On the use of Trial and Error Traffic Engineering techniques in the Internet

Pedro Andrés Aranda Gutiérrez
University of Paderborn
Paderborn, Germany
Email: paaguti@hotmail.com

***Abstract*—The Border Gateway Protocol governs the overall routing in the Internet. With the time, it has been overloaded with functions which is was not initially designed for. The main example is Traffic Engineering. Internet Service Providers need to adjust the traffic on their peerings and in absence of a better alternative, use Trial and Error techniques based on manipulating the Autonomous System Path attribute in Border Gateway Protocol (BGP-4) updates. This paper shows sequences of BGP-4 updates where the use of Trial and Error Traffic Engineering techniques have led to instability and the countermeasures used to minimise their impact impact. The analysis methods presented also partially reveal the way Internet Service Providers use AS_PATH Prepending.**

***Index Terms*—Routing protocols, Network Operations, Network management, Network monitoring, Public networks**

I. Introduction

The Internet is partitioned in Autonomous Systems (ASs) interconnected with each other. This interconnection is governed by peering agreements signed between the Internet Service Providers (ISPs) which manage these ASs. Peering agreements started being technically simple documents where ISPs included technical and economical conditions. Technical definitions included the addressing space which would be mutually visible through the links and economical conditions defined the method of translating the traffics exchanged in economical transactions. Service Level Agreements (SLAs) where introduced to guarantee up-times and certain Internet Protocol layer parameters like round trip delay , tolerated levels of packet loss and in- and outbound traffic levels. SLAs penalising non-conforming behaviour introduce an additional incentive for control traffic control mechanisms, i.e. Traffic Engineering.

BGP-4 [1] is the routing protocol which governs the Internet. It implements the information exchange between ASs and has a complex routing decision mechanism to decide which route has to be installed in the routing table. The number of ASs traversed to reach a prefix is one of the attributes it takes into account during this decision process. This information is stored in the Autonomous System Path (AS_PATH) attribute and BGP-4 implementations offer mechanisms to manipulate it.Vendor reference material like [2] or [3] explain how each vendor has implemented the specification and slight differences can be appreciated.

Trial and Error (T&E) is based on a closed loop which involves stepwise tuning of the AS_PATH attribute of the prefix or prefixes announced by an AS in order to adjust the incoming traffic. Colitti et al. recognise the use of T&E techniques in this context [4], but do neither provide any further analysis of the techniques used nor examine the impact on the routing infrastructure. Vendors provide examples of Traffic Engineering (TE) techniques based on the control of the AS_PATH attribute [5].

This paper presents work in progress on the analysis of Trial and Error TE using BGP-4. A previous paper proposing a method to isolate these sequences [6] was presented in the ICNS-2009 congress. Studying the extent to which this technique is used by analysing data in the RIPE's Routing Repository (RIPE RR) [7], a period of unusually intense activity in May 2007 was isolated

This paper studies the distribution of AS_PATH lengths and hop counts before, during and after the incident and identifies significant BGP-4 update sequences responsible for the profile. It is structured as follows: Section II presents the rationale behind the isolation process. Section III studies the behaviour of a set of prefixes before, during and after a period of unusual activity and identifies the sequences of updates which explain the profiles. Section IV presents the conclusion and further work.

II. Trial and Error TE in the Internet

This section presents the basics of peering agreements in the Internet and how these and the tools available to enforce them have shaped Traffic Engineering in the Internet. The state of the art in research on Trial and Error and associated techniques is examined. Finally, previous research on the topic is shortly presented.

A. Peering agreements in the Internet

Internet Service Providers organise their interconnection through peering agreements, which include the definition of technical and economical conditions under which they exchange traffic. The technical definitions include addressing space which is made mutually accessible, the mechanisms to route traffic through the interconnection links and Internet Protocol layer parameters like round trip delay and tolerated levels of packet loss and acceptable traffic levels for the in- and outbound links. The economical terms of a peering agreement include payment policies and SLAs which guarantee up-times and penalise non-conforming behaviour. This

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7180
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7180

introduces an additional incentive for mechanisms to control the in- and outbound traffics of a network and, thus, for the implementation of TE techniques. Since BGP-4 lacks real TE capabilities, all approaches have been based on designing a certain routing configuration and assessing its quality by the traffic distribution it creates in the inter-provider links. Applications ranging from simple load balancing between independent links to a major upstream ISP to load sharing between several upstream ISPs are described in [5].

