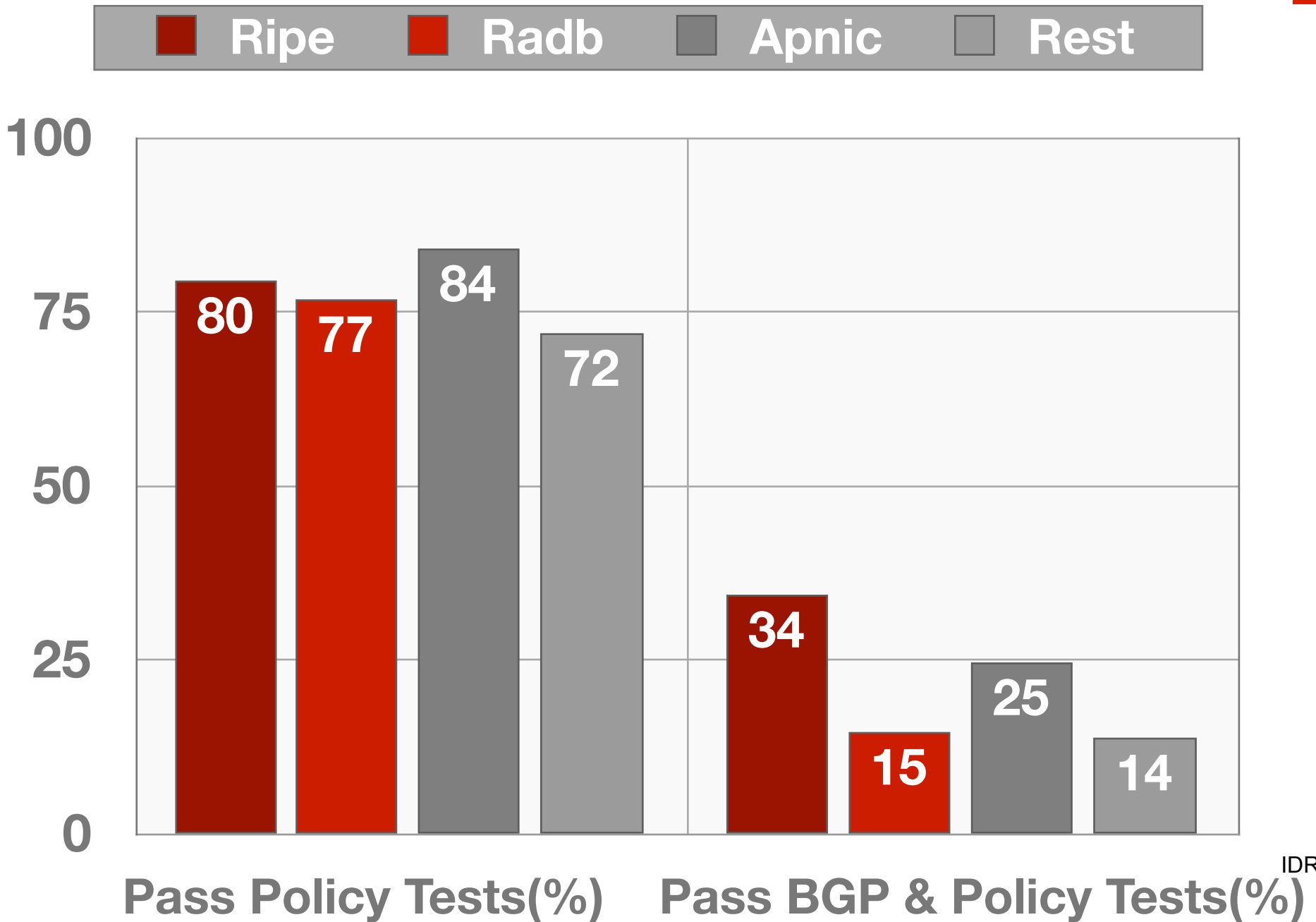
Examples of same filter

```
Policy of AS1:
from AS2 import B
to AS2 export A


Policy of AS2:
from AS1 import A
to AS1 export B
```

Tests for consistency of IRR

- Policy based tests (correctness)
  - import-export consistency
  - Link-Level policy is symmetric
- BGP based tests (freshness)
  - All peers of an AS, as found in BGP, must be registered in IRR.
  - The high-level policy of an AS should be the same in both BGP and in IRR (e.g. Provider to Customer).

Quantify the Accuracy of IRR



The Rest of this talk

- Key concepts of IRR and NEMECIS
- Validation of our approach in practice
- How can we improve IRR?
- Conclusions

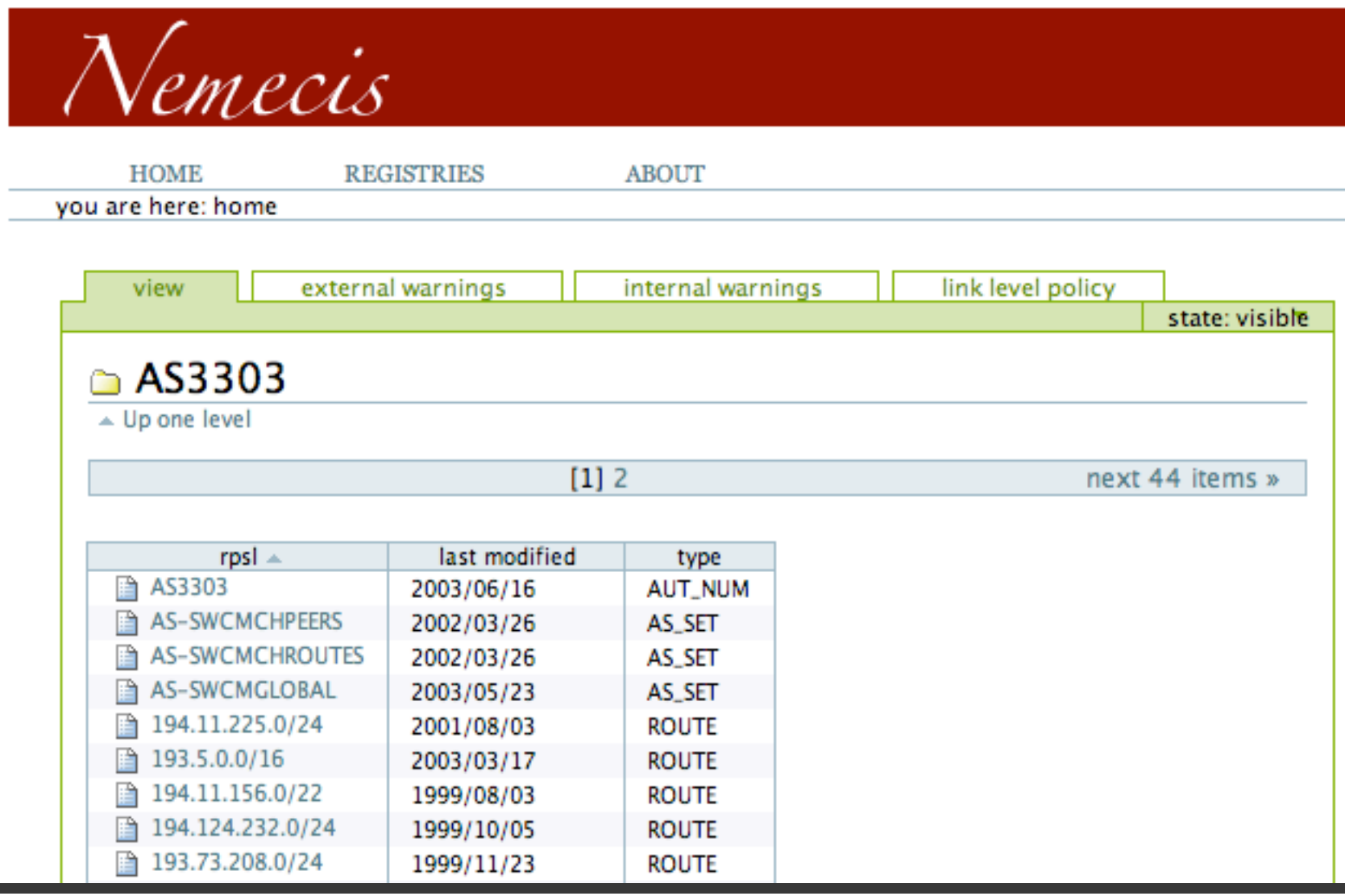
Shortcomings of IRR

- RPSL is used for both configuration and cooperation
  - Too complex (ISPs maintain web pages for policy/communities)
  - Unnecessary details are revealed
- Policy is stored as simple text
  - Difficult to process: no query support
- Usually there are no consistency checks
  - People don’t trust the IRR information
  - Mistakes can occur from stale information
- No variable level of details on the information
  - Either all the information or none
  - Ideally, we want to customize/control who sees how much information (other information for customers, other for peers)

What ‘should’ the role of IRR be?

- Towards a safe and robust Internet:
  - We need an automatic way to detect abnormal routing behavior
  - Prevent IP hijackings
  - Accountability: trace-back of errors
- We lack the tools to analyze the configuration of an Autonomous System.
  - Detecting of errors should be automatic

Snapshot of Nemecis



Conclusions

- The first effort to quantify the consistency of IRR
  - RIPE is the best registry (over 2100 ASes)
  - Useful information exists in the Registries.
  - This could renew the interest for IRR and AS collaboration
- We believe that IRR has an important role in the future operation of Internet.
- To use our demo please visit:

http://ira.cs.ucr.edu:8080/Nemecis

Introduction to compact routing

Dmitri Krioukov

UCSD/CAIDA

dima@caida.org

IDRWS 2004

Initial interest: theoretical (fundamental) aspects of routing on graphs

Interest crystallization history:

- Scalability concerns
  - Convergence
  - Routing table size
- Immediate causes
  - Routing policies
  - Increasing topology density
    - Multihoming
  - Address allocation policies
  - Inbound traffic engineering, etc.
- Various short-term fixes
  - Let's consider one of them

Routing on AS#s (ISLAY,atoms)

Disregarding practical problems associated with it, this idea does not solve anything in the long run: small multihomed networks requiring $O(1)$ IP addresses will lead to the situation with the total number of ASs being of the same order as the number of IP addresses.

Crystallization history (contd.)

- Put aside routing policies (another interesting problem tackled by others☺)
- Level of abstraction: AS graph, which is a fat-tailed and scale-free small-world
- Problem becomes: theoretical lower and upper bounds for routing on massive fat-tailed scale-free small-world graphs

Fat-tailed scale-free small-worlds

- "Small-world" = there is virtually no long paths ('remote' nodes), i.e. the distance distribution has small average and dispersion
- "Fat tail" (e.g. power-law) of the node degree distribution = there is a noticeable amount of high-degree ('hubby') nodes ⇒ the graph has a 'core' ⇒ small-world
- "Scale-free" node degree distribution (e.g. power-law) = there is no 'hill' (characteristic scale) in it ⇒ there is a lot of low-degree ('edgy') nodes ⇒ the graph is 'hairy'
- Colloquially: scale-free = power-law

Assessment of known facts: networking community

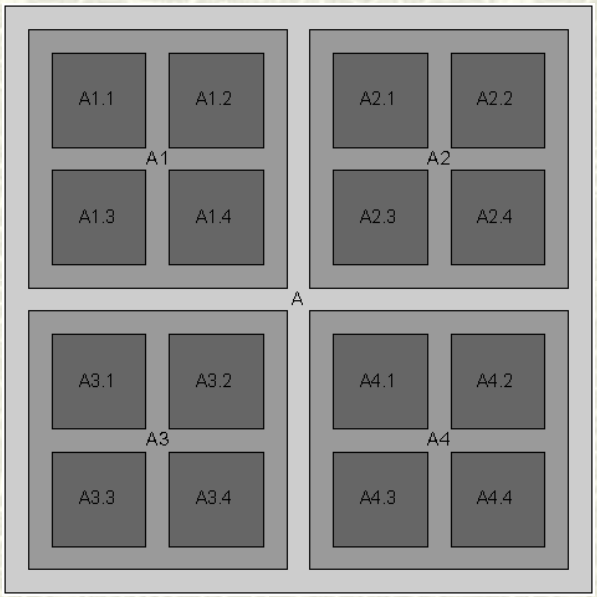
Hierarchical aggregation, multiple level of abstraction, i.e. Nimrod, MLOSPF, ISLAY, i.e. Kleinrock-Kamoun's hierarchical routing scheme of 1977 (KK).

But: there is a cost associated with KK routing table size reduction: path length increase. It depends strongly on a particular topology

KK path length increase

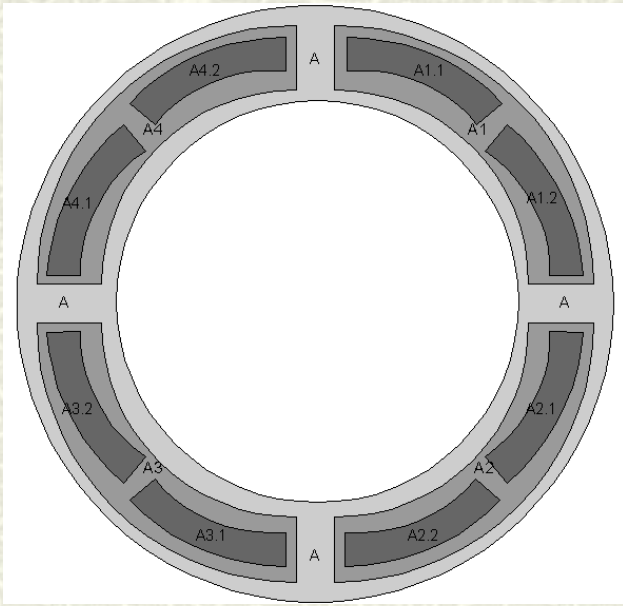
Sparse topology

- $\langle L(n) \rangle \to \infty$, $\langle L_{kk}(n) \rangle \to \infty$ s.t. $\langle L_{kk} \rangle / \langle L \rangle \to$ const.
- There are remote points



Dense topology

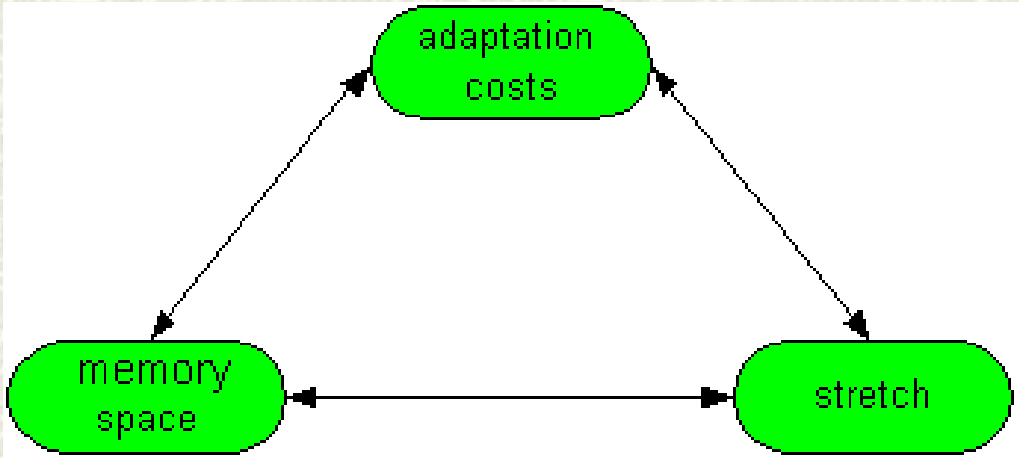
- $\langle L(n) \rangle$=const. ($\langle$degree$\rangle \to \infty$ instead) but $\langle L_{kk} \rangle \to \infty$ so that $\langle L_{kk} \rangle / \langle L \rangle \to \infty$
- There are no remote points, so that one cannot usefully aggregate, abstract, etc., anything remote—everything is close



What does path length increase mean in practice?

- Consider a couple of peering ASs. Their peering link is the shortest path between them. Non-shortest path routing may not allow them to use it, which is unacceptable.
- BGP is shortest path **if we 'subtract'** policies (there is no view of global topology anyway). Distance and path vector algorithms are 'shortest path' algorithms by definition.
- Path length increase associated with routing table size decrease is a concern. On the AS topology, the KK scheme produces 15-times path length increase. Can anyone do better?

Assessment of known facts: distributed computation theory



Triangle of trade-offs:

- Adaptation costs = convergence measures (e.g. number of messages per topology change)
- Memory space = routing table size
- Stretch = path length inflation

Crystallization history (contd.)

Simplify the task: put adaptation costs aside, i.e. assume they are unbounded, i.e. consider the static case. Reasons include:

- BGP adaptation costs are unbounded (persistent oscillations)
- The negative answer (memory space and stretch cannot be made simultaneously small on scale-free graphs) was expected. Reasons:
  - KK stretch on the Internet
  - High stretch of other schemes on complete network and classical random graphs

End of story

- Done: considered the “best” routing scheme known today (by Thorup and Zwick) and analyzed its memory-stretch trade-offs on Internet-like topologies.
- Found:
  - Both stretch and memory can be made extremely small simultaneously but **only** on scale-free graphs
  - A number of other unexpected interesting phenomena suggesting that there are some profound yet unknown laws of the Internet (and maybe some other networks) topology evolution

References

- Presentation:
  http://www.caida.org/~dima/pub/crig-ppt.pdf
- Infocom version:
  http://www.caida.org/~dima/pub/crig-infocom.pdf
- Technical report version:
  http://www.caida.org/~dima/pub/crig.pdf

Inter-Domain IP Protection

Christoph Reichert
Email: reichert@fokus.fraunhofer.de

Competence Center for
Next Generation Network Infrastructure (NGNI)

Fraunhofer Institute for Open Communication Systems

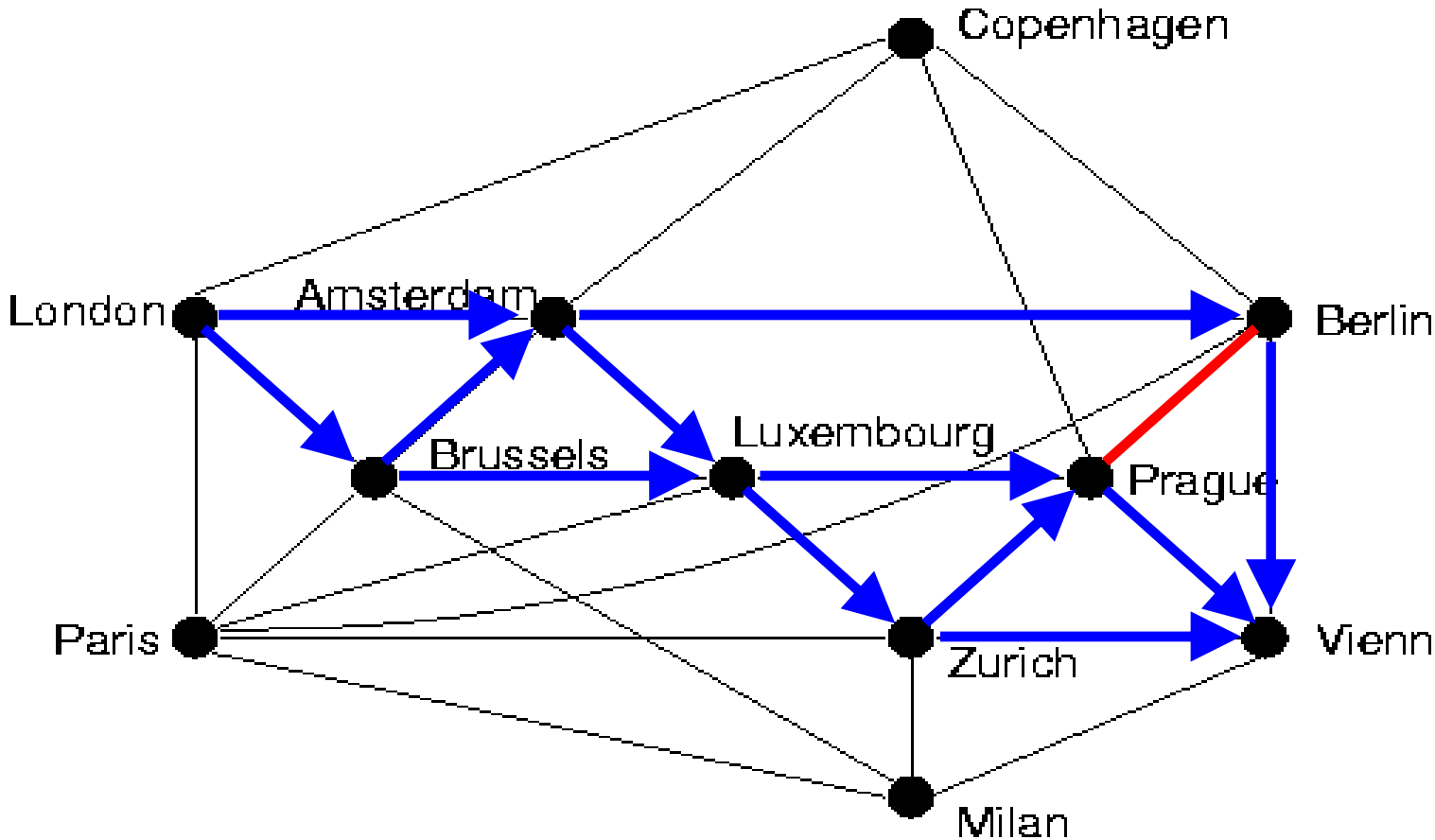
Overview

- Motivation and Approach
- Hammocks
- Topology Requirements
- Inter-domain Issues

Motivation and Approach

- Goal: high availability (“five nines”)
- IP Protection means
  - to enable a *Fast Local Reaction* in case of failures,
  - by providing a backup route *before* the primary route fails, and
  - to do this for connection-less datagrams.
- Axiomatic Approach: No Signaling after failures (fast local reaction!)
  - Rationale: It’s the fastest approach (proactive).
- New Intra-Domain Routing scheme required
  - Provide *two* next hops for each destination at each router $\Rightarrow$ O2 Routing.
  - Keep link state *protocols*, but replace Dijkstra’s SPT *algorithm*.

Example: Hammock from London to Vienna



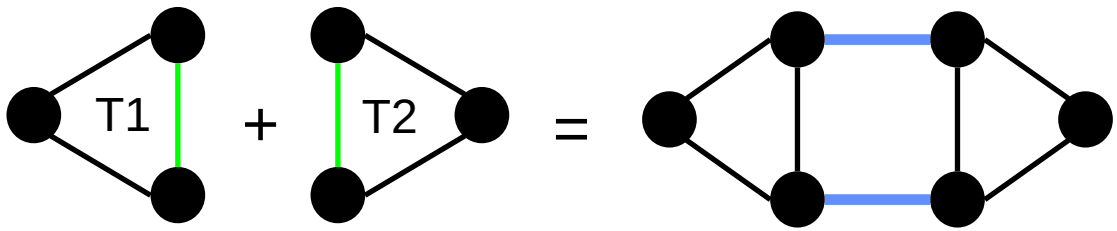
Classification of Resilience Mechanisms

	Reactive (“slow”)	Proactive (“fast”)
Connection-oriented/ Circuits	Path Restoration	Path Protection
Connection-less/ Datagrams	Dynamic Rerouting	O2 Routing

Topology Requirements

- Definition:
  - A topology is *O2 capable*, if and only if hammocks can be constructed between every pair of nodes.
  - Assumption: Each node is source and destination.
- A topology is O2 capable, if
  1. the topology is bi-connected (no single failure disconnects the network),
  2. each node belongs to a triangle, and
  3. the removal of any simple cycle of length 5 or higher leaves the remaining network connected.

A Graph Operation preserving O2 Capability



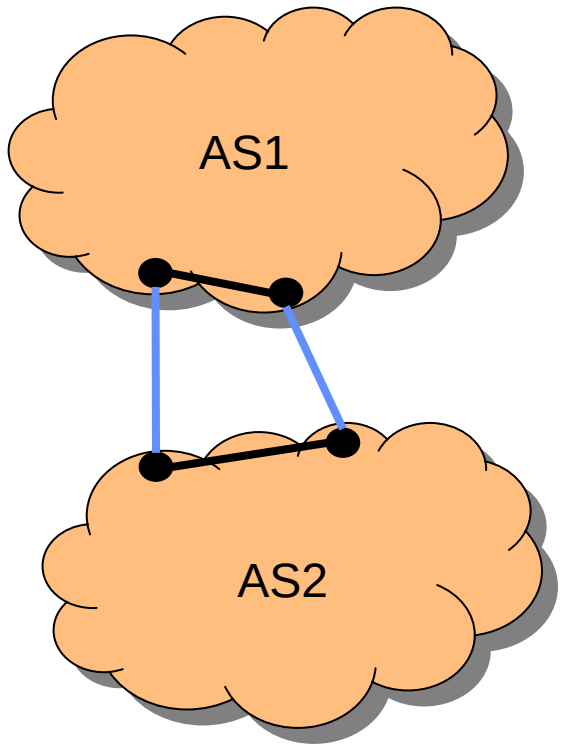
- T1, T2 are O2 capable ⇒ T1 + T2 is also O2 capable.

Inter-domain Issues

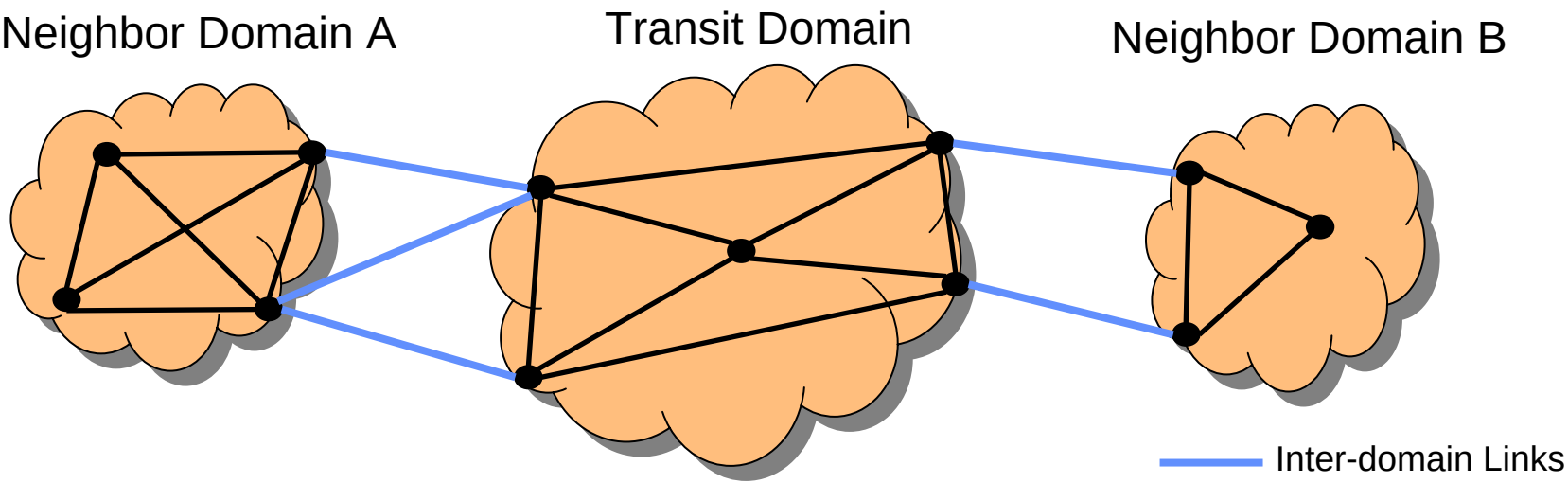
- Goal:
  - enable Fast Local Reaction at AS boundaries.
- Issues:
  - Peering
  - Representation of neighbor domains

Peering

- Obviously
  - at least two peering links are required,
  - these links must end at different nodes at both sides.
- New Requirement
  - Border routers of a domain must be neighbors!
  - “Border Twins”
  - that’s all!

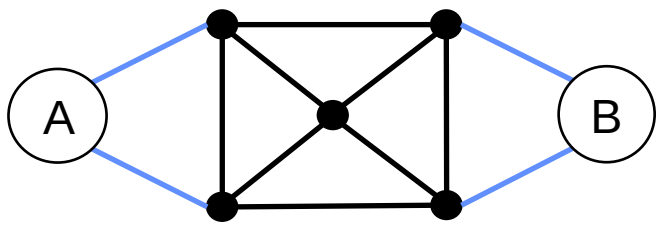


Representation of Neighbor Domains as Single Nodes



- E-BGP speakers announce "Virtual-AS-Node" LSAs into the link state IGP!

View of Transit Domain:



Fraunhofer FOKUS Institute for Open Communication Systems

Conclusion

- Given two domains are already O2 capable
  - there are additional topology requirements for peering,
  - but they are easy to meet.
- BGP/IGP interaction needs to be extended
  - to represent neighbor domains as single nodes in the topology.

References

- G. Schollmeier et al.,”Improving the Resilience of IP Networks”, IEEE High Performance Switching and Routing, 2003
- C. Reichert, “Topology Requirements for Resilient IP Networks”, submitted to MMB & PGTS 04

Stabilizing the BGP control plane

IDRWS 2004

Götz Lichtwald, Roland Bless
Institute of Telematics
University of Karlsruhe
Germany

Motivation

BGP suffers from:

- Too many unnecessary BGP update messages [1], due to …
  - … router OS-Bugs
  - … human factors
  - … or physical line problems
- Propagation scope of BGP update messages is not restricted
- Global BGP updates stress routers unnecessarily

Existing BGP improvements:

- Route Flap Damping [RFC 2439]
- Graceful Restart Mechanism for BGP [raft-ietf-idr-restart-09]
- NOPEER Attribute [RFC 3765]

[1] – R. Mahajan, D. Wetherall, T Anderson "Understanding BGP Misconfiguration"; Sigcomm 2002

Objectives

- Do **NOT** change BGP, but improve it
- Provide a fast inter-domain failure reaction (faster than pure BGP)
- Do not immediately *broadcast* any inter-domain failure (➔ Limit notification scope to affected ASes)
- Reduction of foreign determined resource consumption

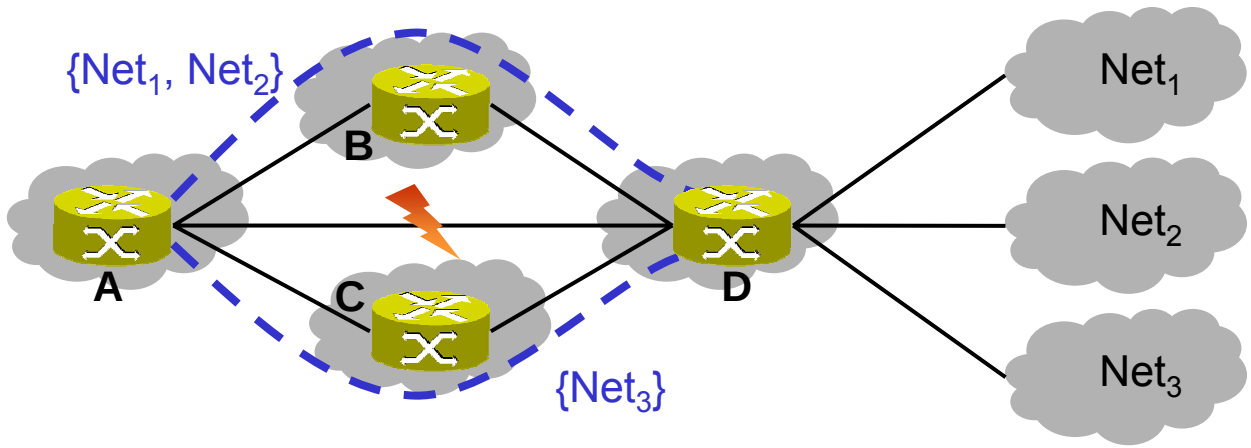
Basic Fast Scoped Rerouting Concept

- No inter-domain path failure ➔ $BGP_{Fast\ Scoped\ Rerouting} \approx BGPv4$
- Inter-domain path failure ➔ Fast Scoped Rerouting (FaSRo) takes over
- Failure handling on **two time scales**:
  - **Fine granular time scale** (≤ **T** min) ➔ **BGP_{FaSRo}**
    - Setting up the FaSRo-Path
    - Traffic is redirected to the FaSRo-Path
    - If link recovers ➔ switch back to BGP
      - ☺ BGP update process was not needed
      - ☺ Only affected peers a stressed
    - link failures seems persistent ➔ switch back to BGP and start BGP update process
      - ☺ BGP takes control of the failure handling
      - ☺ BGP update concept is not broken—only delayed
  - **Coarse granular time scale** (> **T** min) ➔ **BGP**

FaSRo Path variation (1/2)

FaSRo-Fan

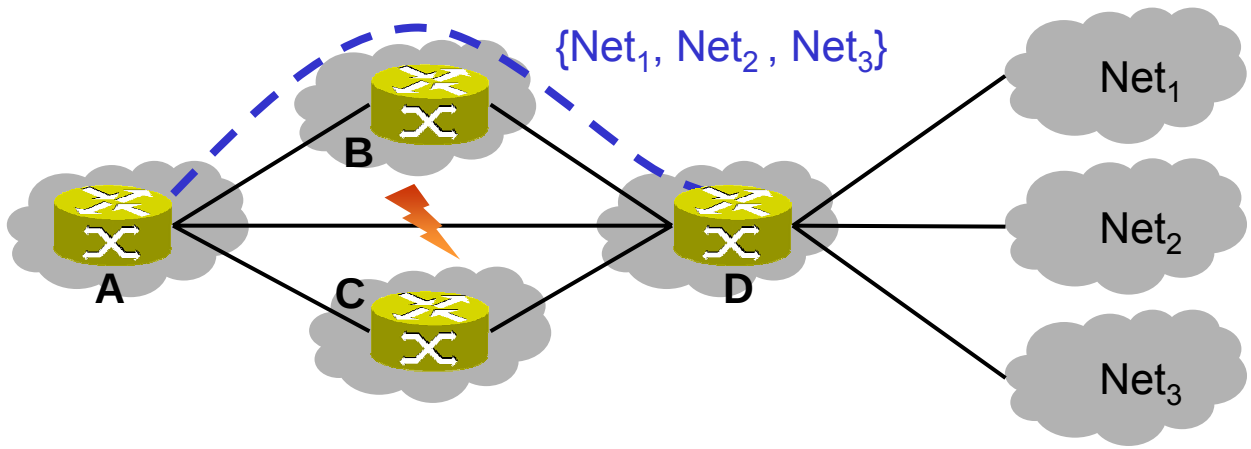
- Setting up a FaSRo-Path per destination network



FaSRo Path variation (2/2)

Only ONE FaSRo-Path

- Providing only one FaSRo-Path for all destination networks



FaSRo Path

Autonomous System

Pros and Cons of the FaSRo-Variations

Fan-Variant

- ☺ No policy violation
- ☺ Switching traffic to alternative path
- ☺ Simpler than pure BGPv4
- ☹ Signaling overhead (FaSRo-Path per destination network)
- ☹ Per destination network a FaSRo-Path has to be maintained
- ☹ Not optimal routes for a short period of time

One-Path-Variant

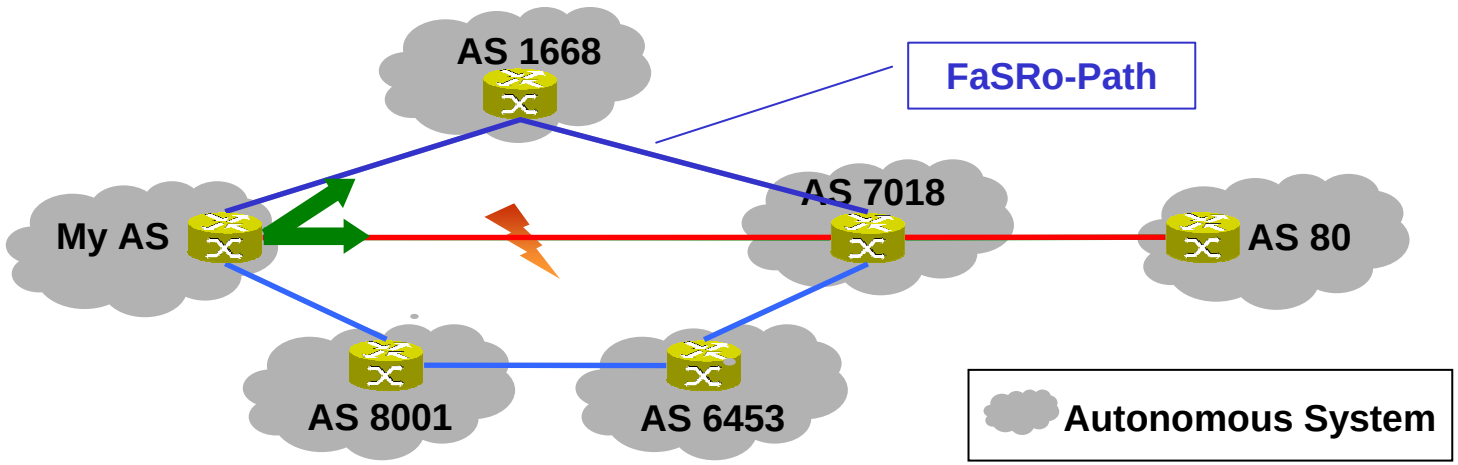
- ☺ Only one substitution for the broken AS Path
- ☺ Less signaling effort to set up and maintain the FaSRo-Path
- ☺ Simpler than pure BGPv4
- ☹ Short time policy violations
- ☹ Risk of bandwidth scarcity
- ☹ Not optimal routes for a short period of time

One-Path-Variant makes sense, as only short time failures are handled!

What is a FaSRo-Path

Network	Next Hop	AS-Path
*> 3.0.0.0	12.0.1.63	7018 80
	66.185.128.48	**1668 7018 80**
	209.123.12.51	8001 6453 7018 80
…		

RouteView.org



(1) Failure (from “My AS” point of view)
(2) Look for substitution
  (a) Scan routing table for another AS-Path that ends at AS 7018
  (b) Select shortest path of those (=FaSRo-Path)
(3) Redirect Traffic
  (a) Scan FIB for all entries that have 12.0.1.63 as next hop
  (b) Rewrite entry to FaSRo-Path

Evaluation

- Simulation run with SSFNet [2]
  - Simulation setup without policies
  - Changed amount of Autonomous Systems from 3 to 500 ASes Topology was …
    - … hand made
    - … extracted from AS-Topology from Route-View [3]
    - … generated by Brite [4]

- Real Results
  - Running code for Quagga [5] (former Zebra [6])
  - Signaling of FaSRo-Path works
  - Traffic is redirected

[2] – SSFNet http://www.ssfnet.org/
[3] – Route-View http://www.routeview.org/
[4] – Brite http://www.cs.bu.edu/brite/
[5] – Quagga http://www.quagga.net/
[6] – Zebra http://www.zebra.org/

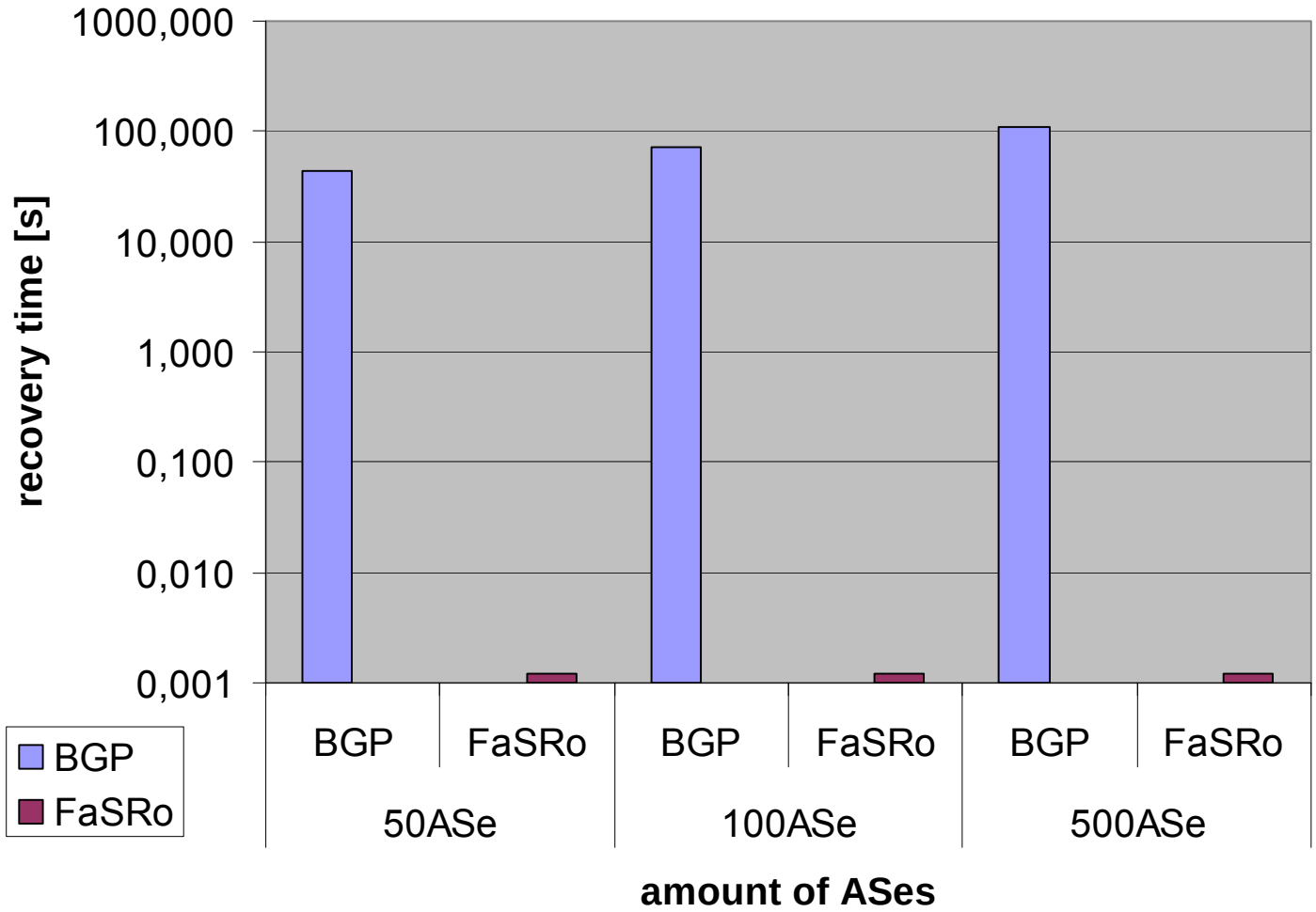
What was measured

- Observed scenario:
  Inter-Domain loss of connectivity
  - What happens with pure BGP:
    (1) T_{detect} – Describes the time until BGP recognizes the failure
    (2) $T_{convergence}$ – Time until the failure is “fixed”
  - What happens with BGP_{FaSRo}:
    (1) T_{detect} – Describes the time until BGP recognizes the failure
    (2) $T_{FaSRoPathSetup}$ – Time until the FaSRo-Path is setup

- Influence on the
  - Network:
    - BGP – Broken path is recovered quite fast, but network is unnecessarily stressed with updates
    - BGP_{FaSRo} – Quick substitution of broken path and **no further** implications on the network
  - User:
    - BGP – Possible route changes until network is converged
    - BGP_{FaSRo} – Only **ONE** short rerouting

Simulation of BGP vs. BGP_{FaSRo}

- Motivation
- Objectives
- Basic Concept
- **Evaluation**
- Conclusion



Scenario: Inter-Domain loss of connectivity

Measured for

- BGP : Time from first seen update to last seen update (= convergence)
- FaSRo : Time until the traffic was redirected to the FaSRo-Path

Conclusion

- FaSRo is a simple mechanism
- Depending on duration of the fine granular time scale 40% – 60% of inter-domain path instabilities could be hidden [7]
- Provides a fast inter-domain failure reaction
- Stability of control plane could significantly improved by FaSRo

[7] – K. Singh; “A survey of Internet routing reliability”; IRT internal talk, Columbia Computer Science; April 2003

Feedback appreciated – for example …

- … from operators and others:

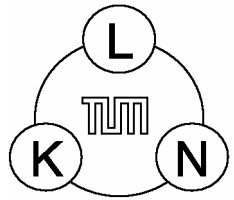
  Apart from “We don’t like BGP changes” and it is a new potential for further mis-configurations, what do YOU think about this concept? For whom might FaSRo be sensible?

- ... from vendors and others:

  Can this kind of BGP extension be easily be implemented, does it affect the stability of the current BGP implementation, are there—from your side—any efforts to do something similar?

- … from academics and others:

  Any concerns, wrong assumptions?



Independence of Inter-Domain QoS Signaling and Routing*

Thomas Schwabe, TU München
(thomas.schwabe@ei.tum.de)

Thomas Engel, Siemens AG
(thomas.engel@siemens.com)

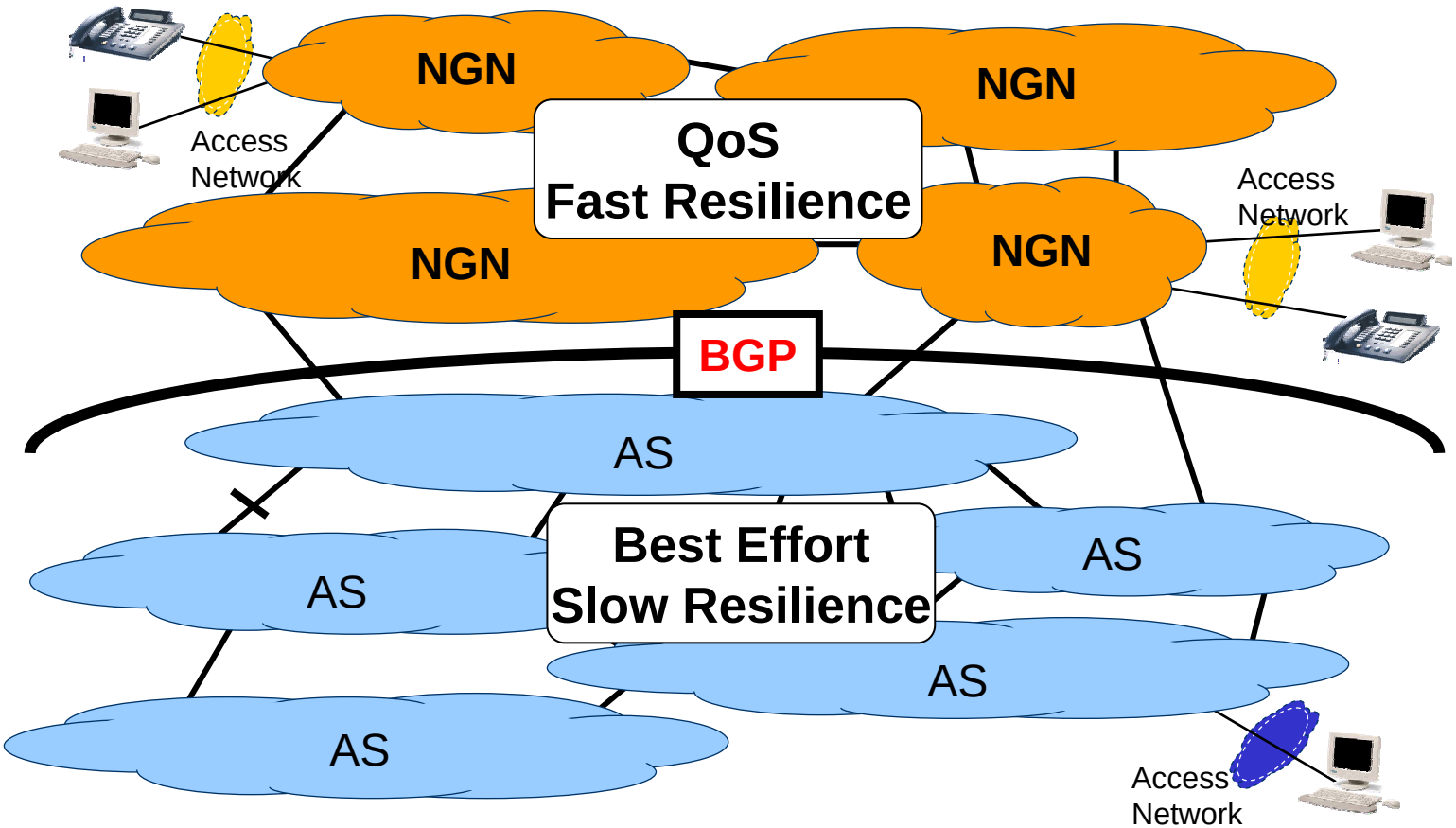
***This work was partially funded by the Bundesministerium für Bildung und Forschung (ministry of education and research) of the Federal Republic of Germany under contract 01AK045. The authors alone are responsible for the content of the slides.**

Motivation

- Convergence of networks
- QoS and resilience support
  - Mechanisms for differentiation of QoS and BE traffic
  - Network admission control
  - Resource reservation (Intra- and Inter-domain)
  - ...

Assumptions

Goal - End-to-end QoS support



Resource reservation

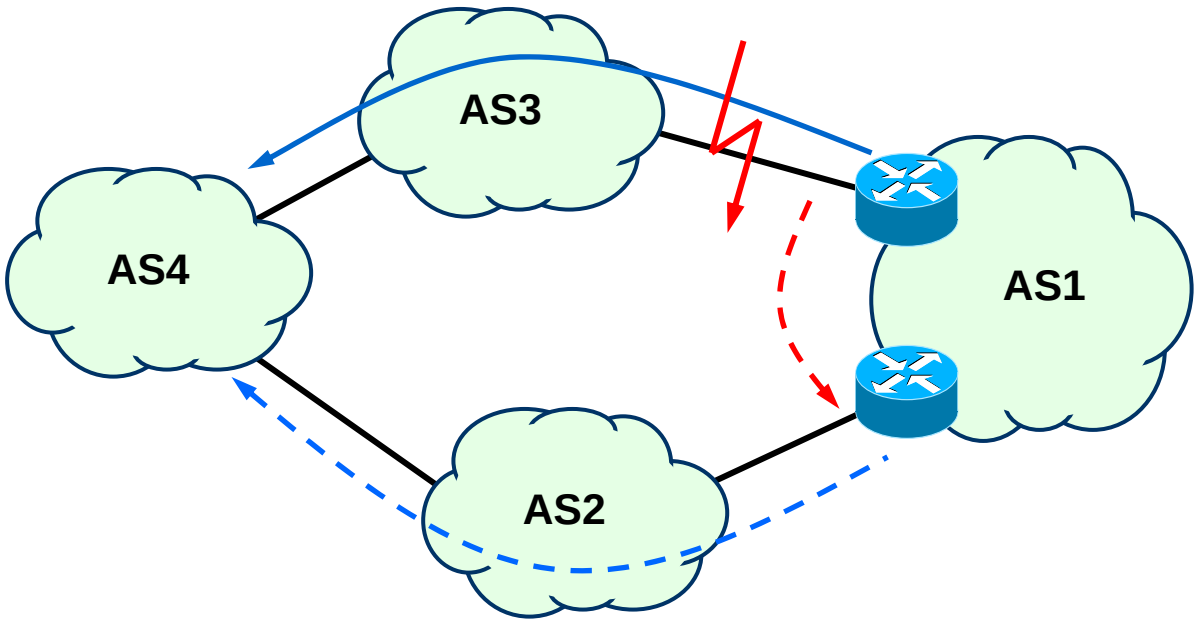
- Intra-Domain – out of scope
- Inter-Domain resource reservation
  - Different solutions like BGRP, SICAP etc.
  - Independent of an implementation
  - Dependency of the routing
- Challenges:
  - Interworking with BGP
  - Requirements on BGP
  - ...

Normal Case

- Stable routing – no changes
- Resource reservation (RR)
  - Setup of a new flow
  - Enough resource are available
  - Everything is ok!

Worst Case- Link failure

- Shift all QoS traffic from the broken link to the backup link



Possible procedure

1. BGP rerouting
   - Well known
   - needs time for convergence (in the range of minutes)
2. Resource re-allocation:
   - Has to be done after the rerouting
   - RR for all effected flows (or aggregated flows)
   - Takes time for signaling too (minimum one RTT per flow)

Challenges

- When is BGP converged?
  - Introduction of a timer
  - Value – multiple of the MRAI timer?
- RR doesn't wait on the convergence of BGP ⇒ waste of bandwidth
  - Reserve resources on temporary paths
  - Timer based de-allocation ⇒ extra time

Consequences

- No fast Inter-domain rerouting
  - Rerouting of QoS traffic will be in the range of seconds or minutes
- During the rerouting - no support of QoS
- Need for:
  - Fast convergence of BGP
  - Fast convergence of the RR
  - **Optimal interworking of signaling and routing**

Another case - Traffic Engineering

- Reaction on overload situation
  - Exceed a limit of reserved QoS traffic
  - Congestion of the signaling traffic
- Different situation:
  - Traffic can be forwarded on the old path
  - Time for preparation:
    - Setup of an alternative path
    - RR for the effected QoS traffic
    - Shifting of the QoS traffic
  - But then need for a differentiation between failure and TE reaction

What happens if the RR fails?