The process aims at an ideal traffic distribution with respect to some objective, i.e. minimisation of peering costs, uniform traffic distribution, etc. which is achieved by approximation with a more or less closed loop of configuration and observation, which starts with an initial configuration being deployed on the routers of an AS and the inter-provider traffic being determined. Eventually the configuration is refined and redistributed to the routers and the inter-provider links are monitored again. This process is repeated stepwise, the objective of the iterations being to come as close as possible the traffic distribution which best fulfils the SLAs subscribed by the ISP.

All the steps in this loop can be automatised by publicly available tools. Route servers, which help to define routing policies and adjust the inter-domain routing according to them, are widely available [8]. Looking glasses [9] to check the impact of routing configurations on the Internet are also widely deployed. Routers export traffic information using the IPFIX [10] standard, which allows to include BGP-4 information to the traffic information records. This information format is also used by other traffic measurement tools. A fully automated T&E TE loop is thus feasible.

In order to achieve the best approximation to the ideal traffic distribution, ISPs fraction their addressing space. G. Huston [11] recognises the use of this technique and examines its impact on the routing tables of the Default Free Routing area of the Internet. Quoitin et al. [12] provide a detailed analysis of Inter-domain Traffic Engineering techniques and propose the definition of a new BGP-4 attribute to provide a common way of expressing TE operations. The impact on Quality of Service (QoS) of Trial and Error TE was shown in the IST-INTERMON project [13], which developed a graphical tool to correlate routing actions and their impact on QoS [14].

B. AS_PATH Prepending

BGP-4 [1] is the inter-domain routing protocol of the Internet. It has a complex routing decision mechanism, which takes into account the number of ASs traversed to reach a prefix [2], [3]. This information is exchanged in the AS_PATH attribute. By default, BGP-4 speakers append their Autonomous System Number (ASN) to the AS_PATH only once and only in their exterior sessions, i.e. in the BGP-4 sessions between ASs. BGP-4 implementations offer a mechanism to manipulate this attribute, which is known as AS_PATH Prepending. It consists in adding the ASN several times at the beginning of the AS_PATH attribute. Different vendor manuals provide examples of TE techniques based on the control of the AS_PATH attribute [5].

BGP-4 implements loop prevention at AS level. In sessions between different ASs, incoming advertisements are checked and discarded if the AS_PATH attribute contains the ASN of the router they arrive at. It is thus impossible, that an interaction between BGP-4 speakers with default configurations (i.e. configurations without AS_PATH Prepending produces an advertisement such that it's AS_PATH attribute contains a repeated Autonomous System Number. AS_PATH Prepending only appears as the result of specific configuration commands, [1] which can be applied as part of the implementation of a policy. S. Halabbi [5] provides many examples of the use of AS_PATH Prepending.

AS_PATH Prepending has triggered some adverse effects. A public report from Cisco Systems [15] points out that their routers have trouble processing very long AS_PATHs. A. Kapela and A.Pilisov analysed the problem of hijacking Internet prefixes using BGP-4 at DefCon'2008 [16]. They show how to implement a man in the middle attack using BGP-4 traffic generation tools like SBGP or BGPSIM included in the Multithreaded Routing Toolkit [17] which can inject forged AS_PATHs in the Internets routing tables. AS_PATH Prepending makes it harder for network operators to detect and eliminate these forged AS_PATHs, because of the complexity involved in programming and maintaining a correct AS_PATH filters. This complexity grows with the proximity of ASs to the core of the Internet.

C. Definition of the Canonical AS_PATH

Since by manipulating the AS_PATH attribute, providers fine-tune the preference of a given path, the length of the AS_PATH attribute cannot be used to determine the number of Autonomous Systems between the originating AS and the collector which feeds the data to the RIPE RR.

To determine the distance of a network prefix in terms of the number of physical Autonomous Systems which have to be traversed, the concept of Canonical AS_PATH is introduced. The Canonical AS_PATH is derived from the AS_PATH attribute by removing all AS_PATH Prependings. It represents the path followed by the advertisement from the source to the observation point and its length is the number of ASs the update traversed.