- What means fails?
- No resources can be allocated
  - Looking for a new path
- Only a part of the resource can be reserved
  - Need for a new path too
  - Completely shift of the QoS traffic
  - Splitting of prefixes $\Rightarrow$ how?
  - .....

Consequences

- Testing of new paths for enough free resources
  - Enlarge the convergence time
- QoS impairment for a long time
  - May be ok for TE reaction, but not if a failure occurs

Conclusion

- Different reaction on a failure and for TE reasons
- Failure reaction – always QoS impairment
- TE rerouting – subsecond traffic shift should be possible
- No idea – how to handle limited QoS resources

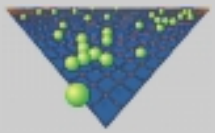
Open questions

- Is the independence of the resource signaling the right way of thinking?
- Do we need one combined routing and signaling protocol for QoS traffic?
- Is routing based on traffic load and QoS class a solution?

Inter-domain Resilience for QoS Traffic

IDRWS'04

Thomas Engel, Siemens AG, engel.thomas@siemens.com
Thomas Schwabe, Munich University of Technology, thomas.schwabe@tum.de

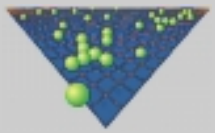


This work was partially funded by the Bundesministerium für Bildung und Forschung (ministry of education and research) of the Federal Republic of Germany under contract 01AK045. The authors alone are responsible for the content of the slides.

Inter-domain Resilience for QoS Traffic

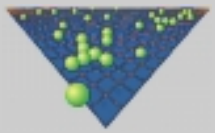
Outline

- **Resilience target**
- **Failure reaction**
- **Approaches**
  - fast convergence
  - stable routes



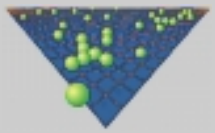
Goal

- **Next generation IP networks will provide both QoS and resilience**
  - QoS
    - a small number of DiffServ based QoS classes
    - admission control
    - reservation requests
    - resource management
  - Resilience
    - link, interface and node failures (hardware and software) do not affect QoS traffic
    - carrier grade availability of QoS for QoS traffic: 99.999%
    - less than 5,26 minutes of QoS violation due to link, interface and node failures



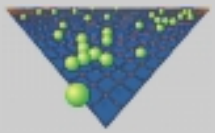
Intra-domain Resilience

- **Failure recovery**
  - failure event: node, interface or link failure
  - failure detection
    - frequent hello messages
    - lower layer failure notification
  - distribution of topology changes and route calculation by OSPF or IS-IS
  - resource management
    - pro-active configuration of admission control guarantees agreed QoS for accepted traffic even after a link failure
    - adaptation of admission control after a failure
- **fast sub-second recovery**
- **a low number of sub-second QoS violations are tolerated (SLA)**



Inter-domain QoS Traffic

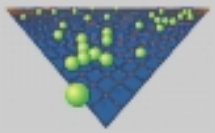
- **Assumptions**
  - BGP selects inter-domain routes
  - an inter-domain resource management (RM) cares about resource provisioning for QoS traffic
    - request based admission control and resource provisioning similar to RSVP at AS path from origin to destination AS
    - for details see presentation of Thomas Schwabe
  - RM follows route selection of BGP
  - BGP selects routes independently of RM



Inter-domain Failure Recovery

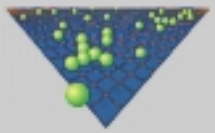
- **recovery from a link or node failure, best case:**
  - failure detection via missing KEEPALIVE messages
  - short BGP convergence process
  - RM detects route change
  - RM allocates the required resource at the new route

- **recovery, worse case:**
  - failure detection via missing KEEPALIVE messages
  - BGP reroutes QoS traffic
  - RM detects rerouting and adapts resource allocation
  - BGP reroutes QoS traffic
  - RM detects rerouting and adapts resource allocation
  - BGP reroutes QoS traffic
  - ...



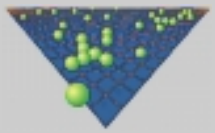
Inter-domain Failure Recovery

- **Problems**
  - Can agreed QoS be provided during BGP convergence processes?
    - QoS provisioning requires resource allocation by RM as a reaction on BGP rerouting activities
    - there is no QoS guarantee in the time period after a route change until RM has allocated the required resources at the new route
    - even worse redirected QoS traffic interferes with QoS traffic already using sections of a new route
    - QoS will be violated during convergence time
  - mean convergence time is 3 minutes according to Craig Labovitz



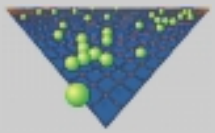
Optimising Convergence Time Through MRAI

- **BGP convergence time depends on MRAI**
  (Minimum Route Advertisement Interval)
- **improved BGP convergence by MRAI reduction (see T. G. Griffin)**
  - default MRAI = 30 sec
  - scenarios evaluated by T. G. Griffin (clique of size 15):
    - optimal MRAI: 7 sec
    - ratio of convergence time with optimal MRAI to convergence time with default MRAI = 0.3
  - How many rerouting events are possible in 5,26 minutes?
    - simple model using results from T. G. Griffin
    - optimised mean convergence time: 3min*0.3 = 54sec
    - number of rerouting events in 5.26min: 5.26 / 54sec = 5.8
- **A low number of rerouting events per year will violate the resilience target!**



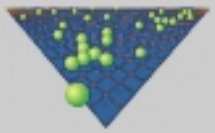
Inter-domain Failure Recovery

- **To reach the resilience target either:**
  - improve BGP convergence
    - speed up information transfer: disable MRAI
    - backup paths
    - mult-path routing
  - avoid route changes
    - resilient chains
    - local rerouting
  - avoid QoS impairments during BGP convergence
    - for further study



Improvement of BGP Convergence Time

- **set MRAI = 0**
  - ☹ huge processing load
  - ☹ does not improve convergence time according to investigations of T. G. Griffin
- **prepare a backup path**
  - ☺ new route is already prepared in the failure event
    i.e. convergence time = 0
  - ☺ simple resource allocation at backup path
  - ☺ further improvements by pre-allocation of resources
  - ☹ size of BGP routing table
  - ☹ BGP has to be changed to enable backup paths
- **multi-path routing**
  - similar to backup routes with backup routes utilised all time

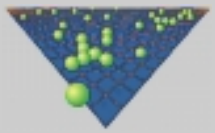


Stable Routes

- **Resilient Chains**

  - each AS provides resilience
  - next hops are resilient, i.e. intra-domain link and node failures are not recovered by redirecting QoS traffic to a different neighbour
  - each exchange points X provides resilience

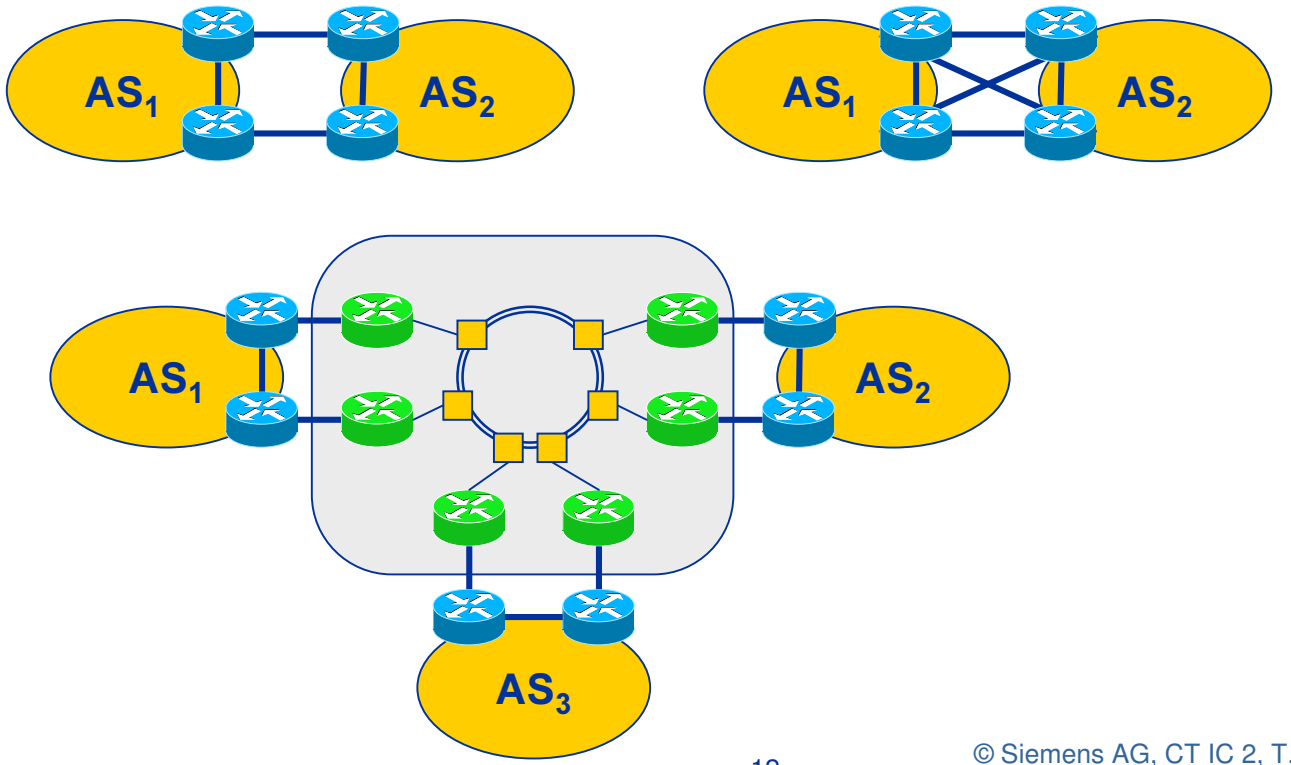
- **QoS traffic streams follow highly available, stable routes**
  - a next hop is changed due to intra-domain failures with very low probability, e.g. 10^{-5}
  - an exchange point is unavailable with very low probability, e.g. 10^{-5}
  - routes ≤ 10 AS hops are available except for 1.7 hours a year
  - with repair times in the range of minutes to hours rerouting events due to link and node failures are rare events
  - with both, resilient chains and optimised MRAI timers, the resilience target seems reachable



Information & Communications
Networks & Multimedia Communications

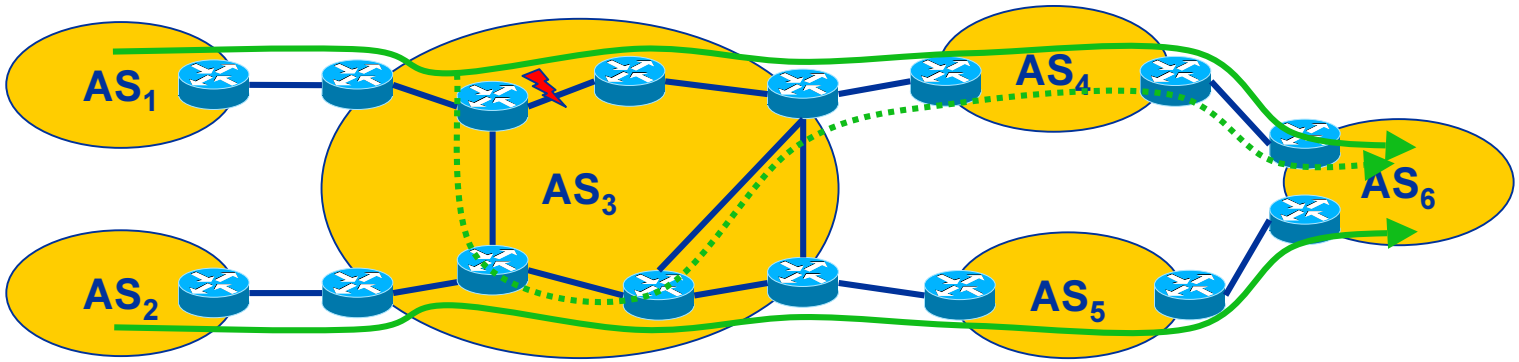
Resilient Exchange Points

- **redundant connections**
- **fast local failure recovery**



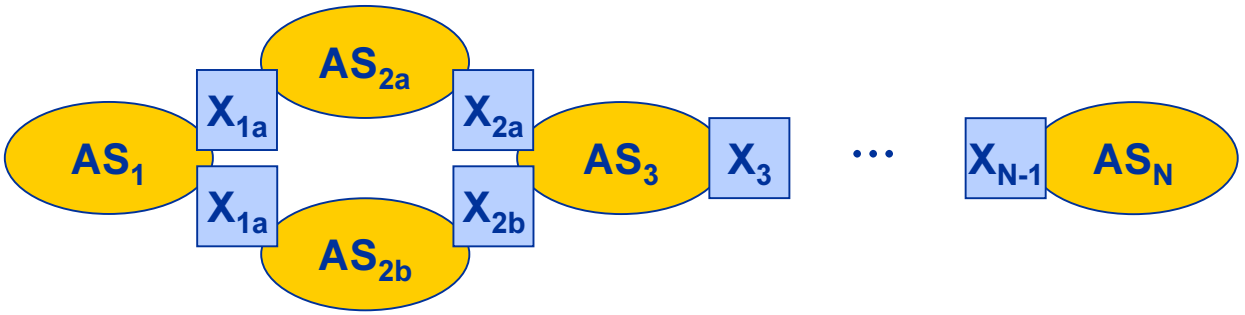
Resilient AS with Resilient Next Hops

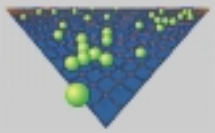
- **fast intra-domain failure recovery providing high QoS availability**
- **QoS traffic is not redirected to a different next hop because of internal link and node failures (with a very high probability)**



Local Rerouting

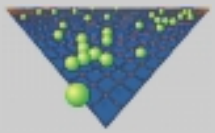
- **AS3 does not propagate whether it routes traffic destined for AS1 via AS2a or AS2b**





Conclusions

- **To enable next generation IP networks to provide QoS and resilience:**
  - either BGP convergence must be substantially improved
  - or rerouting frequency has to be reduced through AS stable paths
- **We propose to base end-to-end resilience across multiple next generation networks on resilient chains and optimised MRAI timers**
  - much more easier to implement than alternatives
- **Open issues**
  - detailed analysis of availability of resilient chains
  - how to optimise MRAI in large AS topologies
  - effect of other BGP parameters on convergence: route flap damping, ...
  - how to reduce the effect of inter-domain rerouting on QoS traffic
  - how to damp or reduce the effects of BGP rerouting activity not caused by link or node failures



Information
and Communication
Networks

SIEMENS

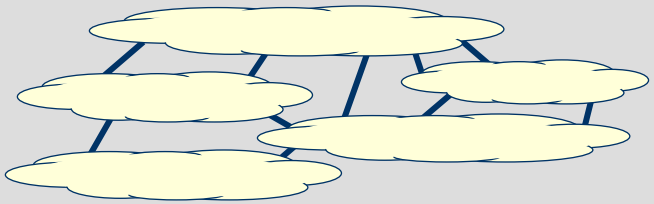
Inter-Domain Routing Issues in Next Generation Networks

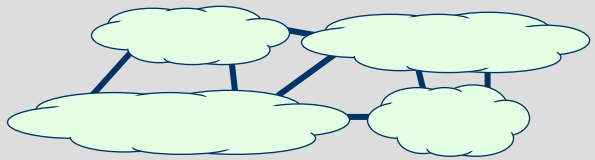
Karl J. Schrodi, Siemens AG

karl.schrodi@siemens.com

Next Generation Networks (NGN) – what makes the difference?

SIEMENS





The Internet

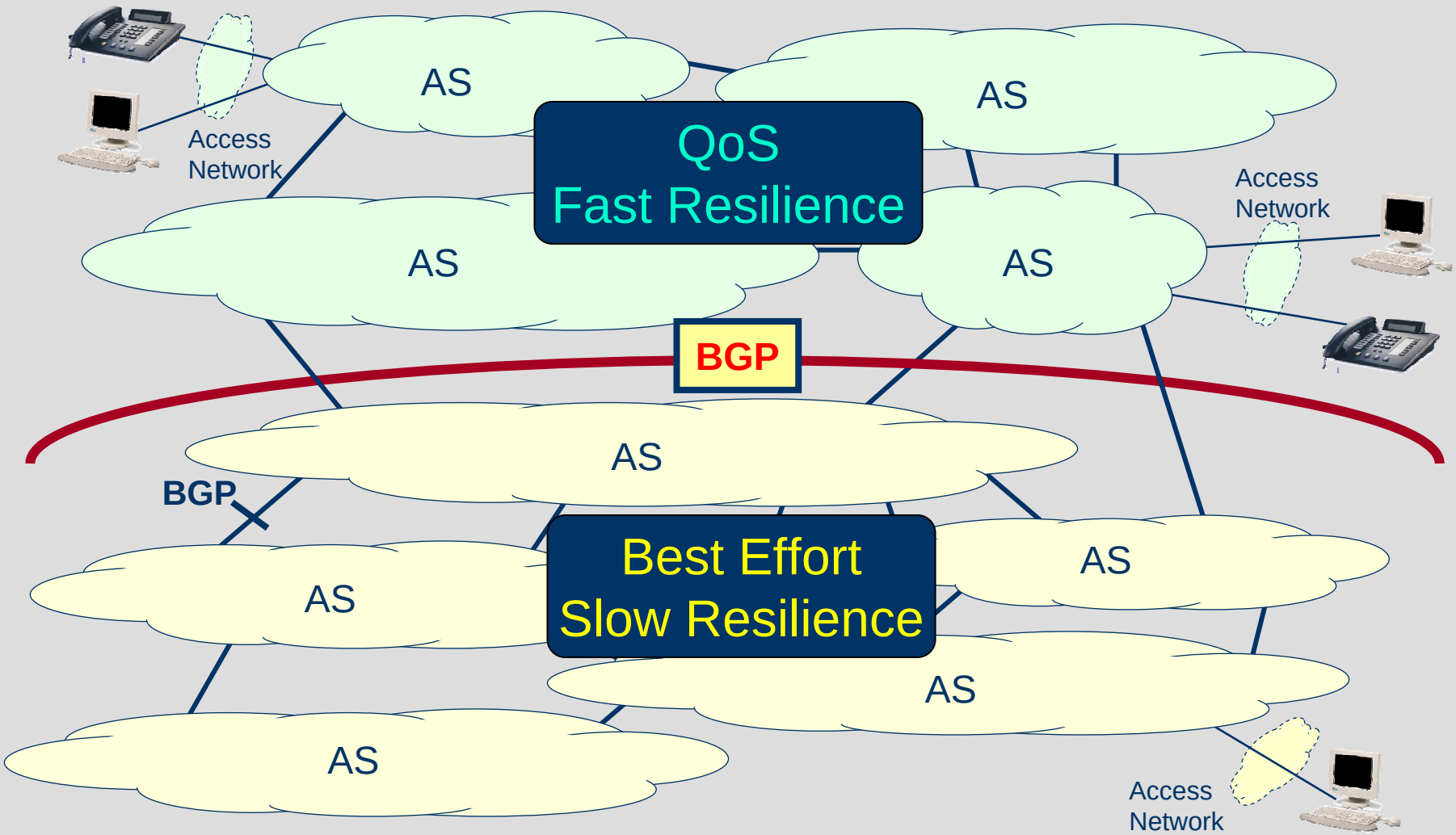
- . . .
- fair ´best effort´ service for all kinds of applications
- undetermined QoS
- good resilience (but sometimes slow)
- . . .

Next Generation Network

- . . .
- variety of services, e.g. with low delay for interactive voice and video, on top of the Internet´s
- differentiated, assured QoS
- ´five nines´ of service availability (→ ´fast´ resilience)
- . . .

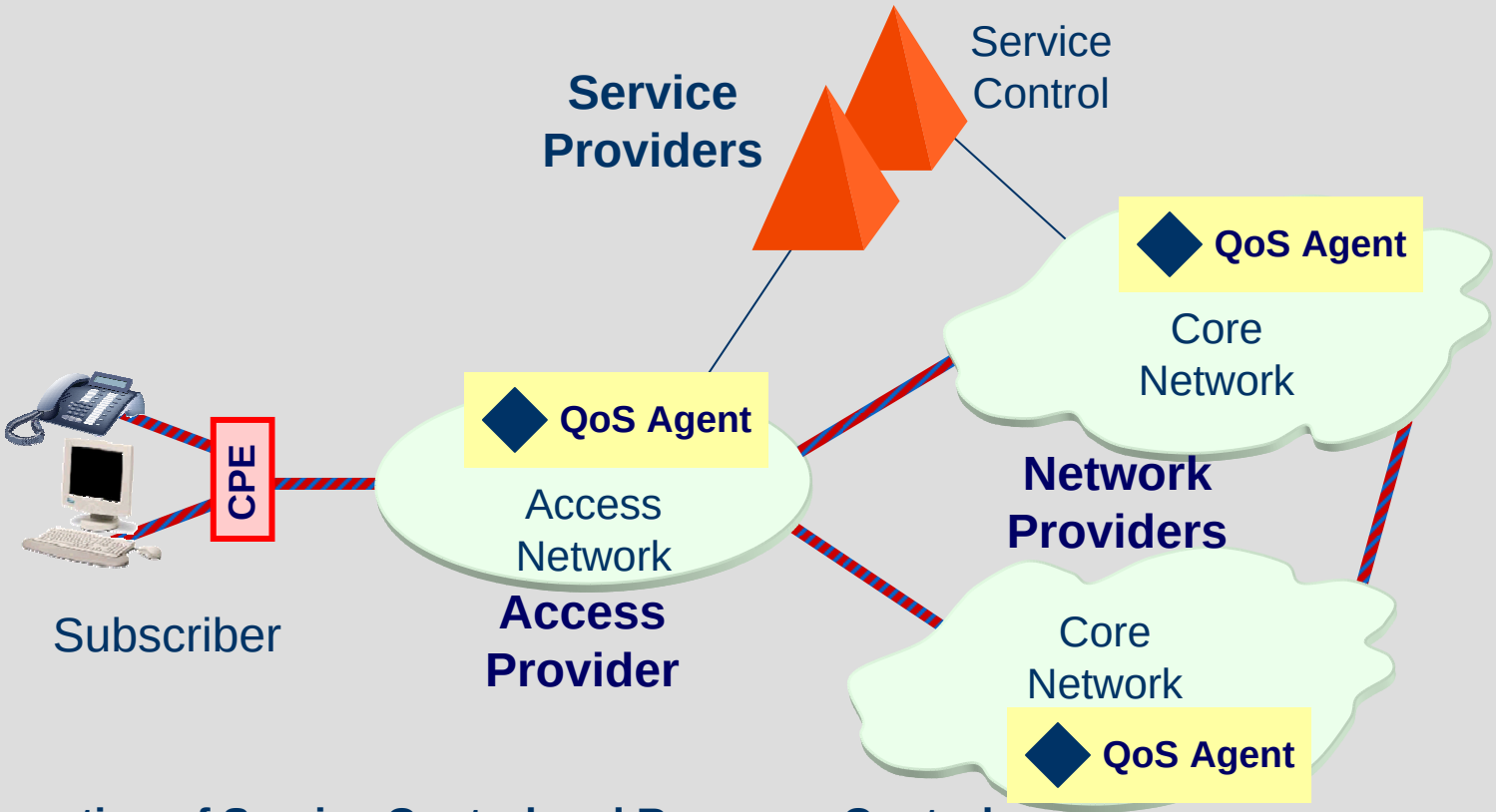
NGN subscribers use all services of the Internet
→ NGN and Internet are coupled using BGP

SIEMENS



Assured QoS requires control of network resources

SIEMENS



- Separation of Service Control and Resource Control
- Network domains need at least one instance of resource management
- Qos Agent may be centralized or distributed (functionally/physically)

Note: Overprovisioning can only provide undifferentiated and not assured QoS. It will completely fail in disaster situations.

Assumptions and issues related to routing in NGNs

SIEMENS

Assumption:

Decentralized, autonomous, connectionless routing has proven to be application independent, scalable, robust and economical in the Internet.

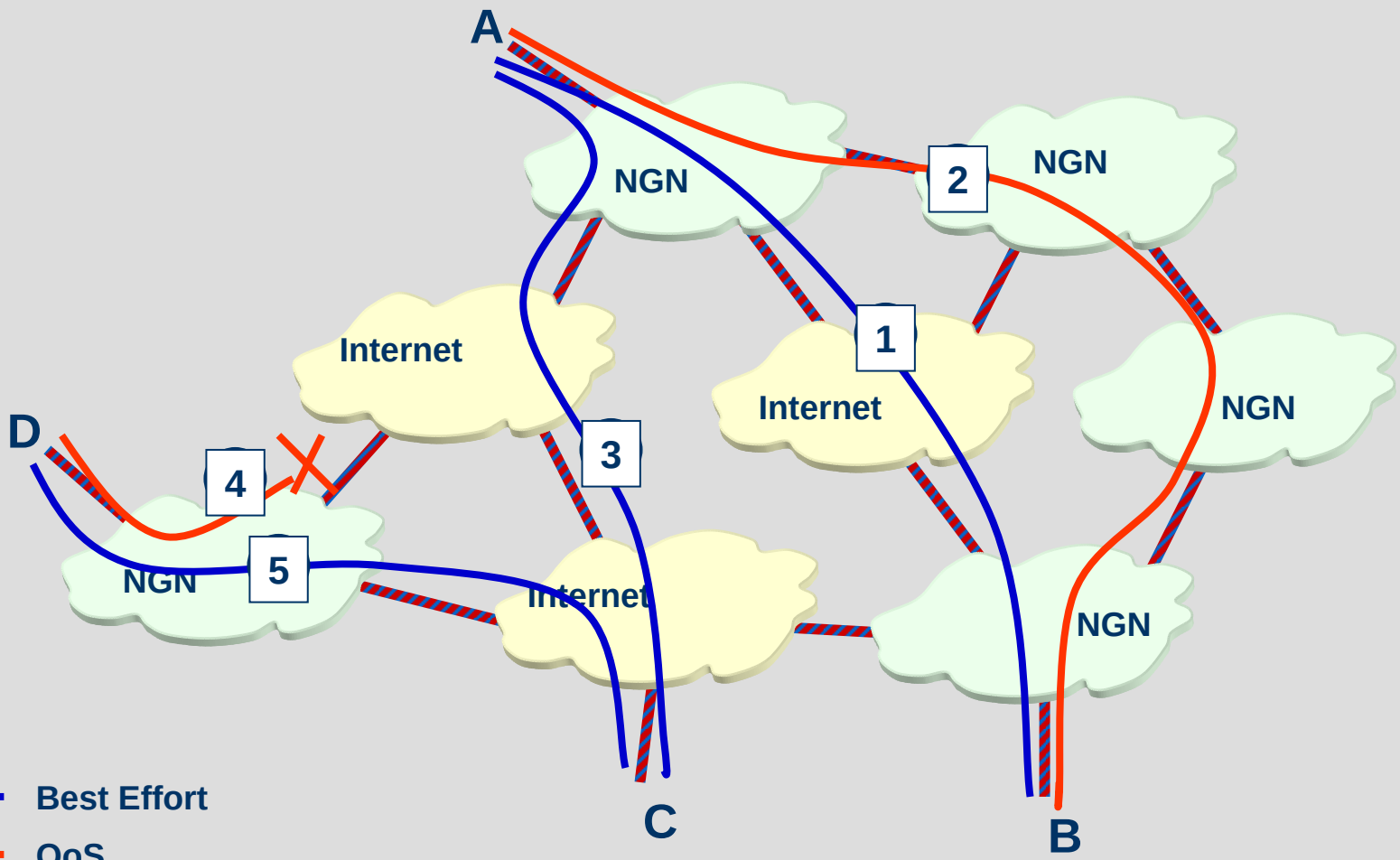
We want to adhere to these principles for NGNs, too.

Some issues (to be explained (but not solved) in subsequent slides)**:**

- **To ensure that QoS traffic remains in NGN domains: Do we need service dependent routing?**
- **How do we ensure resource/data path consistency, i.e. that resources are allocated to the actual traffic route?**
- **What happens in case of routing updates?**
- **How fast do we have to react (and converge) after a link or node failure?**

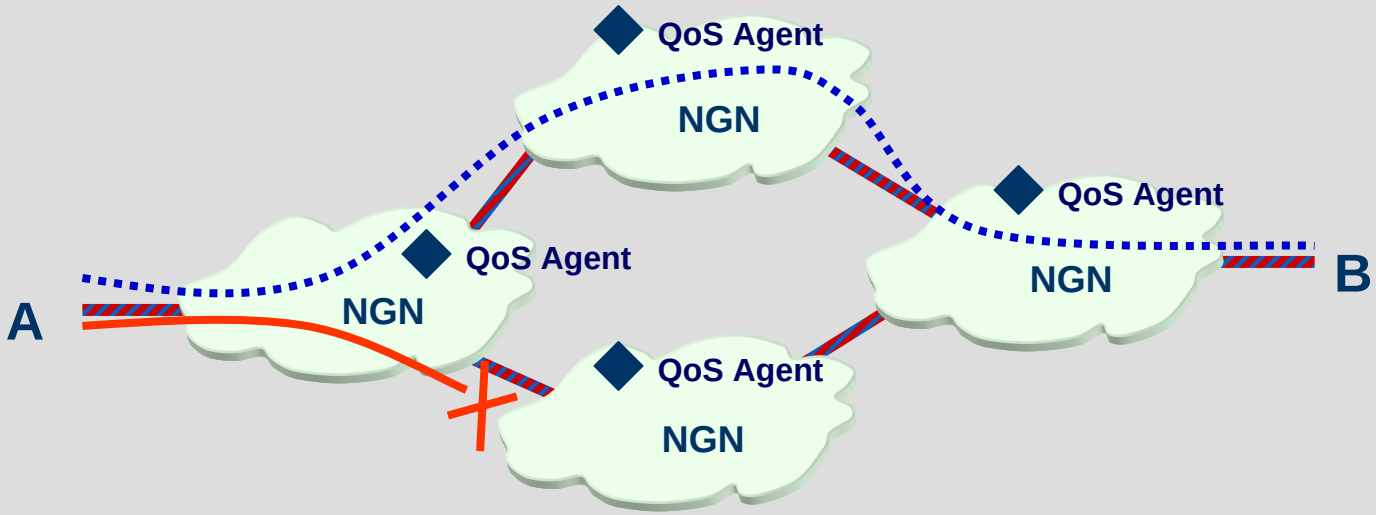
Issues (1):
Service dependent routing?

SIEMENS



Issues (2): Resource/data path consistency

SIEMENS



´Reservation path´

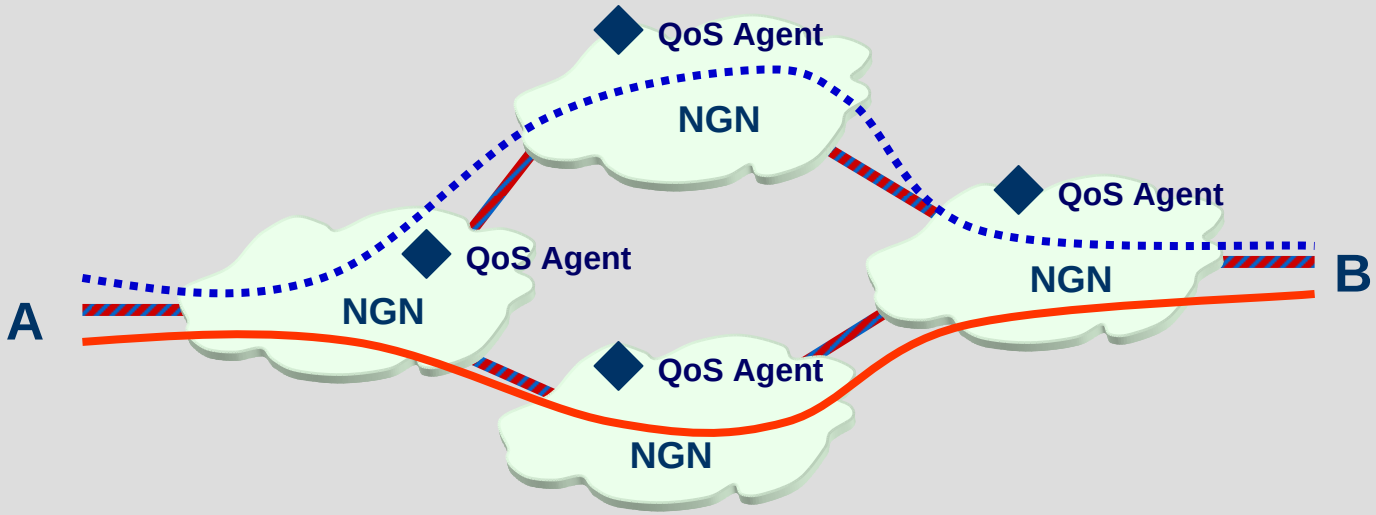
Data path

RSVP sticks to the destination routed path – but has ist weaknesses.

In case of ´path decoupled´ signaling, how do we lock the resource reservation to the actual route?

Issues (3): Routing update

SIEMENS



´Reserved´ route

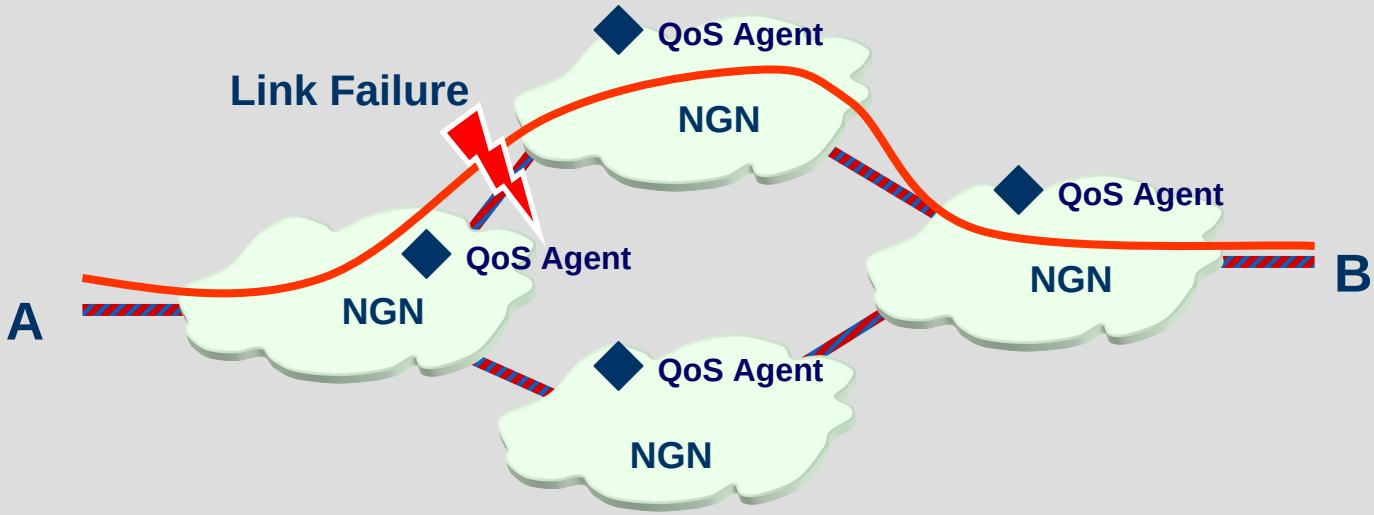
New route

How does resource management recognize route updates?

How – and how fast – does it find the new route?

Issues (4): Failure reaction

SIEMENS



´Reserved´ and used route

How fast can we detect a failure?

How fast can we react on a failure?

How fast does our reaction converge?

Thank You !

restraining route leakages

Stefan Mink

IDRWS 2

Amsterdam, 02.05.2004

quotes from ixp mailing lists

“I apologise for the leak this morning. A tiny mistype in the outgoing filters produced this leak which has been fixed.”

“Due to a router crash earlier today and a subsequent loss of config we have leaked routes and tripped the max-prefix counters on a number of peers.”

problem description

- unintended route leakages occur on a regular base
  - unintended route origination
    - e.g. IXP prefix leaks
  - unintended (**third party**) route **re**distribution
    - redistributing full table to peers: „free transit“ (and free beer at the next IXP-meeting ;)
    - redistributing former client routes (which may now be a peer)
    - **often undetected on receiver side**

root cause analysis

sender side

- configuration error
- missing maintenance

receiver side

- mu^h^hshould check **authorization** to use a received route
- ***transitive authorization***
  - authorization to use third parties routes

transitive authorization check

check of transitive authorization is a hard problem:

- RPSL
  - consistency/correctness of policies unclear
  - timeliness issues, scaling issues
- max-prefix
  - emergency stop (maybe on all links)
- future security architectures don't fix it either
  - SBGP: RA covers only direct authorization
  - soBGP: only checks ASPolicyCerts for path existance

quote

Dear peering members,

We've added some additional prefixes to AS-IS, please reset the peering for those where we've tripped max-prefix.

Please update your filters if you do that manually.

transitive authorization schemes

What primary authorization schemes exist in IDR?

- peering
  - authorization to pass received routes on **to clients**
- transit
  - client authorizes provider to pass its own and its client routes on **to everybody**
  - provider authorizes client to pass routes on **to clients**

formalize it: edge-types

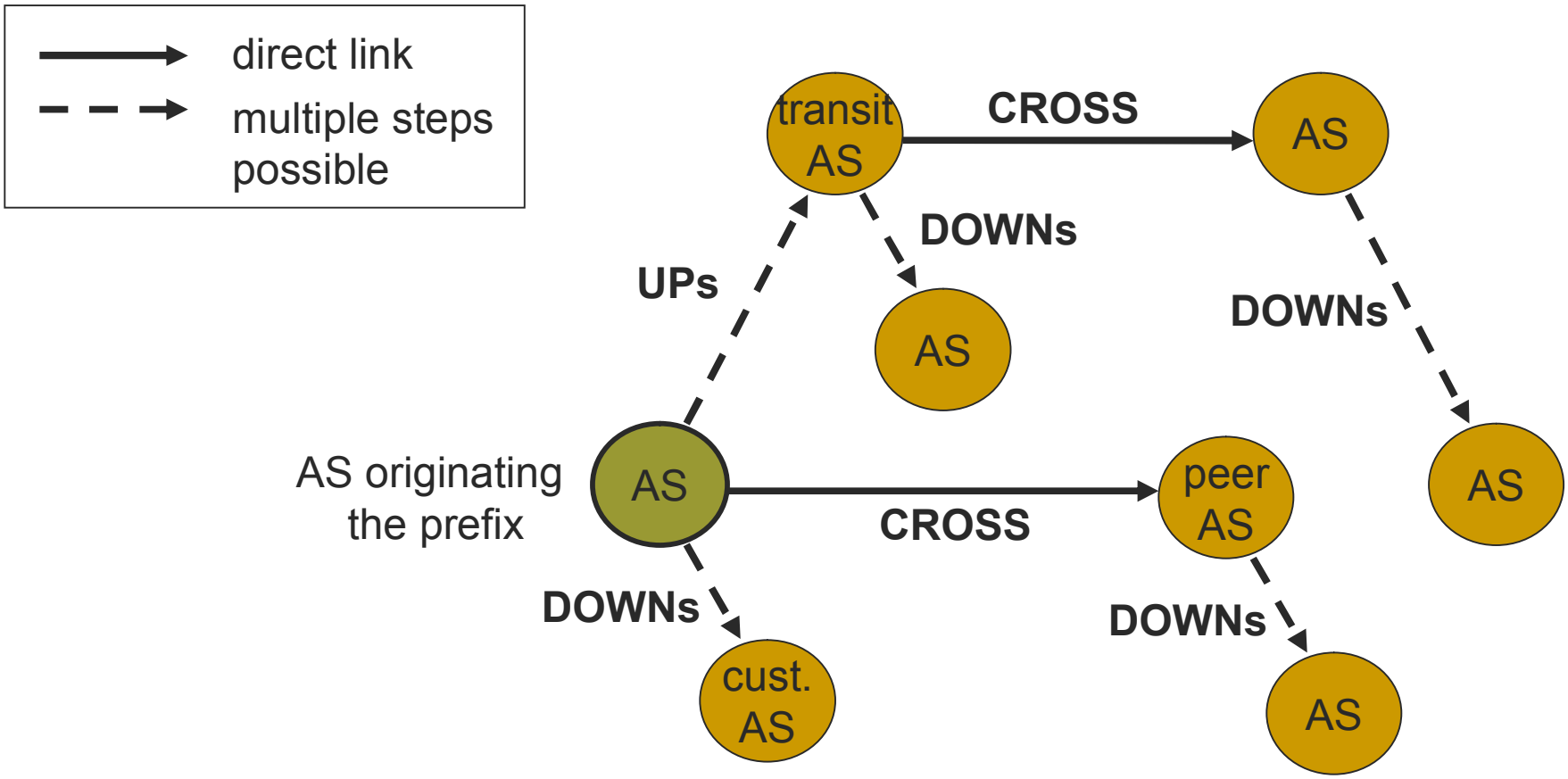
- Internet is a graph
- edge-types
  - transit
  - peering
- directed edge-types (route travel)
  - customer – provider: **UP**
  - peer – peer: **CROSS**
  - provider – customer: **DOWN**

valley freeness

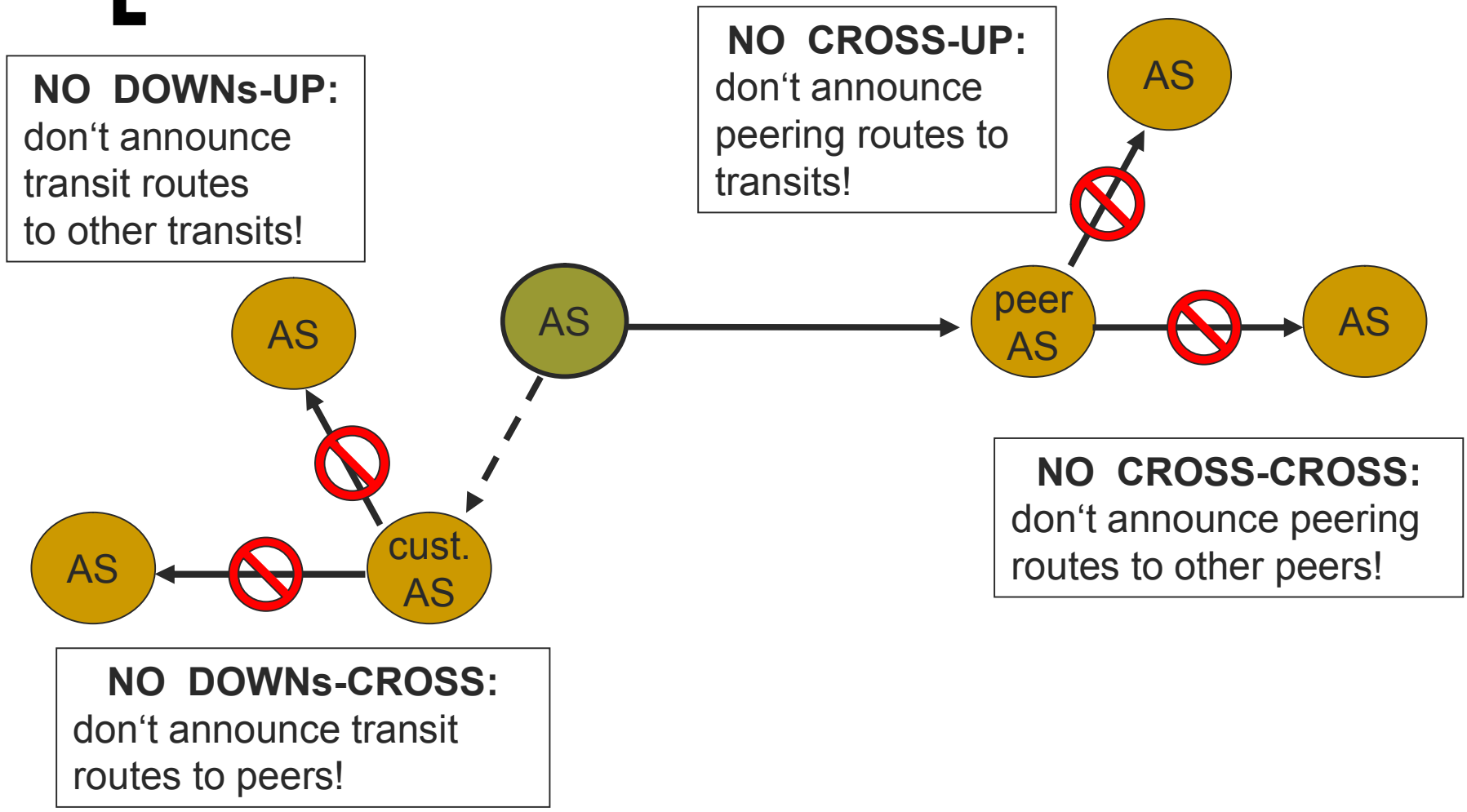
based on listed authorizations follows the **valley freeness property** of the Internet:

- when a route was passed **CROSS** or **DOWN**, it must only be passed **DOWN** further on
- better seen on a picture

valid paths: „valley free paths“



invalid paths: „valley paths“



my proposal/short term patch

- detecting leaks ⇔ detecting routes traveling non-valley-free paths
- ➔ detecting leaks therefore can be done by **recording edge-types** and **checking them for valley freeness**
- ➔ mark via communities, check via route filter mechnisms

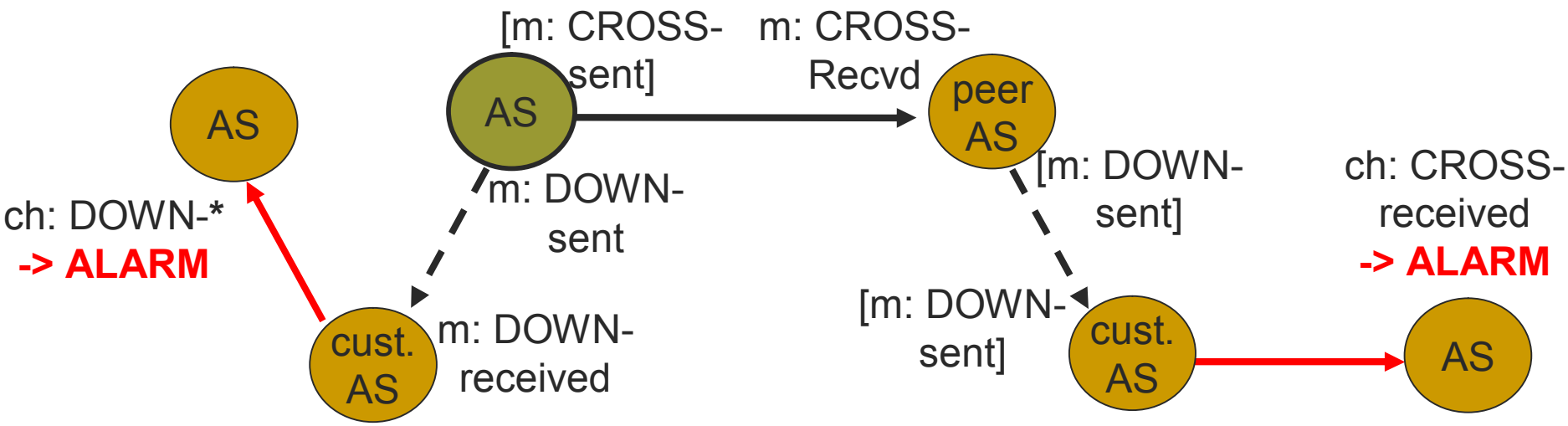
detect unauthorized paths

- before announcing a route, the sender
  - **checks** route for edge-type-marks incompatible with current link
  - **marks** route with edge-type when not and announces it
- before accepting a route, the receiver
  - **checks** route for edge-type-marks incompatible with current link
  - **marks** route with edge-type when not and announces it

marking specifics

- marking is being done by both parties
  - sender: <edge-type>-sent (e.g. **UP-sent**)
  - receiver: <edge-type>-received (e.g. **UP-received**)
- why?
  - resilience (for UP and DOWN)
  - “multiple cross link problem” would otherwise make receiver-only marking necessary (see example)
  - enables detection of inconsistent marking (=wrong configuration?)

examples



disclaimer:
- only new **m**arks are shown
- only alarming **ch**ecks are shown
- non critical marks for this example are shown in [ ]

implementation: marking – I

via extended communities: use existing **type high** definitions:

- two-octet AS specific
  - type[2 bytes]:**AS[2 bytes]**:data[4 bytes]
- four-octet AS specific
  - type[2 bytes]:**AS[4 bytes]**:data[2 bytes]

these types are used for the new communities by specifiying a **new „type low“** for both.

impl.: marking – II

data bytes of new community:

- only the last byte is used to encode the following values
  - UP-sent, UP-received
  - CROSS-sent, CROSS-received
  - DOWN-sent, DOWN-received
  - more funny relationships (e.g. sibling? partial transit? non-transit?)

impl.: checking

- via standard route filtering mechanisms (route-maps, policy-statements)
  - log alarm message
  - give bad LocPref (emergency use only)
  - discard

making it “foolproof”

- users will mess up things when doing this by hand, so
  - help via template configs
  - better: vendors provide knobs
    - to entitle a session as UP, CROSS, DOWN
    - automatically apply marking/checking
    - provide reporting/counting mechanisms

comparison

filtering based on this proposal

- versus max-prefix shutdown
  - avoids session shutdowns in many (all regularly occuring?) cases
- versus filtering via RPSL-DB
  - is “realtime capable” (no sync)
  - is easier to use/maintain (no DB)
  - scales better (class based)

We´re done!

objections please ☺

risks

- filtering of marks
  - proposal only usable before and after filtering
  - filtering usually is beeing done by small providers (on the edge) -> limited damage
- setting wrong/false marks
  - could lead to restricted route distribution (on discard action, not on setting low local pref.)
  - maybe be detected if other party applies marking & checking

What about ressources?