D. Trial and Error TE detection algorithm for RIPE's Routing Repository data

To study the extent to which this technique is used, data in the RIPE's Routing Repository were analysed. The RIPE RR stores data coming from different routers which are connected to significant points in the Internet, which include several Internet Exchanges (IXPs) like the London Internet Exchange (LINX) and the RIPE's Default Free Routing area (DFR).The algorithm to detect Trial and Error used is described in [6]. It is based on splitting the RIPE RR data set into smaller sequences which contain information about one specific prefix as seen

[1] `set as-path-prepend ...` for Cisco Systems routers

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7180
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7180

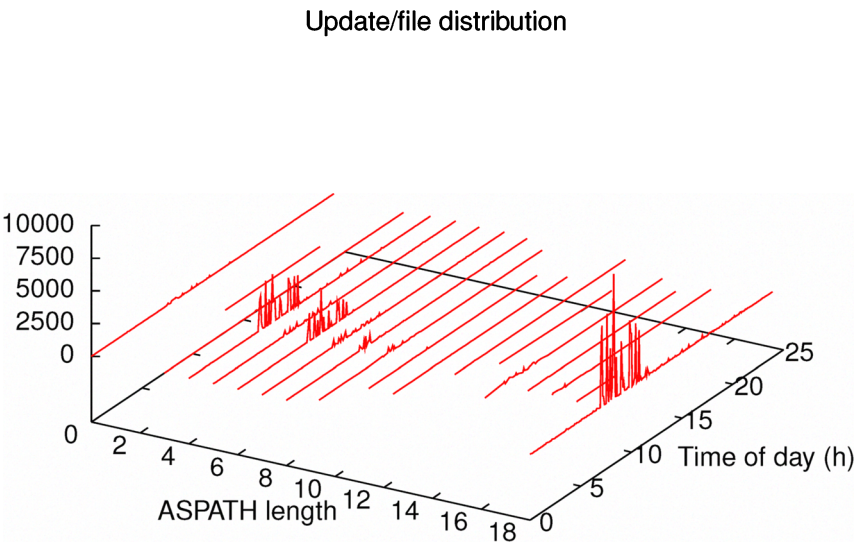

Figure 1. Length of the AS_PATH the day of Trial and Error Traffic Engineering incident

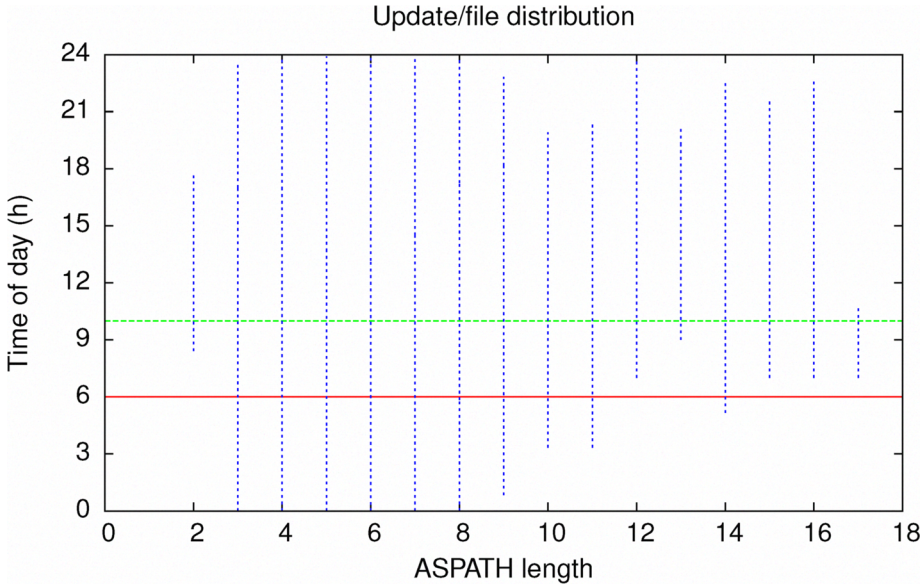

Figure 2. Existance of a given AS_PATH during Trial and Error Traffic Engineering incident

from one of the contributing routers. After this step, BGP-4 update sequences which have followed a common path in the Internet are isolated. If the length of the AS_PATH attribute in advertisements fluctuates, it is a proof that AS_PATH Prepending is being used to control the preference of a given path with respect to others, since for the reasons given above (BGP-4 loop detection, etc.) this sequence of updates can not result from an automatic protocol behaviour.