- problems may be
  - memory consumption
    - prefixes in AS8560 have in average 2.79 (=3) ASes in the path, so they traversed 2 edges
    - lets do the math for full table:

      130 K routes * 3 bytes attr.header * 3 edges

      *2 communities *8 bytes < 18 MB
  - CPU consumption
    - Less work is to do compared to filtering on RPSL

Some aspects of more specific prefixes routing

Volodymyr Yakovenko UMC NOC vovik@umc.com.ua

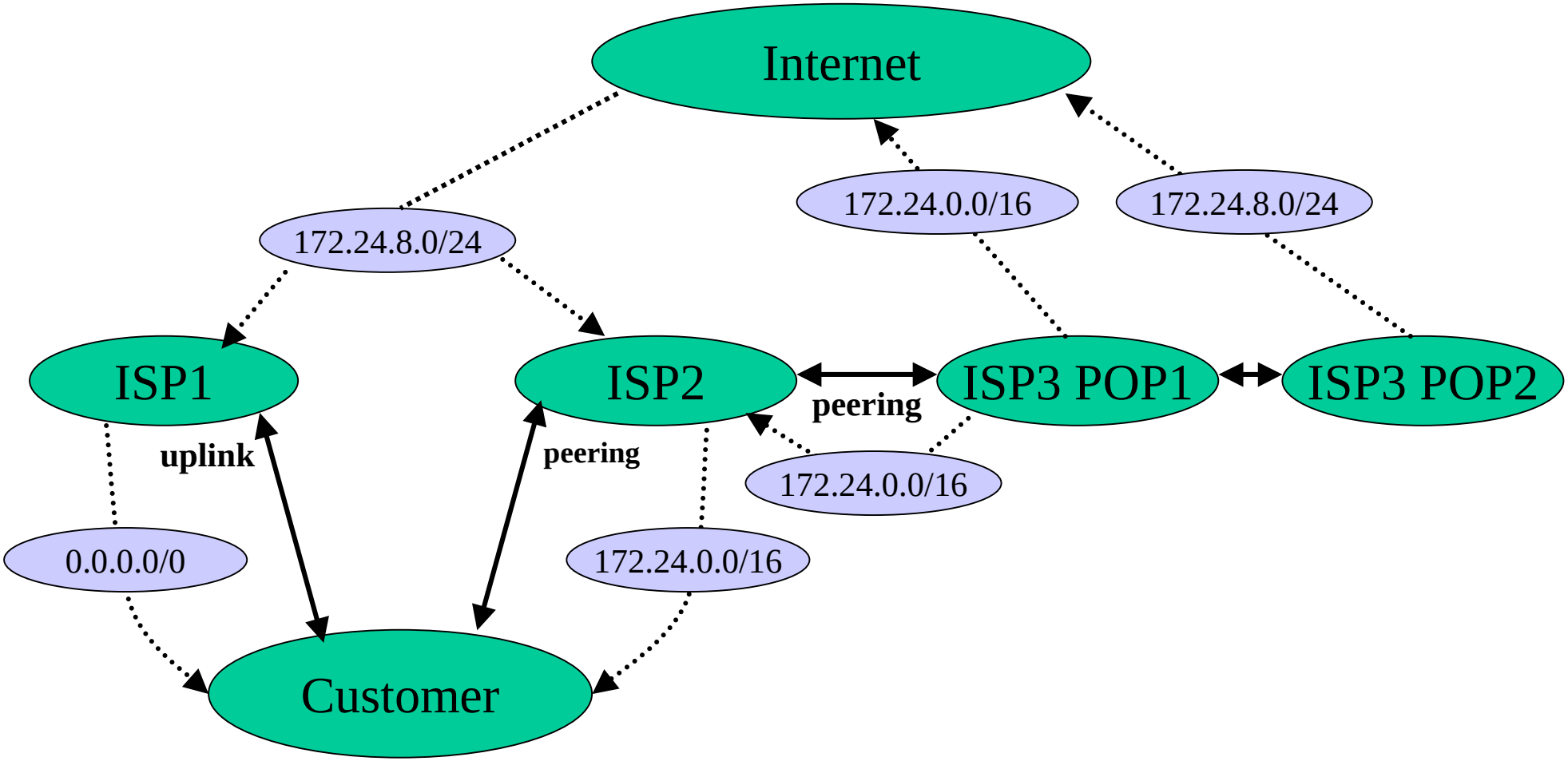
Agenda

- Reasons for more specific routes usage
- Possible drawbacks:
  - Inconsistent routing
  - Traffic Fraud
- Ways to overcome the problem

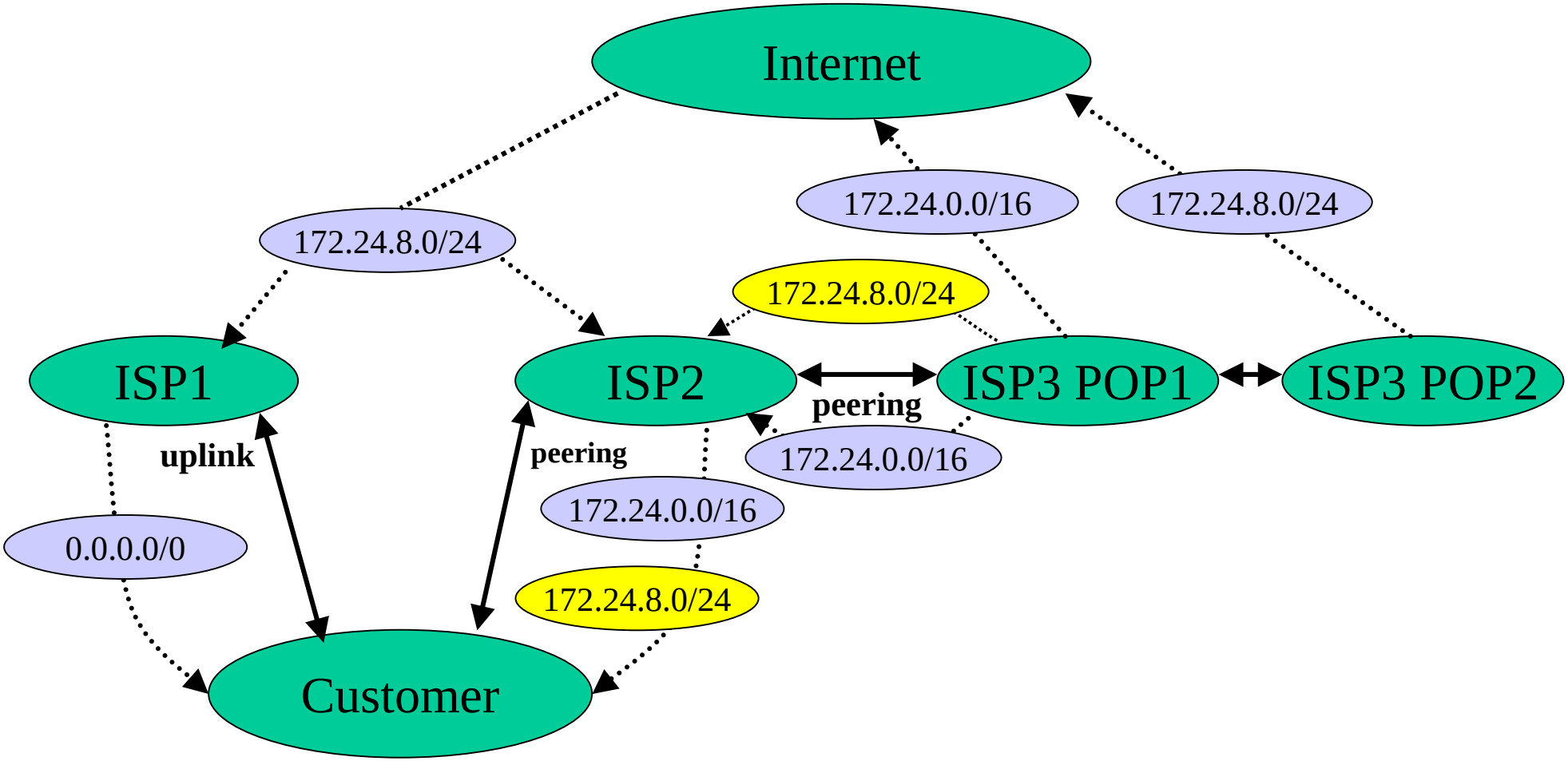
Probably the most common reasons for more specific routes usage today are:

- Inbound traffic distribution across multiply links
- Address space distribution between different sites
- PA-addresses based multi-homing
- Historical or political reasons
- Configuration errors

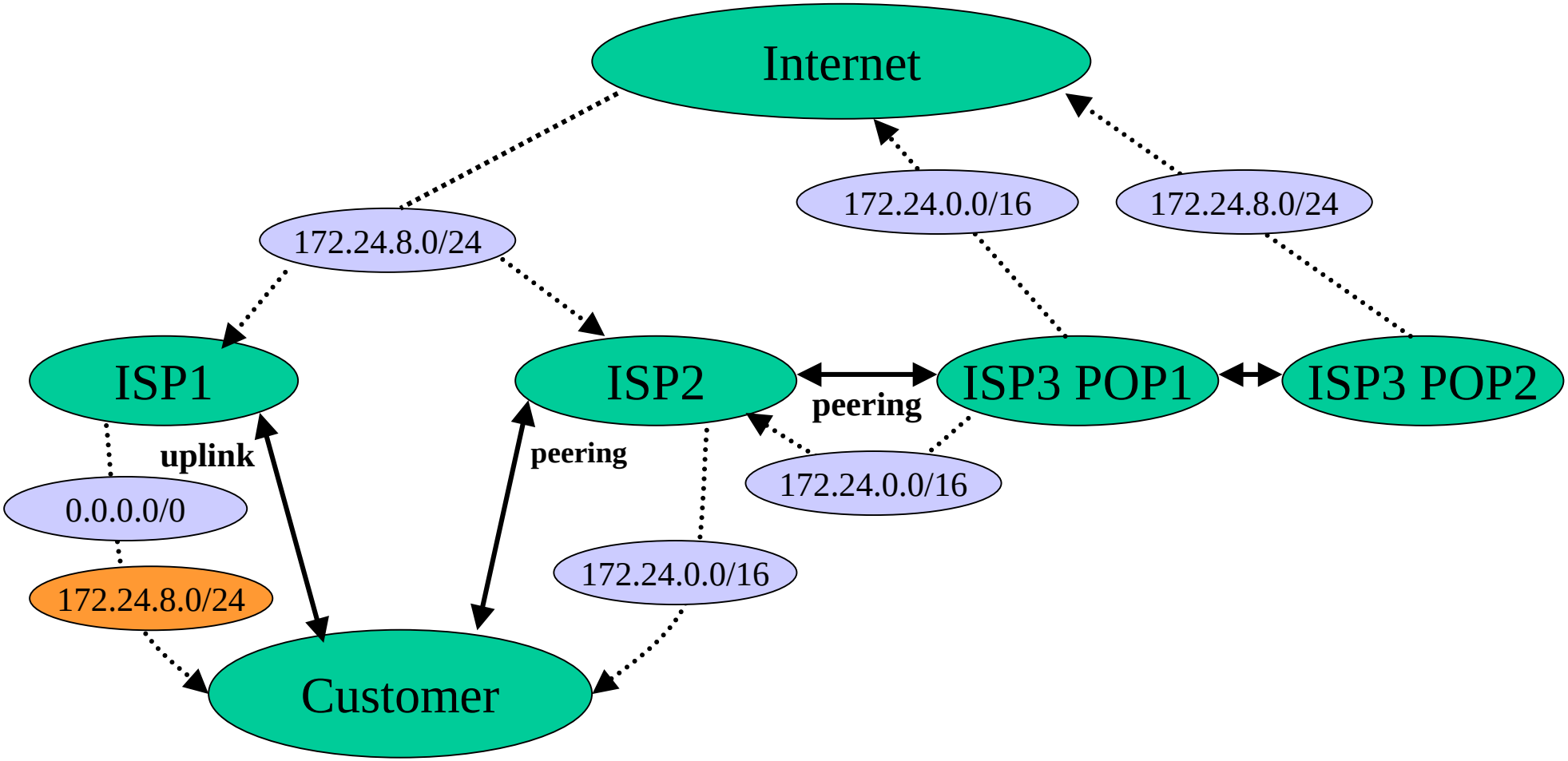
Inconsistent Routing Case



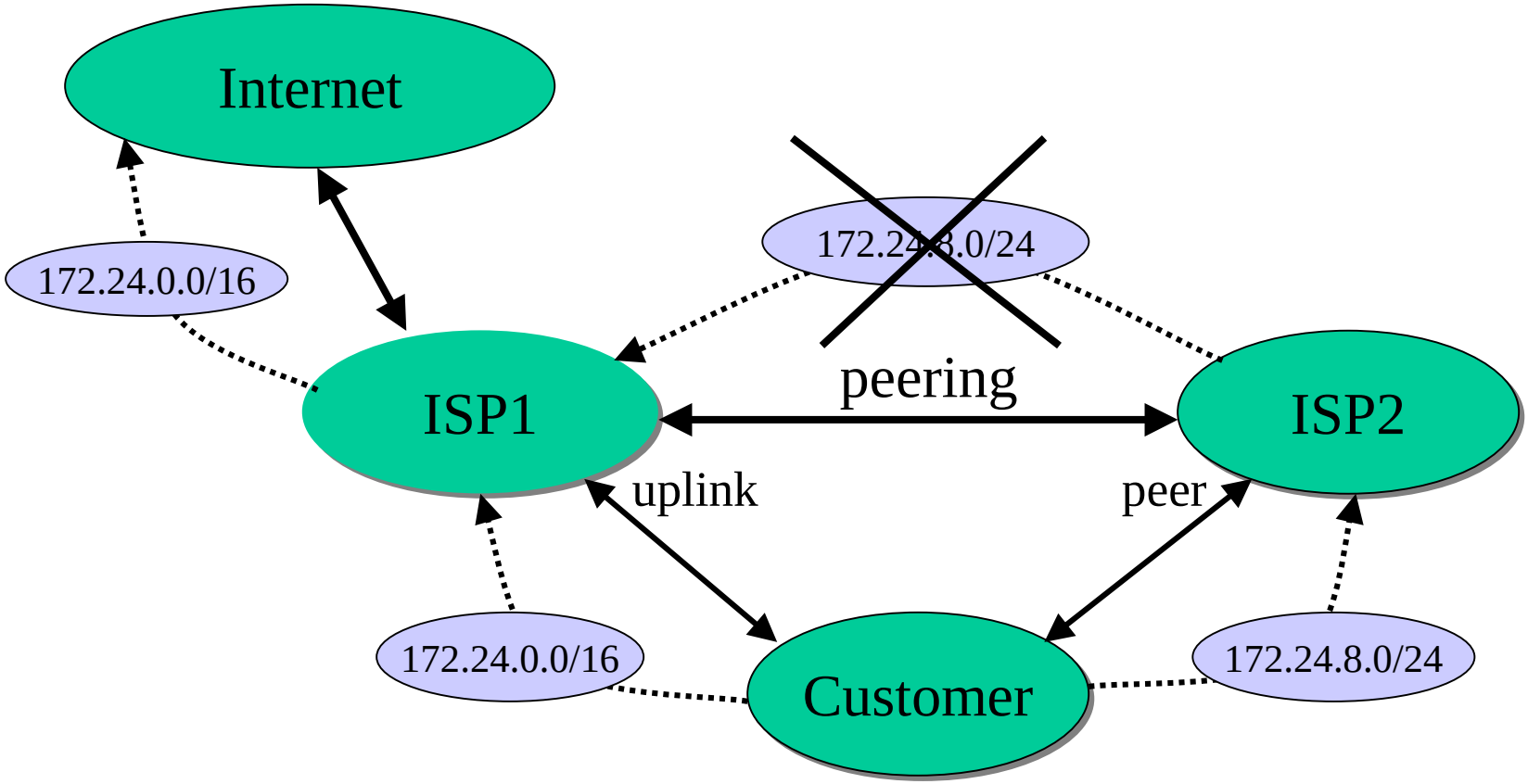
Inconsistent Routing Case: solution I



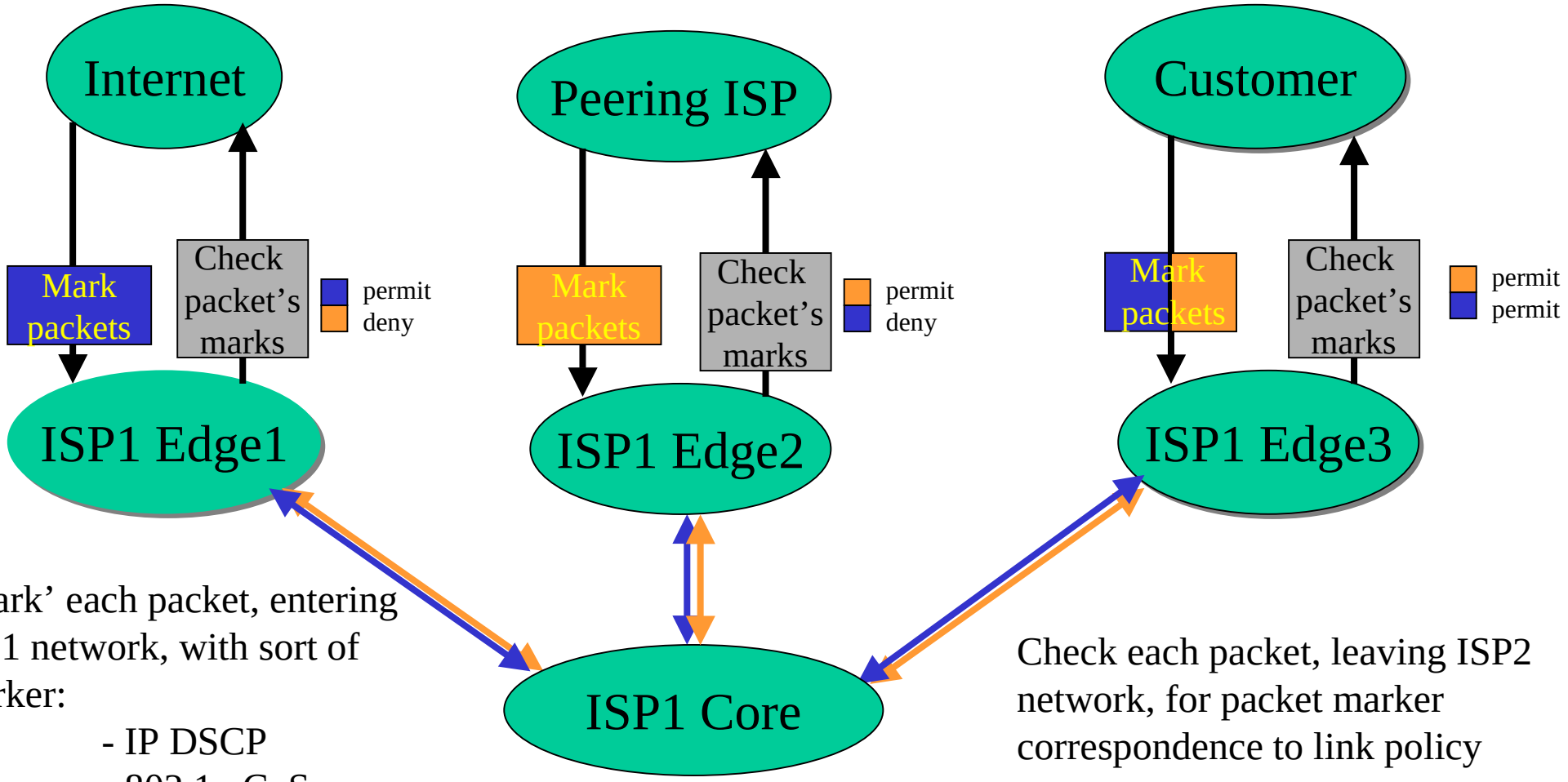
Inconsistent Routing Case: solution II



Traffic Fraud



Packets Marking



'Mark' each packet, entering ISP1 network, with sort of marker:

- IP DSCP
- 802.1q CoS

Check each packet, leaving ISP2 network, for packet marker correspondence to link policy

Ways to overcome the problem:

- Traffic fraud case
  - Do always check that Internet sourced traffic does not leak throw peering links
  - On peering links do not accept prefixes, shorter than you allowed to accept from your customer
- Inconsistent routing case
  - Always do consistent announces (less and more specific prefixes together) in all directions
  - If you receive certain prefixes over peering link always allow same and shorter prefixes throw uplink(s)

Questions?

Thank you for your time!

Embedded BGP Routing Monitoring

Th. Lévy
O. Marcé

Living with Partial Routing

IDRWS 2004

Simon Leinen, SWITCH <simon@switch.ch>

AS3303 stuff stolen from Andre Chapuis <chapuis@ip-plus.net>
See SwiNOG 7 presentation http://www.swinog.ch/

Real ISPs Have Full Routing

That’s 134113 routes right now (April 30, 2004)

Reasons not to want 134113 routes

- RIB-challenged routers
  - (but RAM is cheap)
- FIB-challenged routers
  - especially with per-line card forwarding (VIP2 etc)
- TCAM-challenged routers
  - large TCAMs are expensive and run hot
- slow routers

Approaches to reduce # of routes

filtering

- suppressing certain types of routes e.g.
  - bogons
  - too-specifics

aggregation

- using less-specific routes
  - in particular, default (0.0.0.0/0 or ::/0)

Ideally, aggregation ensures reachability in spite of filtering

Route reduction examples

SWITCH (AS559)

- No multihomed customers
  - Customers use eBGP with private AS numbers
- Two main external hubs
  - Zurich: AS1299 transit, IXEurope TIX
  - Geneva: AS3549, AS20965 (GEANT), CIXP
- Other minor non-customer eBGP locations

Swisscom IP-Plus (AS3303)

- Many multihomed customers
- Many transit providers in Europe and US
- Present on many IXPs worldwide

AS559 approach I: Partial routing

Aggregate most of the Internet under 0.0.0.0/0

For SWITCH (AS559), this is how it works:

- Default to upstreams (AS1299/AS3549)
- Accept routes from peers (martian-filtered)
- and GEANT (AS20965, unfiltered)

This leaves about 25000 IPv4 prefixes

AS559 approach II: Filtering

Policy: don’t accept more-specifics in PA space

Implementation: filter on well-known allocation boundaries per /8

- special handling of 195.0.0.0/8

AS559 Partial Routing Issues

- Maintaining prefix filters (could be automated)
- Balance/optimization between upstreams
  - closest-exit for default route (Zurich/Geneva)
  - some traffic engineering:
    - Accept AS3549 customer routes
    - This causes weird routes to be preferred (10*AS-prepend)
- Asymmetric routes
  - You'll get them anyway with multiple upstreams
- Still too many routes...
  - Those dirt-cheap GigE L3 switches handle only 16K in HW

AS3303 approach I: Filtering

Strict filtering on

- RIR allocation boundaries
- Historical classful addresses (A: /21; B: /22)
- Ad-hoc filters based on size/region

Exceptions

- Customer prefixes
- Chosen prefixes (Google, Hotmail etc. peerings)
- Domestic (Swiss) peerings

AS3303 approach II: "Semi-Defaults"

Ensure reachability to too-specifics without 0.0.0.0/0

Aggregates created to cover RIR space:

- 62/8, 80/7, 212/7, 217/8 -> EU transit ISP
- ARIN/APNIC/LACNIC space -> US transit

Class A/B

- Class B: 128/3, 160/5 and 168/6 -> US transit
- No semi-default for class A

Announced to customers only, marked 3303:9999

Semi-defaults have to be generated internally

- Transit ISPs unwilling to send them

AS3303 Results

- ~62000 routes
  - ~65000 internally with customer more-specifics
- Update noise reduced by about 40%
- Low traffic via "semi-default" routes
  - e.g. 204.0.0.0/8
    - 500 kbps for 10000 aggregated routes

Conclusions

Partial Routing can be done

- Even if you're multihomed
- Even if you have multihomed customers
- As long as you can point (partial) default(s)

Steps Toward Large-Scale Meaningful BGP Simulation

Kihong Park

Network Systems Lab
Department of Computer Sciences
Purdue University

Team: Hyojeong Kim, Bhagya Bethala, Humayun Khan, Ali Selcuk

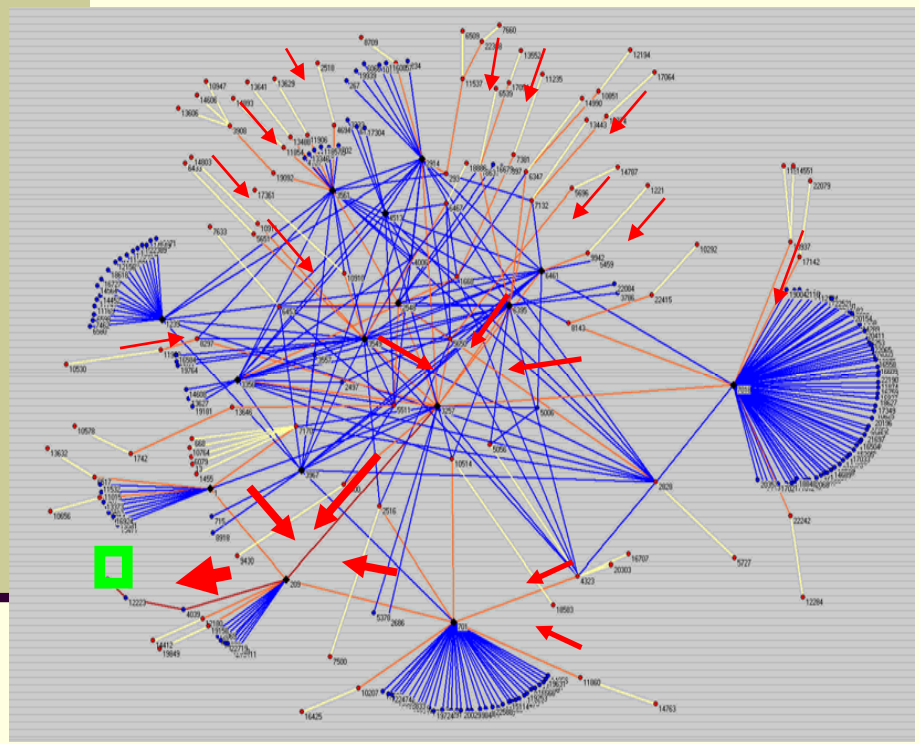
… where

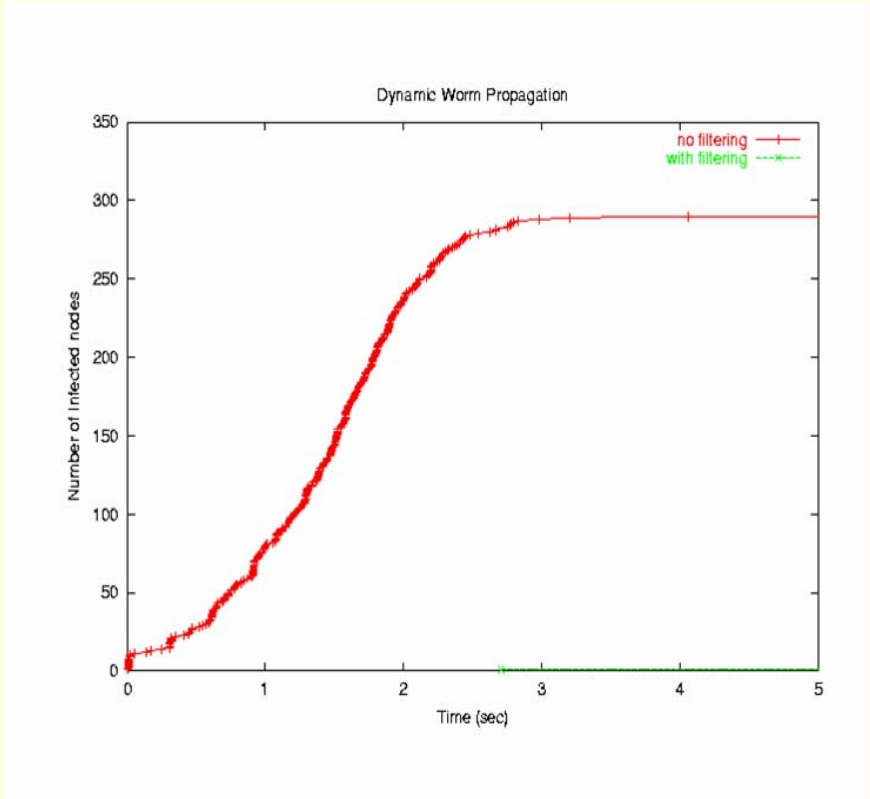
- Large-scale: thousands of ASes
- Meaningful: incorporate policy constraints

- Application:
  - DDoS attack prevention
  - Worm attack protection

→ time-varying routing subsystem: BGP simulation

Application: DDoS & Worm Attack Protection





Today

- Overview of parallel/distributed simulation environment
- Performance evaluation
- Discussion: “meaningful” BGP simulation

PDSSF Overview

- Scalable simulation environment: PDSSF
  - Substrate: DaSSF simulation kernel (C++, workstation cluster) with DML

    → collaboration with David Nicol (Dartmouth/UIUC)

  - Add on: **Tools and algorithms for**
    - **automated configuration support**
    - **performance monitoring and tuning support**

    → kind of like “TeX vs. LaTeX”

PDSSF Overview

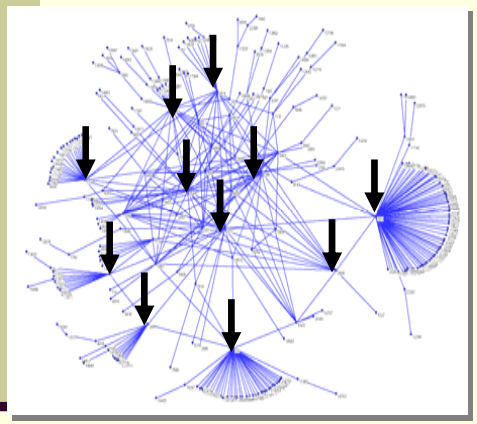
- Scalable simulation environment: PDSSF
  - Key features:
    - Meta-DML
    - Measurement subsystem
    - Partitioning subsystem
  - Meta-DML components:
    - Network topology
    - Protocol stack
      - → traffic generator suite, attacker apps, BGP, DPF, etc.
    - Measurement configuration

PDSSF Overview

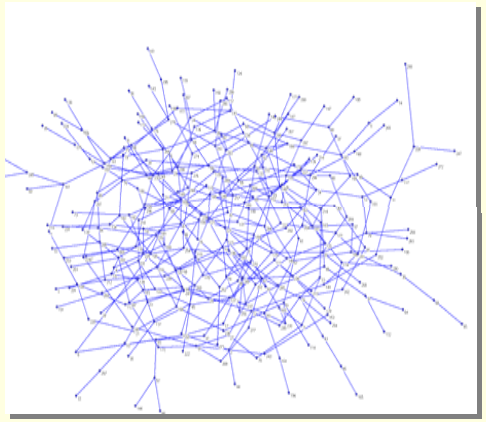
- Scalable simulation environment: PDSSF
  - Meta-DML components (cont.):
    - Power-law topology partitioning
    - Fault model
  - Accurate queueing model
  - Trace-driven visualization

PDSSF Overview

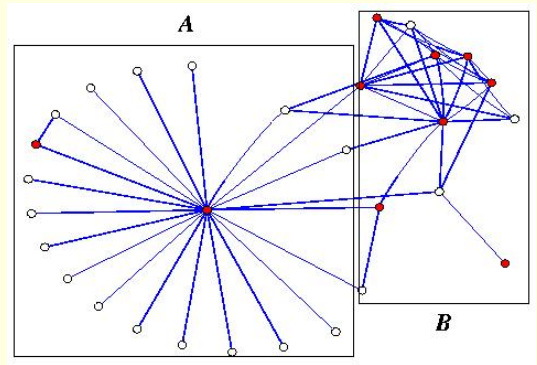
- Exploit power-law connectivity to effect joint load-communication partitioning



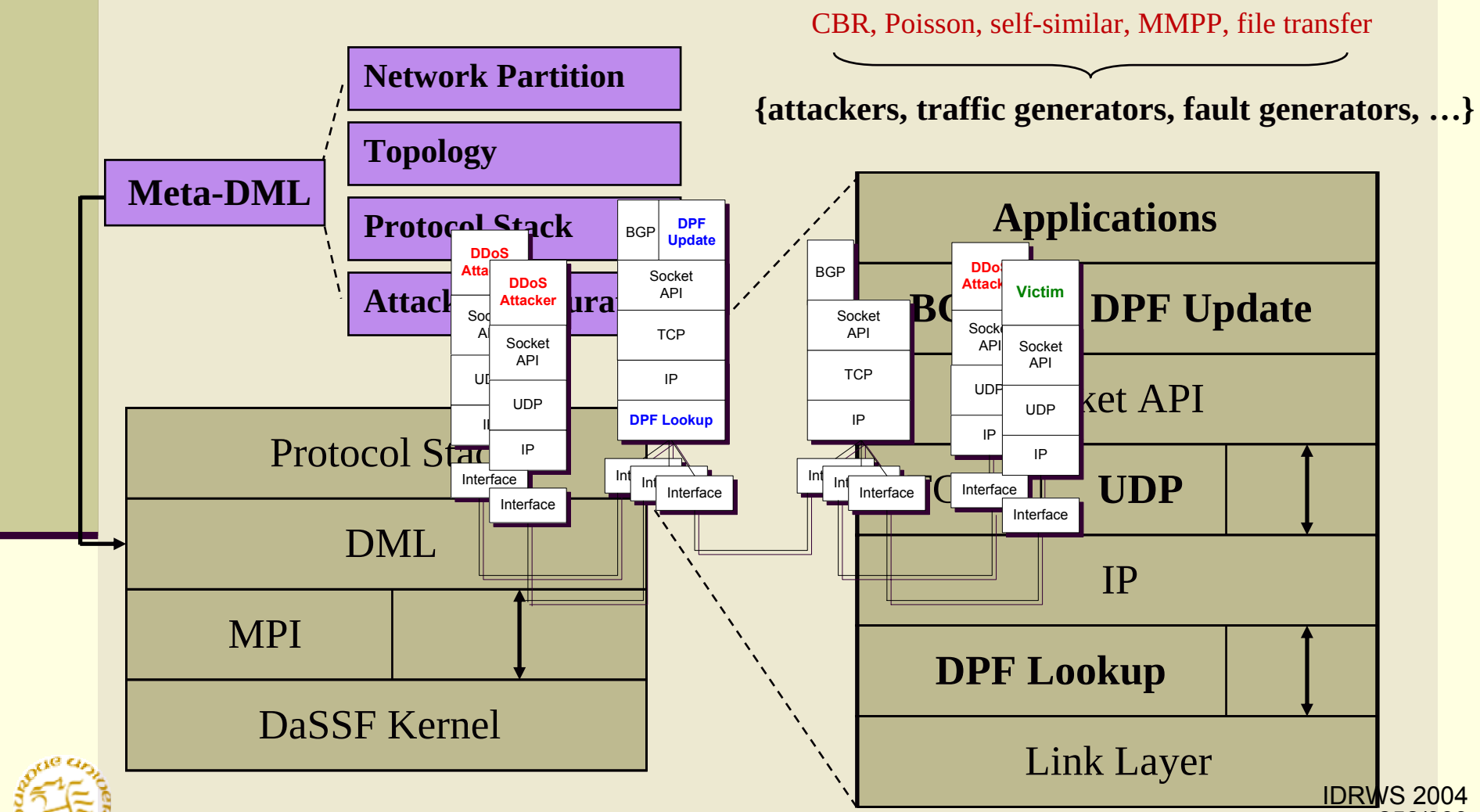
power-law topology



random topology



PDSSF: Architecture

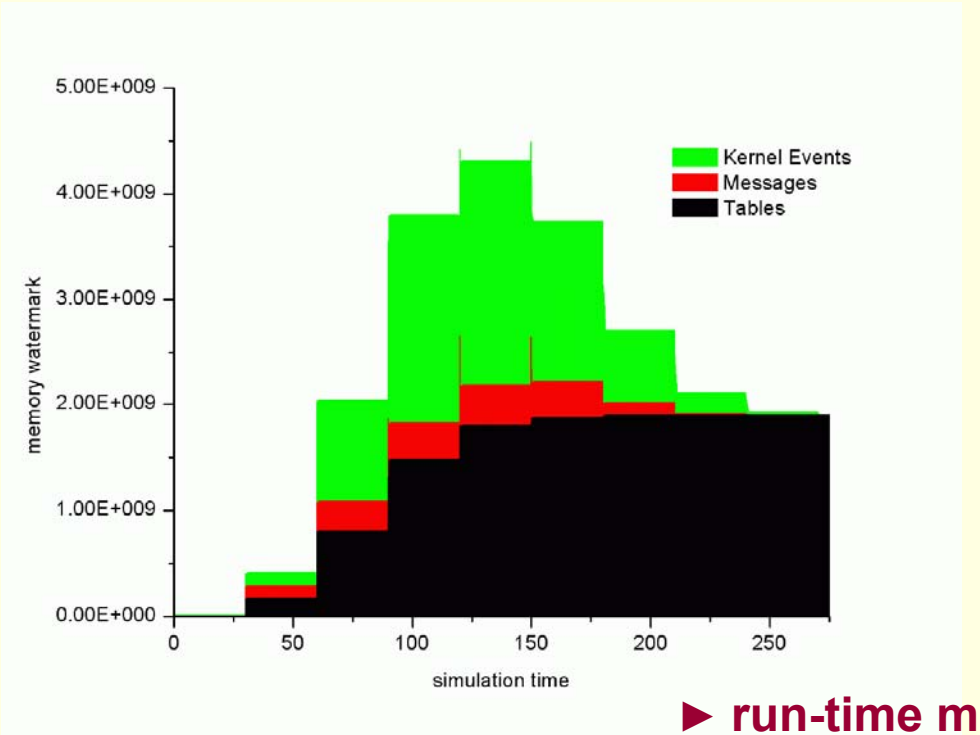


Performance Evaluation

- Scalable simulation environment: PDSSF
  - Measurement subsystem:
    - User level (i.e., protocol stack) vs. kernel events
    - Event counting vs. memory consumption
    - User configurable and extensible
    - Sampling and measurement integration support
      → distributed workstation cluster platform
  - Power-law topology partitioning
    → uniform vs. nonuniform

Performance Evaluation

- Scalable simulation environment: PDSSF
  - Measurement subsystem benchmark
    - → BGP on 3,015 node Internet AS topology

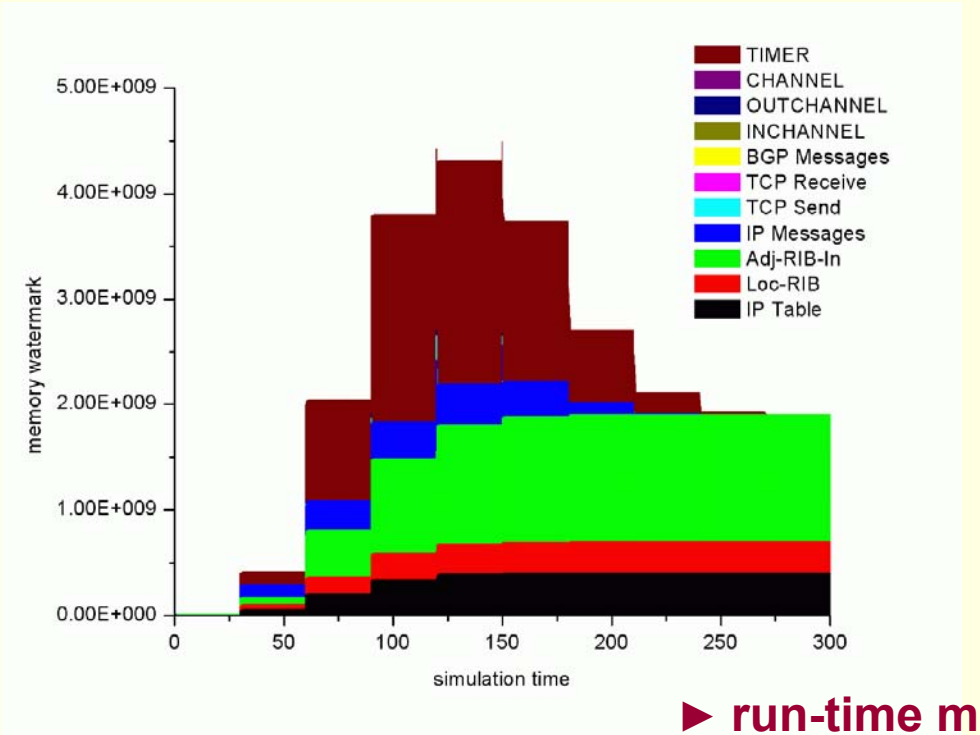


coarse granular

► **run-time memory consumption**

Performance Evaluation

- Scalable simulation environment: PDSSF
  - Measurement subsystem benchmark
    - → BGP on 3,015 node Internet AS topology

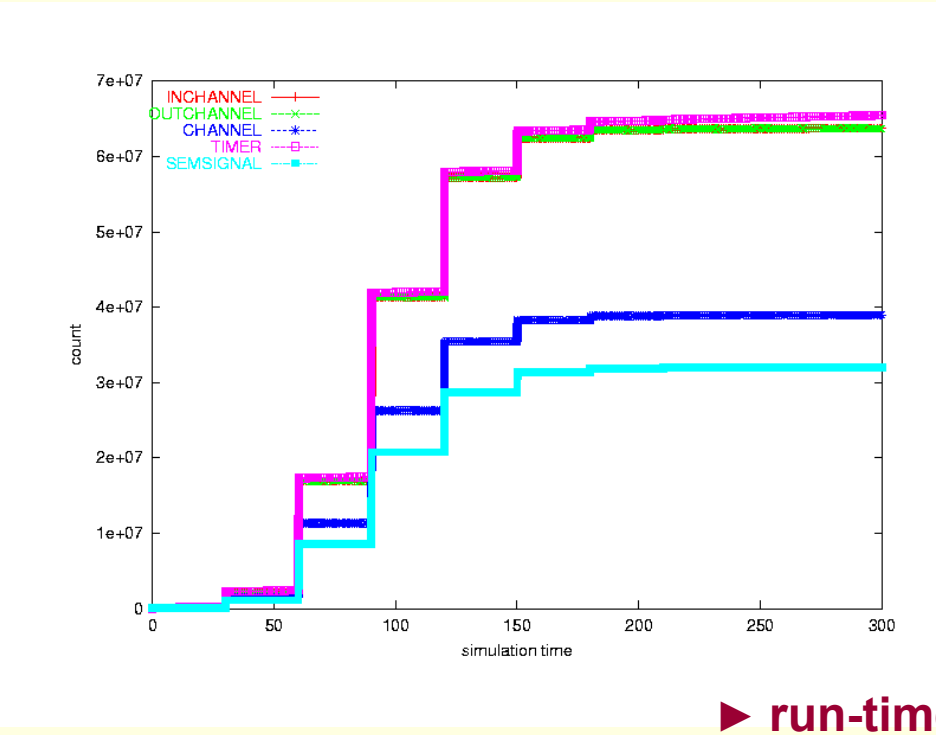


fine granular

► run-time memory consumption

Performance Evaluation

- Scalable simulation environment: PDSSF
  - Measurement subsystem benchmark
    - → BGP on 3,015 node Internet AS topology

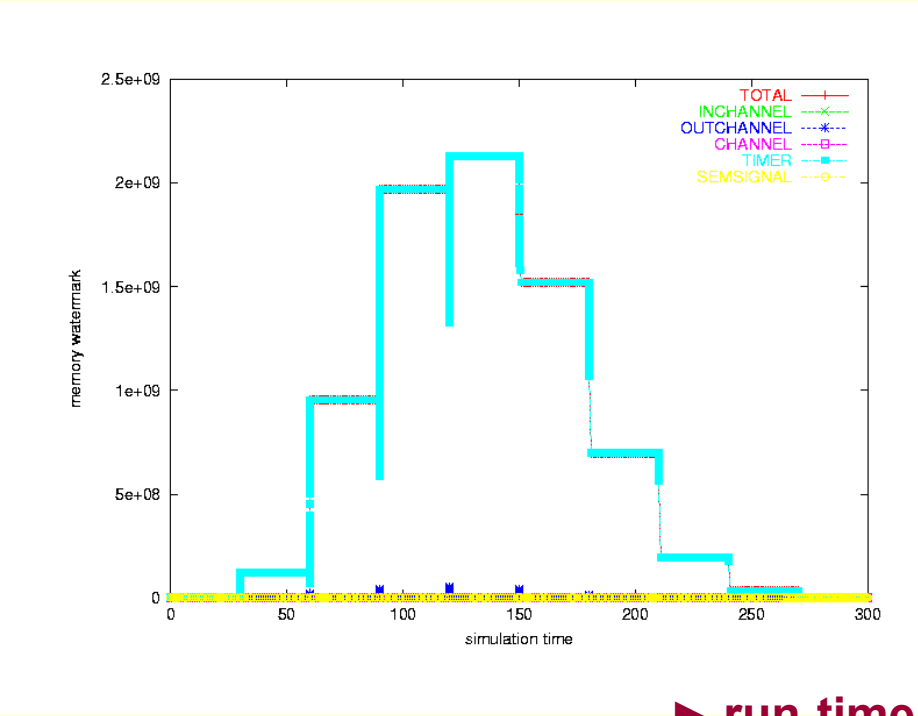


event count

► **run-time event monitoring**

Performance Evaluation

- Scalable simulation environment: PDSSF
  - Measurement subsystem benchmark
    - → BGP on 3,015 node Internet AS topology



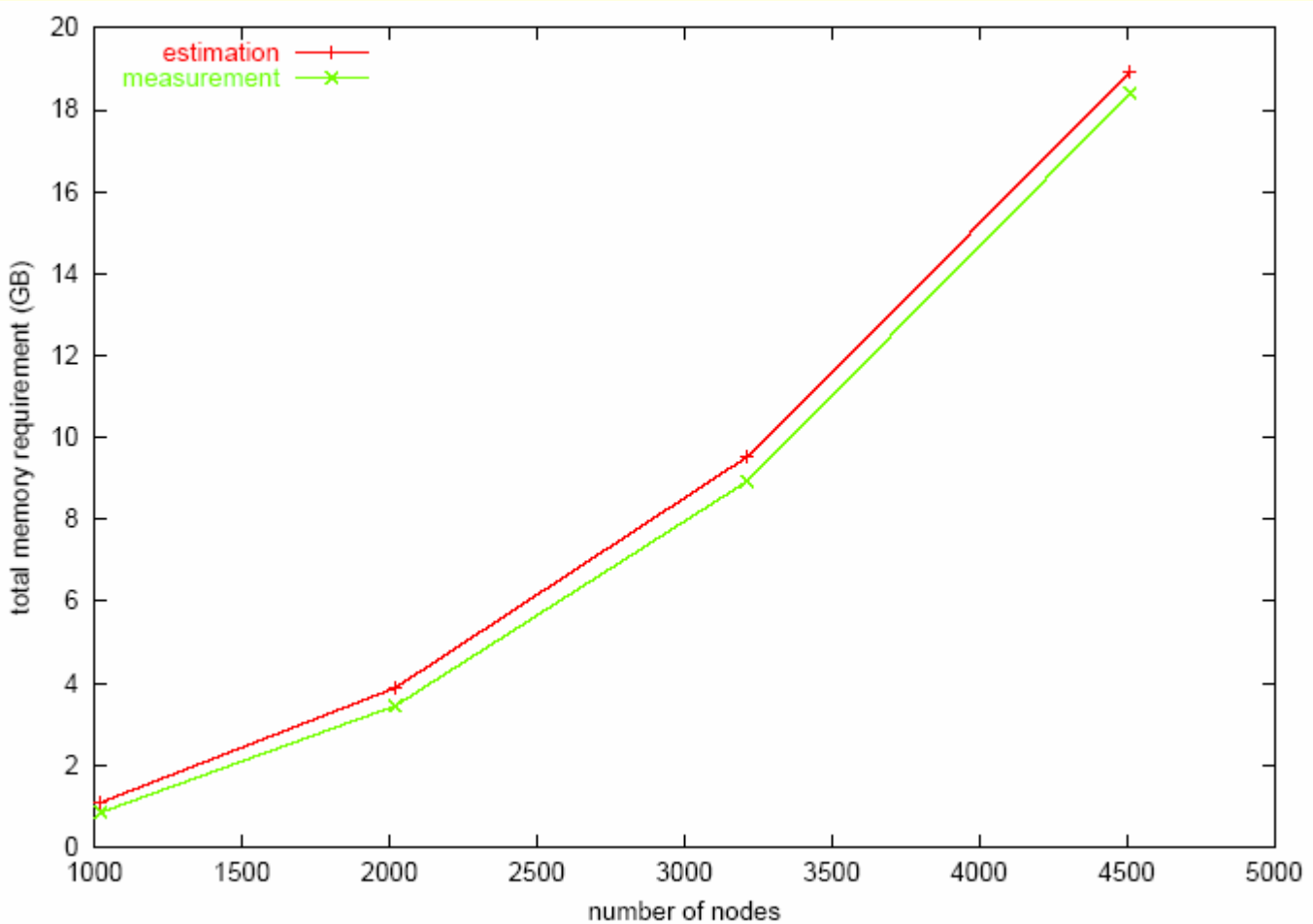
memory occupancy

► **run-time memory monitoring**

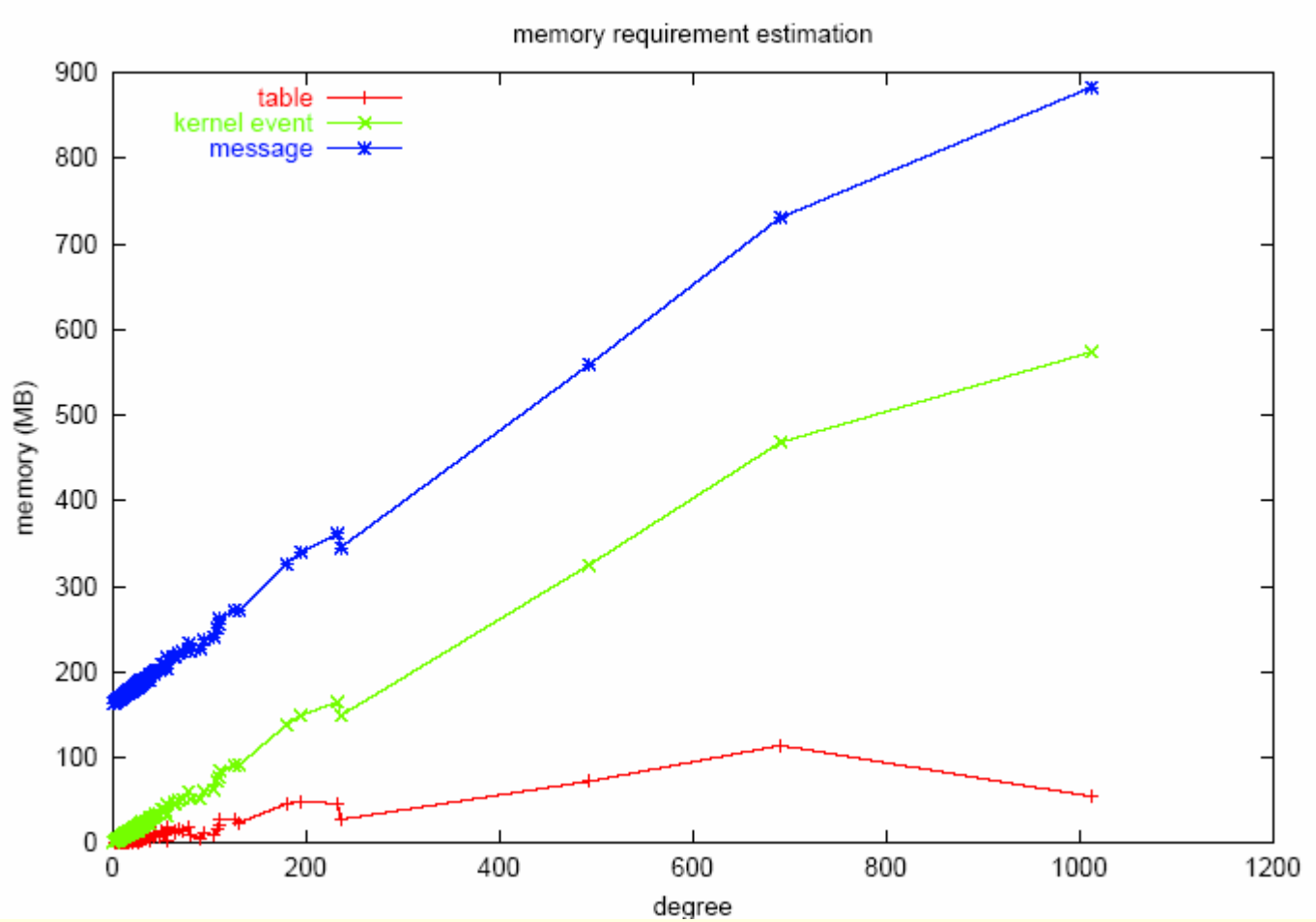
Benchmarking

- Speed-up and memory benchmarks
  - 40+ x86 PCs, 2 GHz, 4GB, 2GB, & 1 GB memory, Linux 2.4+
  - DaSSFNet, MPI