The algorithm also allows withdraws in the sequences for two reasons:

1) Accepted current practises when manually managing the routing of an Autonomous System allow restarting BGP-4 sessions in order make sure that a new configuration is flushed into the network. Restarting a session implies withdrawing all routes before sending the new advertisements.
2) Maximising the information captured by this method: When a change in the AS_PATH occurs, it might trigger the installation of a better route into the routing table. This implies that the old routes are withdrawn. When this change is reverted, the route will reappear in the table. Since we can't exclude that the withdraw isn't

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7180
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7180

signalling a manipulation of the AS_PATH, it is included as part of the candidate sequence.

When applying the algorithm on the data contributed on the month of May, 2007 by the London Internet Exchange routers to the RIPE RR, a period of high activity was detected [6]. The prefixes involved in this period of high activity were identified. Figure 1 shows the BGP-4 traffic related to T&E generated by these prefixes for a whole 24 hour period grouped by AS_PATH lengths on the 29th of May of 2007. The traffic is measured in BGP-4 updates per repository file, which is equivalent to a 5 minute window. The right-most graph shows the global update count. The left-most graph shows the count of withdraws and the other graphs show the update count per file for updates with a given AS_PATH length. A main peak in the global count distribution of approximately 9000 updates per file and several secondary peaks in the range of 4000 to 6000 updates per file show that the phenomenon is not marginal and worth studying.

III. AS_PATH PROFILES FOR TRIAL AND ERROR TRAFFIC ENGINEERING

This section extends the analysis of the behaviour of the set of prefixes involved a Trial and Error TE incident described in [6] to the days before and after the incident.

A. The use of the AS_PATH attribute in Trial and Error TE

The rationale behind this study can be described briefly as follows: upon the detection of a period of unusual activity, the prefixes involved in it were identified. The updates related to these prefixes were extracted from the RIPE's Routing Repository's data-set for a period of time covering the days before, during and after the unusual activity. This set was processed with the Trial and Error detection algorithm and the resulting events were used as the basis for further statistical analysis.

Figure 1 shows the evolution of the density of advertisements for the different AS_PATH lengths in time on the day of the incident. Figure 2 shows the same graph projected on the xy plane, giving a better view of the time periods were advertisements with a given AS_PATH length could be observed. The horizontal lines in the graph delimit the period of time with unusually high activity. The graph for $length = 0$ represents the density of withdraws and the graph for $length = 18$ length represents the overall density of updates of prefixes in this set. It shows that:

1) Most Trial and Error operations involve AS_PATHs with lengths 4 and 6. The graphs follow the profile of the aggregated traffic. Figure 4 shows that they account for approximately 70% of the events.
2) AS_PATH lengths shorter or equal 9 and with lengths 10 and 11 are present during the whole observation period or appear very much in advance with respect to the period of maximum activity. Except for the above mentioned lengths 4 and 6, the traffic is not very significant.
3) AS_PATH lengths 12, 15, 16 and 17 appear after the activity ramps up. As shown in Section III-C, they are used to assign unstable paths a very low priority. Thus, any advertisement with shorter AS_PATH attribute will be preferred in the hope it arrived through a path which is more stable.
4) Events with AS_PATH length equal to 13 appear shortly before the activity ramps up and need further analysis.

Figure 3 and Figure 4 show the probability distributions of the Canonical AS_PATH and the AS_PATH attribute respectively. Figure 3 shows that ASs which are 4 hops away account for more than 80% of the events. Hence, most of the T&E operations detected affect ASs which are physically 4 ASs away from the LINX. The rest of the days, the activity is approximately evenly distributed for Canonical AS_PATH $length \in \{2, 4\}$, as shown in Figure 3.

Figure 4 shows that the AS_PATH length fluctuates between $length = 2$ and $length = 8$. The day of the incident, the distribution has two main peaks at $length = 4$ and $length = 6$. These two peaks together show that most instabilities result from ASs in the path which toggle between announcing without AS_PATH Prepending and prepending by 2. This behaviour is also reflected in the beginning of the sequences shown in Listing 1.

B. The Canonical AS_PATH length distribution as indicator for Trial and Error TE

The probability distribution of the Canonical AS_PATH can be used as an indicator for periods of significant AS_PATH Prepending. Figure 5 shows the Canonical AS_PATH probability analysis and Figures 6 through 9 show the traffic profiles for the same set of prefixes between the 31^{st} of May and the 3^{rd} of June, 2007. When significant peaks in the T&E traffic (above 2500 updates per file) were detected, the probability distribution exhibits a peak for a specific AS_PATH length.