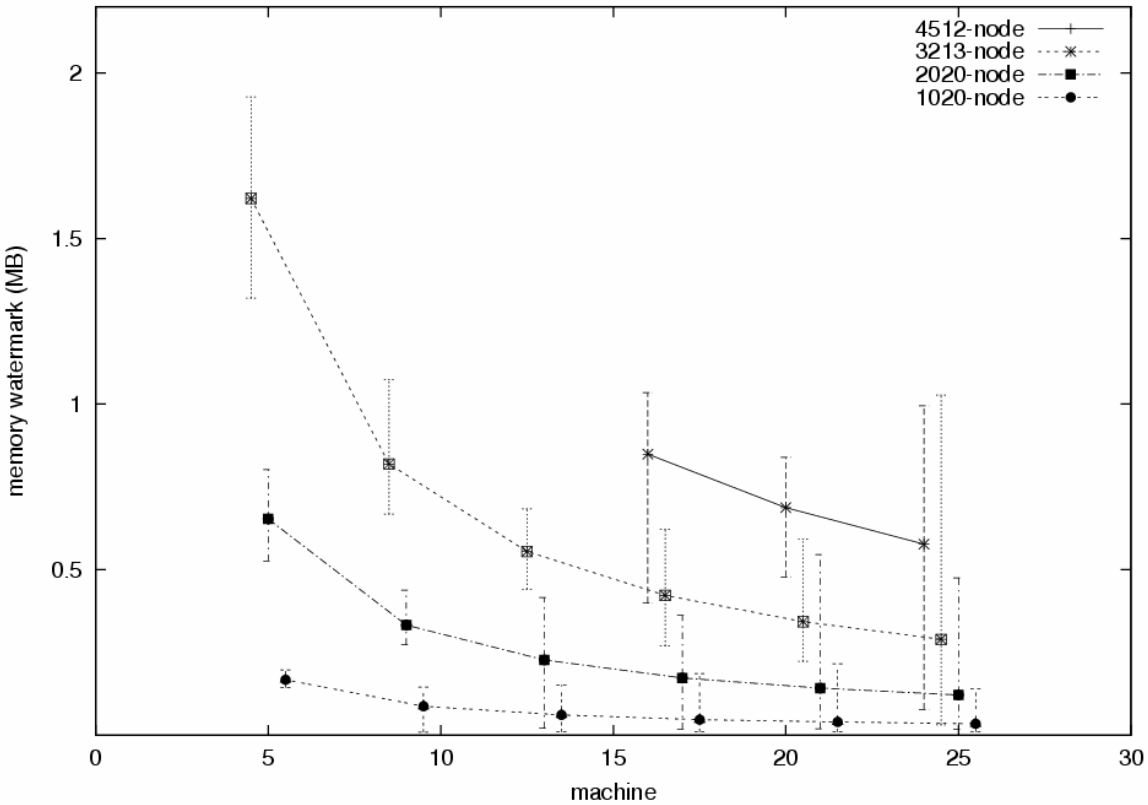
Memory Management



Memory Management

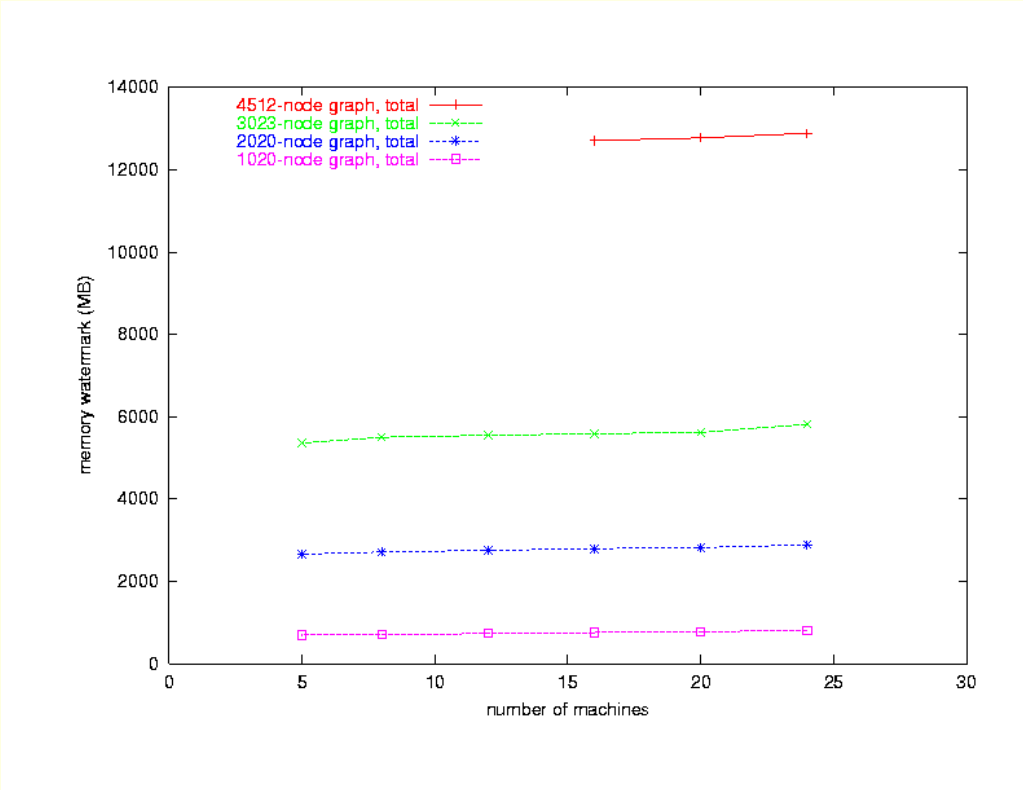


Speed-Up

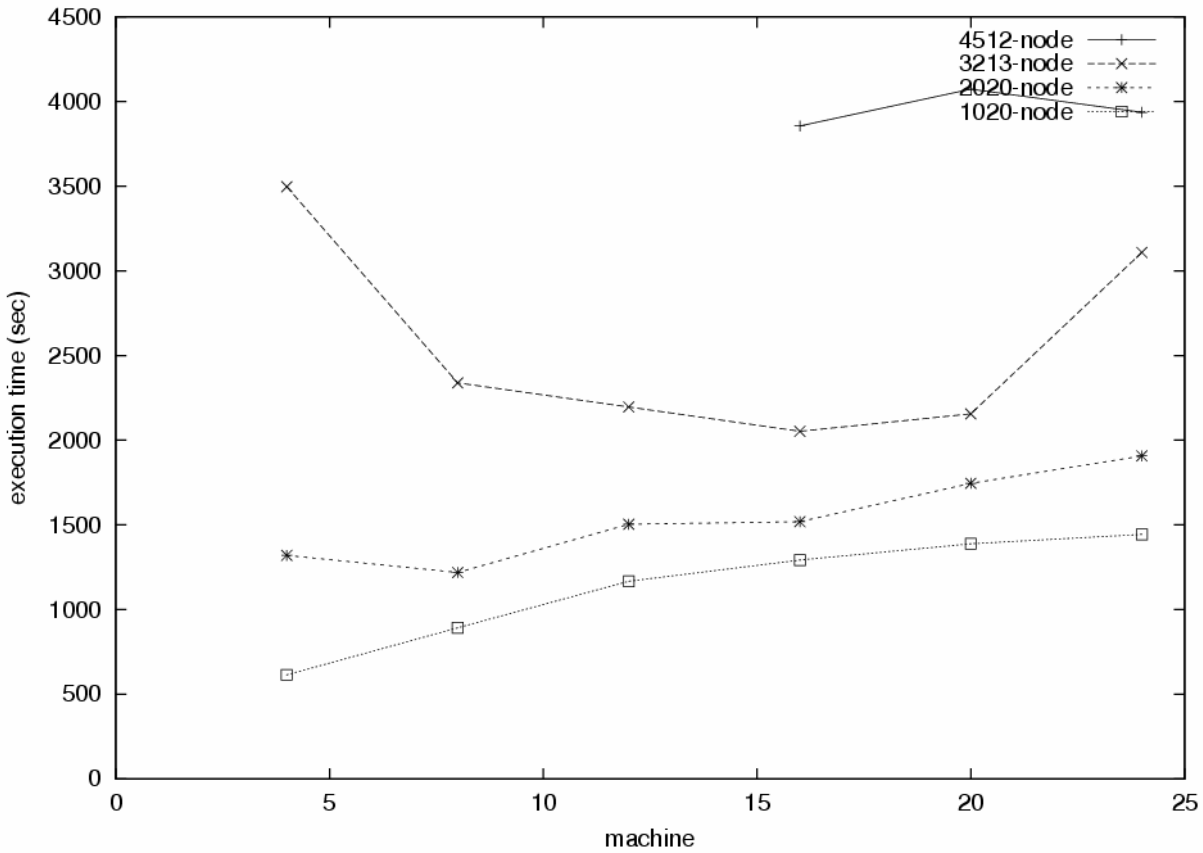

4512-node
3213-node
2020-node
1020-node
memory watermark (MB)
machine
0
5
10
15
20
25
30
0.5
1
1.5
2

Memory Benchmark

- 1020, 2020, 3023, and 4512 Internet AS graphs
- 24 machines

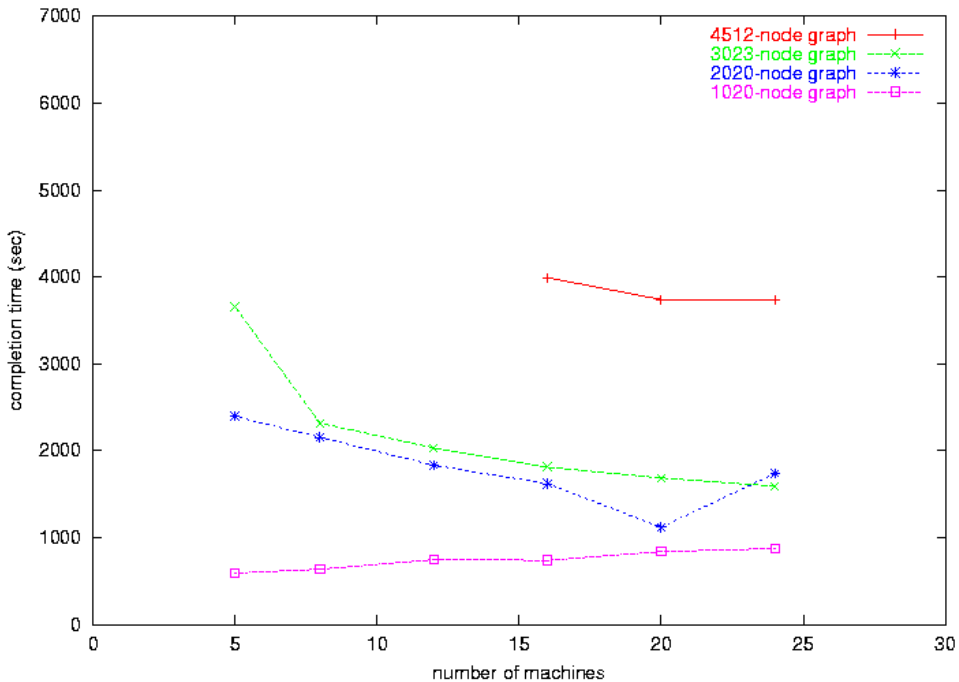


Speed-Up


4512-node
3213-node
2020-node
1020-node
execution time (sec)
machine
0
500
1000
1500
2000
2500
3000
3500
4000
4500
0
5
10
15
20
25

Speed-Up Benchmark

- 1020, 2020, 3023, and 4512 Internet AS graphs
- 24 machines



Infrastructure Attack Protection

- Resilience in the presence of infra attacks:
  - Increased DDoS attack targeted at network infrastructure

    → **e.g., router and name servers**

  - Non-Byzantine failures

    → **e.g., hardware and software faults**

- Key issue:
  - Route-based DPF also protects infrastructure
  - Is there a positive feedback loop?

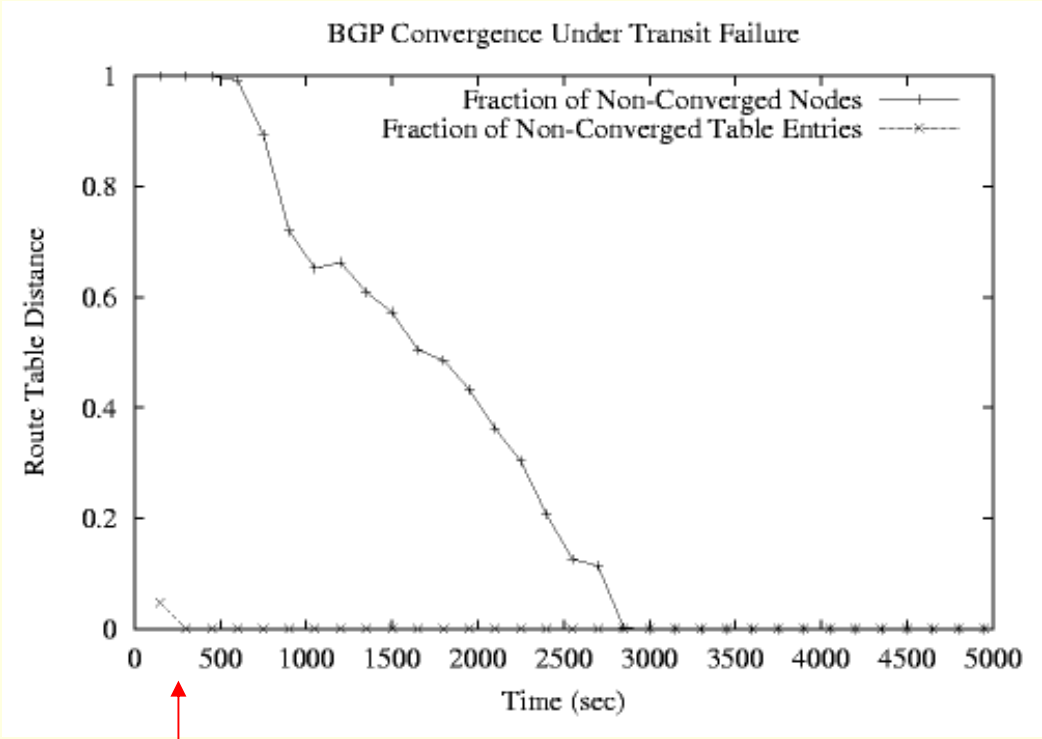
    → **weakened filter net leads to escalation**

Stub and Transit AS Failure

- Catastrophic event
  - → **protective performance under worst-case scenario**
- Key performance metric
  - Safety violation
    - → **discard valid/unspoofed packets due to error in filter table**
  - Staleness
    - → **pass spoofed packet due to inefficiency in filter table**

BGP Convergence

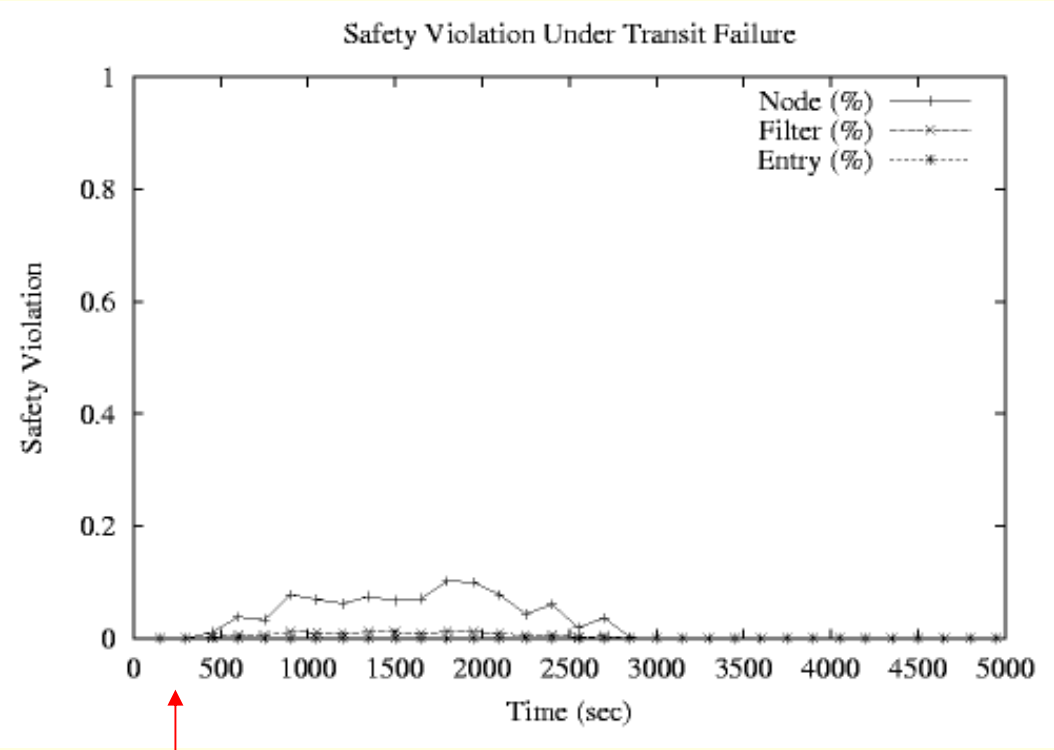
- BGP dynamics under transit AS failure



► **3,015-node Internet AS topology**

Safety Violation

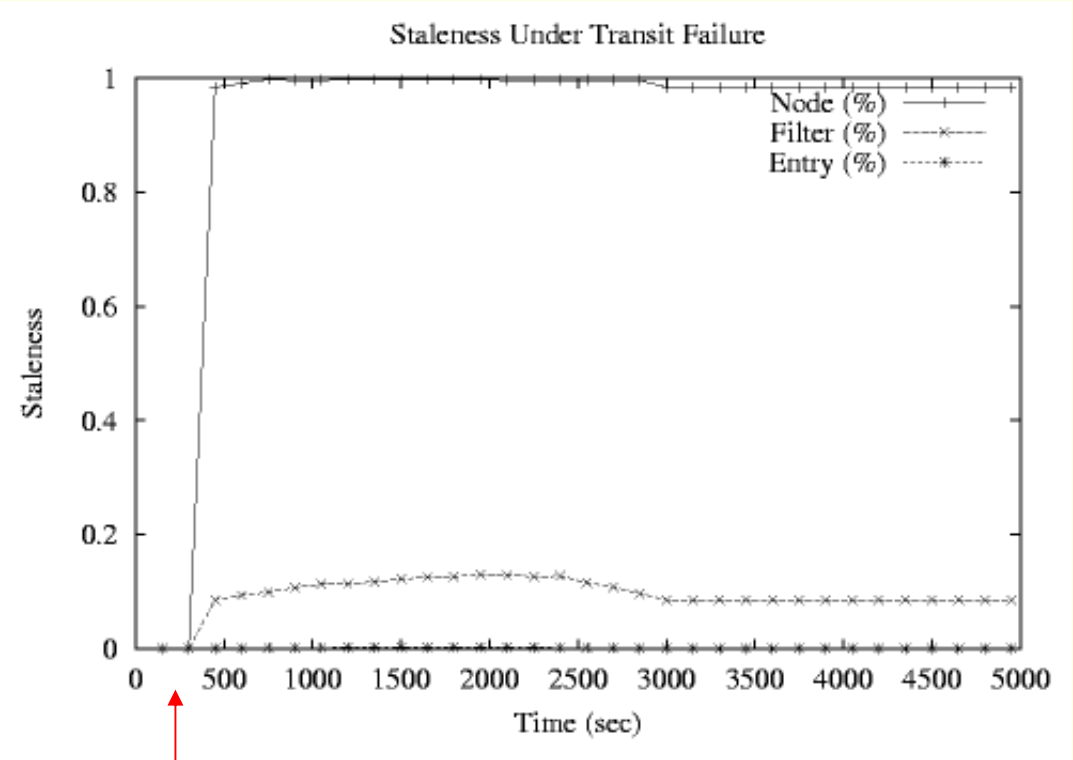
- 3 granularities: entry, filter, node



► 3,015-node Internet AS topology

Staleness

- 3 granularities: entry, filter, node



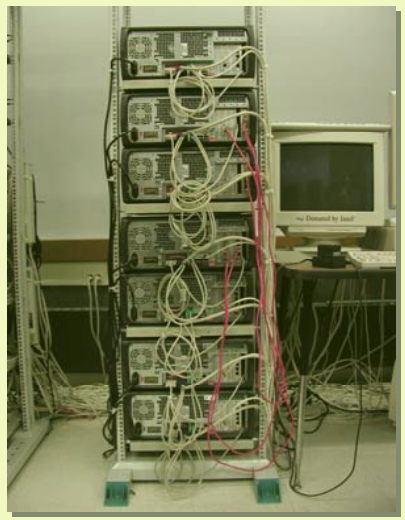
► 3,015-node Internet AS topology

Network Processor Prototyping

- 7-node Intel IXP1200 NP testbed
- Teja development environment

Network Processor

Intel IXP1200



Introduction & Motivations

- Off-line BGP routing monitoring initiatives (i.e based on router logs) already exist:
  - Periodic report : The CIDR Report
- Objective of our work: Study feasibility and accuracy of on-line (or embedded) routing monitoring
- Targeted benefits:
  - Provide valuable & up-to-date results to the local operator?
  - Do the results enable reactions (like route aggregation or filtering)?

Plan

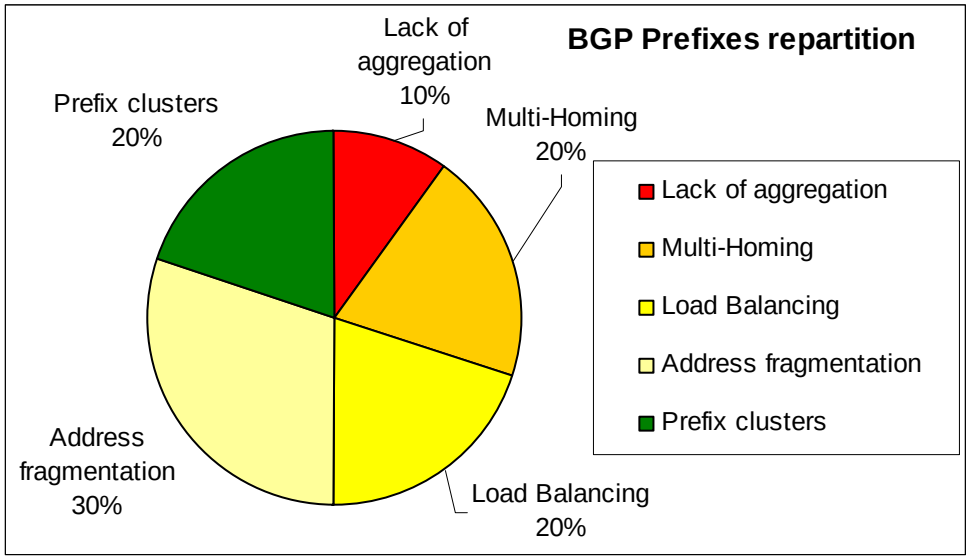
- Scope of the monitoring
- Architecture overview
- Experimental Results
- Possible Reactions
- Conclusion

Scope : Inside the Routing Table (1)

- From a router point of view, BGP prefixes can be classified into following categories:
  1. Lack of aggregation : Prefix could have been aggregated by origin into less specific CIDR prefix.
  2. *Site* Multi-homing : Customer's prefix connected (and announced) through several providers
  3. Load-balancing : Customer shares incoming traffic between several providers.
  4. Address fragmentation : Prefix with same routing characteristics than others but not aggregatable.
  5. Prefix cluster expresses independent routing characteristics.

Scope : Inside the Routing Table (2)

- According to [Bu], average repartition in the Internet core:



- Lack of aggregation , *Site* Multi-homing and Load-balancing correspond to operator's practices.

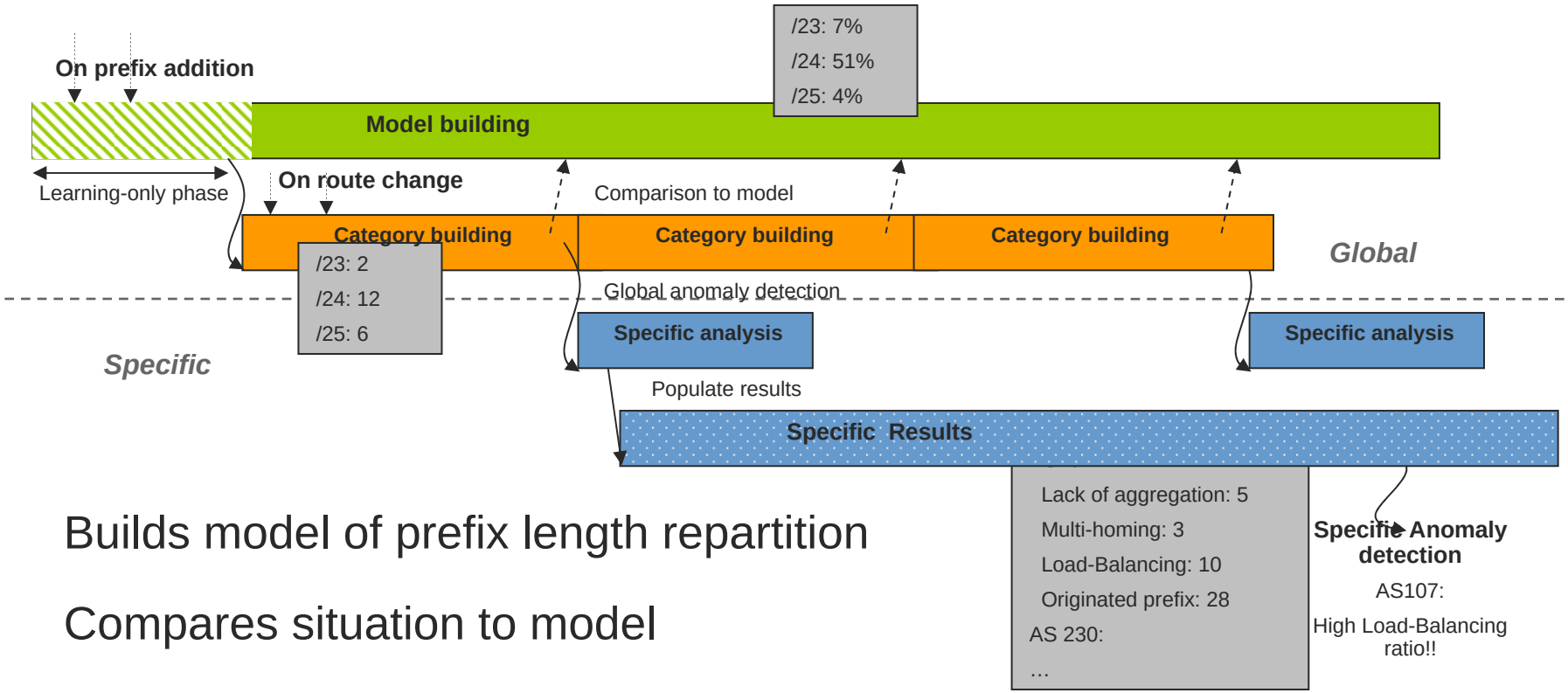
Our approach: embedded routing monitoring

- To provide the operator and/or the manager
  - A view of operator's practices corresponding to local RIB entries.
- Requirements
  - To be able to get diagnostic as soon as the situation appears
  - To be low resources consuming but accurate enough
- Choice
  - Embedded monitoring in the router
  - Use a 2 steps architecture

2-Steps Architecture of the monitoring

- Global monitoring
  - Builds a model of prefixes repartition in the RIB
    - Up to now: Heuristic on prefix length repartition
  - Collects category (sample) and triggers next phase of analysis if shift from the model
    - provides targeted snapshot for analysis
  - Characteristic :
    - Low resource consumption
- Specific analysis
  - Started by global monitoring when potential troubles detected
  - Applies several methods to identify operator's practices
  - Characteristic :
    - Resources consuming, but applied on small subsets

Architecture sequencing *(Informational)*

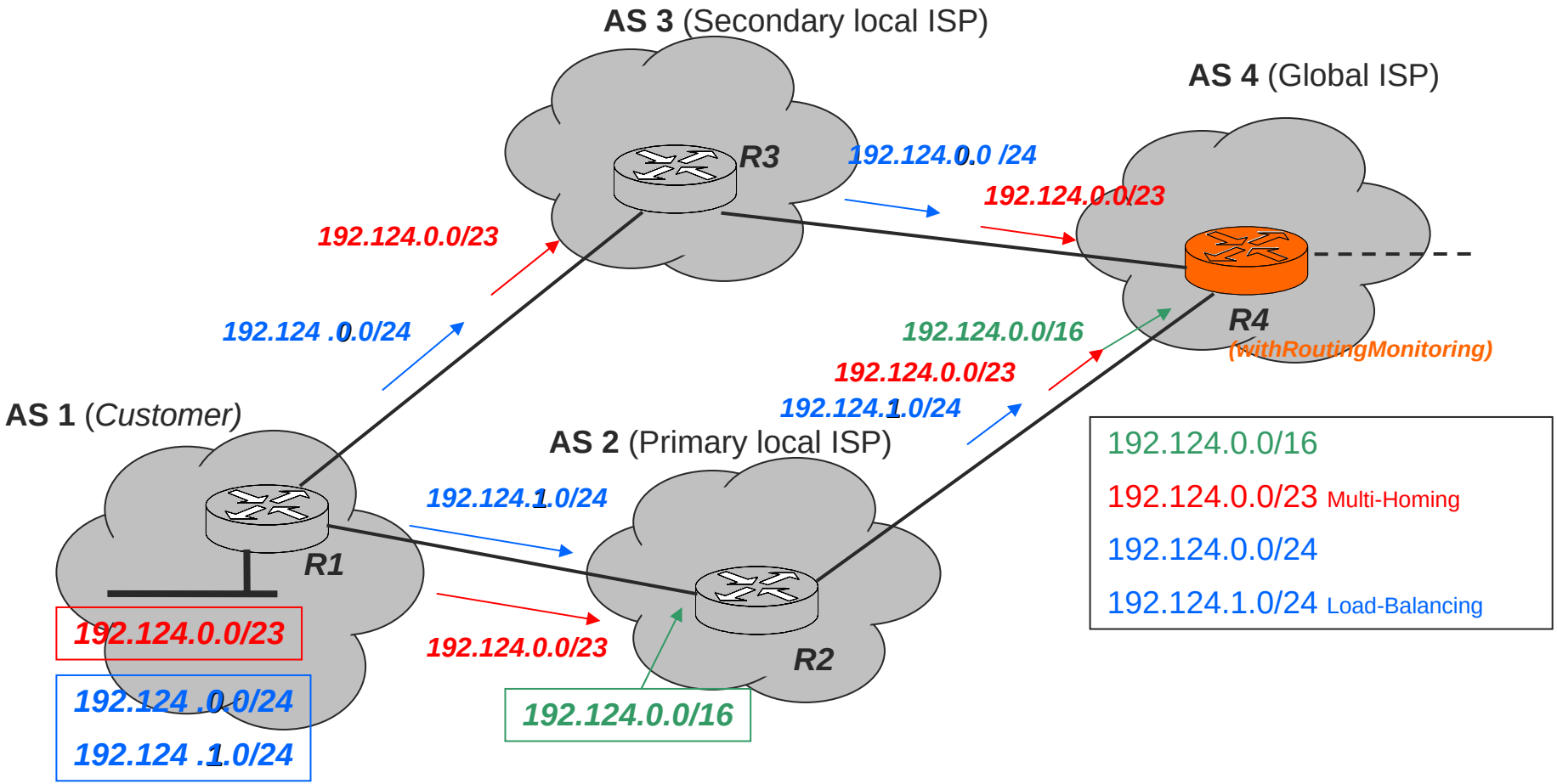


Builds model of prefix length repartition

Compares situation to model

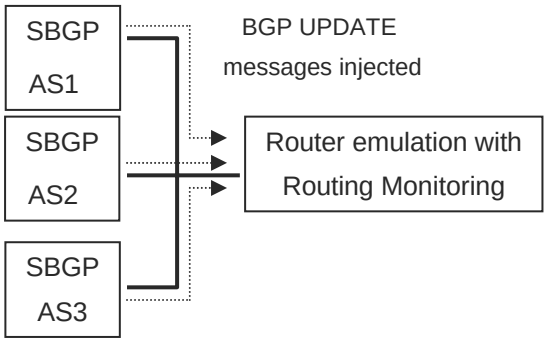
If shift, triggers specific analysis

Specific analysis example

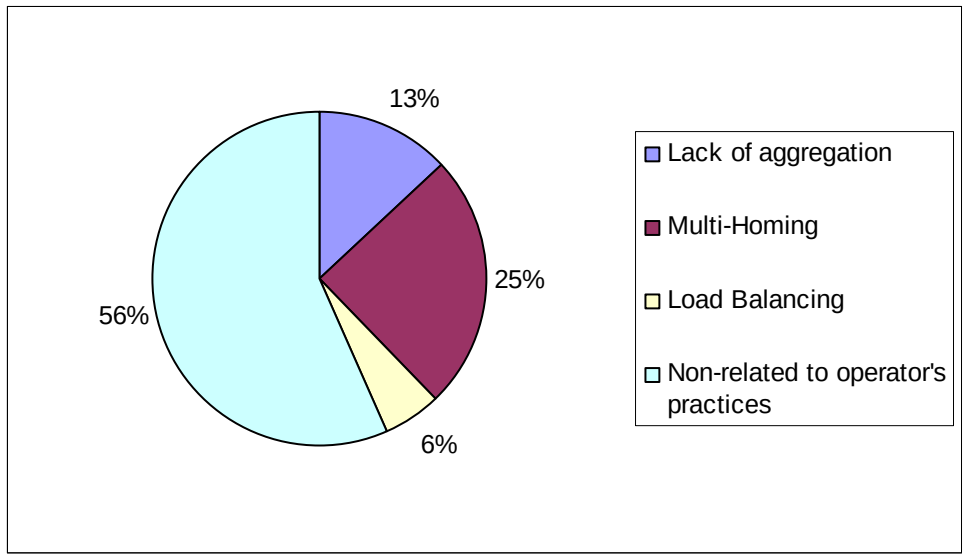


Experimental System & Results

- Based on core IP router snapshots
  - Available on Routing Information Service [RIS]
  - RIB dump transformed into UPDATE messages
- Reinject routes thanks to several BGP speakers (SBGP)



- Early results :
  - Based on RIB
    from RIPE NCC
    (for 204.0.0.0/8 sub-prefixes)



Experiments analysis

- General constatations on detections:
  - Completeness of detection depends on peering relationships of the router with routing monitoring.
  - Prefix repartition comparable to average results from [Bu] (except for Load-Balancing)
- Accuracy of practice detection:
  - Difficult to validate without operator's confirmation
  - But comparison with CIDR report
  - Some ambiguities between Multi-Homing and Lack of Aggregation => Needs for some refinement in methodology
- Low impact on BGP behavior
  - ≅1% of the BGP processing time
  - Two steps architecture is proven to be valid

Possible reactions: the multi-homing case

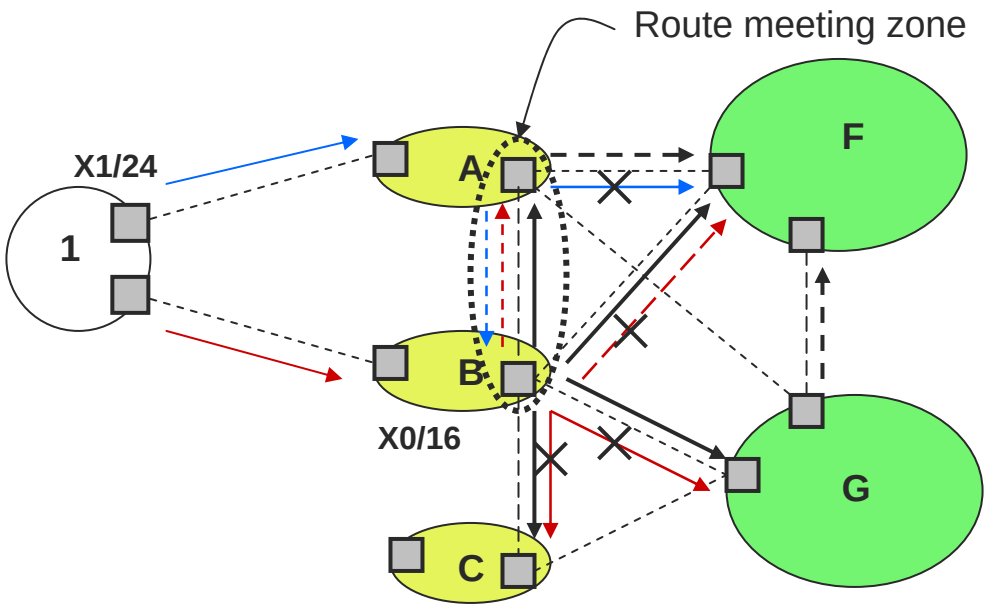
- Causes for multi-homed prefix P1 on BGP routing :

Order of importance

  - Reachability
    - P1 is reachable from anywhere in the Internet.
  - Redundancy
    - From any location, after the failure of one provider, a route (eventually recomputed) is available.
  - Incoming traffic sharing
    - Depending on its origin, the traffic goes trough a particular provider.

- Possible reaction:
  - Restrict the propagation of P1 with no impact on reachability and redundancy but possible impact of traffic sharing

Possible reactions: network example



⇒Globally reduces the number of route entries.
But requires collaborative filtering.

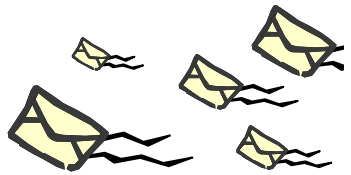
Conclusion

- Routing can be monitored at node level
  - Provides a partial but locally accurate view
- Operator's practices detection can drive prefixes filtering
- Next steps are:
  - Improve accuracy to allow semi-automatic reaction
    - Correlate several monitoring results to improve global view
    - Refine detection algorithms
  - To study impact of prefixes filtering on global BGP dynamics.
  - To study progressive deployment of routing monitoring

References

- [CIDR] The CIDR Report, http://www.cidr-report.org/
- [Bu] “On Characterizing BGP routing table growth”, T. Bu, L. Gao and D. Towsley, Global Internet 2002.
- [RIS] RIPE Routing Information Service, http://www.ripe.net/ris/
- [MRT] Multi-Threaded Routing Toolkit & SBGP, http://www.mrtd.net/

www.alcatel.com

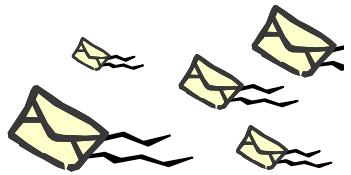


visual analysis of inter-domain routing dynamics

lorenzo colitti, ilaria de marinis,
giuseppe di battista, federico mariani,
maurizio patrignani, and maurizio pizzonia

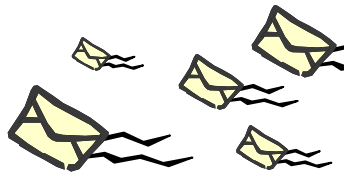
university of rome III
http://www.dia.uniroma3.it/~compunet/

overview



- BGPlay
  - a service for the visualization of inter-domain routing dynamics
- routing classes
  - a way to address the complexity of routing visualization

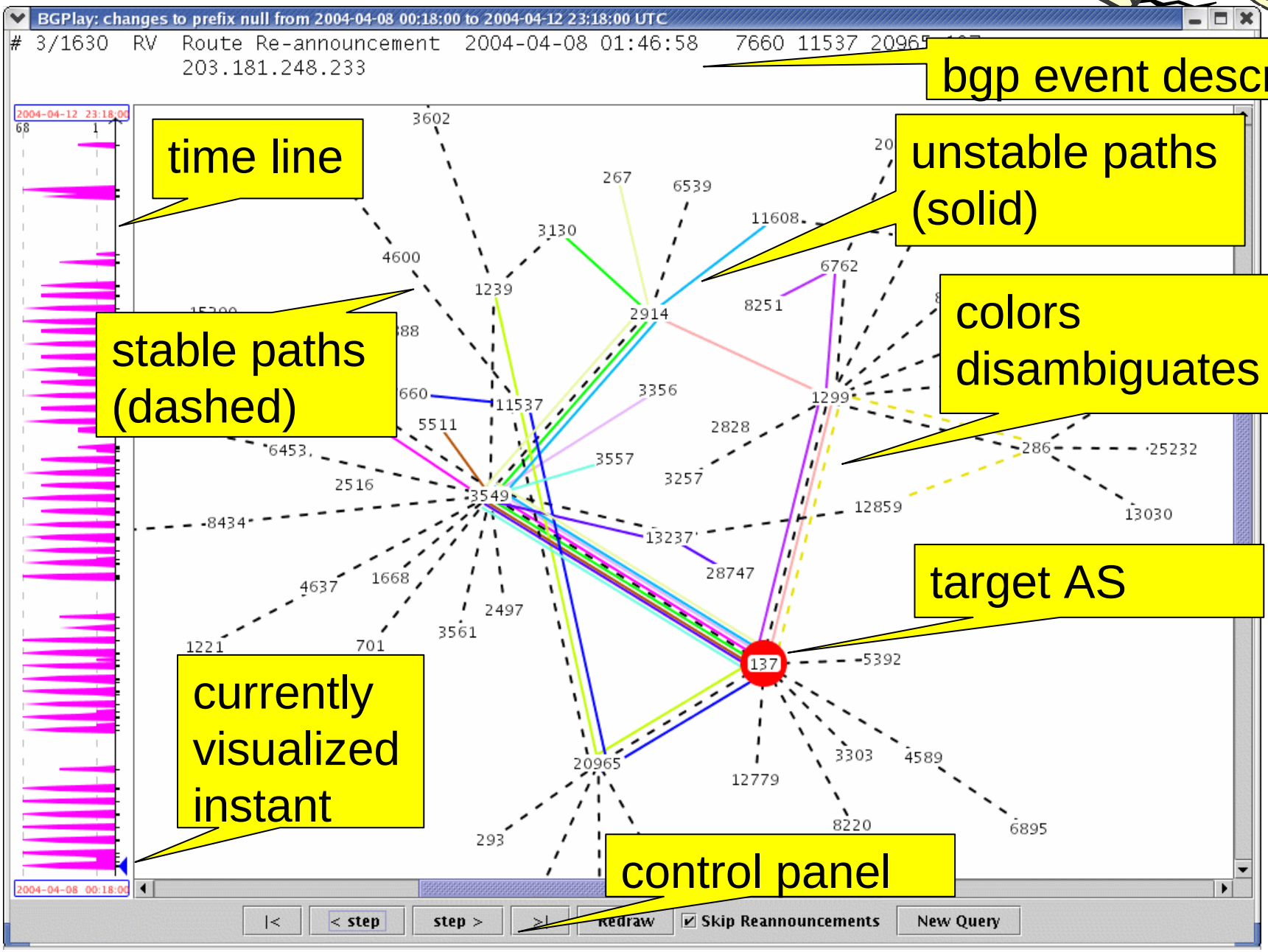
textual representation



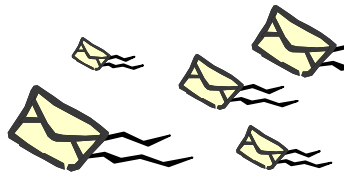
- textual representations of BGP data may be very hard to read

```
RIS DB query result for all RRC boxes.

State of the local RIB on 20040420.

Prefix          Time                    Peer            Next HOP        AS path      ....
193.0.0.0/21    2003-12-15 22:13:58Z    194.153.154.35  194.153.154.35  20854 3333
193.0.0.0/21    2004-01-15 18:01:03Z    193.0.0.56      193.0.0.56      3333
193.0.0.0/21    2004-01-15 18:01:31Z    195.69.144.68   195.69.144.68   3333
193.0.0.0/21    2004-02-19 03:22:48Z    195.69.144.196  195.69.144.68   6762 3333
.....

Updates between 2004-04-20 00:00:00Z  and 2004-04-20 13:08:23Z .

Type    Prefix          Time                    Peer            Next HOP          AS path     ...
A       193.0.0.0/21    2004-04-20 00:55:39Z    64.211.147.146  64.211.147.146    3549 1103 3333
A       193.0.0.0/21    2004-04-20 03:27:57Z    64.211.147.146  64.211.147.146    3549 1103 3333
.....
```

BGPlay


BGPlay: changes to prefix null from 2004-04-08 00:18:00 to 2004-04-12 23:18:00 UTC
3/1630 RV Route Re-announcement 2004-04-08 01:46:58 7660 11537 20965
203.181.248.233
bgp event descr.
time line
unstable paths (solid)
stable paths (dashed)
colors disambiguates
target AS
currently visualized instant
control panel
3602
267
6539
11608
3130
4600
1239
6762
8251
2914
11537
3356
1299
5511
2828
6453
3557
3257
286
25232
2516
3549
12859
13030
8434
13237
28747
4637
1668
2497
3561
1221
701
137
5392
20965
3303
4589
12779
8220
6895
293
|<
< step
step >
>|
Redraw
Skip Reannouncements
New Query

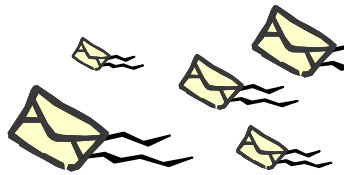
dealing with BGP updates and events

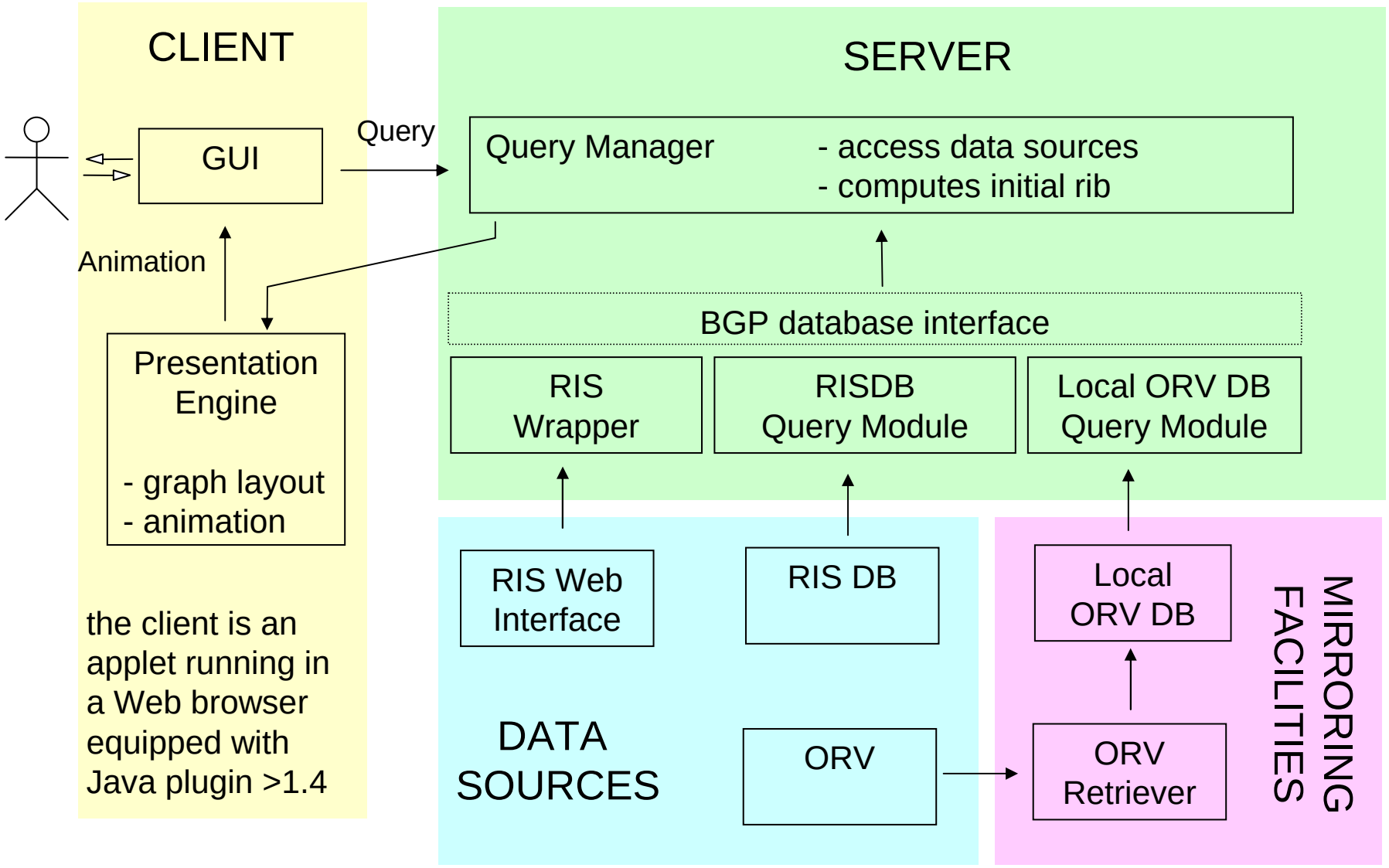


- BGP events are shown by means of animations

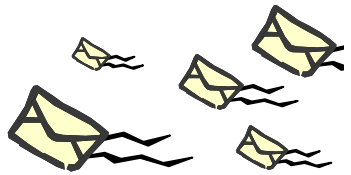
event	AS-path animation
new route: when there is no collector-target path available and one is received	appears and flashes
route re-announcement: when an already known collector-target path is announced again	flashes
route change: when a collector-target path is known and a different one is announced	moves smoothly to the new shape
route withdrawal: when a collector-target path is withdrawn, implies no connectivity until the next “new route” event	flashes and disappears

the BGPlay architecture

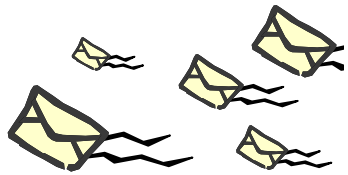




BGPlay@RIS (beta)

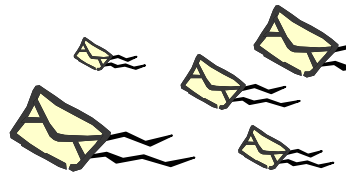


- BGPlay is currently hosted at RIS
  - http://www.ris.ripe.net/bgplay
  - listed under “Tools for Querying the RIS Database”
  - alternative to the “Search by Prefix” service
- provides graphical view of the RIS updates DB
  - 11 route collectors
  - 3 months archive
- alpha version @ Univ. Rome III
  - RIS 11 route collector
    - some of them wrapped, 3 months archive
  - 7 days RouteViews archive (local mirror)



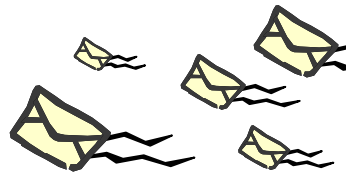
BGPlay demo...