C. Defensive AS_PATH Prepending

In the previous section, the routing behaviour of a set of ASs advertising their prefixes through a mesh of ASs was analysed from the perspective of a single AS which is receiving these announcements. In general, the routing setup is far from static, with one or more ASs in this mesh modifying the AS_PATH attribute of the advertisements. The fluctuations in the AS_PATH may induce the ASs to consolidate different routes depending on the AS_PATH length. This process impacts the Quality of Service (QoS) of the end-to-end paths between the advertising and receiving ASs [14]. ASs prefer stable paths because the QoS parameters stay within predictable limits. Stable paths also reduce the risk of processor overload in the routers as a consequence of continuous path computations. Therefore the routing is fine-tuned, making unstable paths less preferable than stable paths by making them artificially longer than stable alternative paths. The amount of prepending introduced on the advertisements has to guarantee that the announced path will be always longer than the unstable path.

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7180
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7180

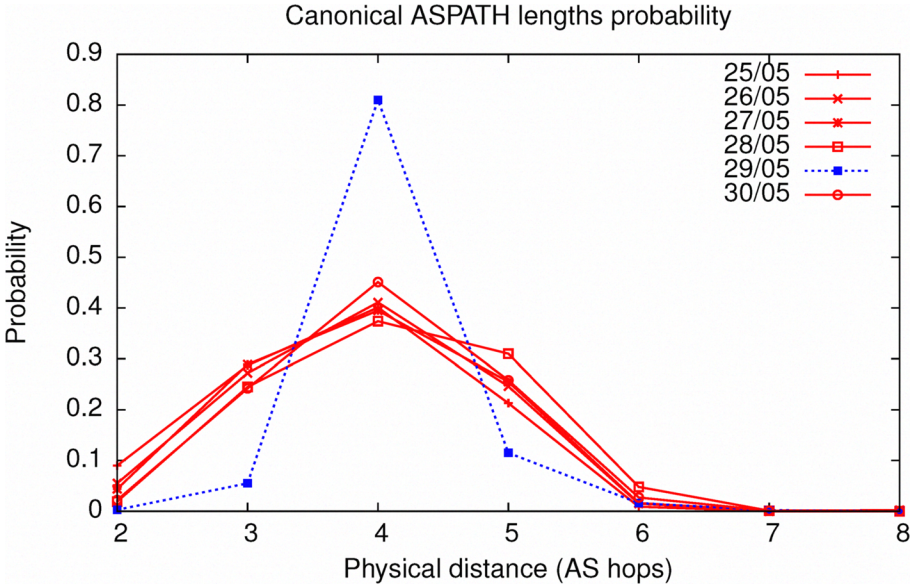

Figure 3. Canonical AS_PATH probability before, during and after the storm

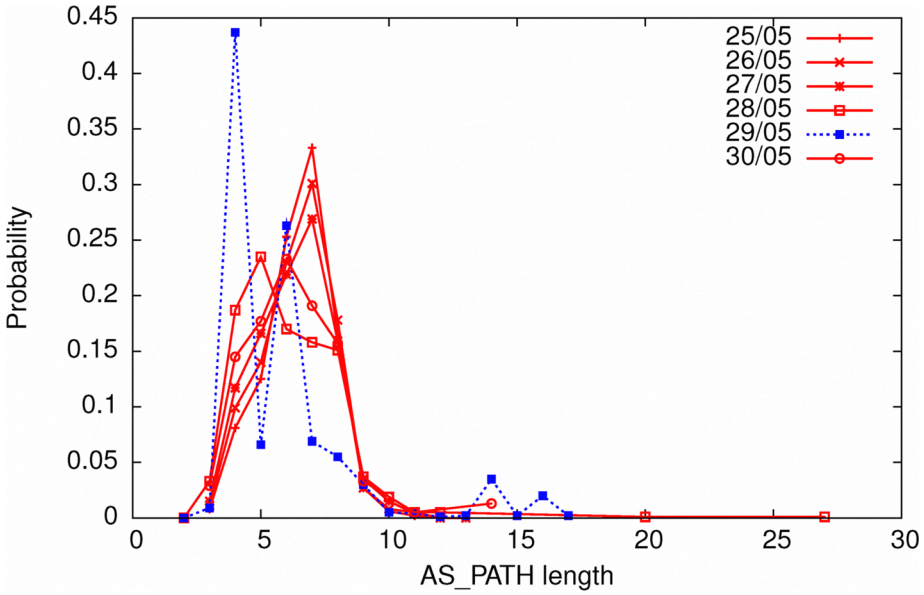

Figure 4. AS_PATH lengths probability, during and after the storm

This behaviour explains the peaks for $AS\_PATHlength = 14$ and $AS\_PATHlength = 16$ in Figure 4 the day of the incident. The day after the incident, a fraction of the udpates still have an $AS\_PATHlength = 14$. They correspond to ISPs which prefer to maintain the defensive re-routing until there is a reasonable level of certainty that incident is over.

Listing 1 was extracted from the output of the analysis program. It contains an abbreviated form of one of the BGP-4 update group which correspond to this defensive AS_PATH Prepending pattern. In the beginning, the source of instability is easily identifiable: AS#51 is prepending by 2 and withdraw-

```
...|A|...|7 2 51 51 51 14|...
...|A|...|7 2 51 14|...
...|A|...|7 2 51 51 51 14|...
...|A|...|7 2 51 14|...
...|A|...|7 2 51 51 51 14|...
...|A|...|7 2 51 14|...
...
...|A|...|7 2 51 51 51 14|...
...|A|...|7 2 51 14|...
...|A|...|7 2 51 14|...
...|A|...|7 2 51 51 51 14|...
...|A|...|7 2 51 14 14 14 14 14 14 14 14 14 14 14|...