BGPlay: possible evolutions



- better user interaction
  - hints on more/less specific prefixes, use colors to convey information (e.g. about activity of the routes), zoom (on timeline and on as-graph), etc.
- highlight faults
  - see for example Caesar, Subramanian, Katz – NANOG30 2004
- more info from registries
- visualize info for many prefixes at once
  - possible for prefixes that behave the same

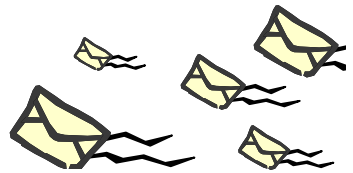
routing class definition

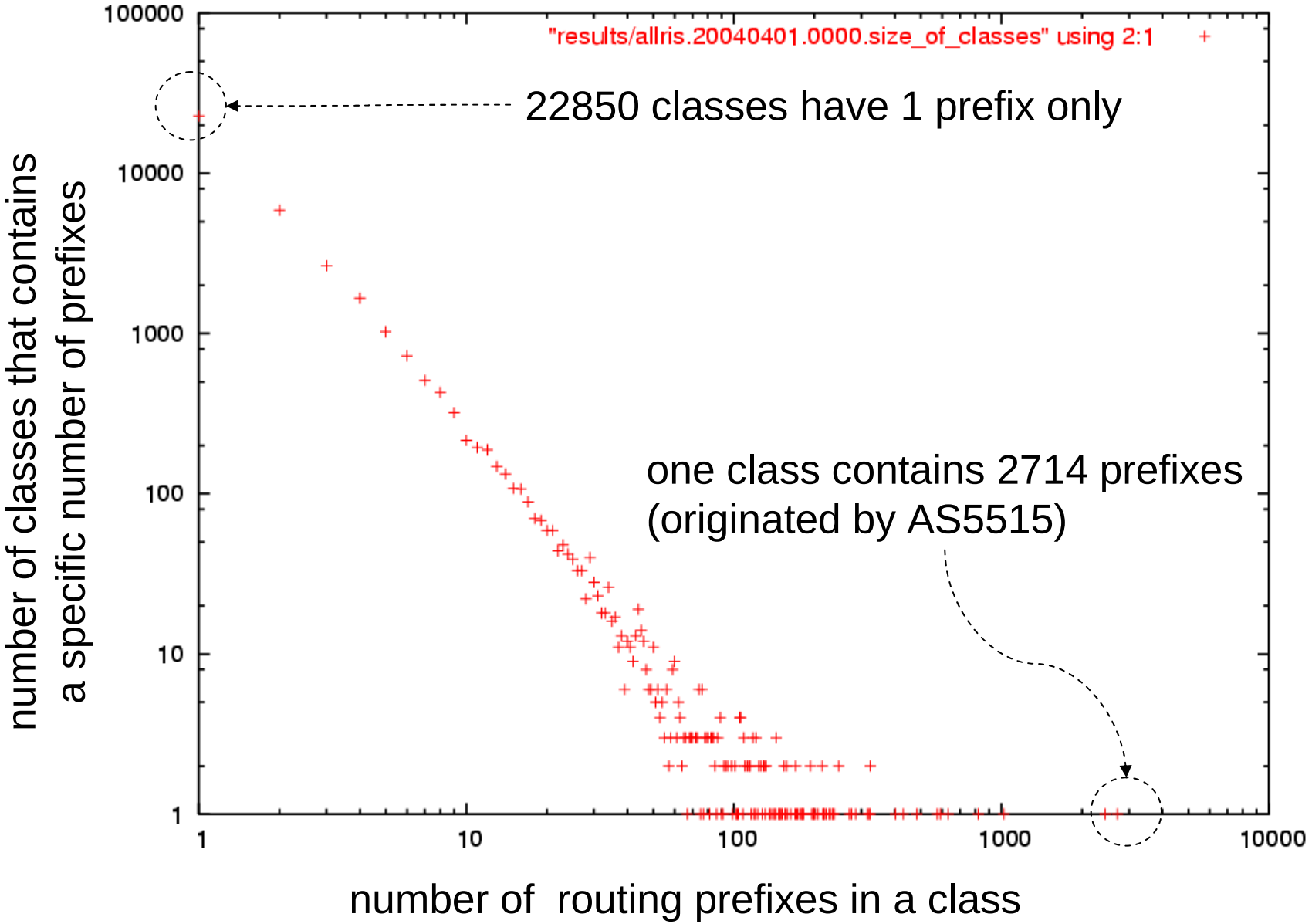


- two prefixes are *equivalent* if all the AS-paths to them are the same
- this *equivalence relationship* induces equivalence classes, that we call *routing classes*

- routing classes depends on
  - the considered instant of time
  - the vantage points from which we gather data

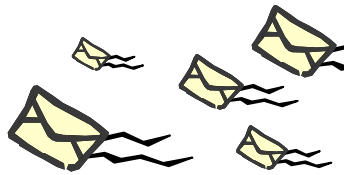
#prefixes within each class





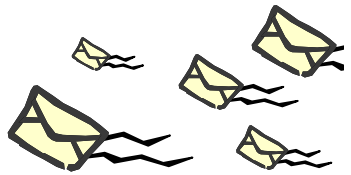
RIS (all collectors) -- 1st april 2004

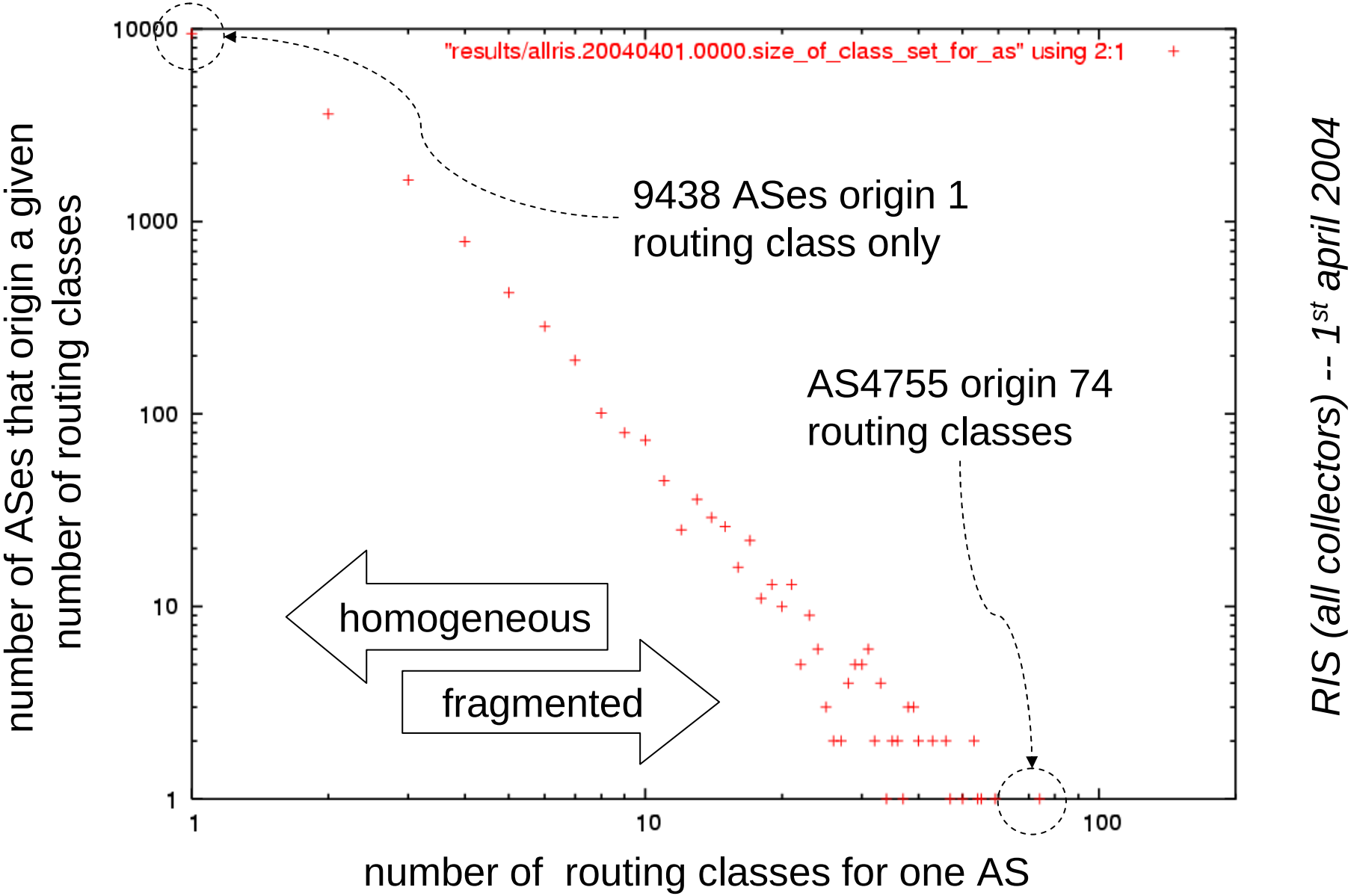
routing classes and ASes



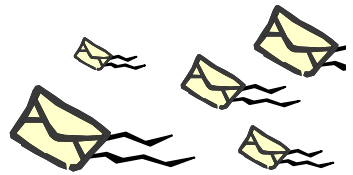
- prefixes in the same class are originated by the same AS
- an AS may origin
  - one (or very few) big routing classes: *homogeneous routing*
  - many small routing classes: *fragmented routing*

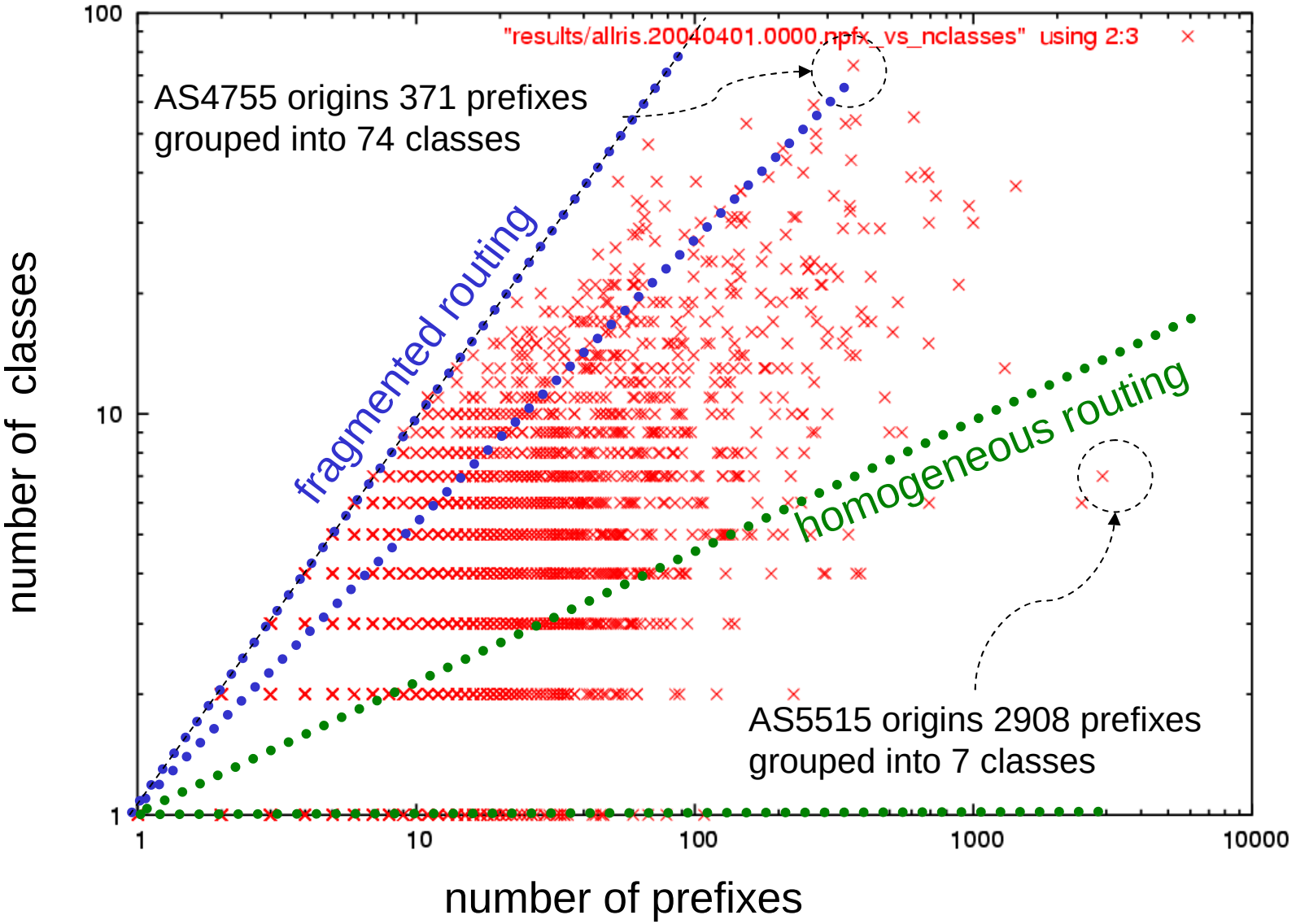
#classes within each AS



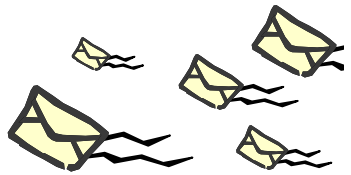


#classes vs. #prefixes



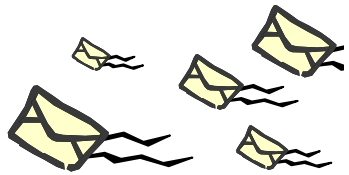


many interesting questions

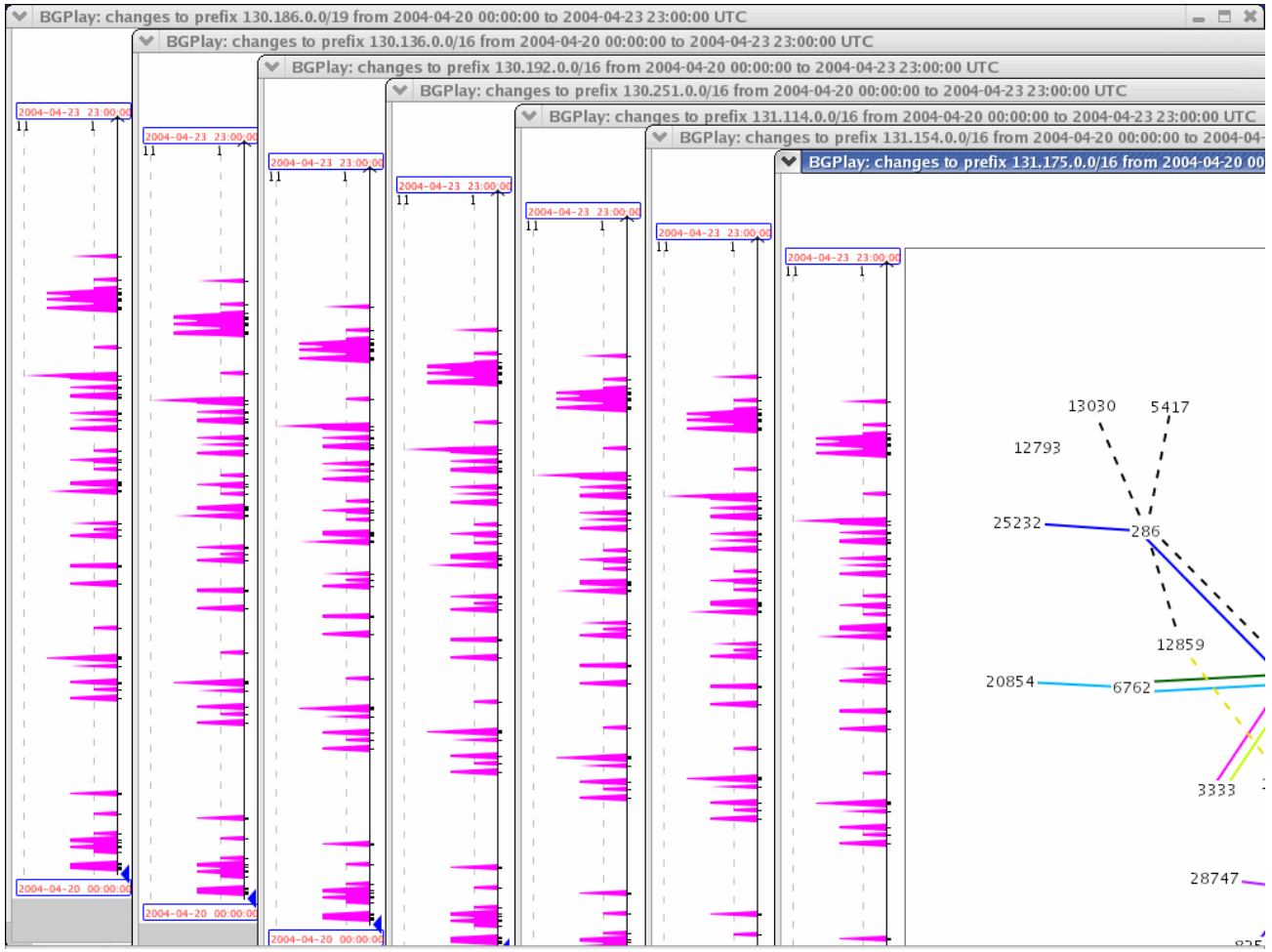


- how stable are routing classes...
  - ...over time? do they split on faults?
  - ...varying the vantage points?
- does fragmented routing affects...
  - ...service quality?
  - ...network management?
- what about routing classes into BGPlay?

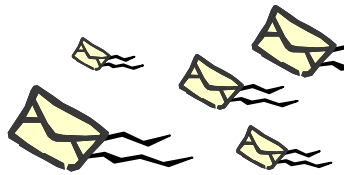
quick routing classes stability test using BGPlay



- one of the routing classes of AS137 over 3 days

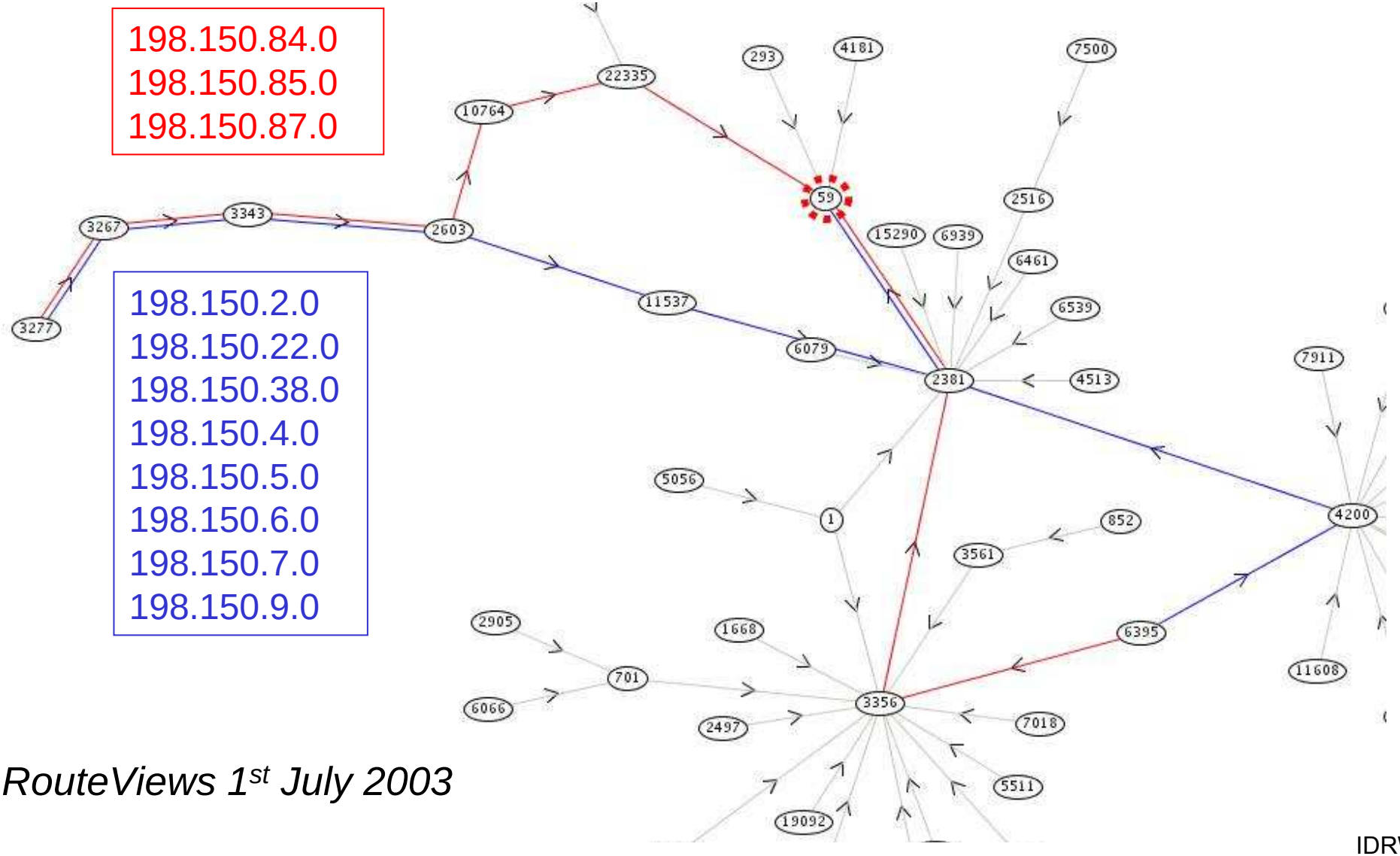


example of non homogeneous routing visualization



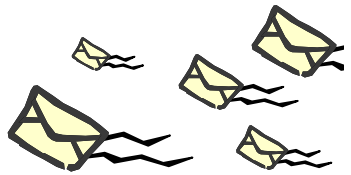
198.150.84.0
198.150.85.0
198.150.87.0

198.150.2.0
198.150.22.0
198.150.38.0
198.150.4.0
198.150.5.0
198.150.6.0
198.150.7.0
198.150.9.0

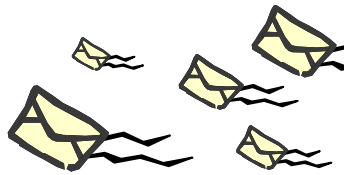


RouteViews 1st July 2003

future work



- rigorous investigation of routing class stability
  - alternative definitions may involve time evolution
- BGPlay improvements
  - routing classes
  - faults
  - reverse paths



questions?

CITE

Cooperative Inbound Traffic Engineering

B. Quoitin and O. Bonaventure
(bqu,obo@info.ucl.ac.be)
Computer Science and Engineering Department
Université Catholique de Louvain, Belgium

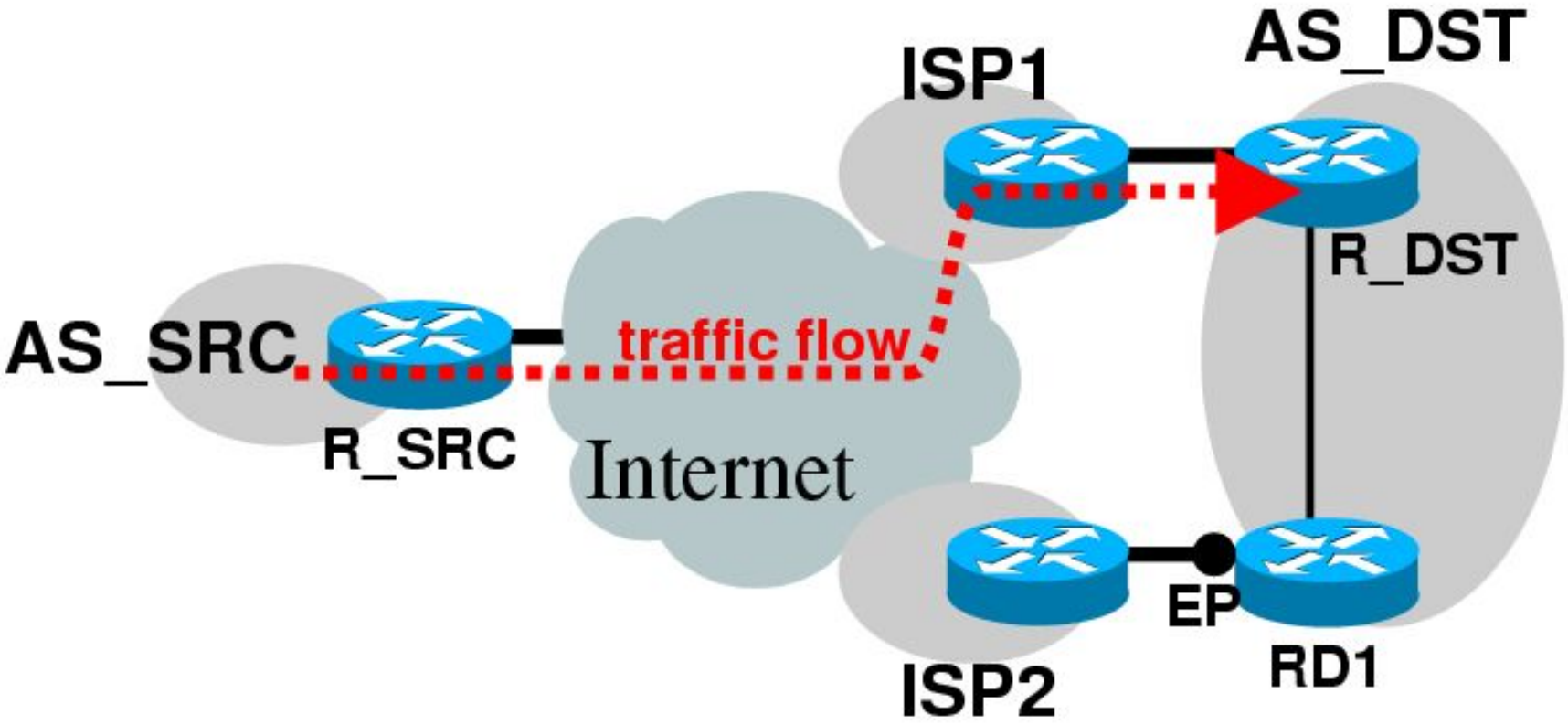
Agenda

- Motivations
- A new approach
  - Example scenario
  - Multi-homed stub
  - Transit domain
- Scalable Inbound TE
- Conclusion and further work

Motivations

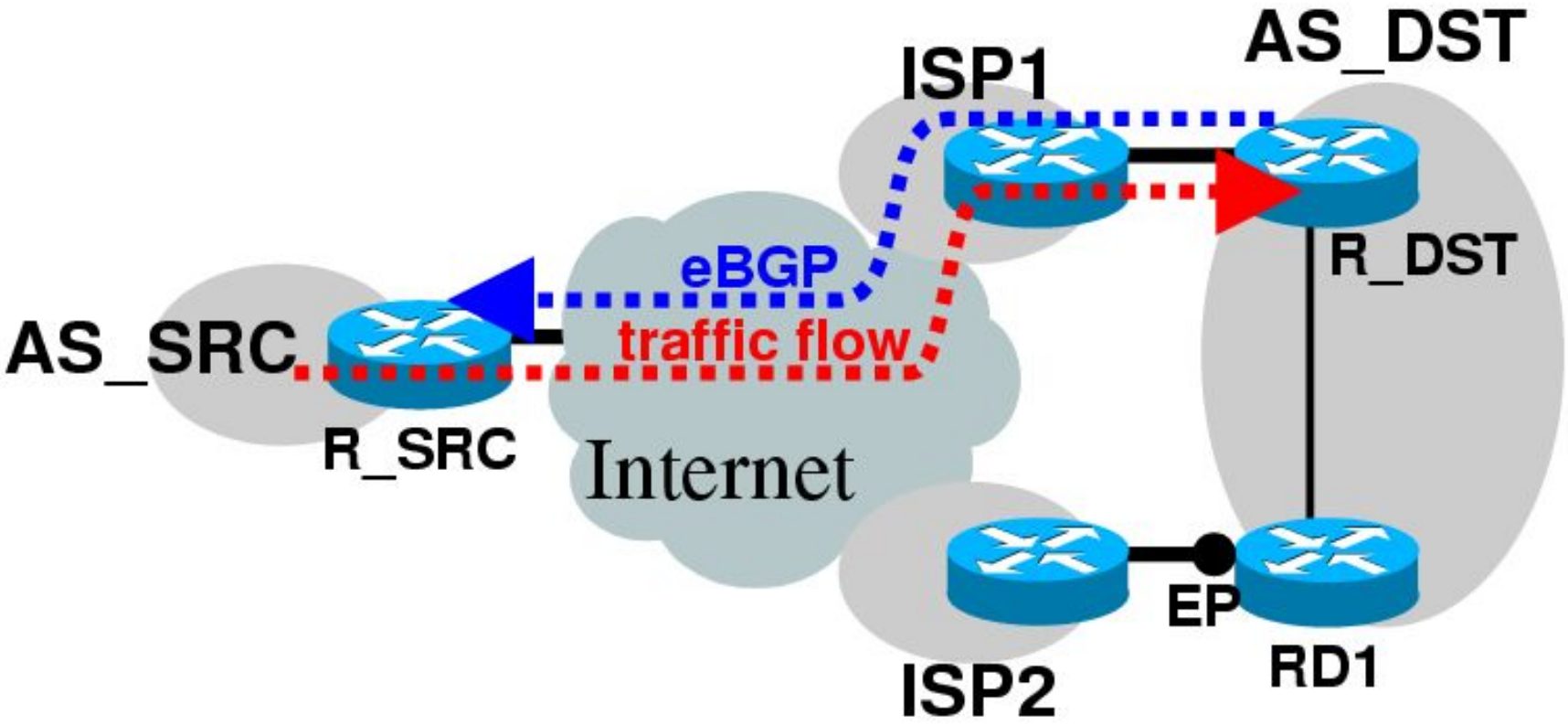
- Target
  - multi-homed stubs (> 60% of stubs)
  - Motivations for multi-homing: Fault-tolerance
- Issue
  - BGP routing often causes imbalance
  - Traffic control is a feature absent in BGP
    - Tweaking of BGP attributes does not work
    - Reduce the cost of transit
    - Reduce congestion (during periods of “abnormal” traffic patterns)
    - Select route based on QoS metric (delay)

Example scenario (1)



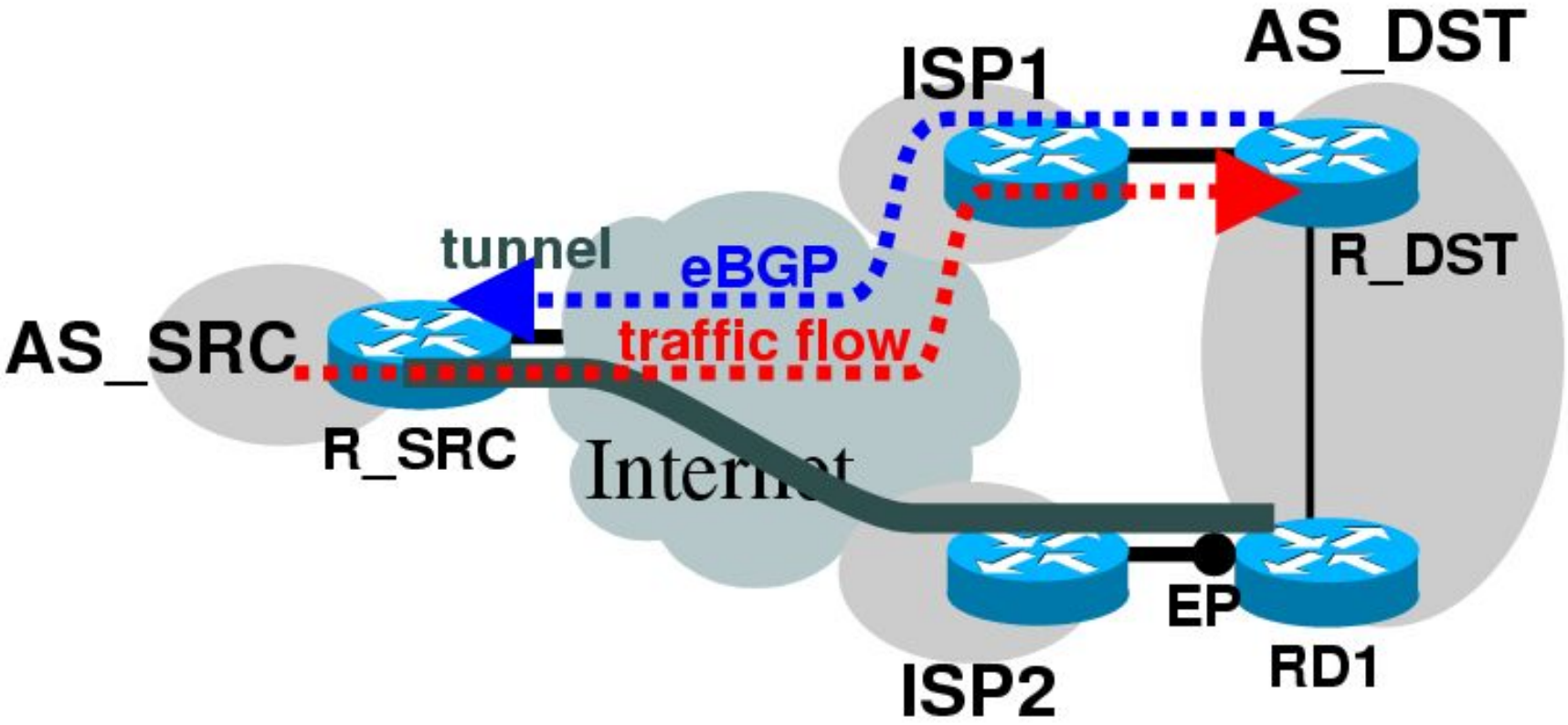
- AS_DST wants to receive traffic from AS_SRC through RD1

Example scenario (2)



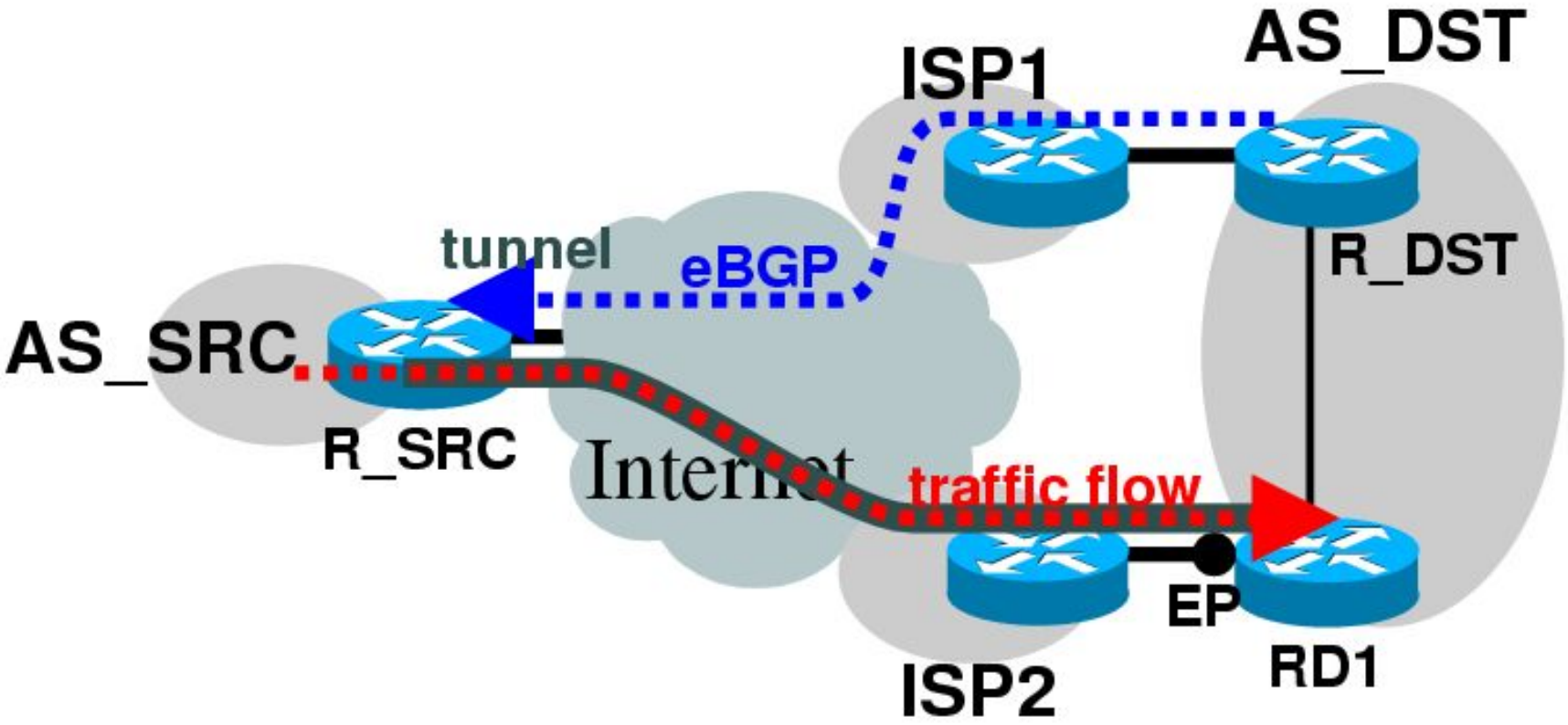
- AS_DST contacts R_SRC in AS_SRC
  - AS-Path = {AS_DST}
  - tunnel end-point EP

Example scenario (3)



- R_SRC establishes a tunnel with EP
  - EP is an IP address of RD1
  - EP belongs to a prefix advertised by ISP2

Example scenario (4)

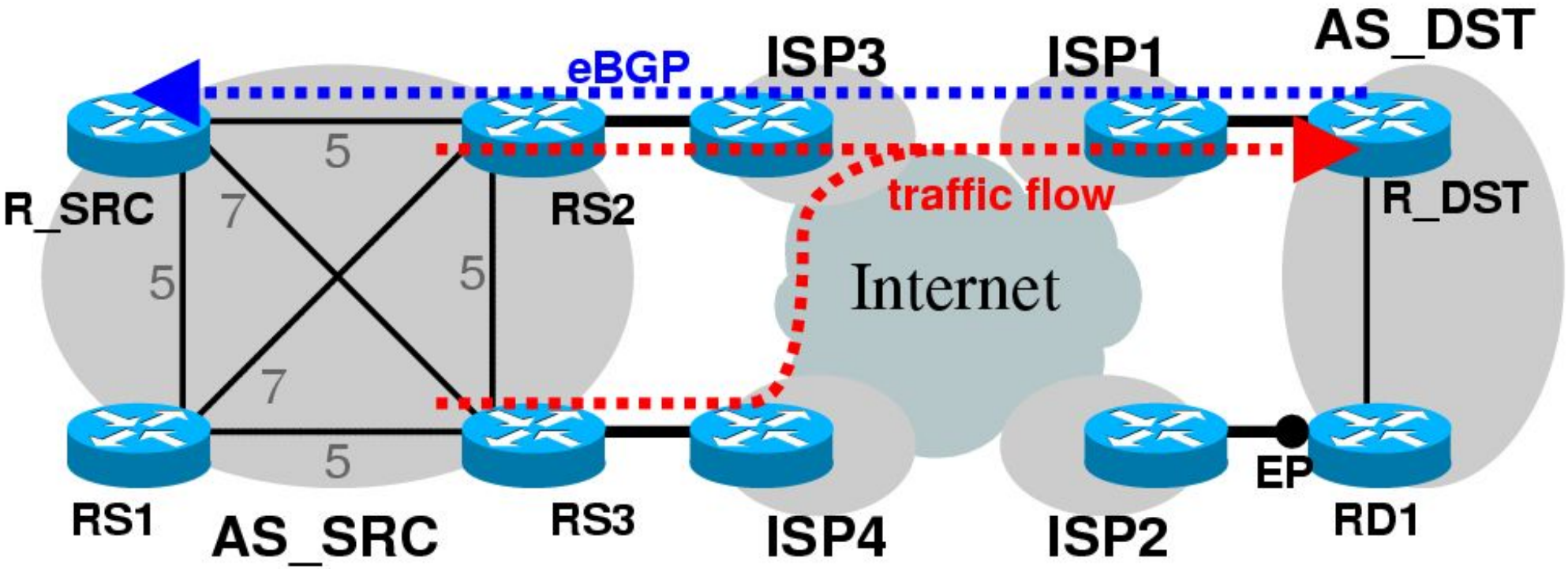


- R_SRC updates its routing table in order to forward packets destined to AS_DST through the tunnel

Issues and solutions

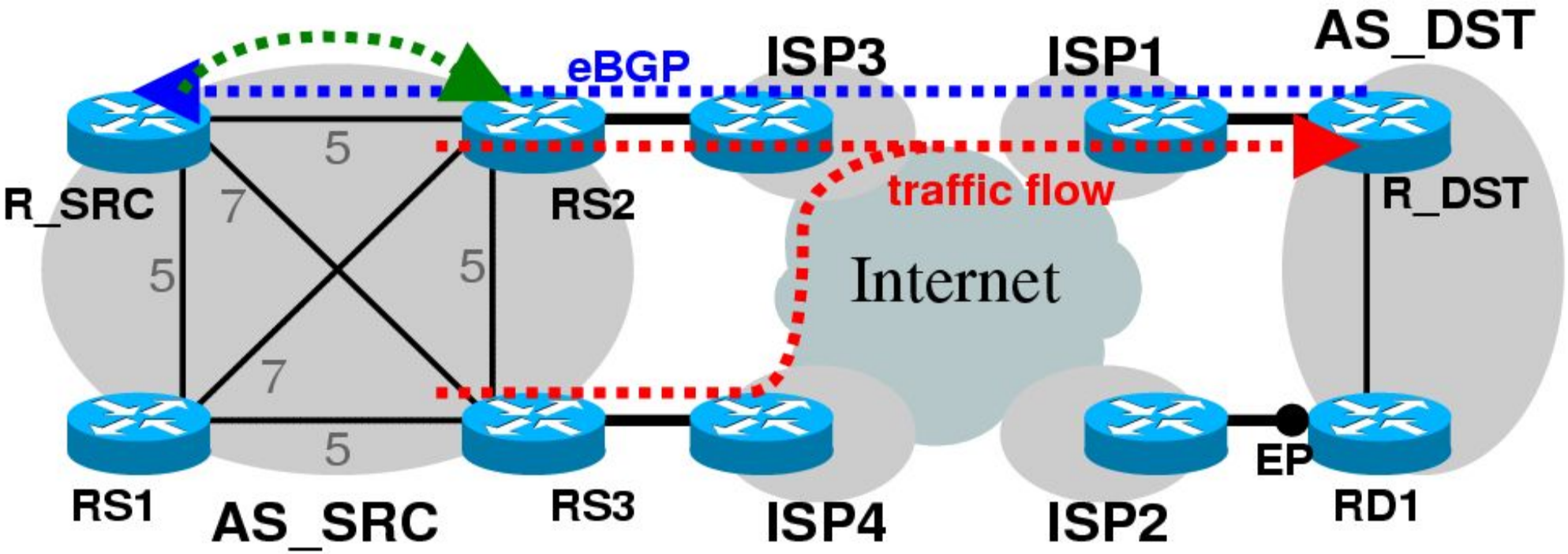
- Dynamic multi-hop eBGP session
  - router R_SRC exposed
  - IPSec for inter-router security
  - MD5 not suitable (requires a passwd)
  - S-BGP or soBGP for router authentication and route validation
- Tunnel information
  - flexible community or tunnel-SAFI (MP-BGP)
  - tunnel types: L2TP, GRE, IPSec...

Multi-homed stub (1)



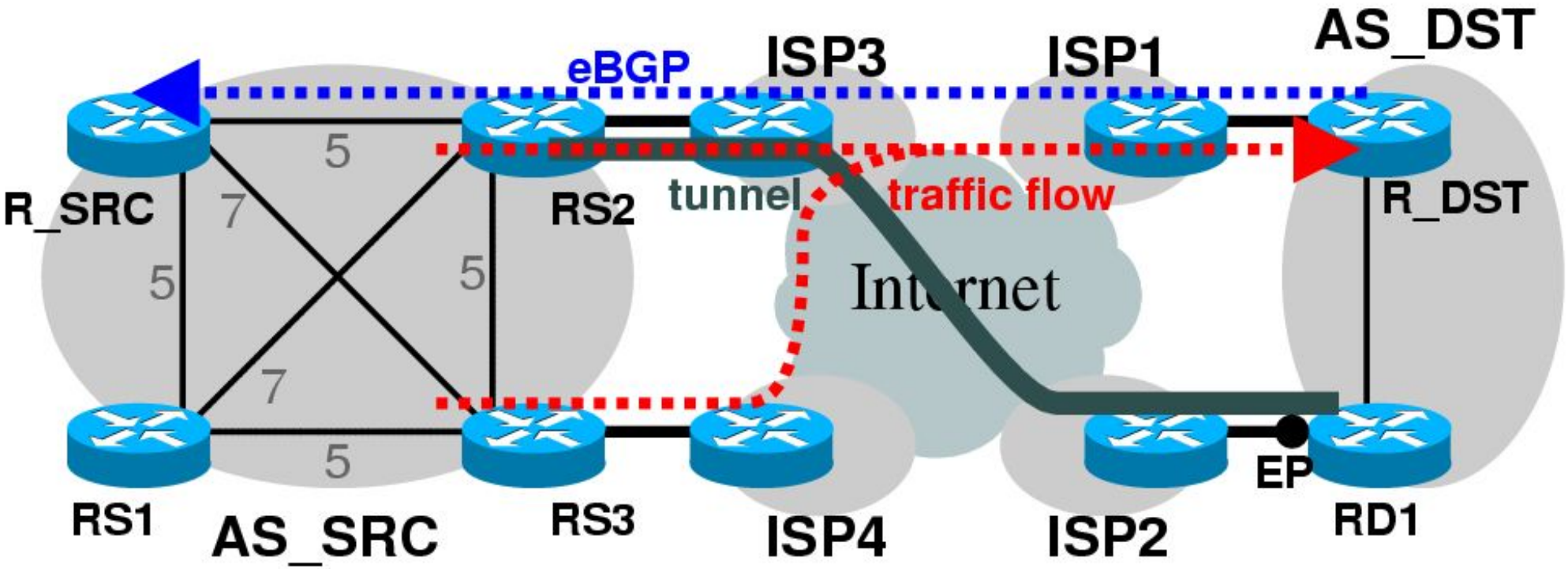
- First case, R_SRC selects a single exit-point
  - How to select an exit point ?
  - looks into its BGP routing table
  - selects best route towards ISP2 (through RS2)

Multi-homed stub (2)



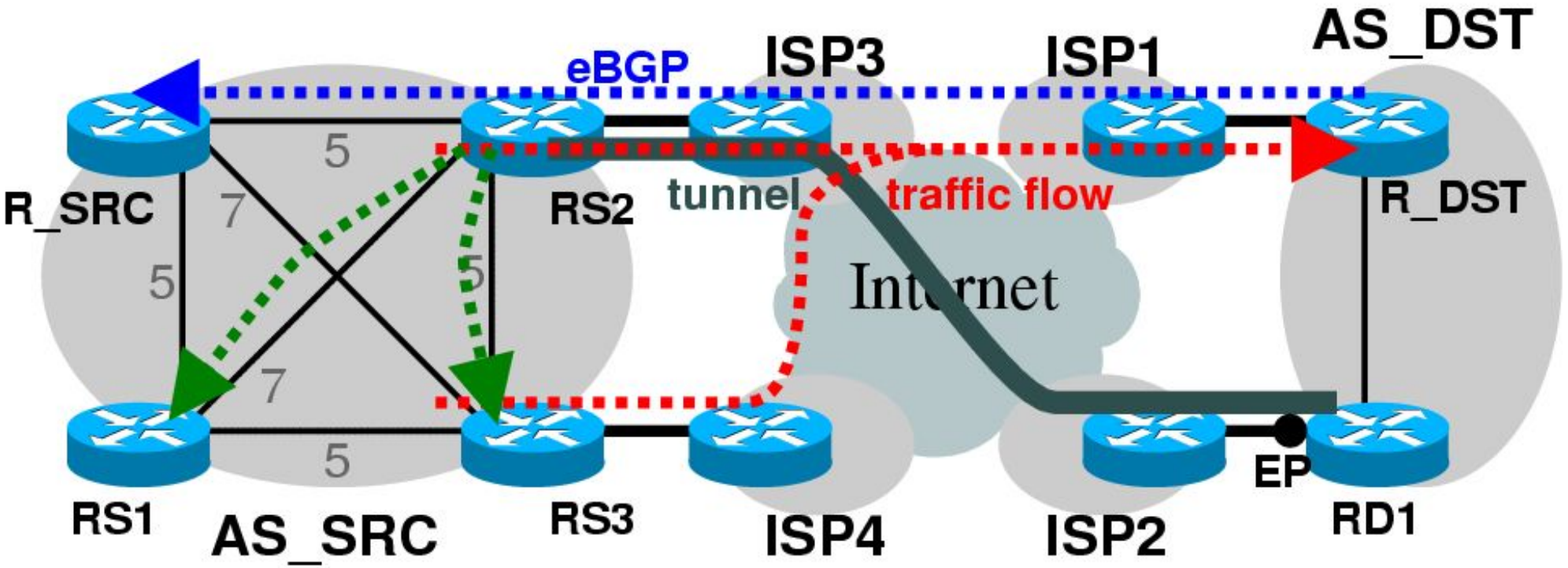
- BGP Update sent to RS2:
  - next-hop = RS2, higher local-pref
  - tunnel end-point EP and parameters
  - not redistributed outside AS_SRC

Multi-homed stub (3)



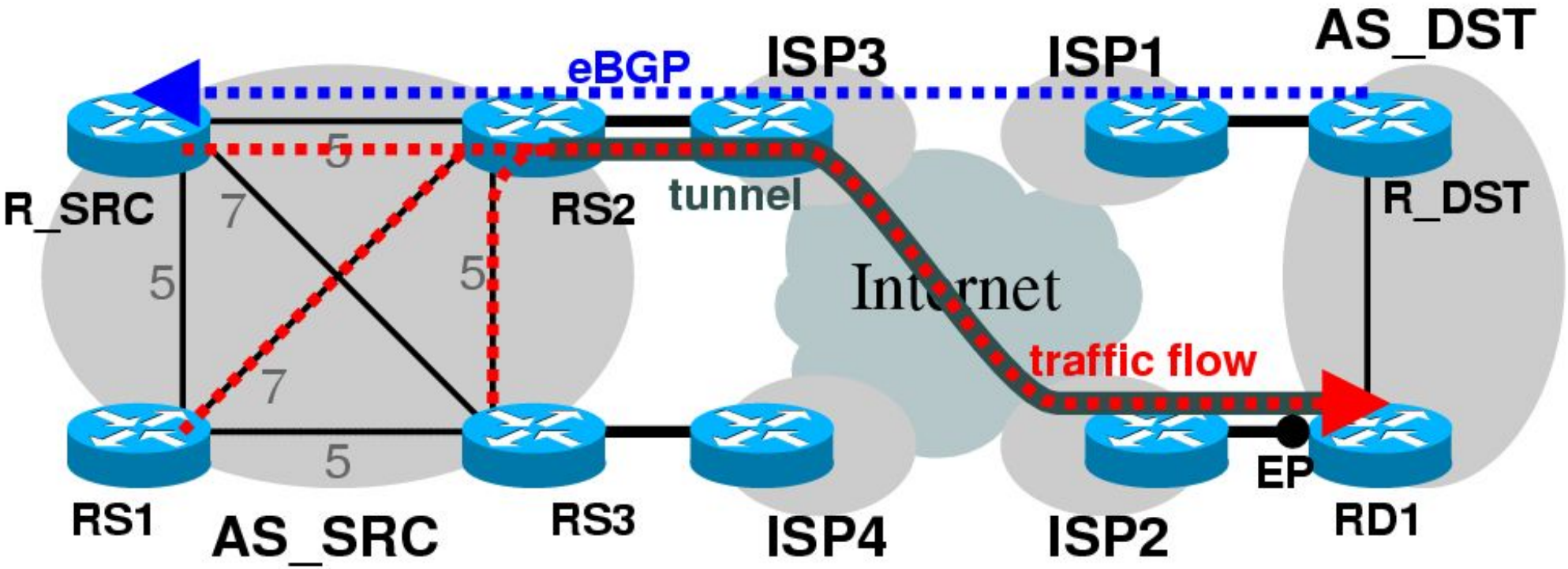
- RS2 establishes tunnel with EP

Multi-homed stub (4)



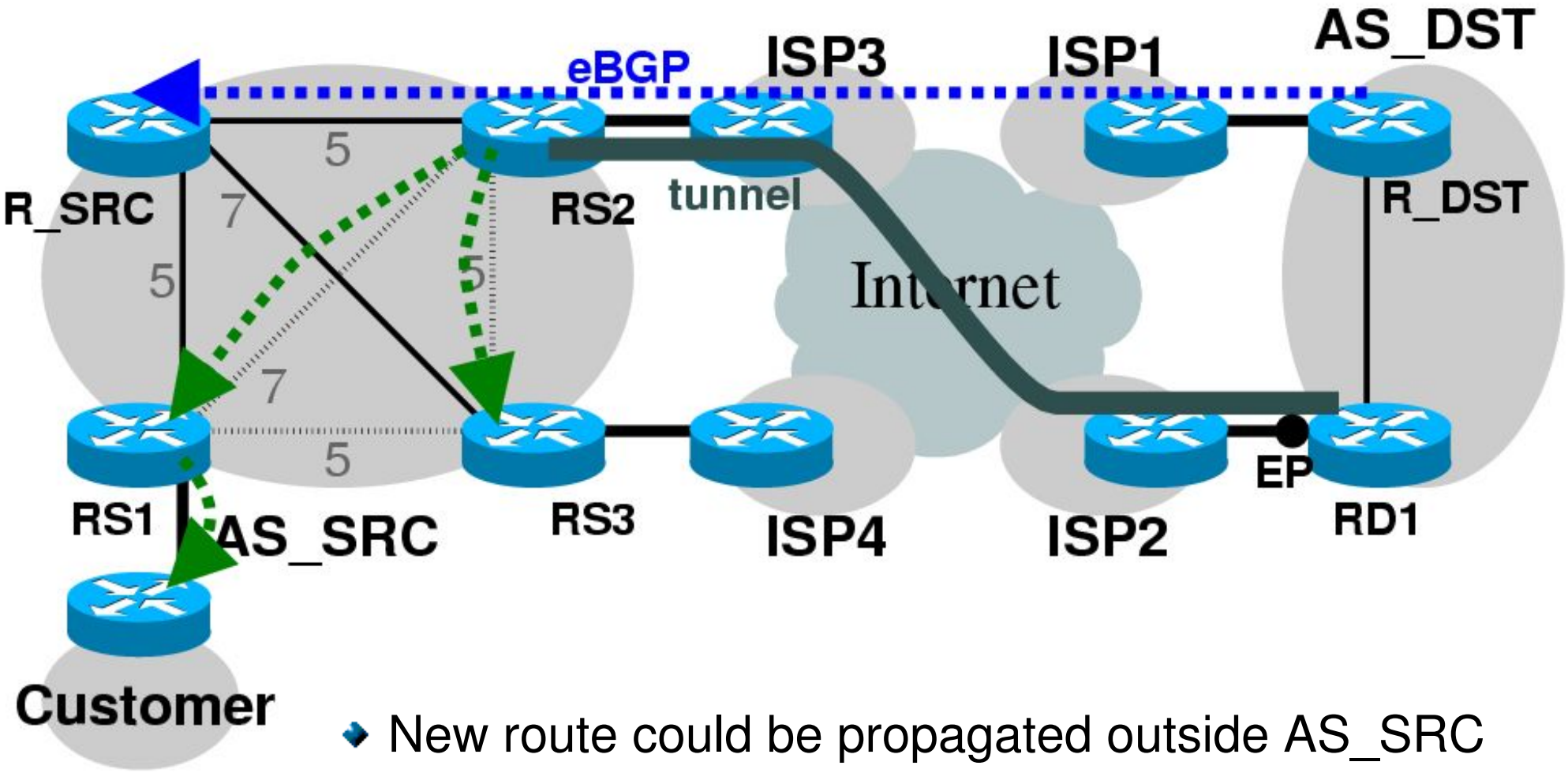
- RS2 redistributes new route inside iBGP and updates its own Loc-RIB
  - next-hop = RS2, higher local-pref
  - not redistributed outside AS_SRC