...|A|...|7 2 51 51 51 14 14 14 14 14 14 14 14 14 14 14 14|...
...|A|...|7 2 51 14 14 14 14 14 14 14 14 14 14 14|...
...|A|...|7 2 51 51 51 14 14 14 14 14 14 14 14 14 14 14 14|...
...
```

Listing 1: Defensive AS_PATH Prepending

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7180
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7180

Table I
AS_PATH PREPENDING BEHAVIOUR (EXCERPT)

AS	fake hops	before	during	after
14	10	2	258	30
16	4		1120	
16	5		1043	
51	2	6	2818	37
11	2			3
11	3	98	108	95
11	4	20	15	3
60	1		2	2
60	2	12	22	13
60	3	115	156	130
60	4	127	178	142
67	3	91	67	60
67	6	60	32	37

ing this prepending. The end of the sequence corresponds to the defensive phase: AS#14 starts to prepend by 10, in order to make this path less preferable compared with other, more stable paths.

Finally, the AS_PATH Prepending behaviour of all ASs was studied. Table I shows an excerpt of the full result and is divided in two sections. The upper section shows Autonomous Systems which exhibit a distinctive behaviour in presence of heavy AS_PATH Prepending. Only one Autonomous System (AS#14) is applying defensive AS_PATH Prepending during and after the incident by introducing 10 fake hops into the AS_PATH. Only two additional Autonomous Systems are involved in significant activity: AS#51 introduces 2 fake hops in a significant amount of events and AS#16 toggles between introducing 4 and 5 fake hops. This happens mainly during the incident.

D. AS_PATH Prepending policies

The lower part of Table I shows other results obtained from Trial and Error TE analysis. The ASs in this part of the table behave with extreme consistency independently of the presence of oscillations. This behaviour is the result of the policies they use under normal circumstances. Thus, AS#11 mainly prepends by 3 and 4, AS#60 has policies which prepend by 2, 3 and 4 and AS#67 consistently prepends by 3 and 6. AS#11 starting to prepend by 2 and AS#60 starting to prepend by one can be considered a result of the AS_PATH Prepending incident.

IV. CONCLUSION AND FURTHER WORK

The AS_PATH attribute is always transmitted in BGP-4 sessions and is directly involved in the decision process which selects the best routes used by the routers in the Internet. AS_PATH Prepending is an artificial technique which modifies this attribute and is the result of a deliberate action of an Autonomous System administrator and thus is a suitable candidate to implement Traffic Engineering in the Internet.