Multi-homed stub (5)



- And traffic now enters AS_DST through RD1

Transit domain

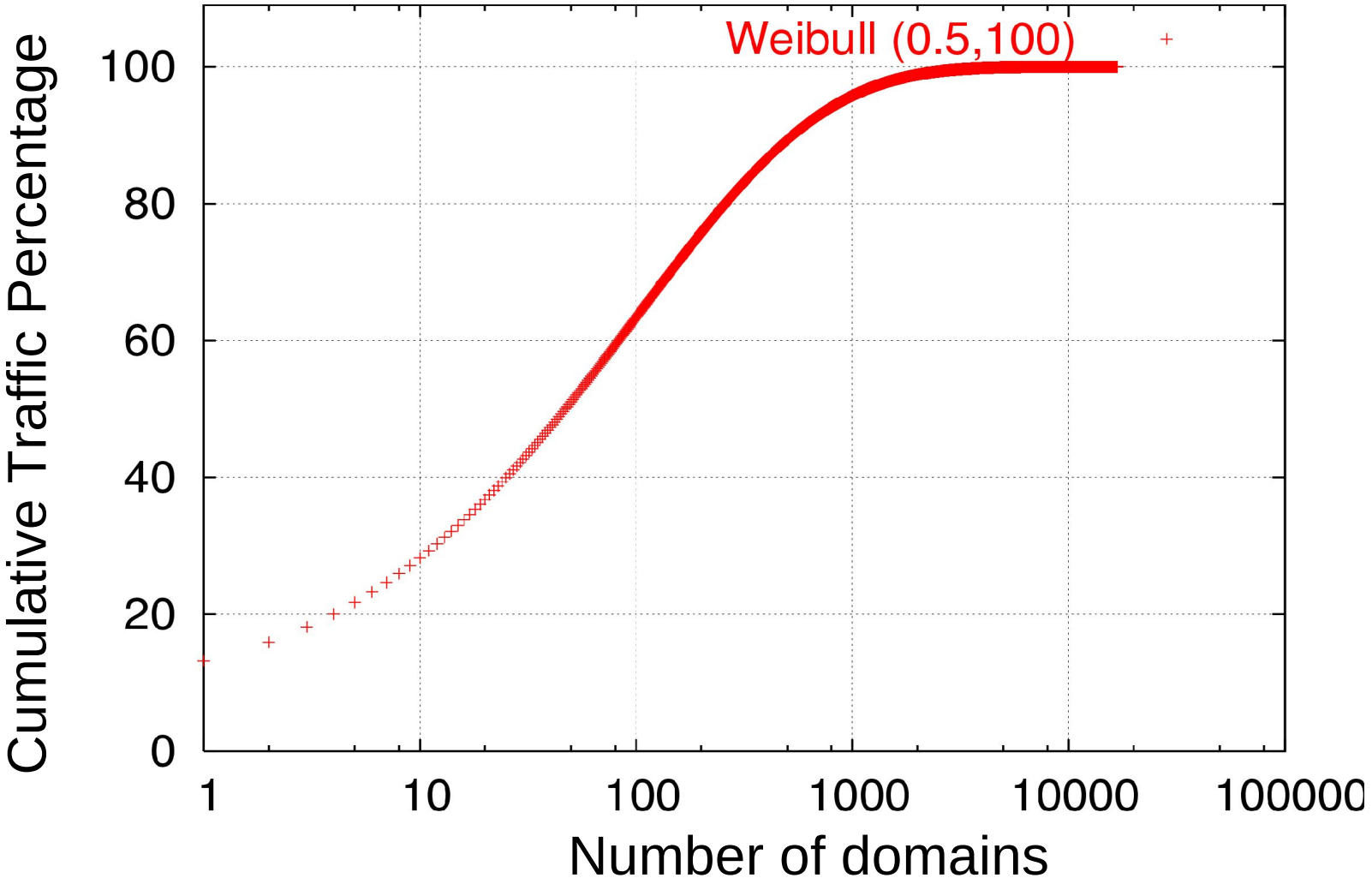


- New route could be propagated outside AS_SRC
- To avoid traffic loops:
  AS-Path = {AS-Path to ISP2} + {AS_DST}

Load-balancing simulation

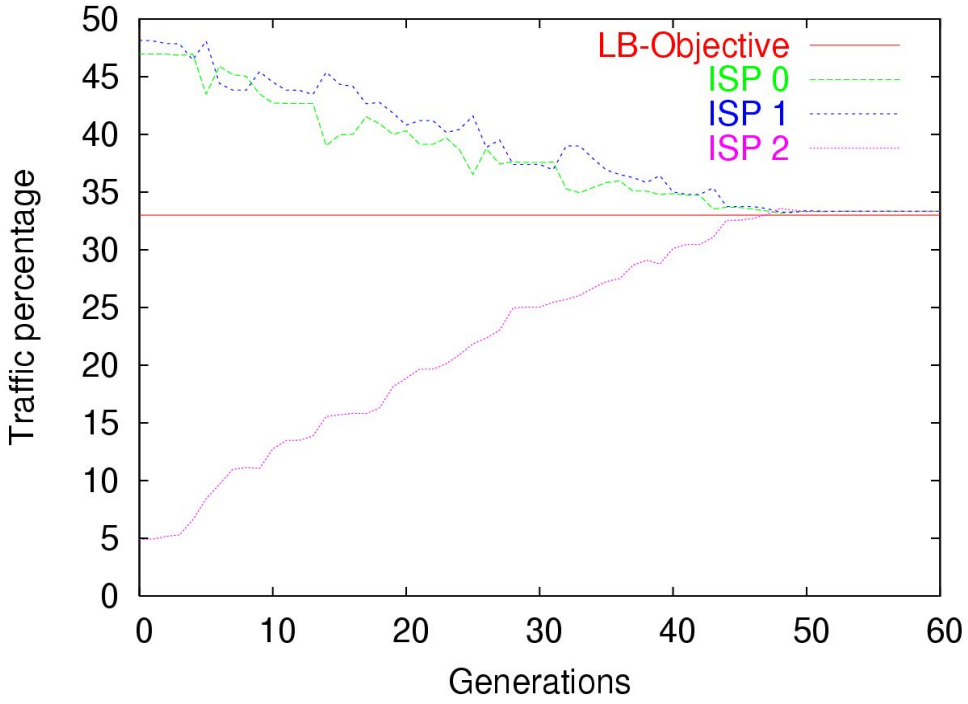
- Scenario
  - Load balancing of traffic on providers
  - Optimization problem
    - allocation of 2-10 providers to ~14.000 sources
    - minimization of imbalance
    - solved by Evolutionary Computing
  - Assumptions
    - transit domains send a neglectible amount of traffic
    - distribution of traffic ~ weibull(0.5)
  - initial allocation of sources computed by simulation of BGP (with C-BGP simulator)

Traffic distribution



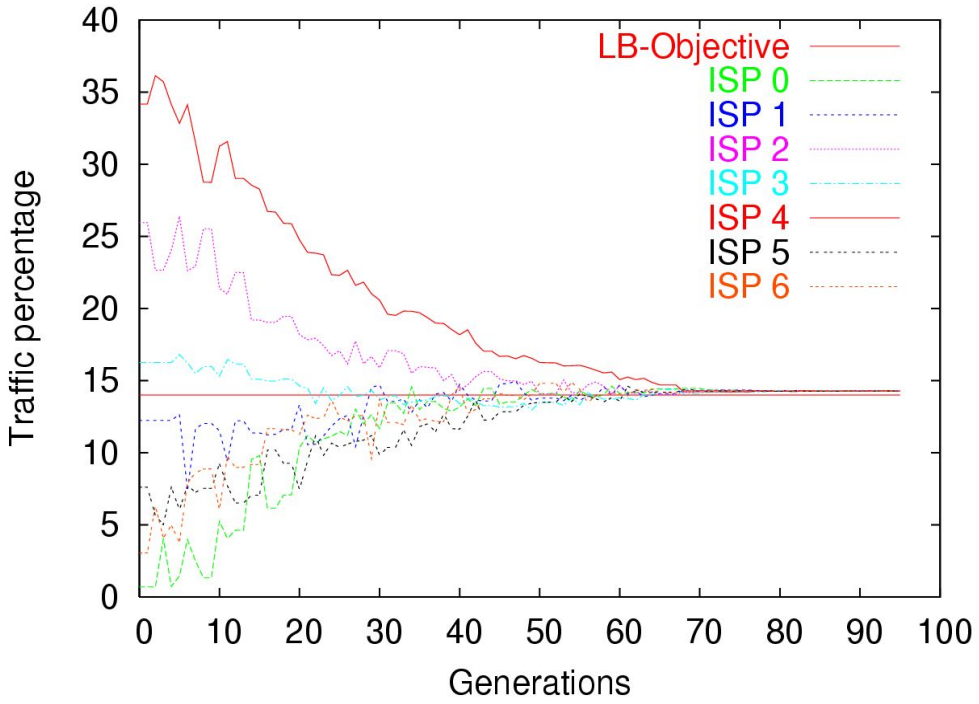
Preliminary results

AS26404



3-homed stubs

AS10794



7 providers

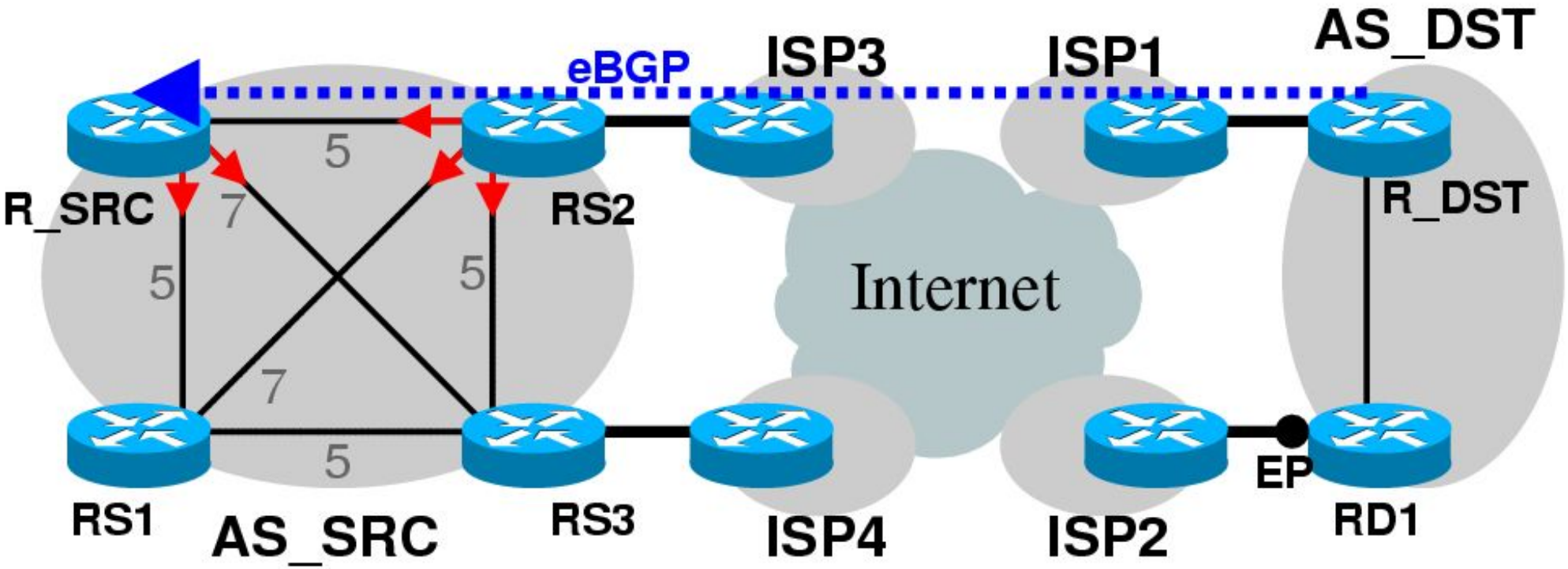
Conclusion

- Solution
  - Deployment
    - use existing protocol with little modifications
  - Scalability: **limited number of tunnels** to use and **limited impact** on BGP stability
  - Determinism
- Further work
  - Move transit sources (on-line)
  - Take traffic dynamics into account
  - Inbound TE for a transit domain

Appendix

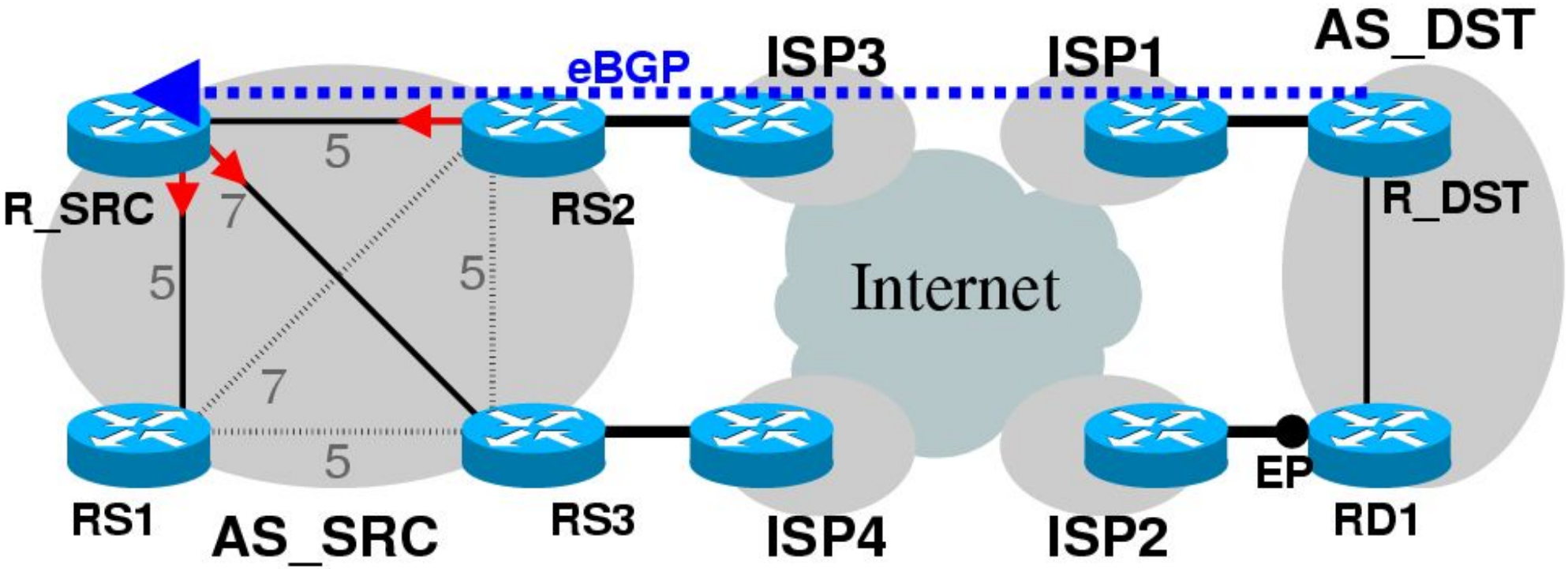
Appendix

Full-mesh of iBGP sessions



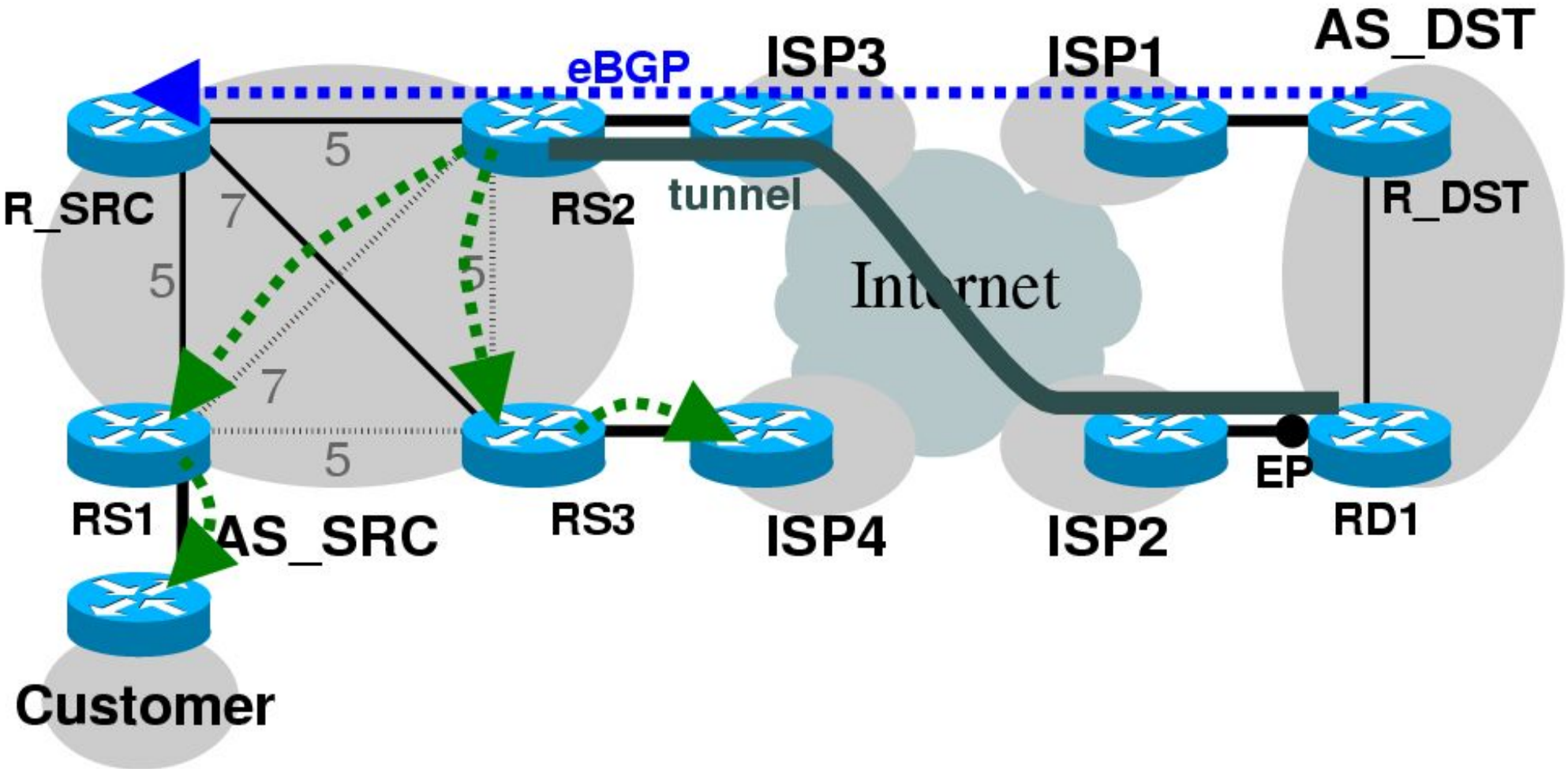
- if AS_SRC uses a **full-mesh of iBGP sessions**
  - R_SRC knows all external routes towards ISP2 that have been chosen by border routers

Route-Reflectors



- AS_SRC uses **route-reflector(s)**
  - R_SRC must be an RR in order to have the more complete view of external routes

Transit in the core



Towards a more systematic approach for interdomain traffic engineering

Steve UHLIG

suh@info.ucl.ac.be
http://www.info.ucl.ac.be/~suh/

Computer Science and Engineering Dept.
Université Catholique de Louvain, Belgium

State-of-the-art of interdomain TE

- **“Route optimization” techniques (InterNap, RouteScience, Opnix, Proficient, Radware,...)**
- **Features in BGP routers for multiple-link load balancing (load-sharing and BGP multipath)**
- **ISP's interdomain TE is primitive :**
  - **change some route's attribute**
  - **check impact on traffic**
  - **accept or try again**

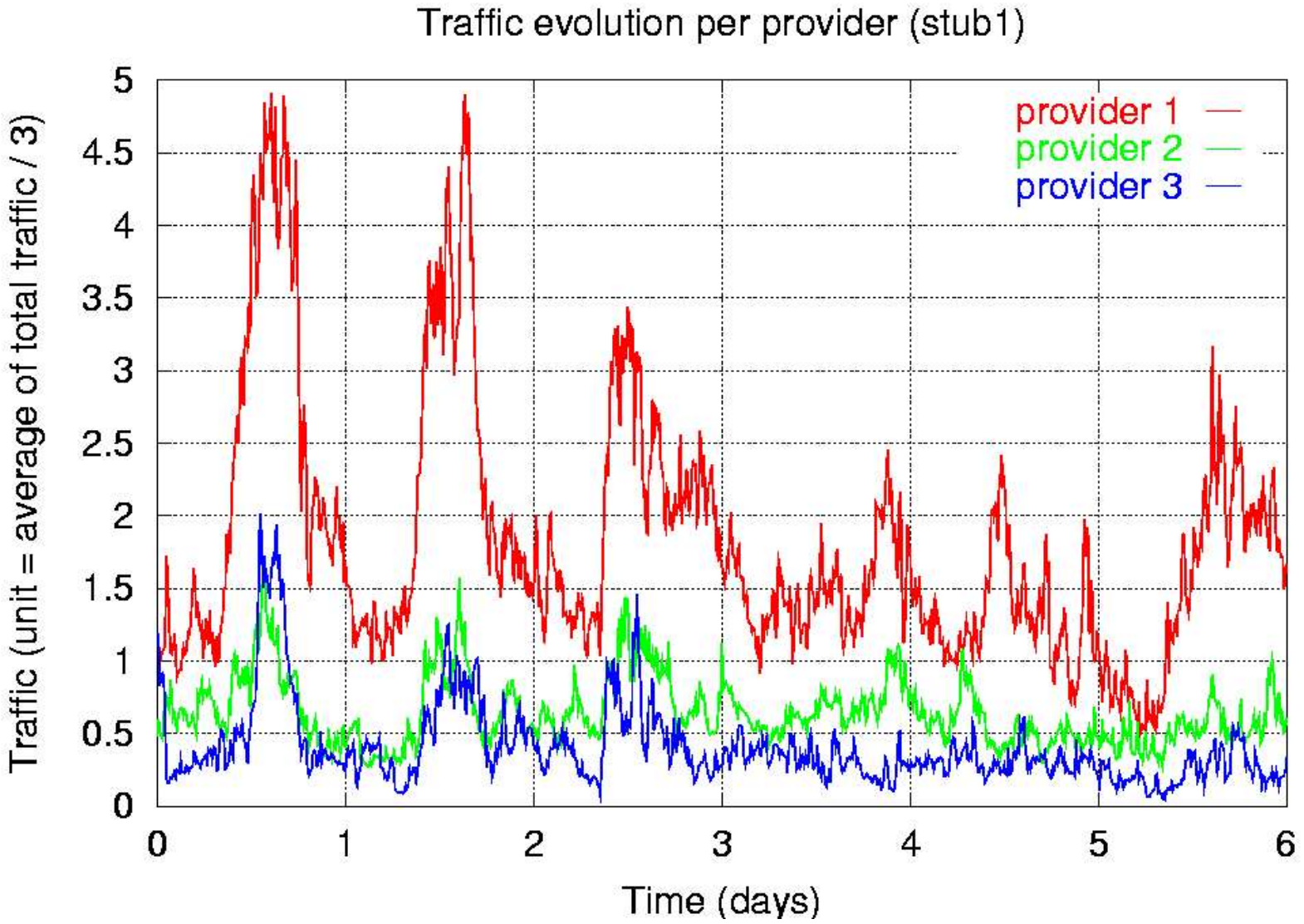
Problem statement

- **Objective 1 : minimize changes to be performed to best route BGP choice**
- **Objective 2 : optimize objective function defined on traffic sent to BGP neighbors (or next hop)**
- **Objective 3 : deal with objectives 1 and 2 in near real-time (a few minutes)**

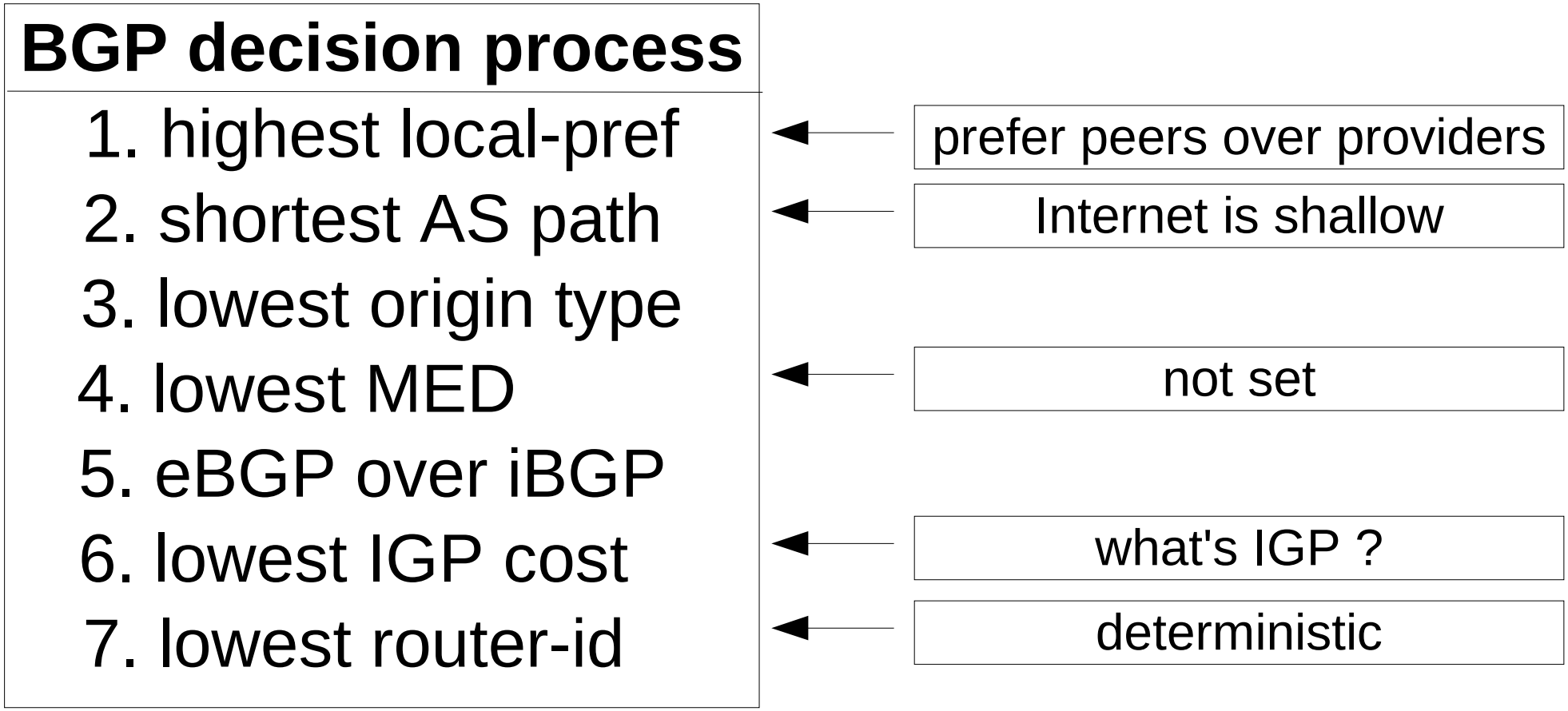
Main issues

- **Optimizing both traffic distribution and minimizing burden on BGP is NP-hard**
- **Tracking traffic over small timescales**
- **Uneven traffic distribution among neighbors found by BGP (tie-breaking)**

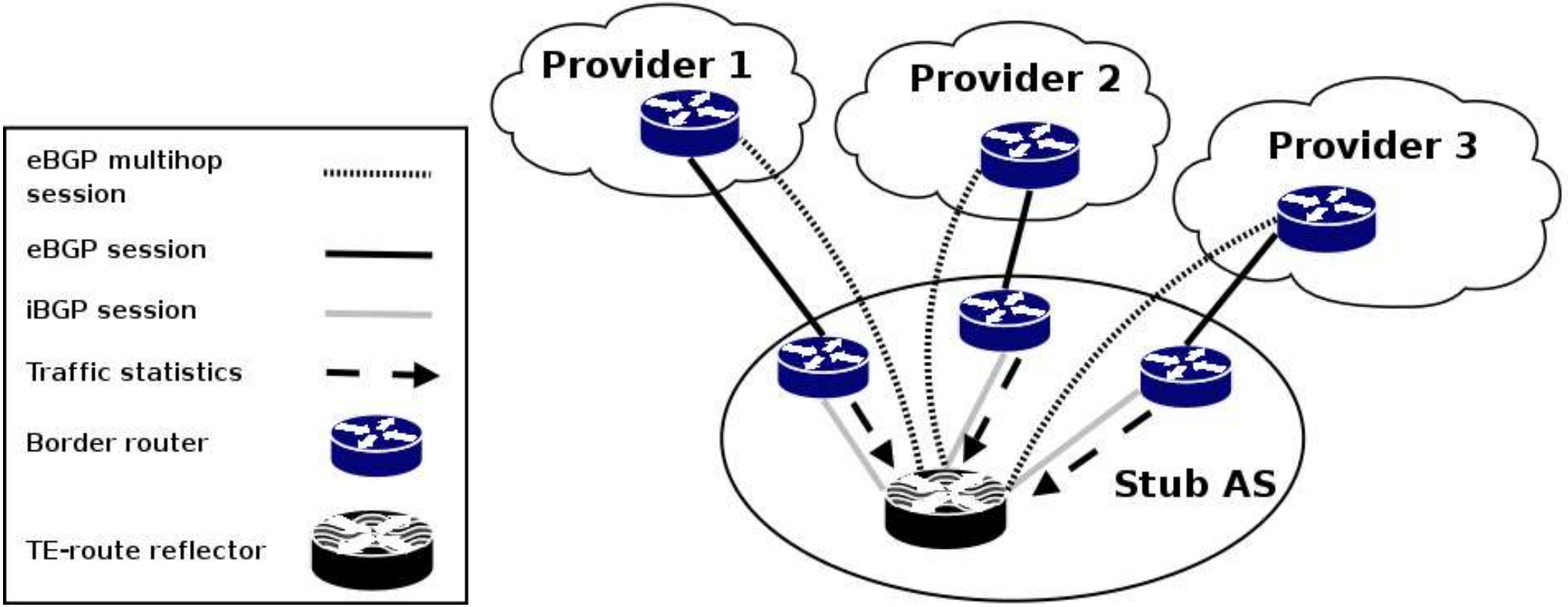
BGP as a poor traffic-balancer



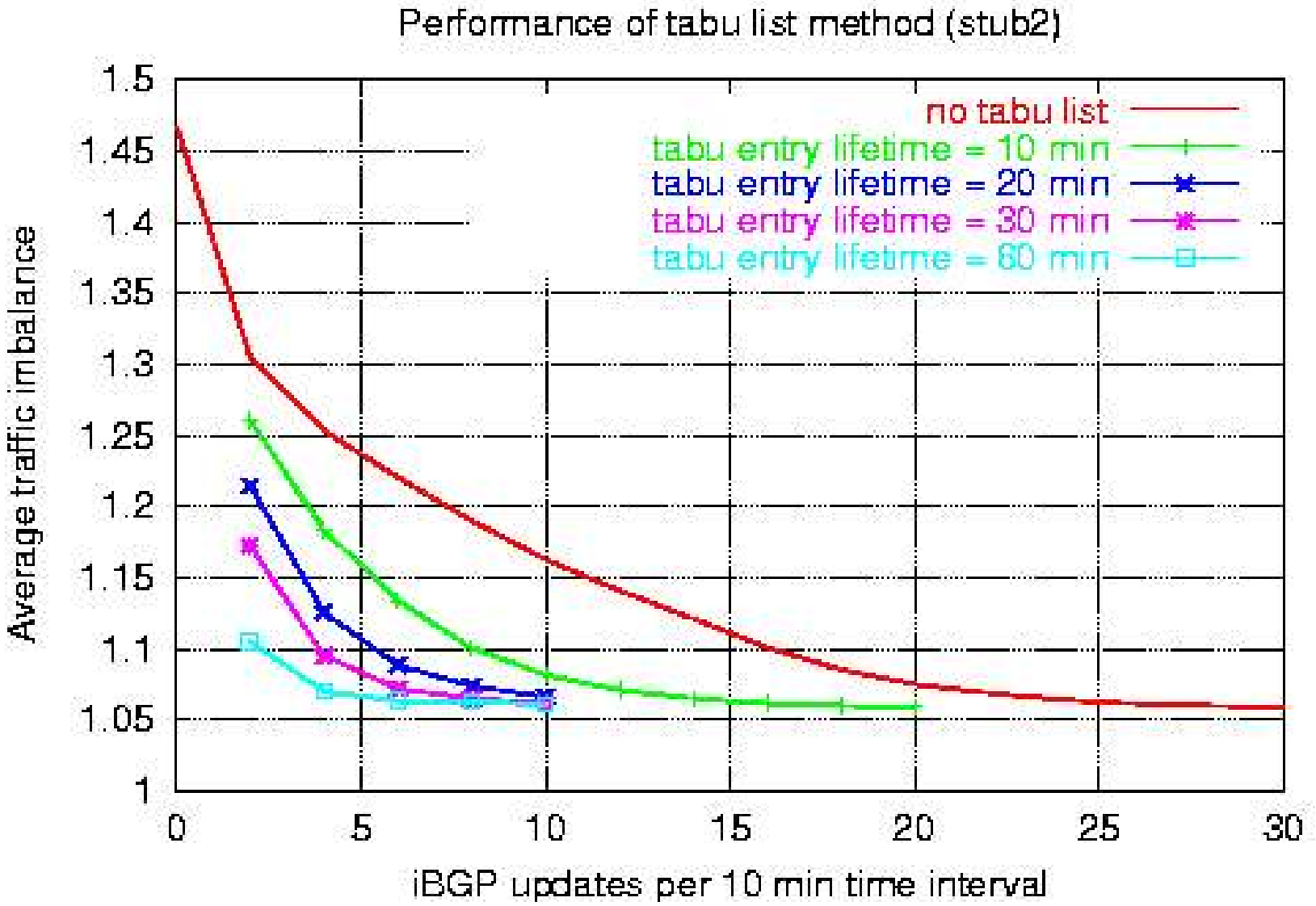
BGP decision process of stubs



Solution for stubs



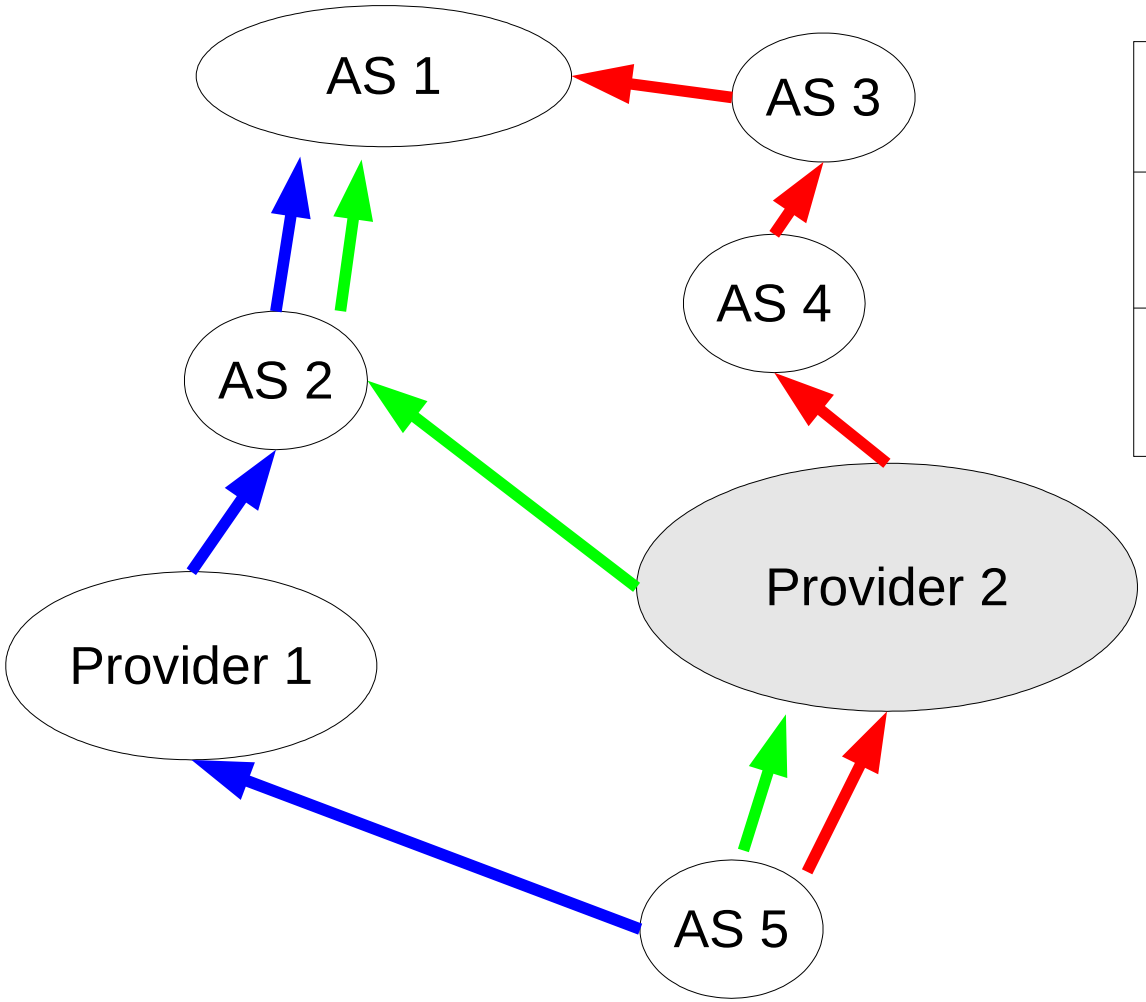
Simulation results



Open issues

- **Global impact of systematic interdomain TE by stubs :**
  - **interaction between outbound and inbound traffic ?**
  - **impact on transit ASes traffic matrix ?**
  - **perverse effects on BGP ?**
- **Is systematic interdomain TE desirable at all ?**

Transit ASes



best route for AS 5 before tweaking	→
best route for AS 5 after tweaking	→
expected best route by provider 2	→

MPLS Traffic Engineering across AS boundaries

CENSORED

Cristel Pelsser
cpe@info.ucl.ac.be
Université Catholique de Louvain
Belgium

Agenda

- Problem statement
- Constrained intra-AS path computation
- Current inter-AS routing
- Proposal for constrained inter-AS path computation
- Remaining issues

Problem statement

- Use of MPLS across AS boundaries
  - VPNs
  - Faster recovery than with BGP
  - QoS
- Requirements are formulated at the IETF
  - ccamp (a lot of new drafts planned)
- Protocol extensions to RSVP -TE already proposed at the IETF
  - Establishment of inter-AS LSPs

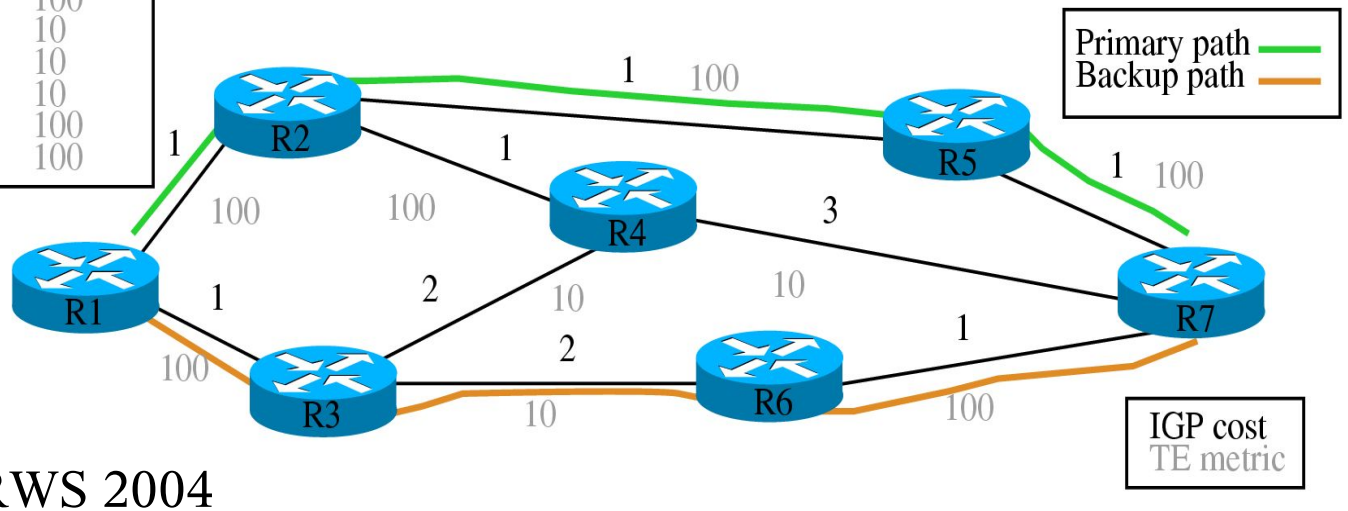
  (`draft-pelsser-rsvp-te-interdomain-lsp-00.txt`)
  - Protection of inter-AS LSPs

  (`draft-decnodder-mpls-interas-protection-01.txt`)

Constrained intra-AS path computation

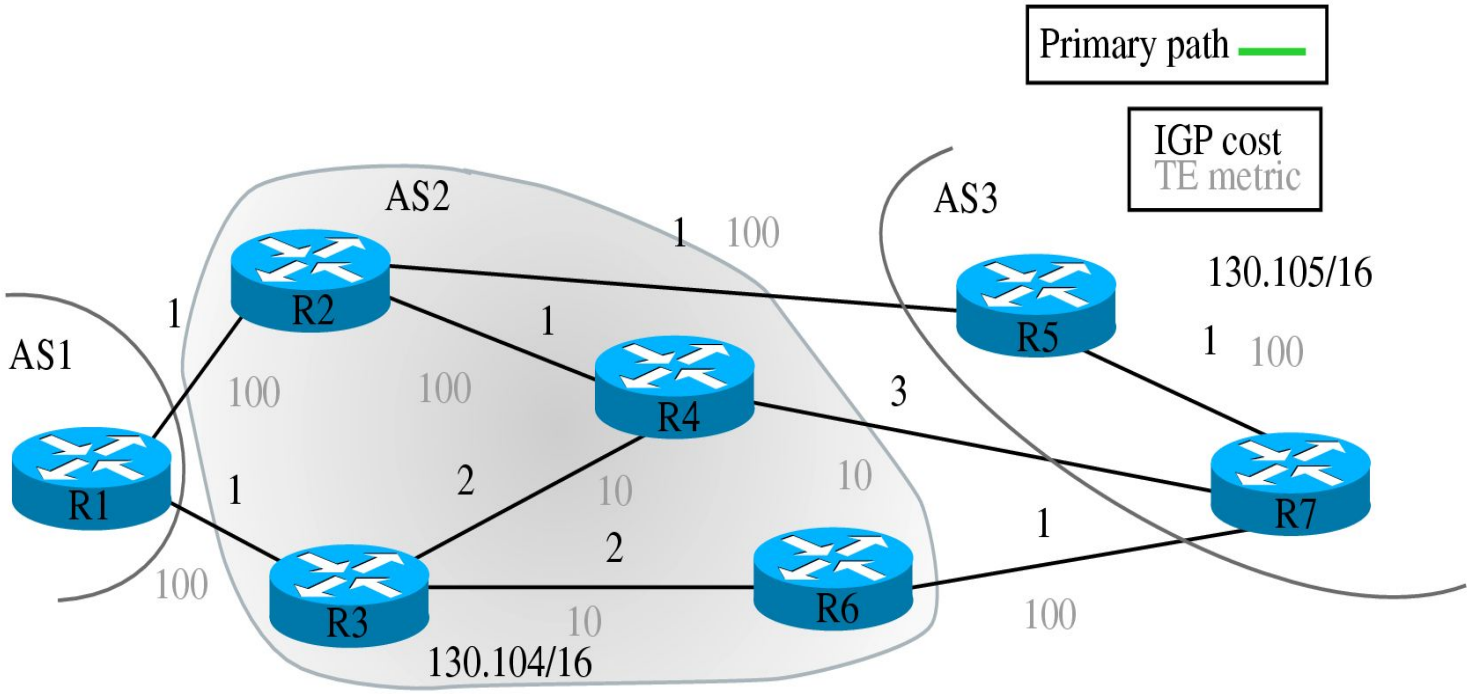
- **Each node** possesses the complete topology of its AS (No areas)
  - Link info:
    - IGP cost
    - TE info with OSPF-TE or IS-IS TE

Link	Cost	TE metric
R1–R2	1	100
R1–R3	1	100
R2–R4	1	100
R2–R5	1	100
R3–R4	2	10
R3–R6	2	10
R4–R7	3	10
R5–R7	1	100
R6–R7	1	100



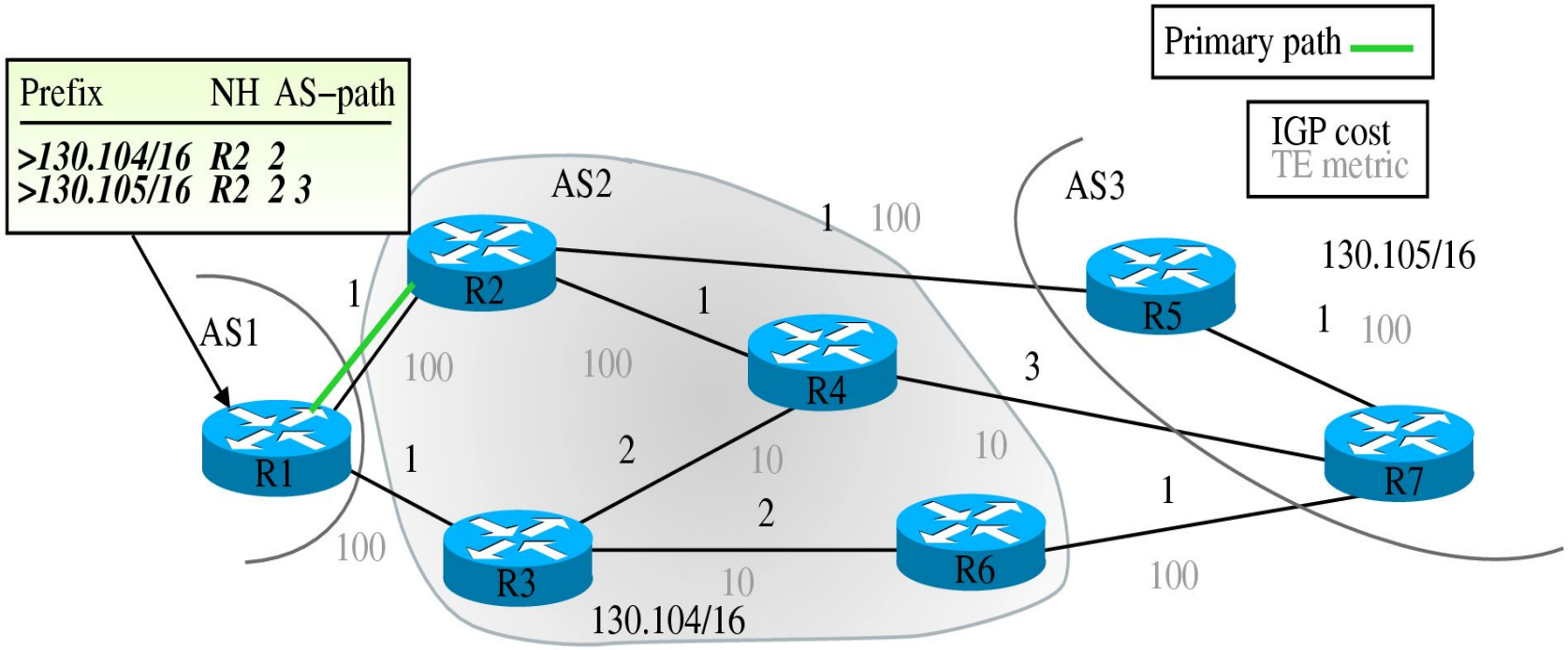
Current inter-AS routing (BGP)

- The nodes only possess **reachability information** for prefixes outside the AS



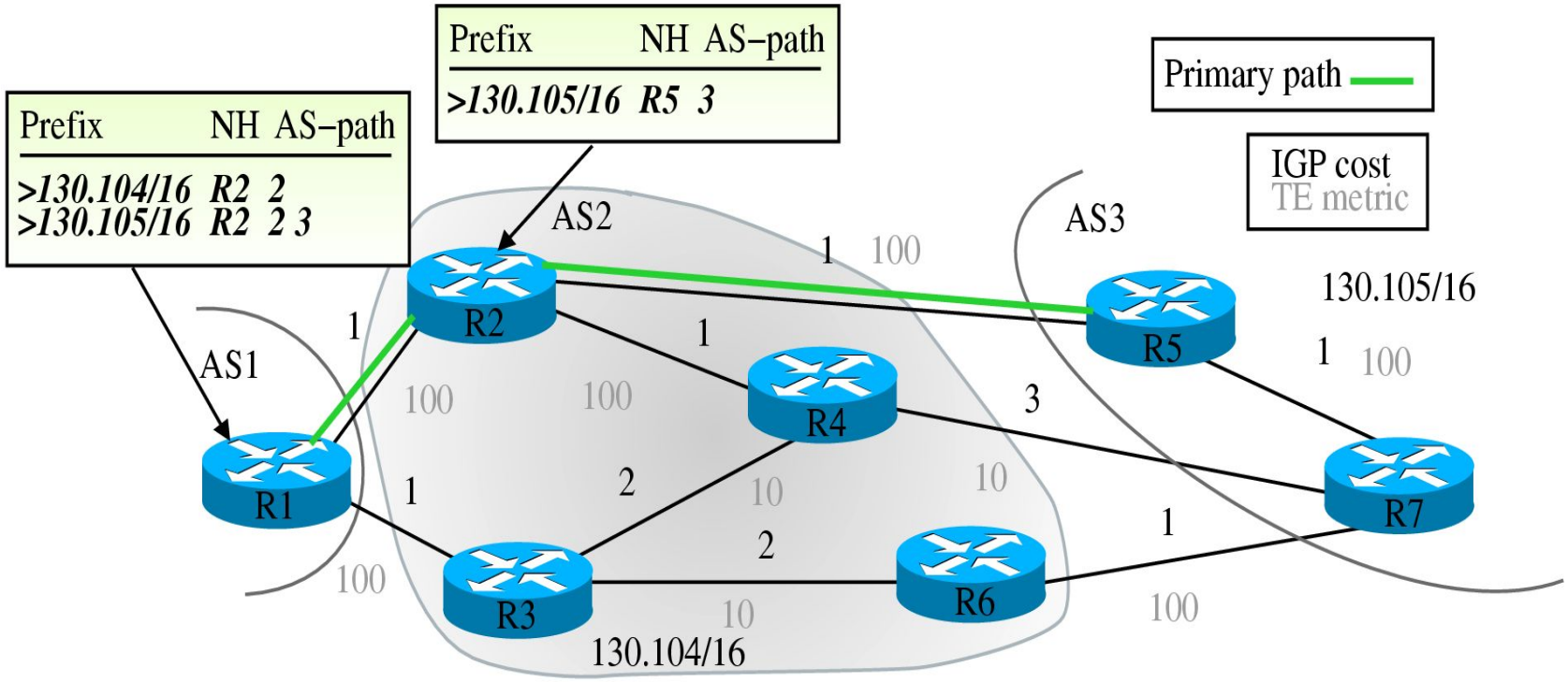
Current inter-AS routing (BGP)