This paper has shown that the mechanism to detect the use of Trial and Error techniques based on AS_PATH Prepending proposed in [6] successfully in isolates sequences of BGP-4 updates which are significant. It shows simple analysis methods which can be applied on the isolated update sequences to help pinpoint incidents, explain their nature and show the role of different Autonomous Systems involved or affected by them. The results reveal defensive methods used an AS which has been the victim of BGP-4 oscillation and the AS_PATH

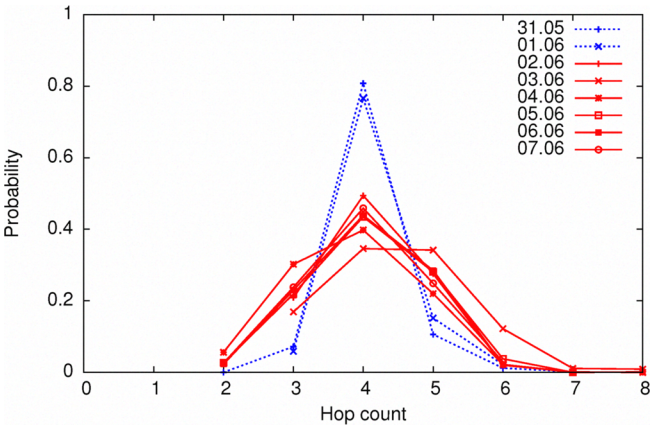

Figure 5. Canonical AS_PATH distributions

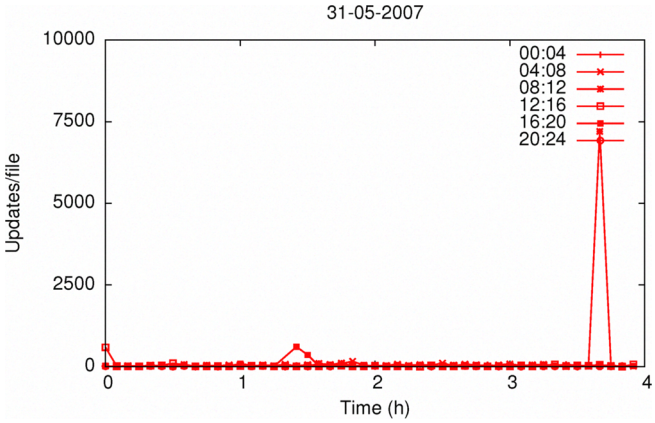

Figure 6. BGP-4 update traffic

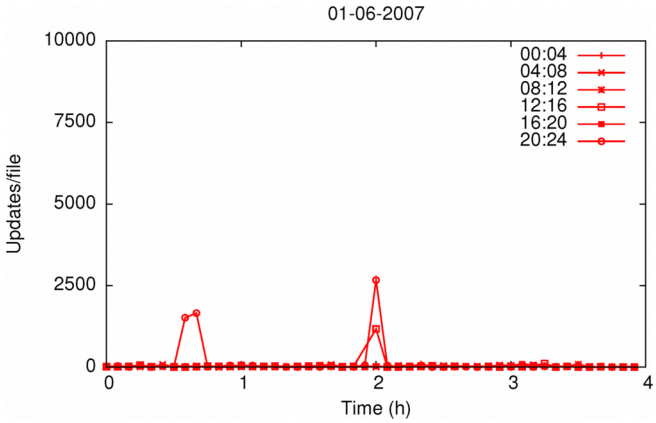

Figure 7. BGP-4 update traffic

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7180
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7180