- The nodes only possess **reachability information** for prefixes outside the AS



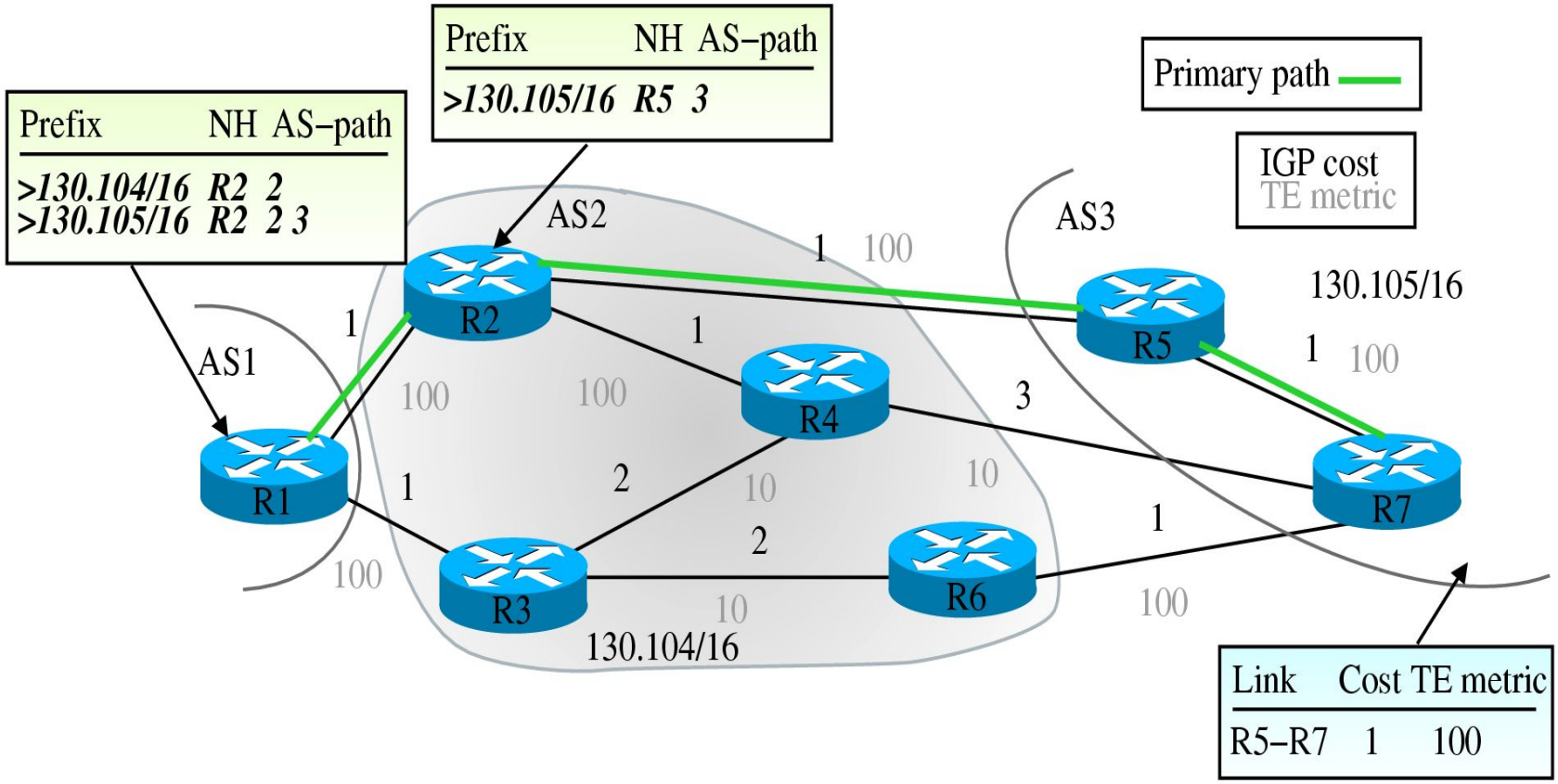
Current inter-AS routing (BGP)

- The nodes only possess **reachability information** for prefixes outside the AS



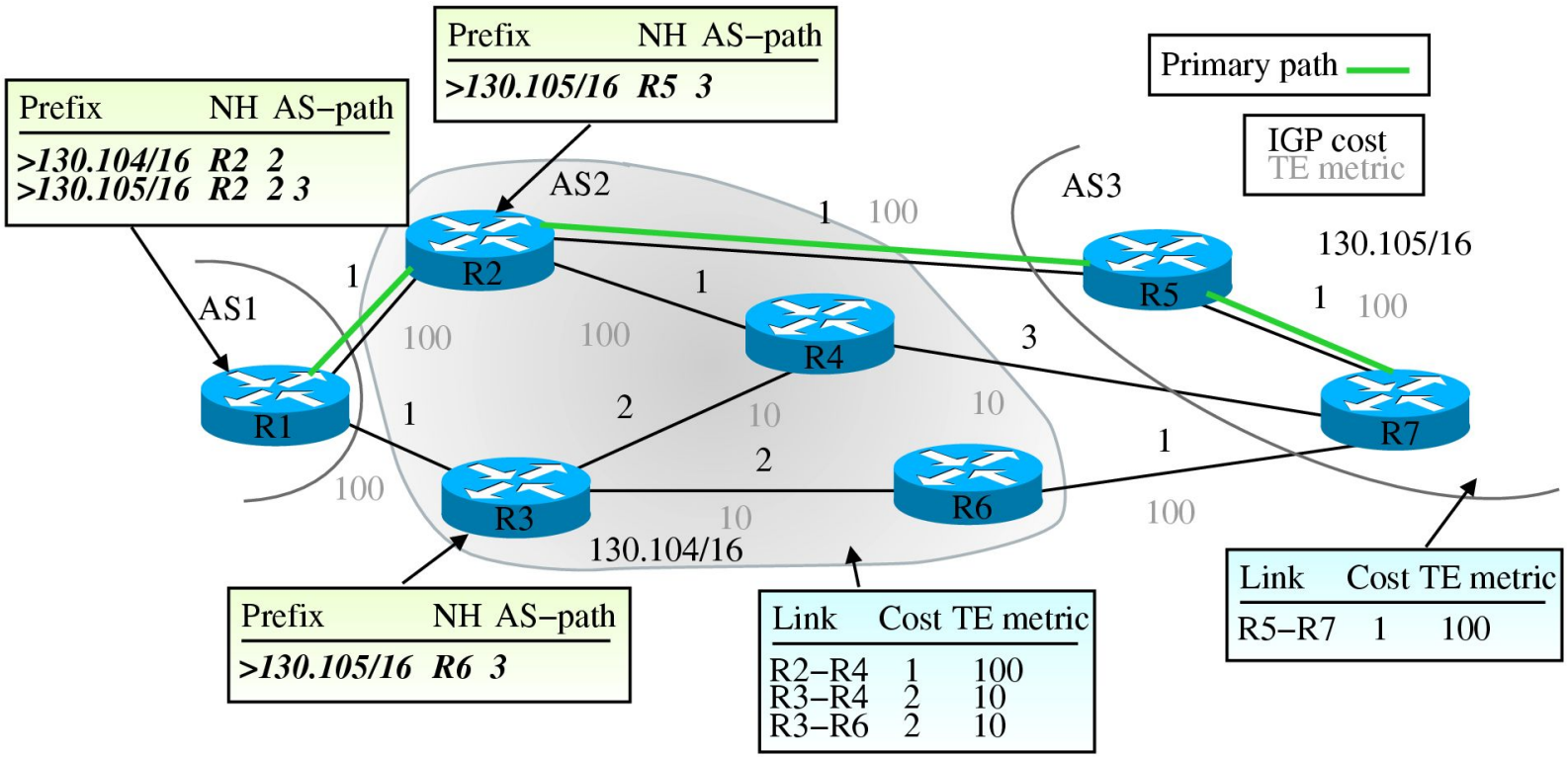
Current inter-AS routing (BGP)

- The nodes only possess **reachability information** for prefixes outside the AS



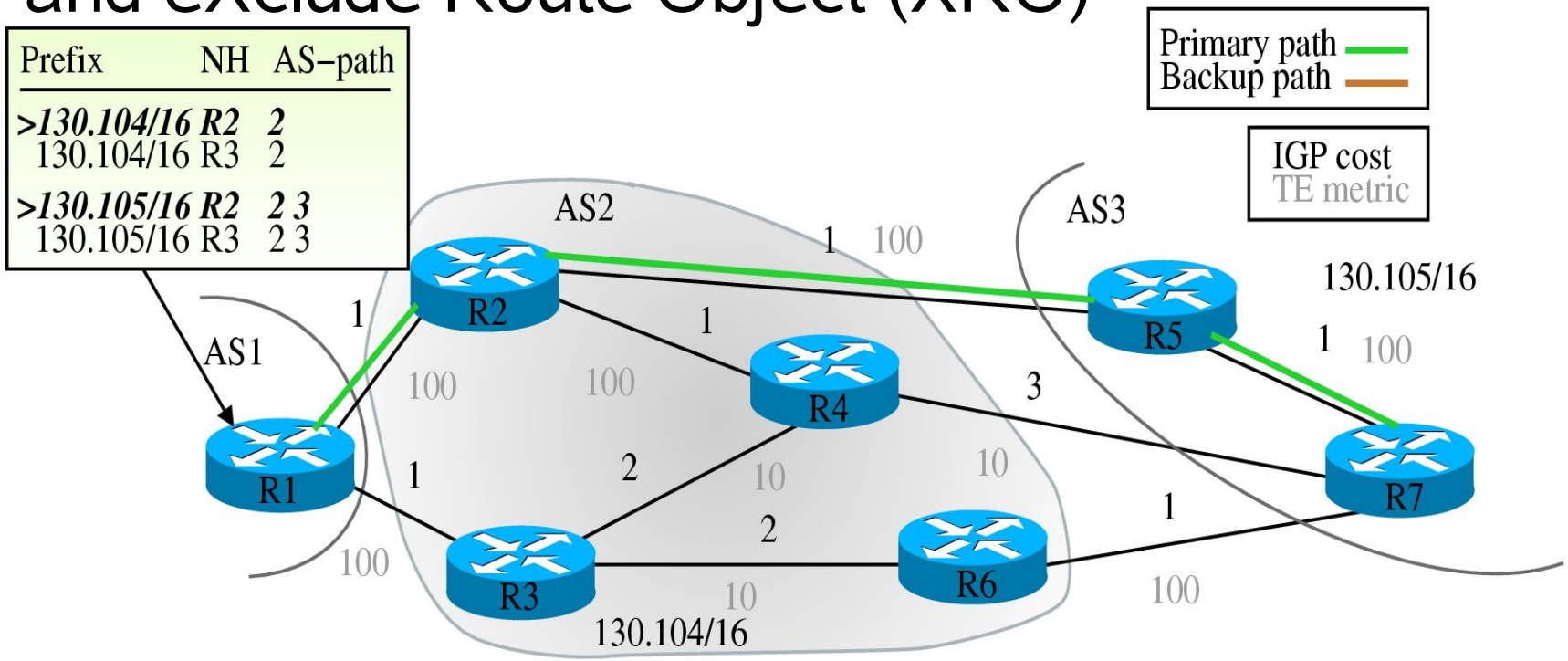
Current inter-AS routing (BGP)

- The nodes only possess **reachability information** for prefixes outside the AS
- Alternate path through R3 never used



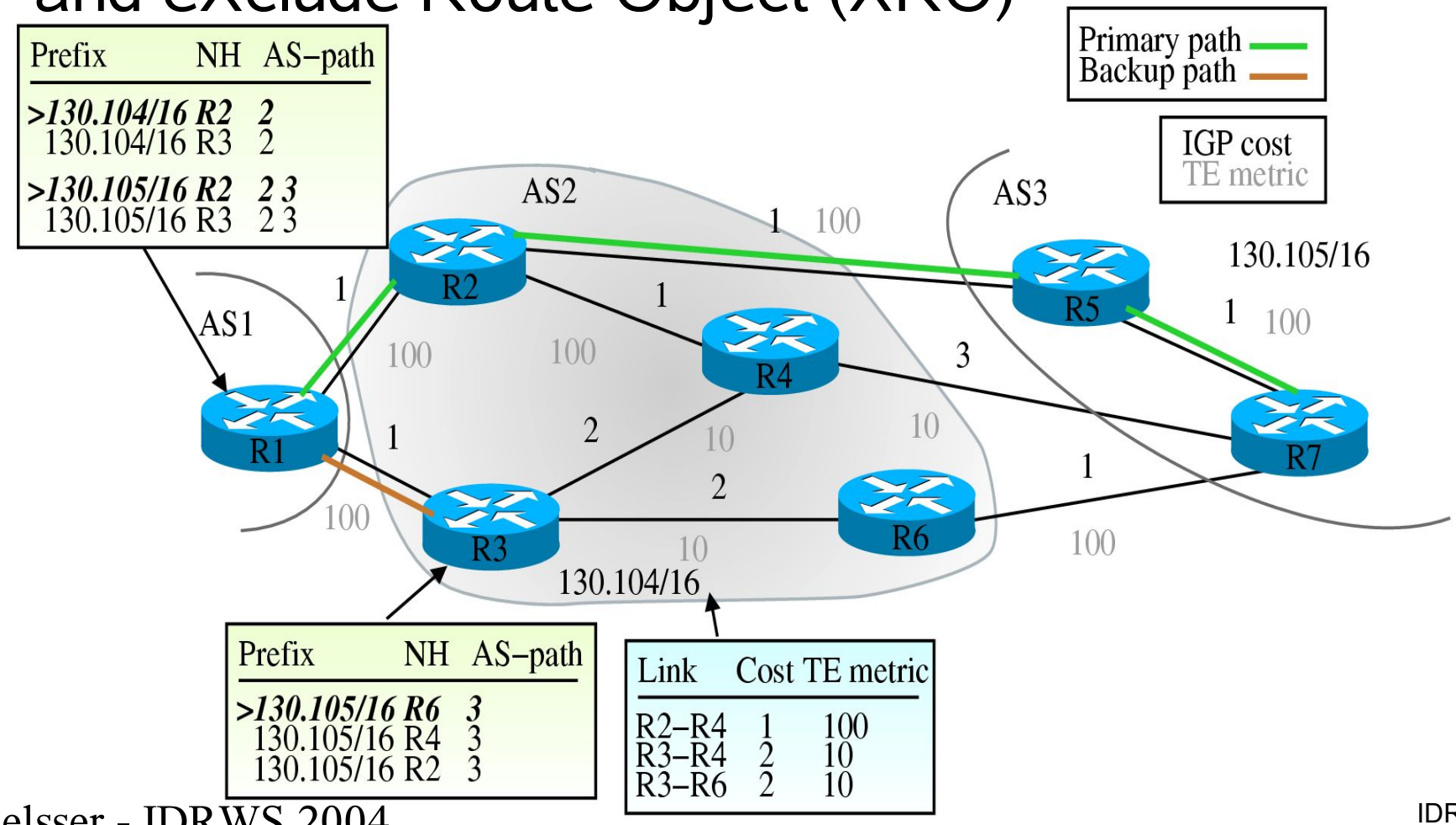
Constrained inter-AS path computation : Proposal

- Compute disjoint path based on **local** Adj-RIB-In and eXclude Route Object (XRO)



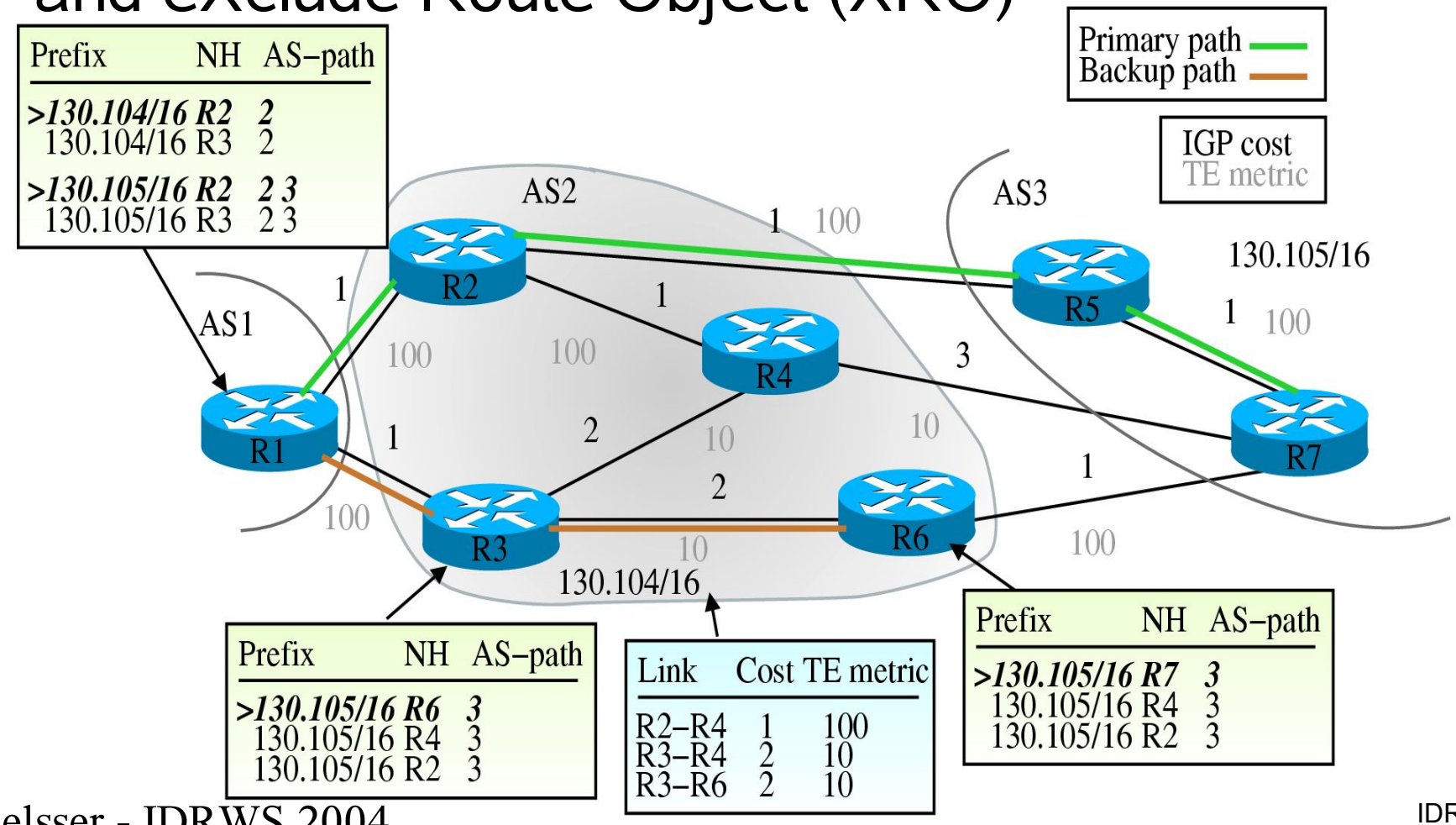
Constrained inter-AS path computation : Proposal

- Compute disjoint path based on **local** Adj-RIB-In and eXclude Route Object (XRO)



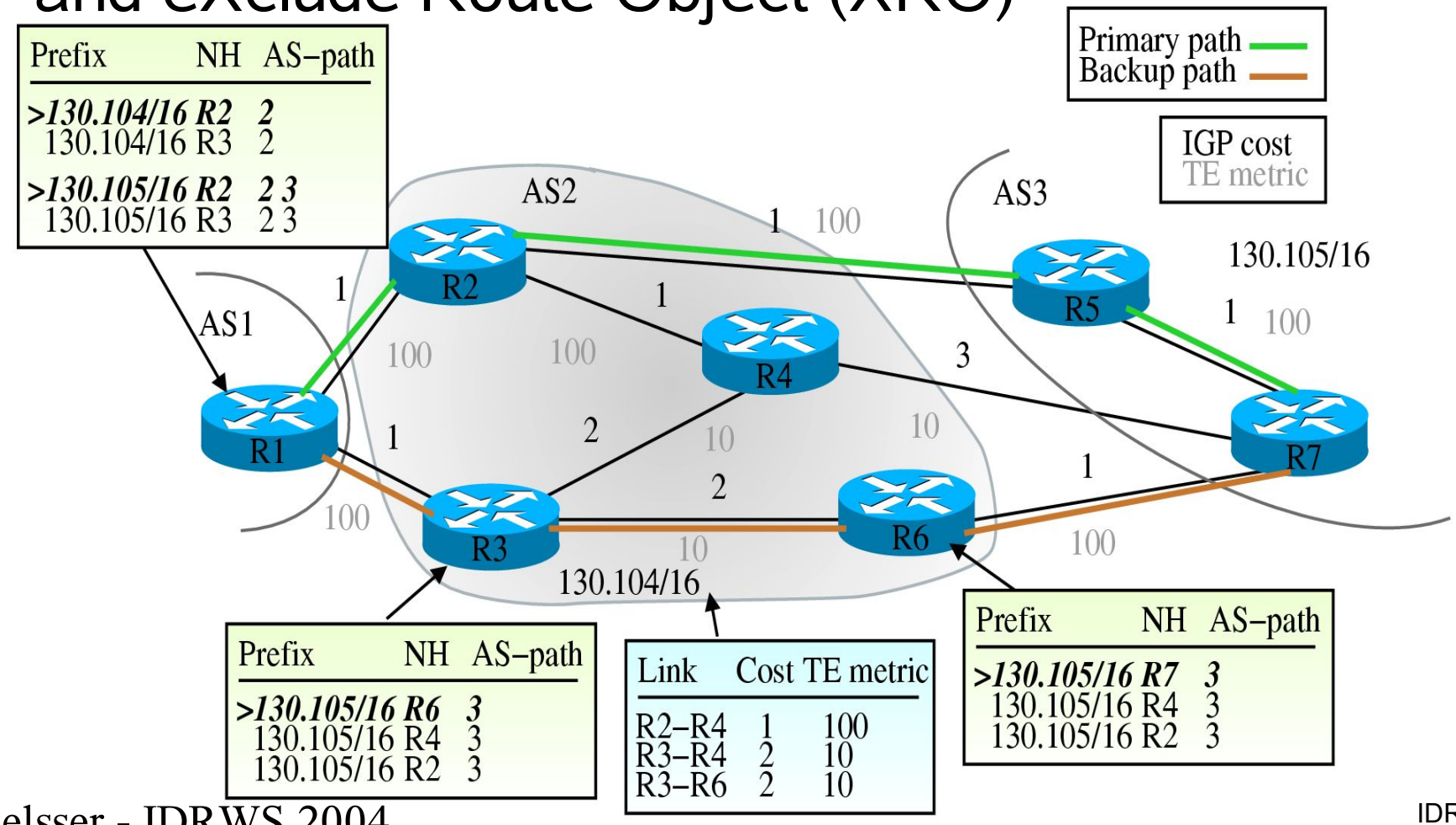
Constrained inter-AS path computation : Proposal

- Compute disjoint path based on **local** Adj-RIB-In and eXclude Route Object (XRO)



Constrained inter-AS path computation : Proposal

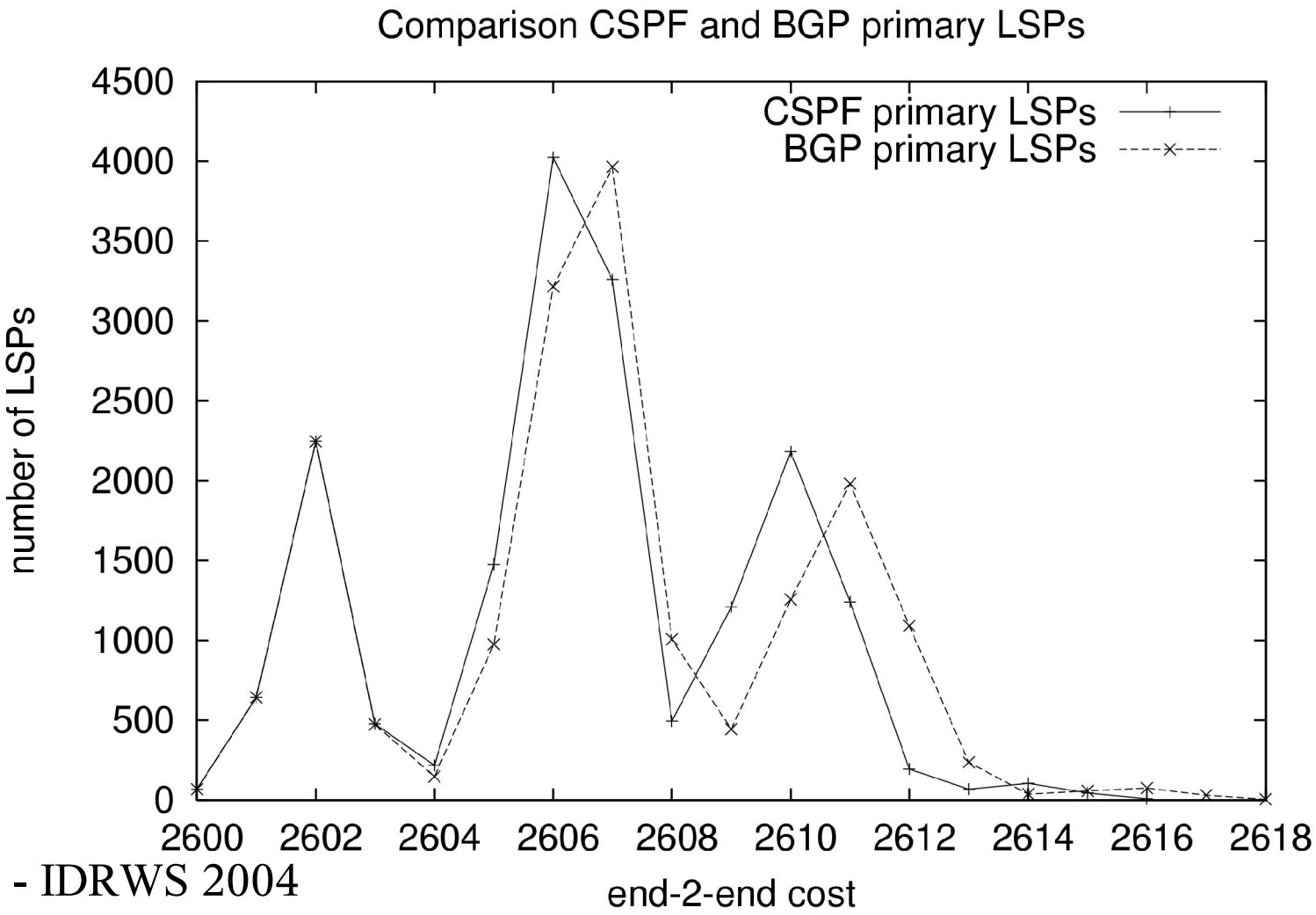
- Compute disjoint path based on **local** Adj-RIB-In and eXclude Route Object (XRO)



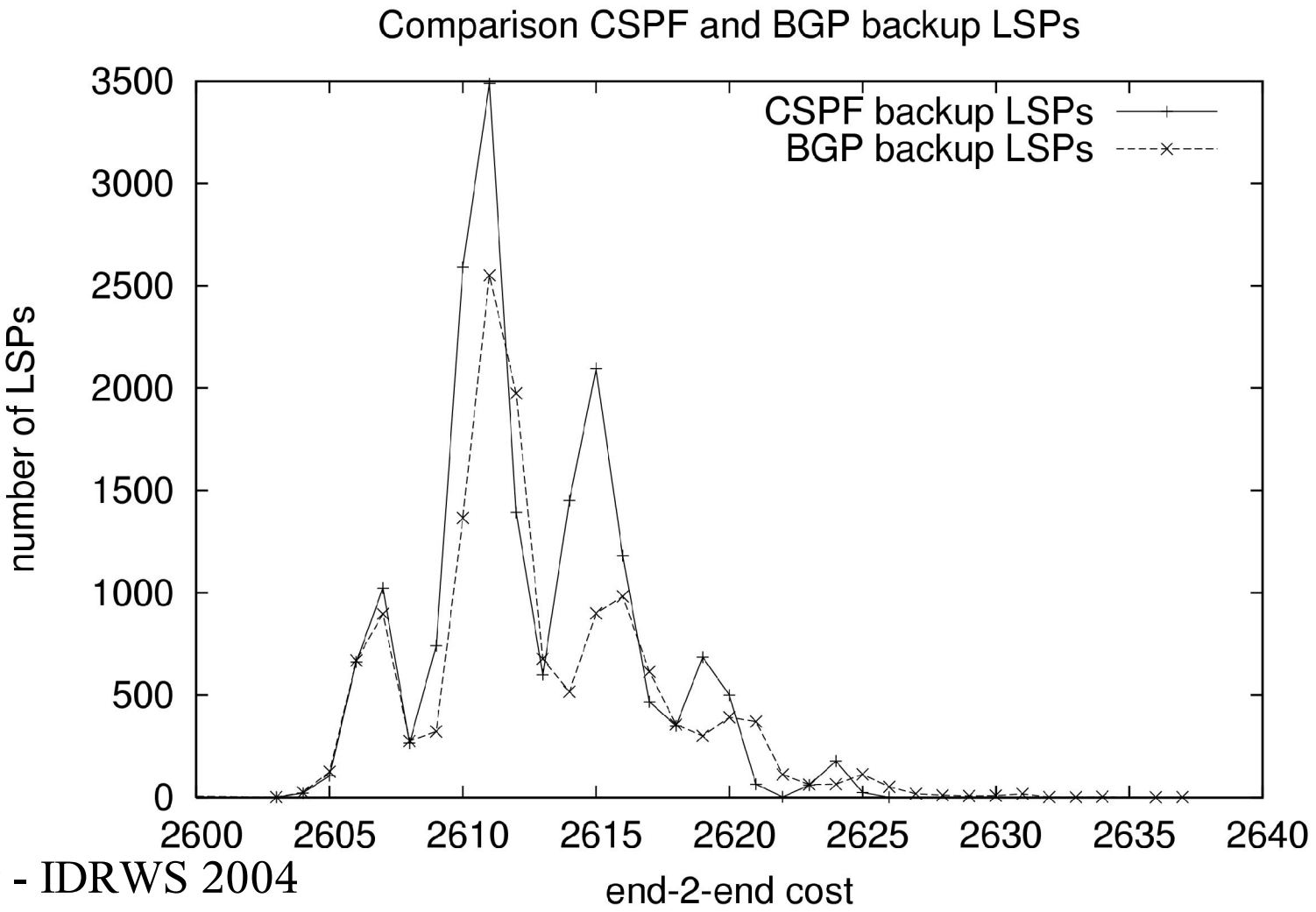
Preliminary results

- Topology with :
  - 20 transits composed of 50 nodes
  - 190 stubs (all possible combinations of dual-homed stubs)
- Customer-provider policies between transit and stubs
- Constraint : node protection
- Optimise end-2-end cost (ex: delay)
- Backtracking (cranckback) when no path available for the required constraint
- No incremental establishment of LSPs
  - available resources are not updated after each LSP establishment

Preliminary results



Preliminary results



Remaining issues

- Work on heuristics for the choice of alternative next-hops (NH)
- All possible NH are not necessarily in the Adj-RIB-In of the local router

  Full-mesh of iBGP session:
  - All routers only know the best route selected by the other routers in the iBGP mesh

  Route-Reflectors (RR):
  - Clients only know the route selected by their RR
  - The RR should make the choice for its clients
- Work on link-state inter-AS routing protocols?

Conclusion

- **Distributed** disjoint path computation possible based on
  - Adj-RIB-Ins

  and
  - eXclude Route Object (XRO)

    (`draft-ietf-ccamp-rsvp-te-exclude-route-01.txt`)
- Applicable for
  - Protection, load-balancing and TE
    - The ISPs can choose the AS-path (difficult with BGP)
  - Establishement of **constrained** inter-AS primary LSP
    - bandwidth, delay, link affinities constraints

BGP Wedgies: Bad Policy Interactions that Cannot be Debugged

Timothy G. Griffin

Intel Research,

Cambridge UK

tim.griffin@intel.com

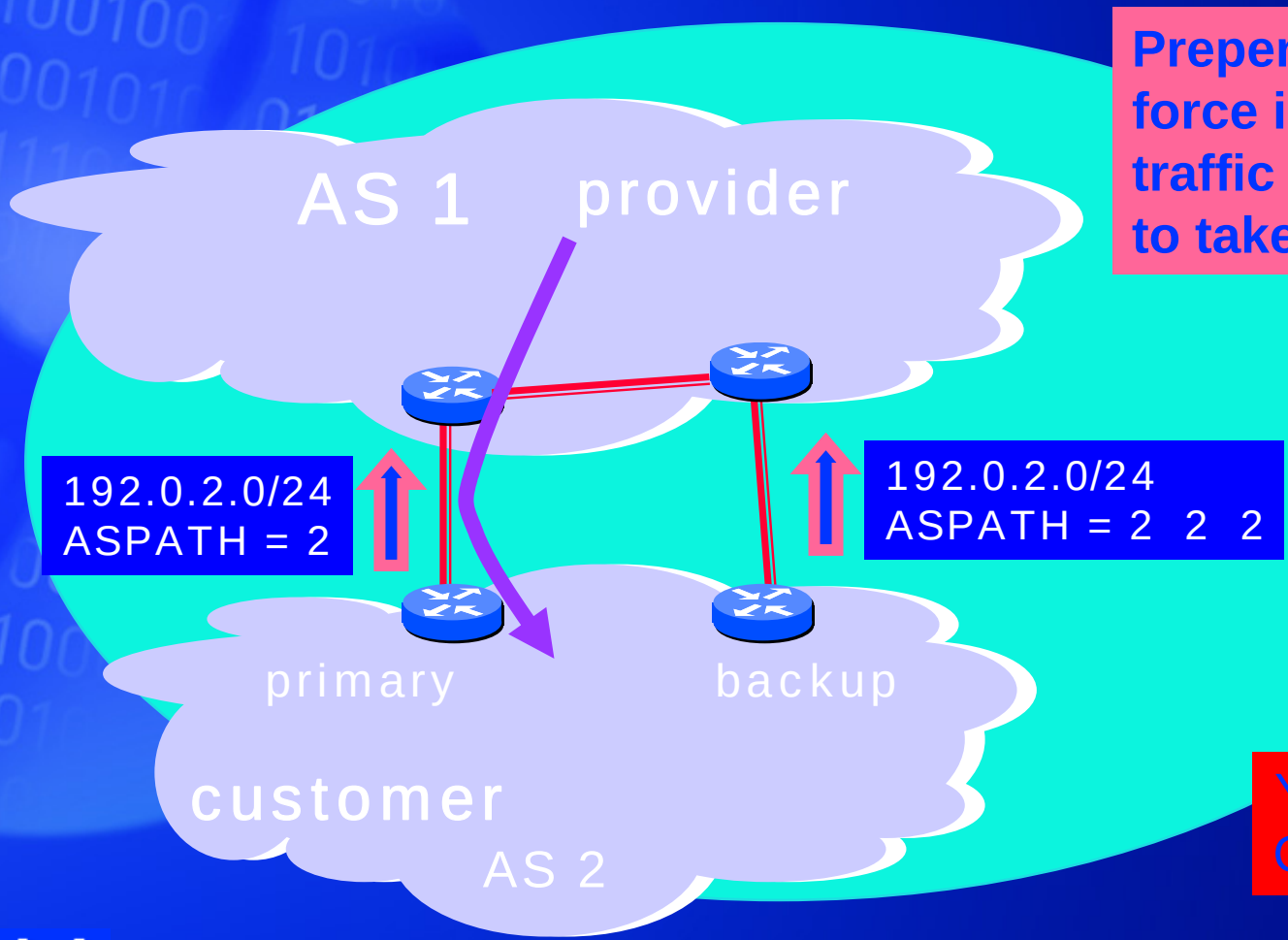
http://www.cambridge.intel-research.net/~tgriffin/

IDRWS II

Amsterdam

May 1-2, 2004

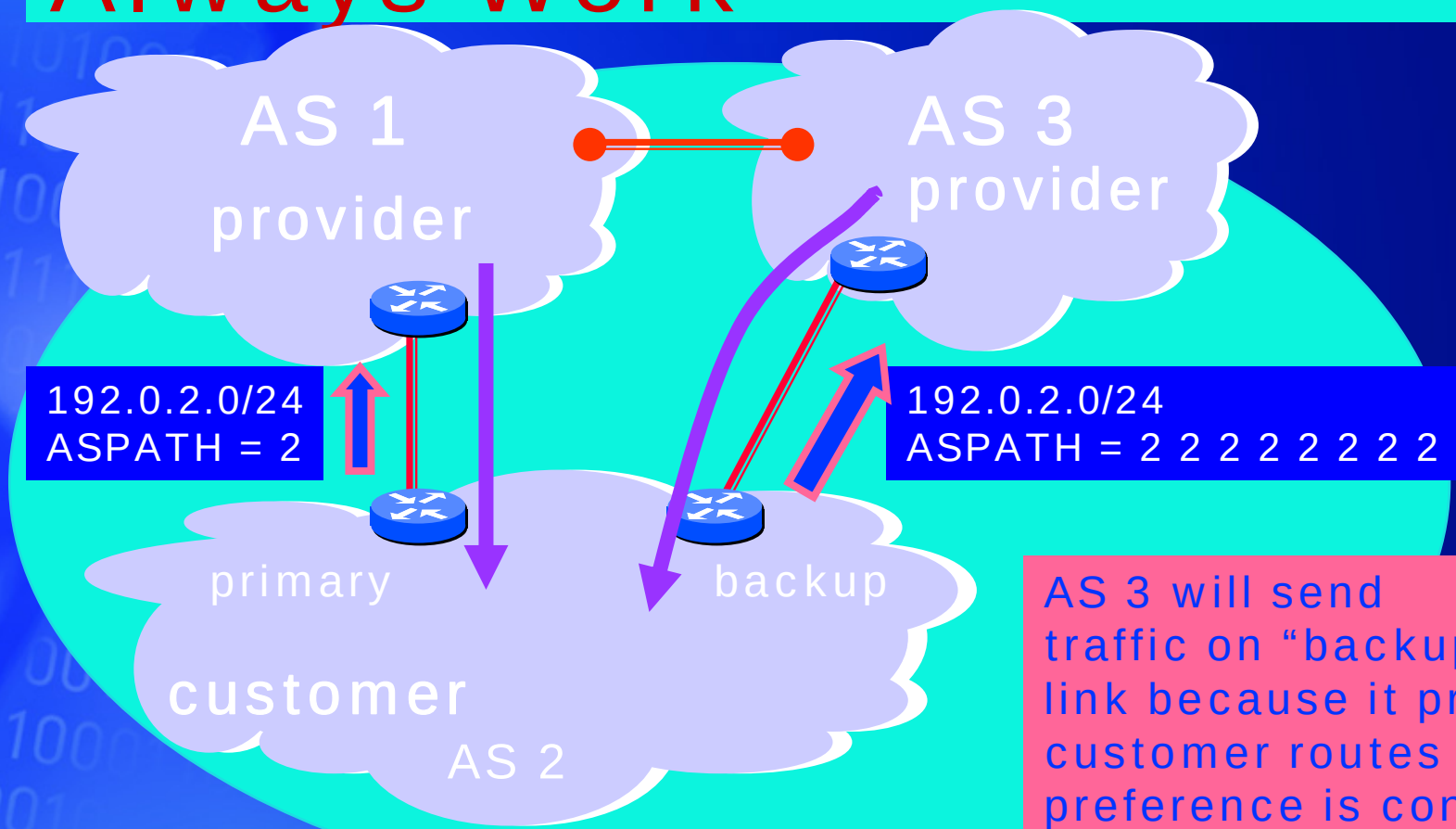
Shedding Inbound Traffic with ASPATH Prepending



Prepending will (usually) force inbound traffic from AS 1 to take primary link

Yes, this is a Glorious Hack ...

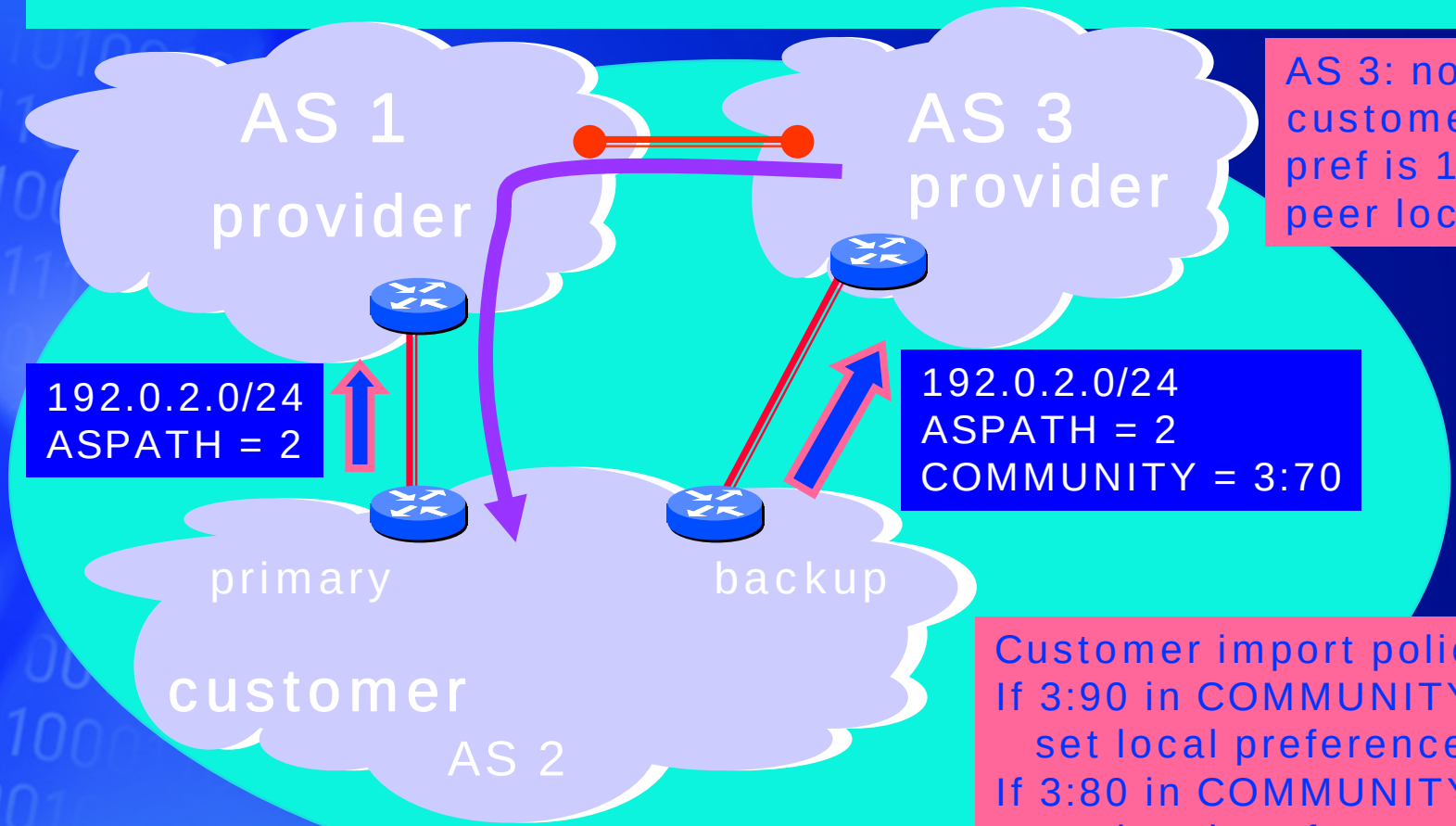
… But Padding Does Not Always Work



AS 3 will send traffic on “backup” link because it prefers customer routes and local preference is considered before ASPATH length!

Padding in this way is often used as a form of load balancing

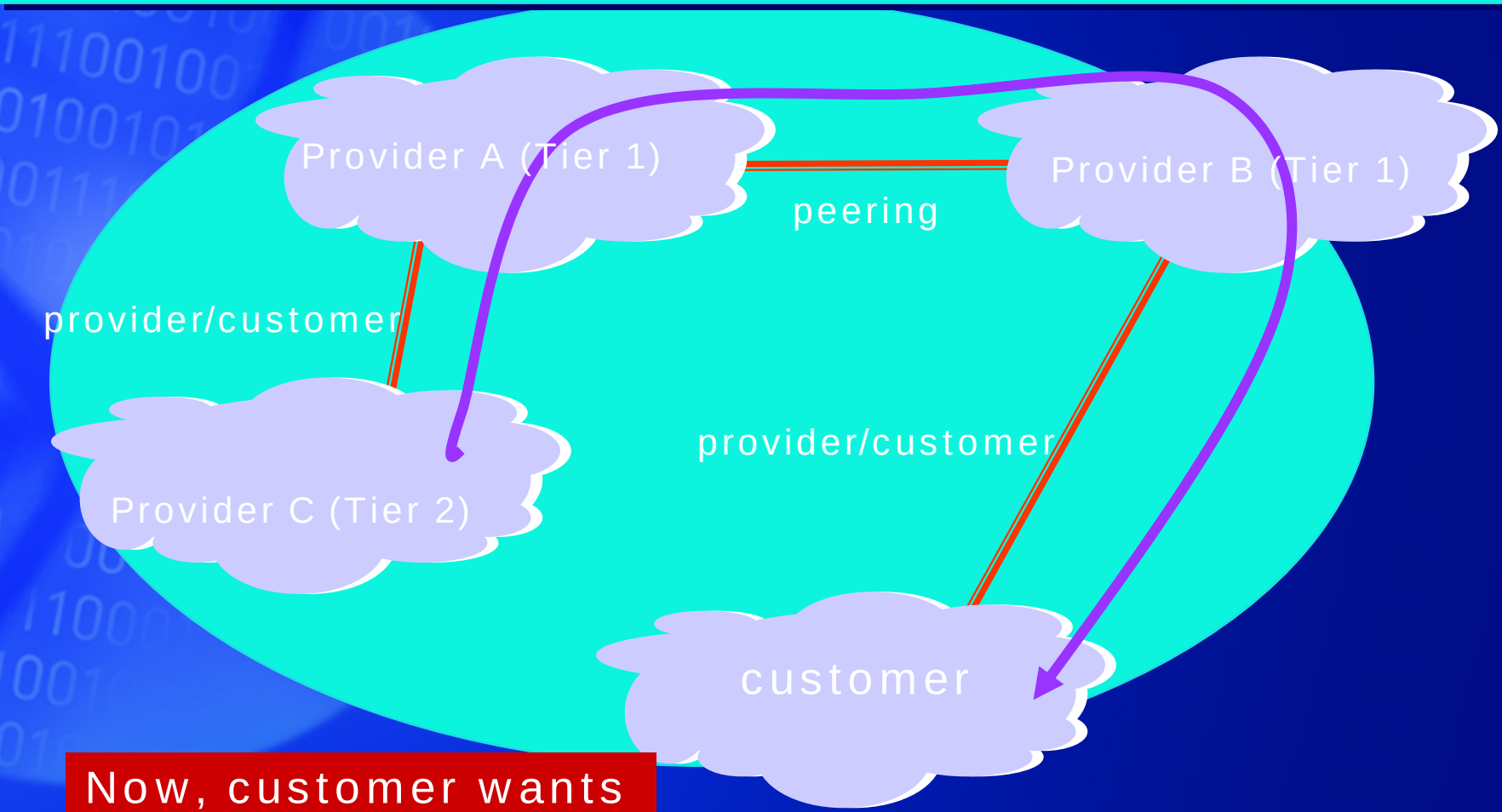
COMMUNITIES to the Rescue!



AS 3: normal customer local pref is 100, peer local pref is 90

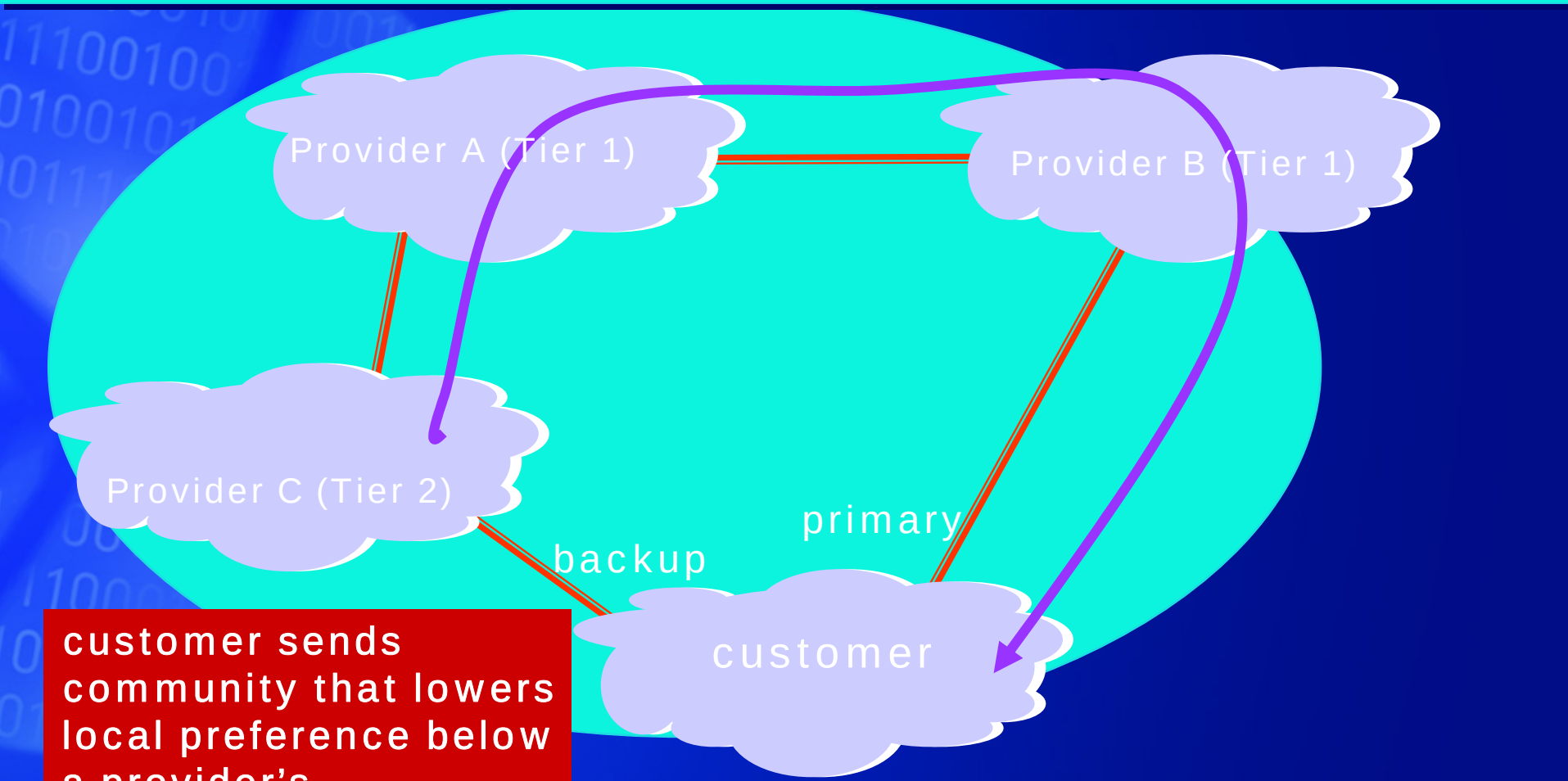
Customer import policy at AS 3:
If 3:90 in COMMUNITY then
set local preference to 90
If 3:80 in COMMUNITY then
set local preference to 80
If 3:70 in COMMUNITY then
set local preference to 70

Don’t Celebrate Just Yet....