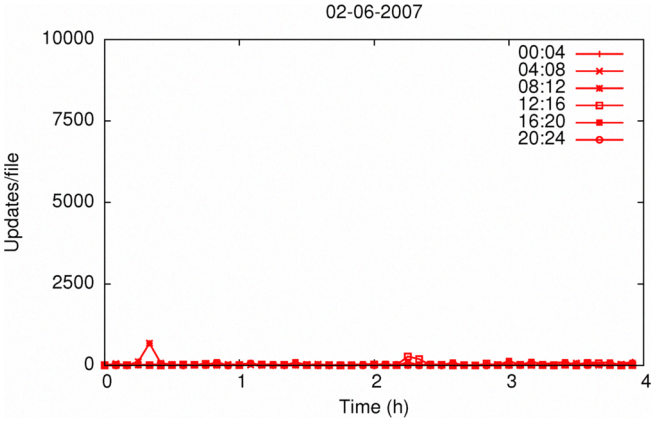

Figure 8. BGP-4 update traffic

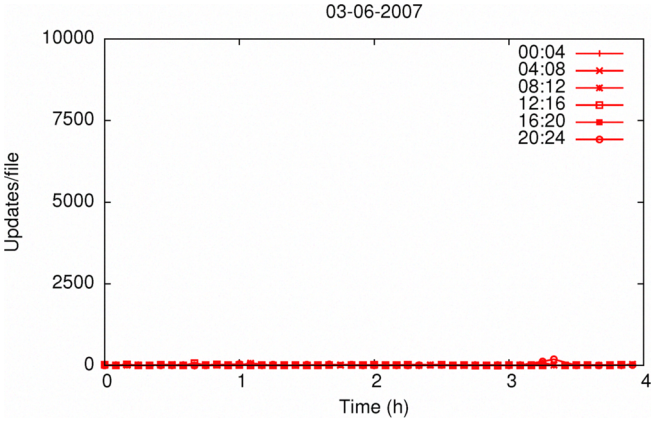

Figure 9. BGP-4 update traffic

Prepending behaviour for some ASs.

A. *Further work*

Next steps in this study are isolation of further BGP-4 update sequences which can be traced back to T&E Traffic Engineering and the exploration of new analysis methods to help detect other instances of T&E. Another research path is related with the detection of BGP-4 soft session restarts in the T&E sequences. Soft session restarts are responsible for massive amounts of BGP-4 update traffic which propagates explosively in the Internet.

ACKNOWLEDGEMENT

This work has been carried out using the extensive BGP-4 data collections of the RIPE's Routing Repository.

REFERENCES

[1] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271 (Draft Standard), Internet Engineering Task Force, Jan. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4271.txt

[2] *BGP Best Path Selection Algorithm*, Cisco Systems. [Online]. Available: http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094431.shtml

[3] *Selecting the Best Path*, Juniper Networks. [Online]. Available: http://www.juniper.net/techpubs/software/erx/erx50x/swconfig-routing-vol2/html/bgp-config10.html

[4] L. Colitti, G. Di Battista, M. Patrignani, M. Pizzonia, and M. Rimondini, "Investigating Prefix Propagation through Active BGP Probing," in *Proc. 11th IEEE Symposium on Computers and Communications ISCC '06*, 2006, pp. 497–504.

[5] S. Halabi, *Internet Routing Architectures*, 2nd ed. Cisco Press, 2000.

[6] P. A. Aranda-Gutiérrez, "Detection of Trial and Error Traffic Engineering with BGP-4," in *The Fifth International Conference on Networking and Services; ICNS 2009*. INRIA, April 2009.

[7] "RIPE Routing Information Service," 2001. [Online]. Available: http://www.ripe.net/projects/ris

[8] Public Route Servers and Looking Glass sites. [Online]. Available: http://www.netdigix.com/servers.html

[9] "BGP Looking Glasses for IPv4/IPv6, Traceroute & BGP Route Servers." [Online]. Available: http://www.bgp4.as/looking-glasses

[10] "IP Flow Information Export (ipfix)." [Online]. Available: http://www.ietf.org/html.charters/ipfix-charter.html

[11] G. Huston, "Analysing the Internet BGP Routing Table," *The Internet Protocol Journal*, vol. Volume 4, no. Number 1, 2001. [Online]. Available: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_4-1/ipj_4-1.pdf

[12] B. Quoitin, S. Uhlig, and O. Bonaventure, "Using Redistribution Communities for Interdomain Traffic Engineering," in *QofIS*, 2002, pp. 125–134.

[13] "Advanced architecture for INTER-domain quality of service MONitoring, modelling and visualisation, EU grant INTERMON-IST-2001-34123," 2001. [Online]. Available: http://www.ist-intermon.org

[14] P. A. Aranda-Gutiérrez, "BGP-4 Protocol Patterns and Their Impact on QoS Behaviour," in *Inter-domain Performance and Simulation Workshop, Budapest*, 2004.

[15] "Update: oversized ASPATH's," 2009. [Online]. Available: http://blog.ioshints.info/2009/02/update-oversized-as-paths.html

[16] A. P. Anton Kapela. (2008, August) Stealing the Internet. DefCon 16. [Online]. Available: https://www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-pilosov-kapela.pdf

[17] "The Multi-threaded Routing Toolkit MRT," http://www.mrtd.net.

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7180
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7180