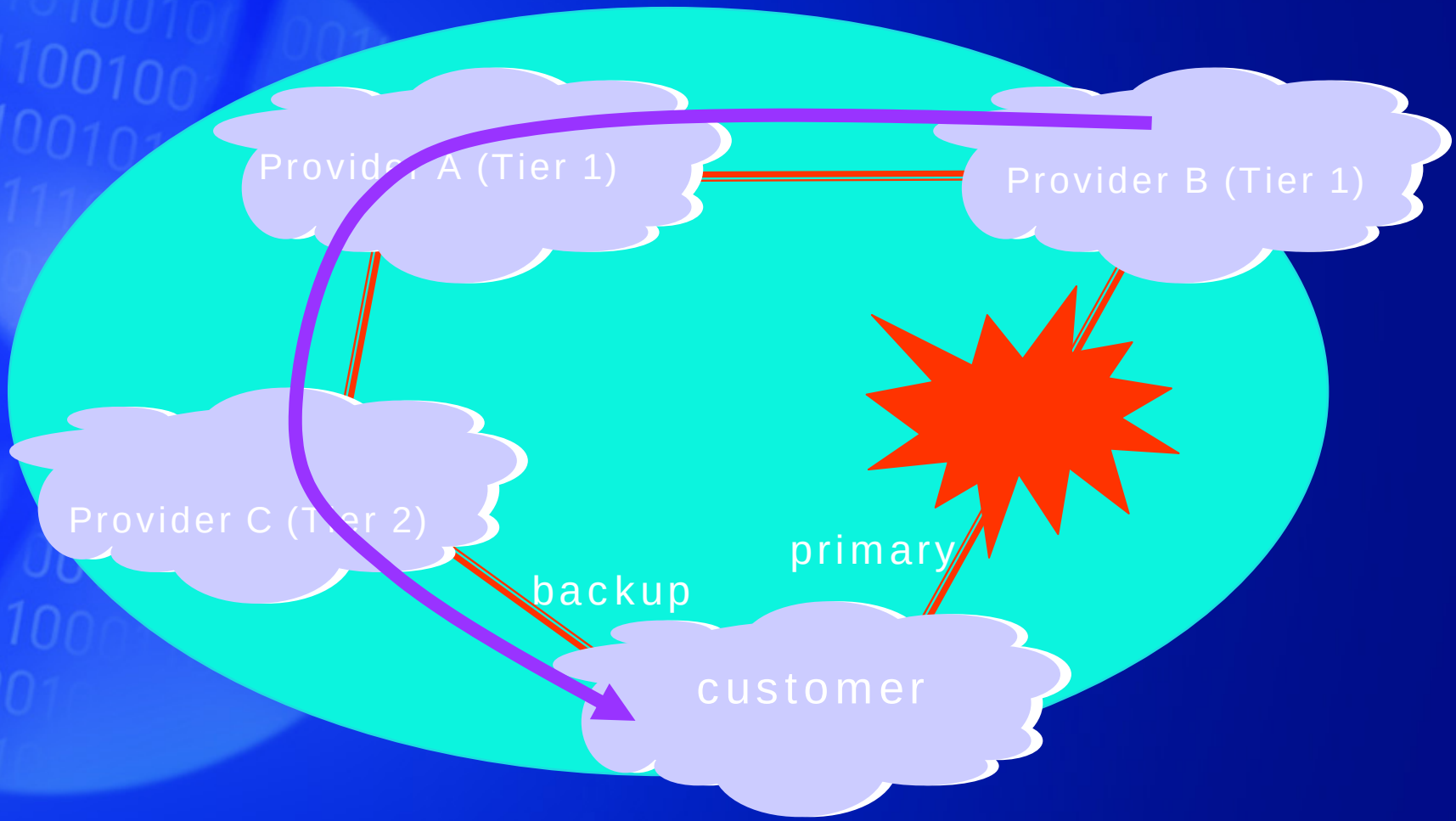
Now, customer wants a backup link to C....

Customer installs a “backup link” …



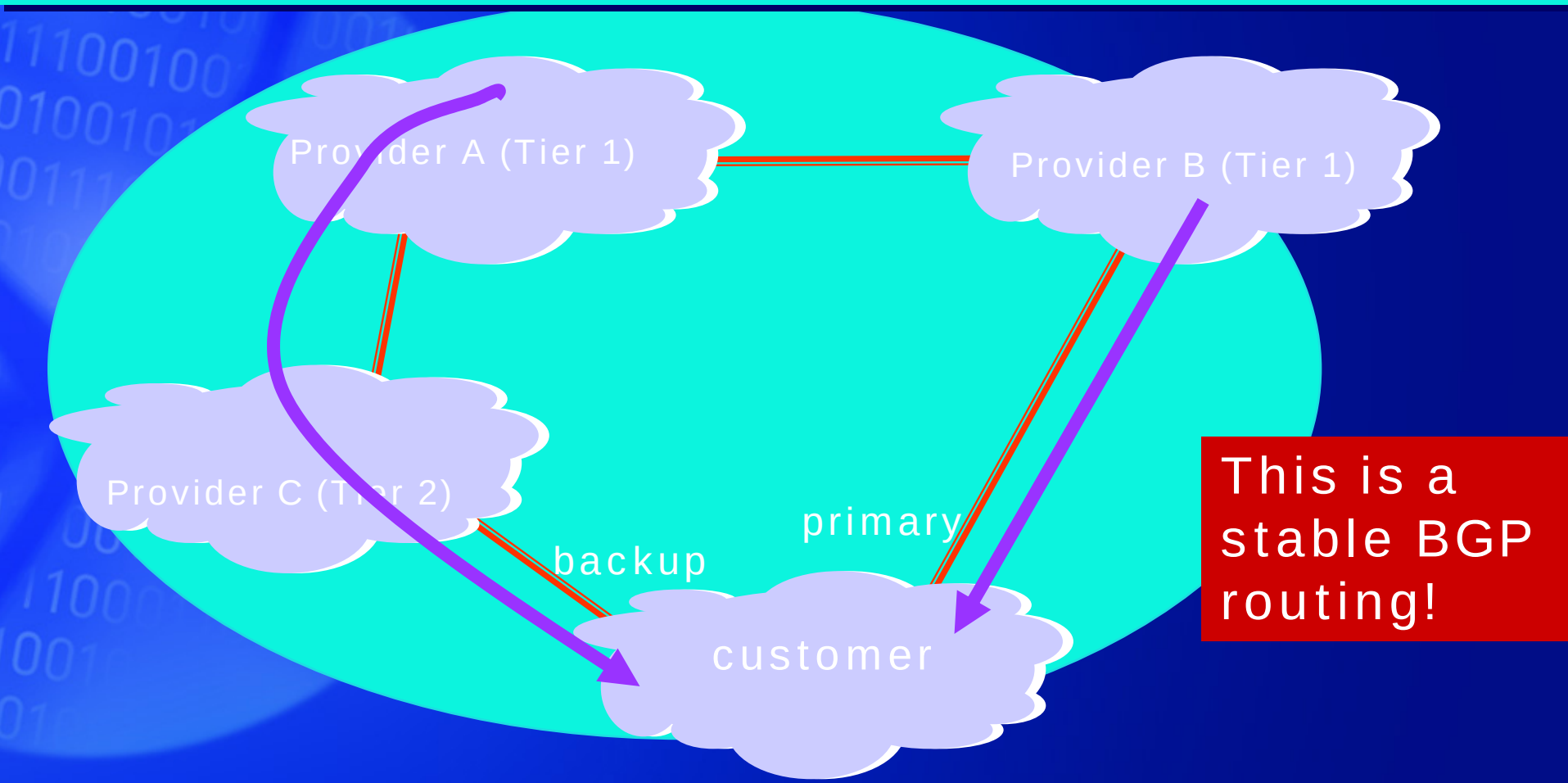
customer sends community that lowers local preference below a provider’s

Disaster Strikes!



customer is happy that backup was installed ...

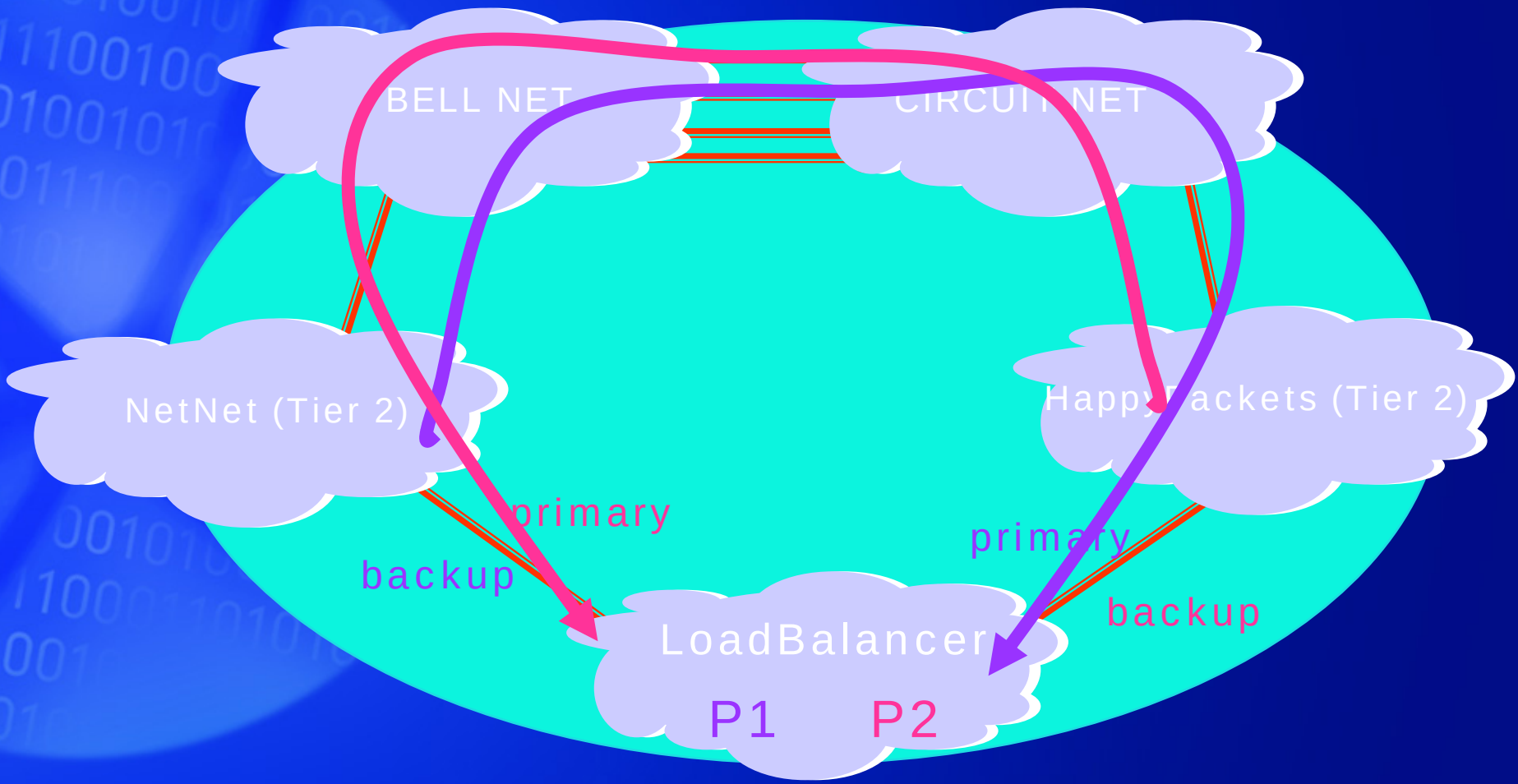
The primary link is repaired, yet routing does repair!



This is a stable BGP routing!

One “solution” --- reset BGP session on backup link!

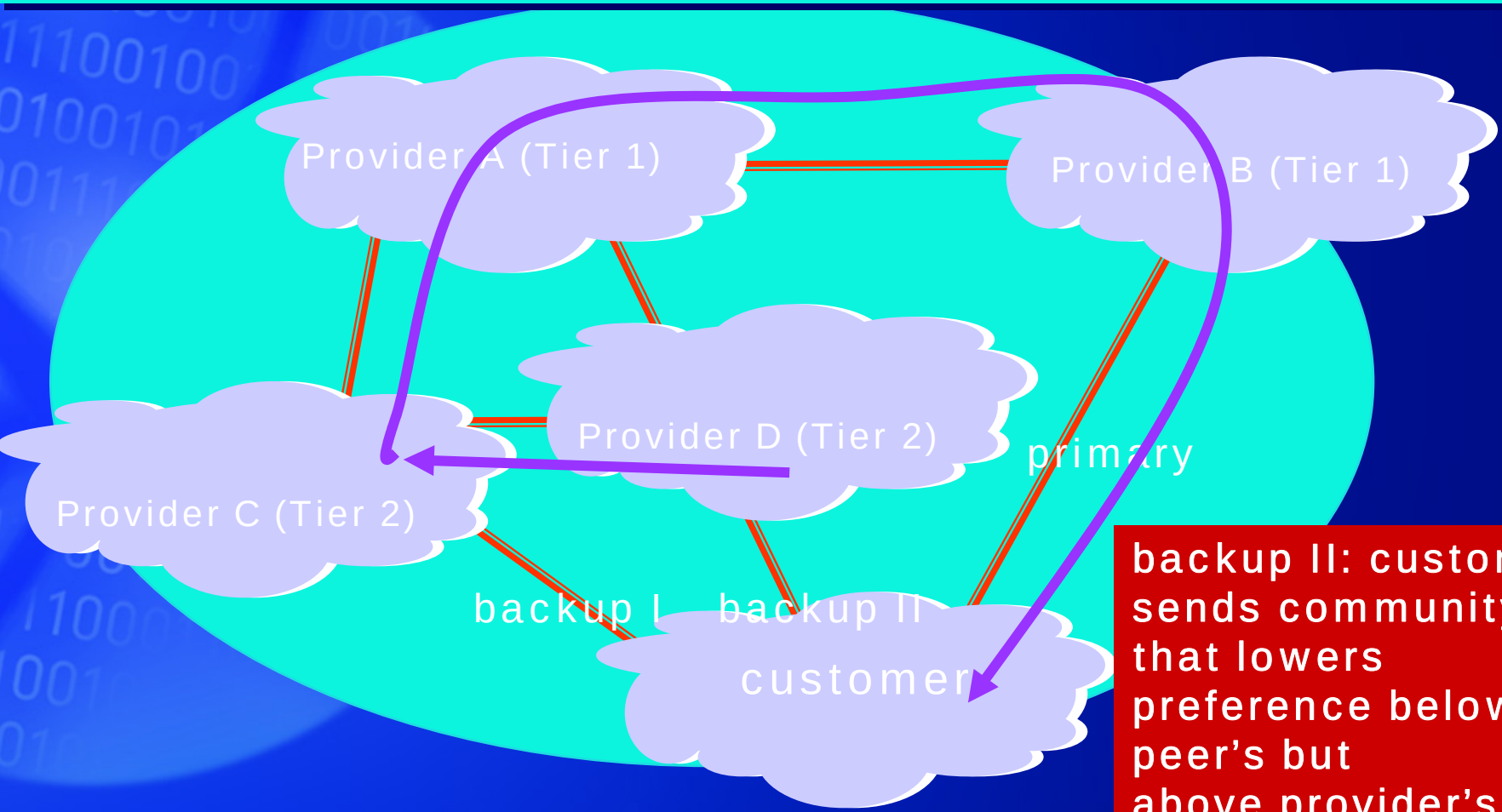
Ouch


BELL NET
CIRCUIT NET
NetNet (Tier 2)
Happy Packets (Tier 2)
primary
backup
primary
backup
LoadBalancer
P1
P2

What the heck is going on?

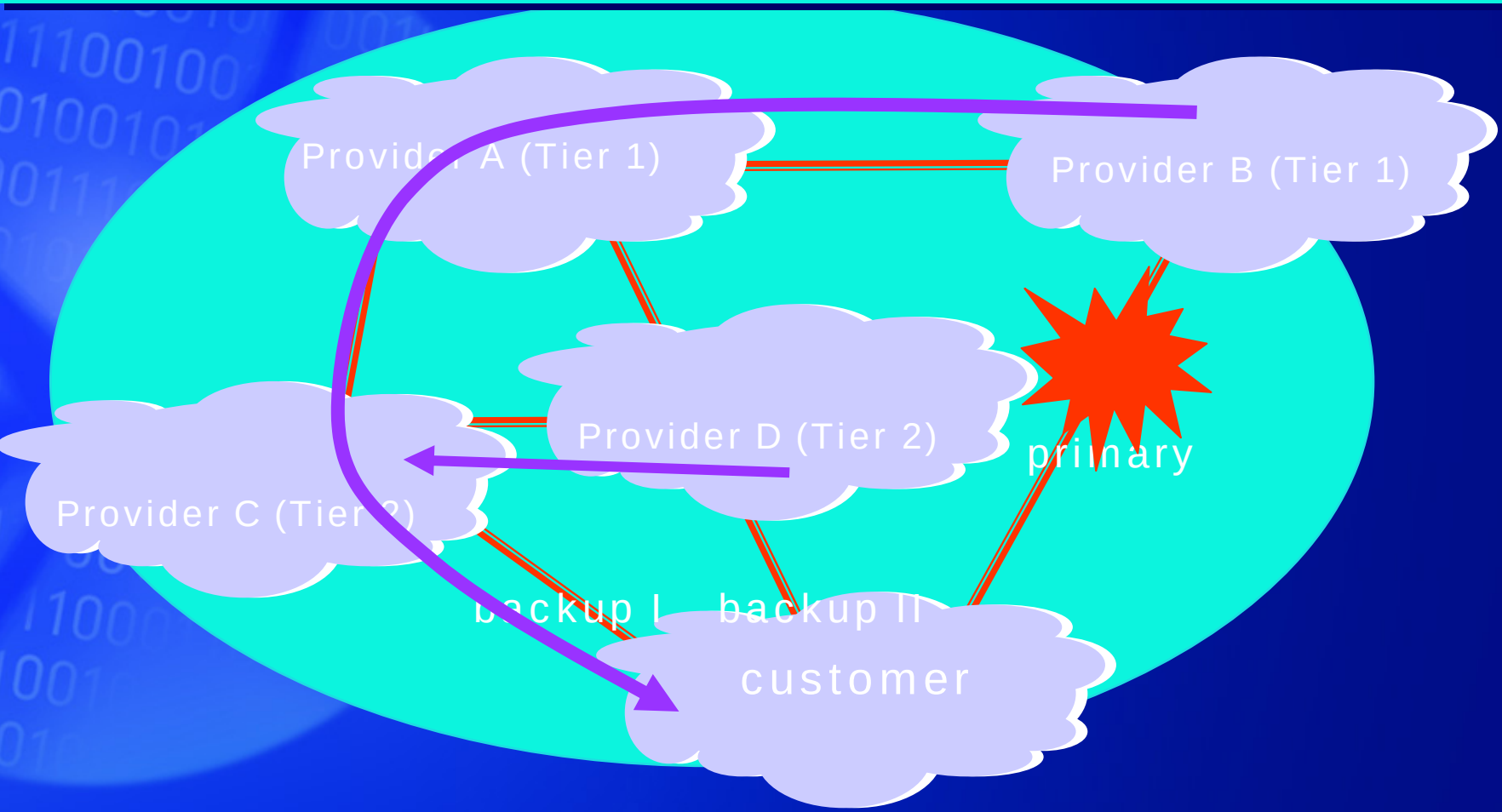
- There is no guarantee that a BGP configuration has a unique routing solution.
  - When multiple solutions exist, the (unpredictable) order of updates will determine which one is wins.
- There is no guarantee that a BGP configuration has any solution!
  - And checking configurations NP-Complete
- Complex policies (weights, communities setting preferences, and so on) increase chances of routing anomalies.
  - … yet this is the current trend!

More fun with communities ....

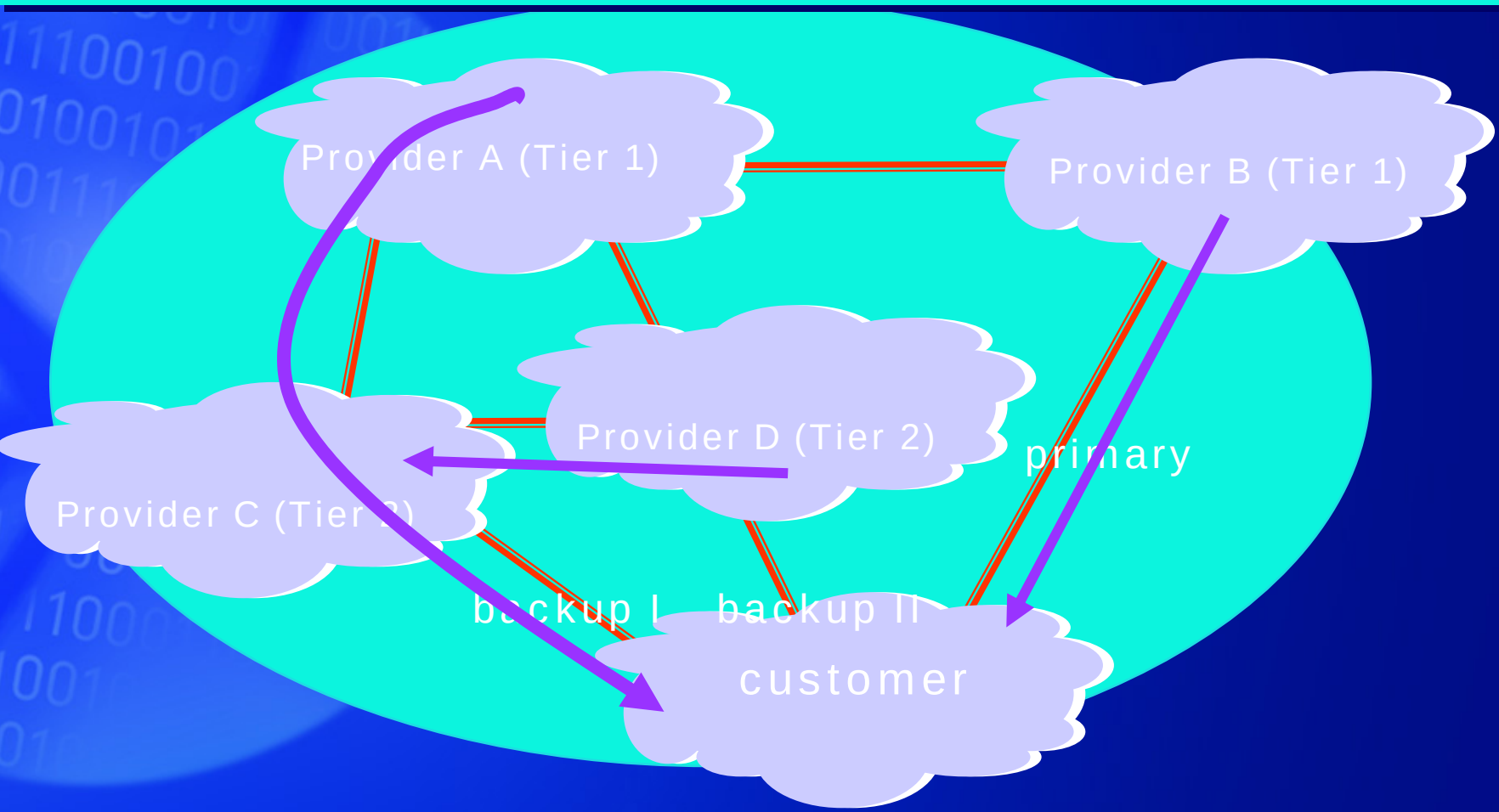


backup II: customer sends community that lowers preference below peer's but above provider's

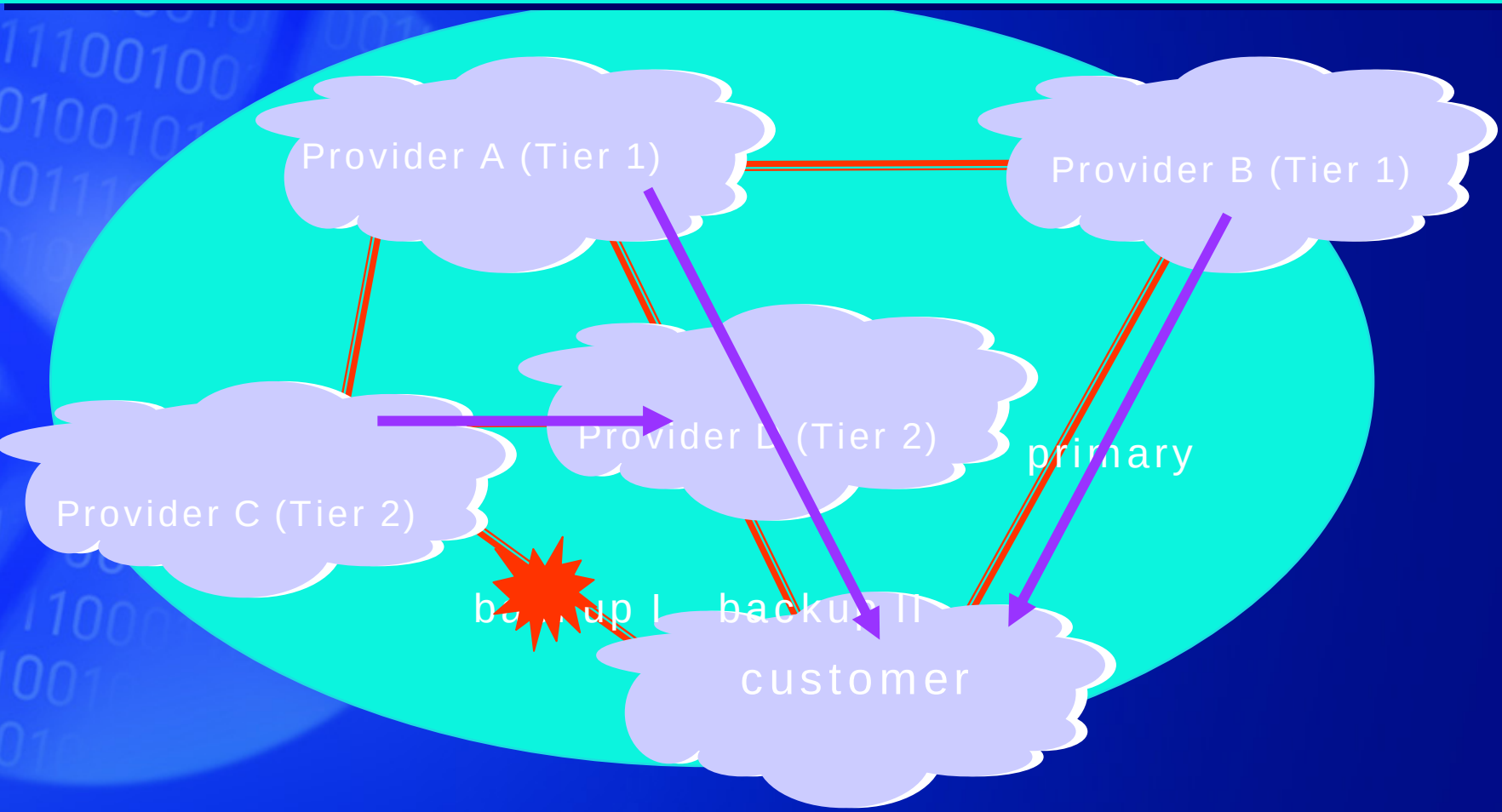
Primary goes down!



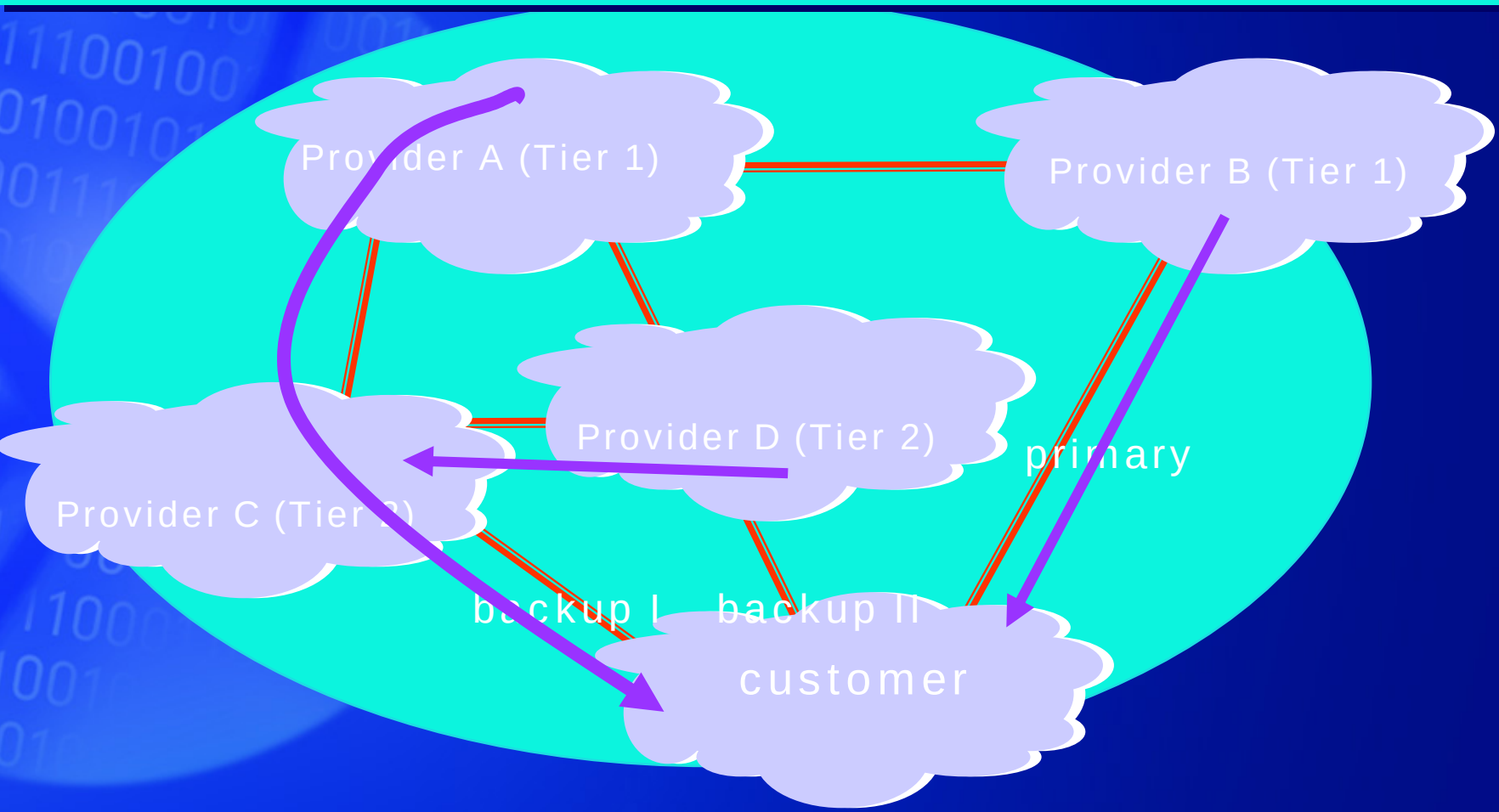
Primary repaired



Reset Backup I session? First, take it down....



Now Bring it up ...



"solution" --- reset BGP session on BOTH backup links simultaneously!

BGP Wedgie

- BBP policies make sense locally
- Sum of local policies allows multiple solutions
- Some solutions are consistent with intended policies, and some are not
- When unintended solutions are installed, no single AS has enough global knowledge to effectively debug the problem

BACKUP SLIDES

What Problem is BGP Solving?

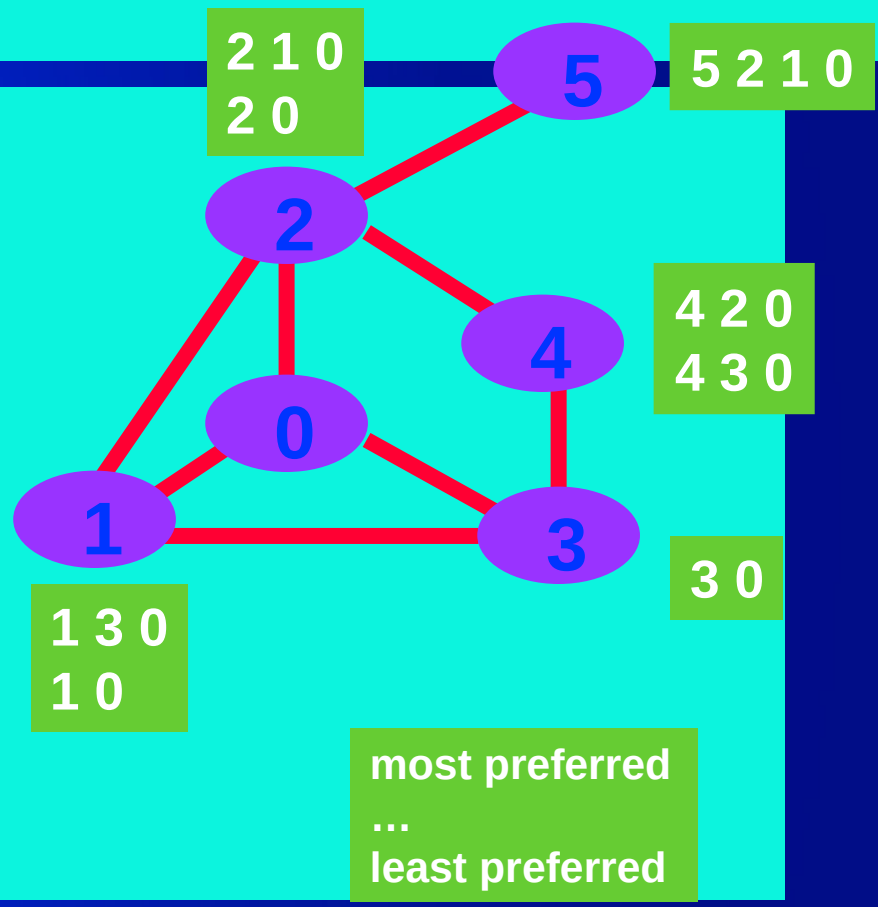
Underlying problem	Distributed means of computing a solution.
Shortest Paths	RIP, OSPF, IS-IS
Stable Paths	BGP

[Griffin, Shepherd, Wilfong. ToN 2002]

An instance of the *Stable Paths Problem* (SPP)

- A graph of nodes and edges,
- Node 0, called *the origin*,
- For each non-zero node, a set or permitted paths to the origin. This set always contains the "null path".
- A ranking of permitted paths at each node. Null path is always least preferred. (Not shown in diagram)

1

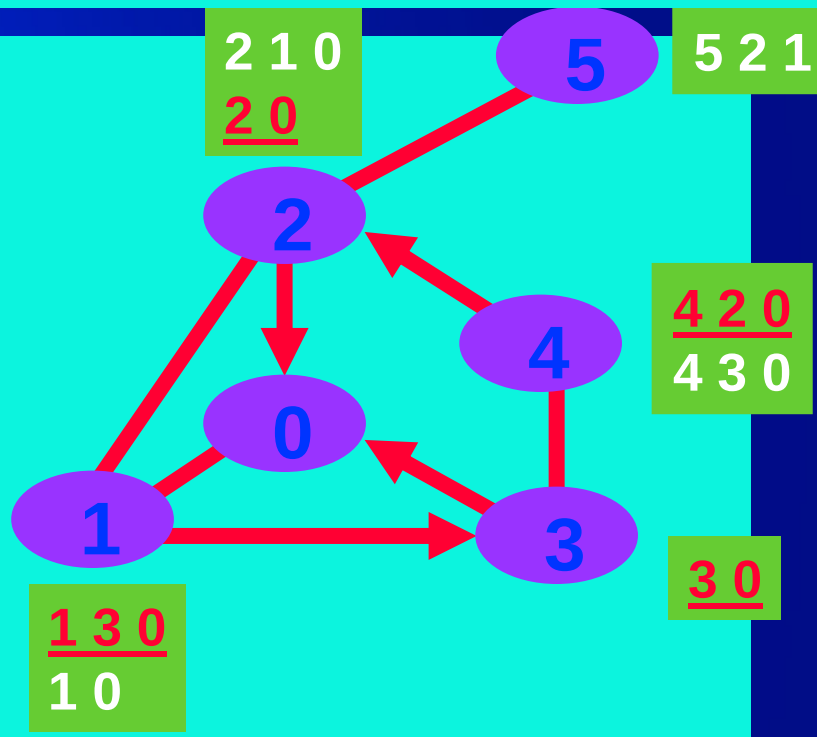


When modeling BGP : nodes represent BGP speaking routers, and 0 represents a node originating some address block

A Solution to a Stable Paths Problem

A *solution* is an assignment of permitted paths to each node such that

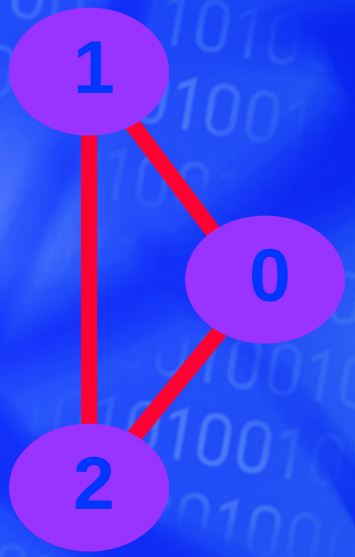
- node u's assigned path is either the null path or is a path uwP, where wP is assigned to node w and {u,w} is an edge in the graph,
- each node is assigned the highest ranked path among those consistent with the paths assigned to its neighbors.



A Solution need not represent a shortest path tree, or a spanning tree.

An SPP may have multiple solutions

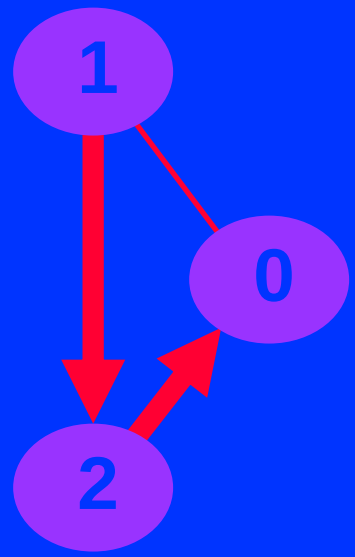
1 2 0
1 0



2 1 0
2 0

DISAGREE

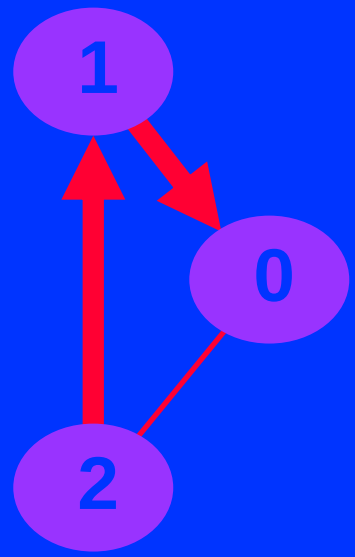
1 2 0
1 0



2 1 0
2 0

First solution

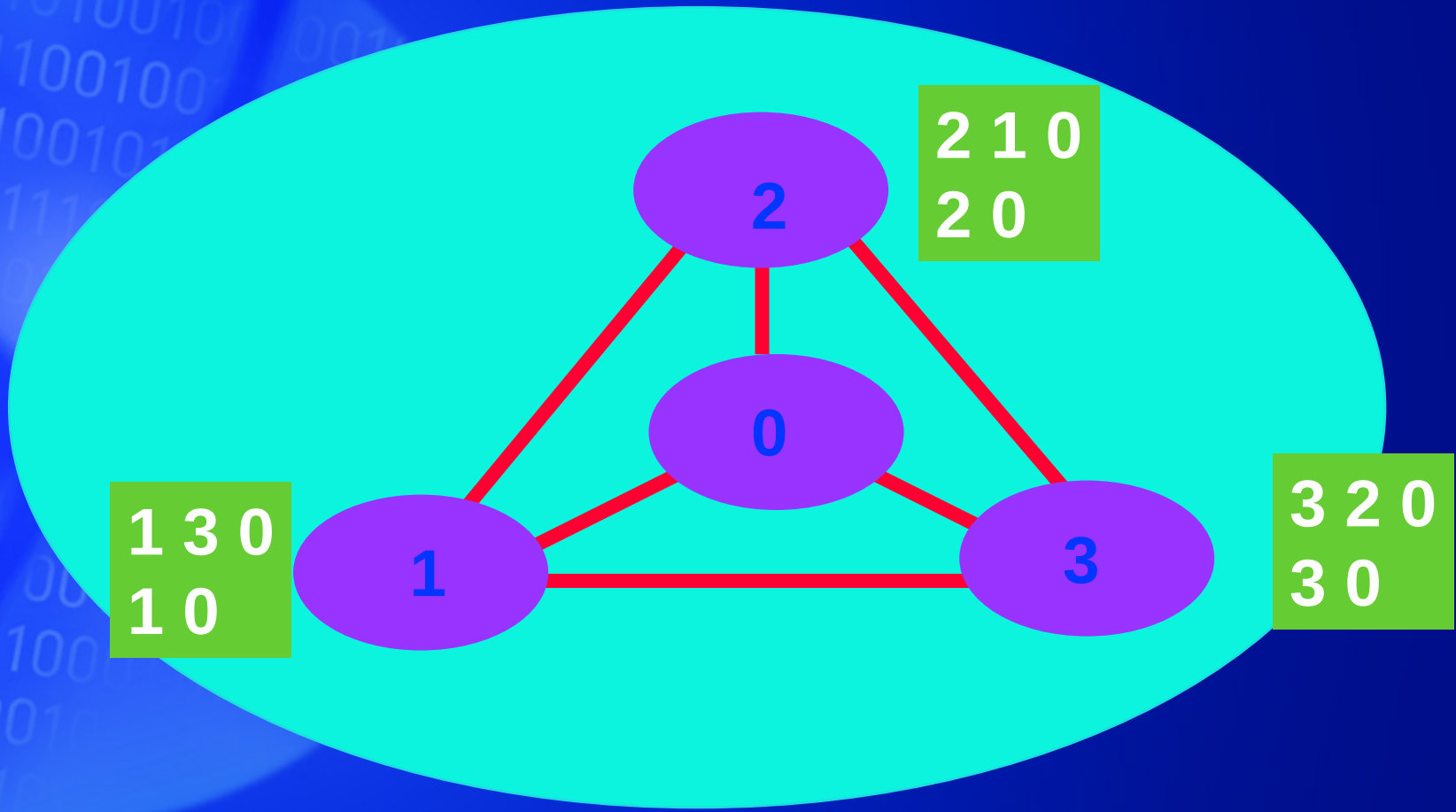
1 2 0
1 0



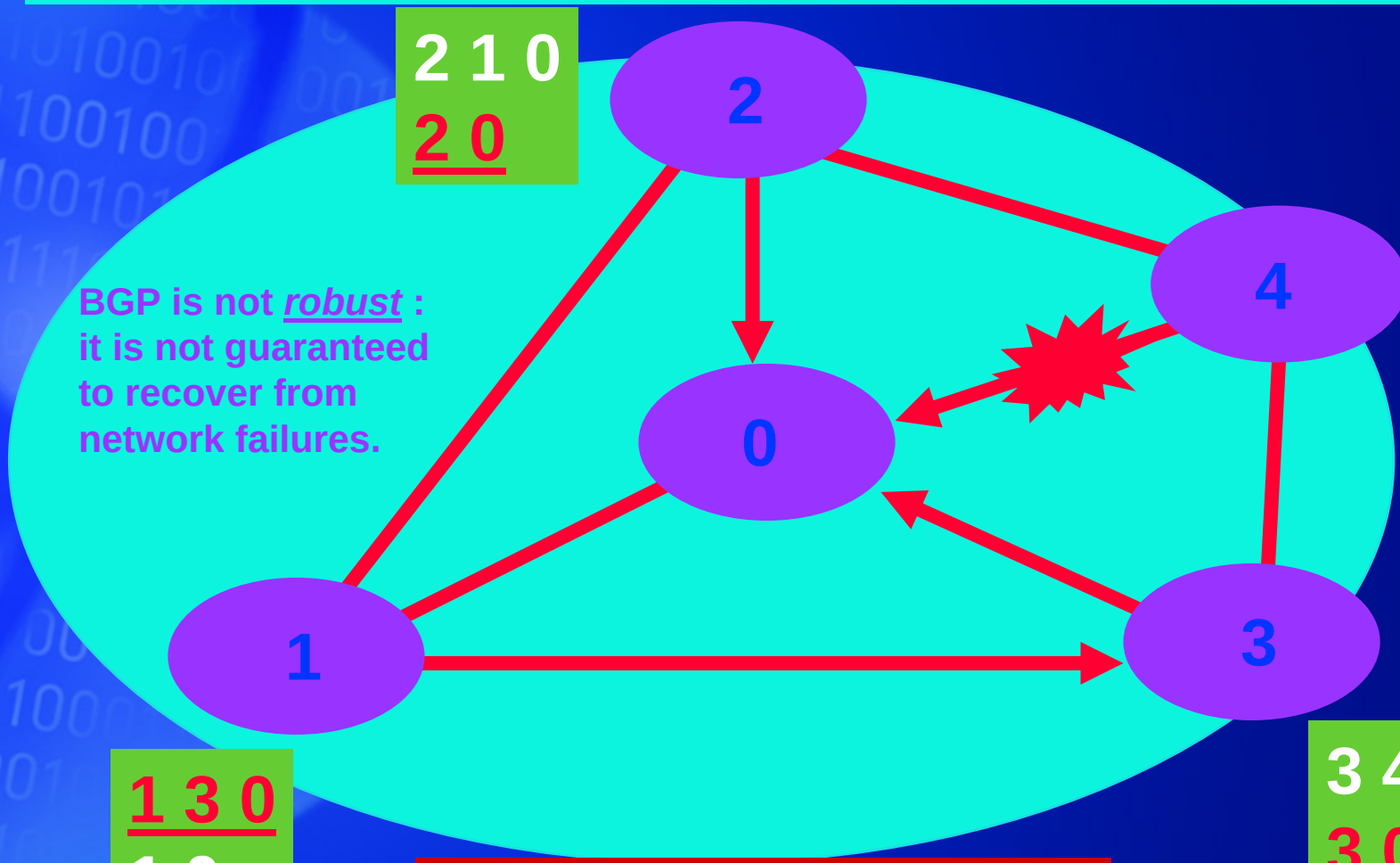
2 1 0
2 0

Second solution

BAD GADGET : No Solution



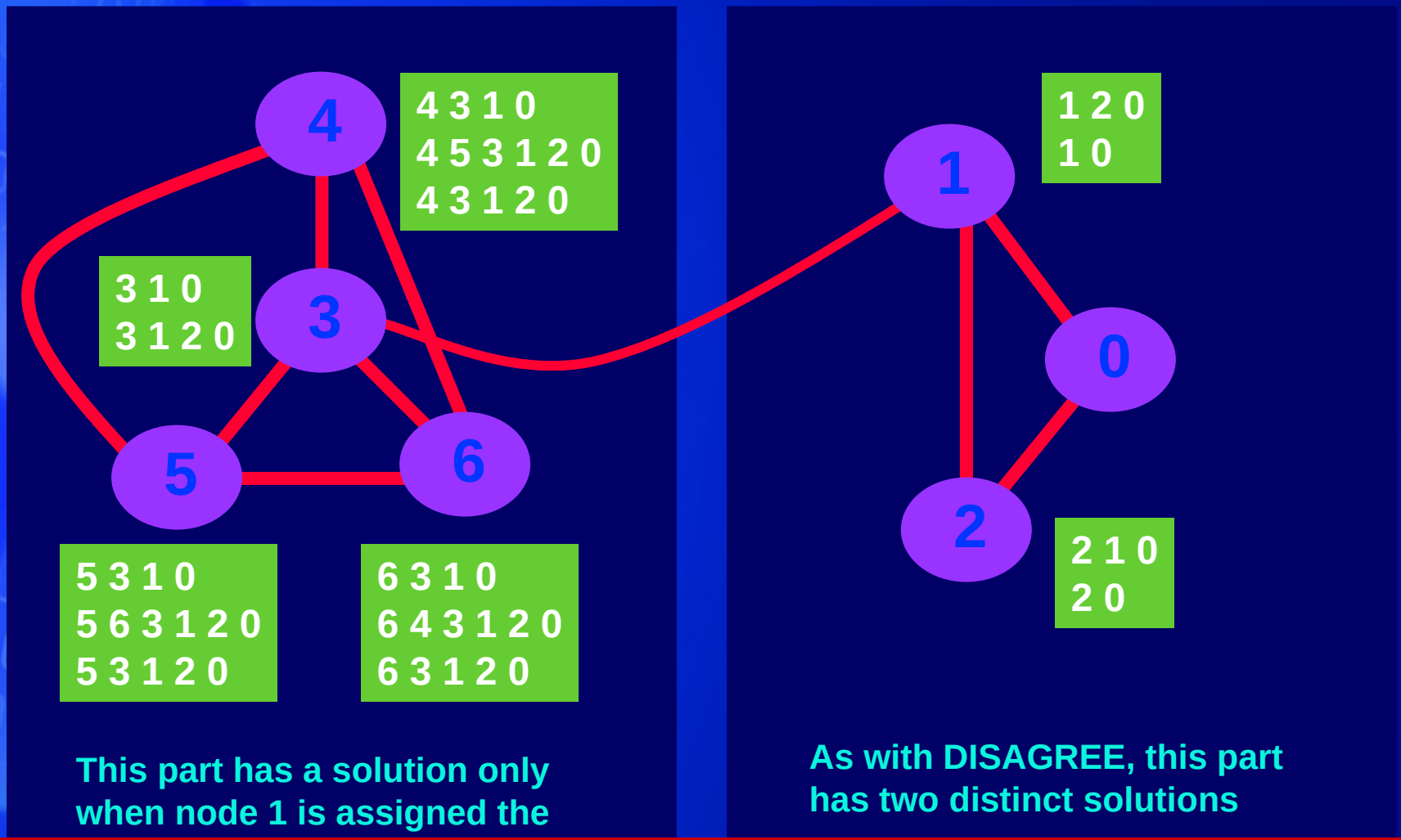
SURPRISE!


2 1 0
2 0
2
4
4 0
4 2 0
4 3 0
BGP is not robust :
it is not guaranteed
to recover from
network failures.
0
1
3
1 3 0
1 0
3 4 2 0
3 0
Becomes a BAD GADGET if link
(4, 0) goes down.

Let's be Clear: The SPP is a BAD model on which to base a routing Protocol

- But, BGP evolved, it wasn't really designed…
- The SPP originated in "reverse engineering" BGP
  - But at least it gives us some insight into the problems of BGP.

PRECARIOUS



Has a solution, but path vector may not find it!

Can BGP be fixed?

- BGP policy languages have evolved organically
- A policy language really should be designed!
- But how?

Joint work with
Aaron Jaggard (UPenn Math) and
Vijay Ramachandran (Yale CS)
to appear at SIGCOMM 2003

Design Dimensions

- Robustness (required!)
- Transparency (required!)
- Expressive Power
- Autonomy (“local wiggle room”)
- Local vs. Global Constraints
- Policy Opacity

Tradeoffs galore

Next?

- Need techniques for constructing policy languages.
- Design of protocols to enforce global constraints.
- Can ad-hocery be avoided?

Some References

- Persistent Route Oscillations in Inter-Domain Routing. Kannan Varadhan, Ramesh Govindan, and Deborah Estrin. Computer Networks, Jan. 2000. (Also USC Tech Report, Feb. 1996)
  - Shows that BGP is not guaranteed to converge
- An Architecture for Stable, Analyzable Internet Routing. Ramesh Govindan, Cengiz Alaettinoglu, George Eddy, David Kessens, Satish Kumar, and WeeSan Lee. IEEE Network Magazine, Jan-Feb 1999.
  - Use RPSL to specify policies. Store them in registries. Use registry for conguration generation and analysis.
- An Analysis of BGP Convergence Properties. Timothy G. Griffin, Gordon Wilfong. SIGCOMM 1999
  - Model BGP, shows static analysis of divergence in policies is NP complete
- Policy Disputes in Path Vector Protocols. Timothy G. Griffin, F. Bruce Shepherd, Gordon Wilfong. ICNP 1999
  - Define Stable Paths Problem and develop sufficient condition for "sanity"
- A Safe Path Vector Protocol. Timothy G. Griffin, Gordon Wilfong. INFOCOM 2001
  - Dynamic solution for SPVP based on histories
- Stable Internet Routing without Global Coordination. Lixin Gao, Jennifer Rexford. SIGMETRICS 2000
  - Show that if certain guidelines are followed, then all is well.
- Inherently safe backup routing with BGP. Lixin Gao, Timothy G. Griffin, Jennifer Rexford. INFOCOM 2001
  - Use SPP to study complex backup policies
- On the Correctness of IBGP Configurations. Griffin and Wilfong.SIGCOMM 2002.
- An Analysis of the MED oscillation Problem. Griffin and Wilfong. ICNP 2002.
- Design Principles of Policy Languages for Path Vector Protocols. Timothy G. Griffin, Aaron D. Jaggard (Department of Mathematics, University of Pennsylvania), Vijay Ramachandran (Department of Computer Science, Yale University). SIGCOMM 2003.
- Network Routing with Path Vector Protocols: Theory and Applications. Joao Luis Sobrinho SIGCOMM 2003