

Implementation of a BGP Route Flap Damping Algorithm for the Bird Routing Project

Alexandre Chappuis, Bastian Marquis and David Klopfenstein @ EPFL.ch

Abstract—Today’s Internet stability strongly relies on the good behavior of dynamic routing protocols such as BGP (Border Gateway Protocol), that enables routing between Autonomous Systems. Route flapping is a well-known and undesirable phenomena occurring in both commercial and private networks. In this report, we carefully explain our implementation of the RFC 2439, BGP Route Flap Damping, for one famous open source routing software suite, the Bird Routing Project.

I. INTRODUCTION

The inter-domain routing protocol BGP is still surviving to the gigantic growth of the Internet that started during the last decade. However, some widely used applications, such as Skype, still suffer from weaknesses of that protocol. The main problems are twofolds: Firstly, BGP has a slow convergence rate, meaning that a change at one location takes quite some time to be propagated throughout the network. Secondly, if a node becomes unstable, for example if its connectivity constantly comes up and down, it will have bad consequences on the network, both in terms of useless processing at BGP routers and unnecessary routing traffic. Routes advertised and withdrawn at regular interval of time are said to be flapping.

Many approaches dealing with route flapping have been developed in the late 90’s. The RFC 2439[1] was the first standard proposed so that routers could inhibit the propagation of bad-behaving routes, until they become stable again. People have used it extensively for many years, in both commercial and open source routers.

Although this standard is not recommended anymore[2] in today’s routers, we wanted to implement it for the Bird Routing Project[3], hoping that it will serve as a good basis for future possible improvements and extension of this RFC. There exist many variants of the Route Flap Damping algorithm and the community has not lost its interest in finding robust mechanisms that could allow BGP to be more resilient.

II. OVERVIEW

RFC 2349 seeks to limit the impact of route flapping by “damping” (*i.e.* ignoring packets of) misbehaving routes. The solution must be able to distinguish flapping routes from good routes, consume few resources, both in terms of memory usage and process time. The RFC solves these problems by assigning each route a penalty term. Whenever this penalty term for a given route reaches a certain threshold, further advertisements for that route are ignored. This penalty term varies over time : it is increased when the route becomes unreachable and decays as long as the route stays stable. As soon as the figure of merit goes below a *reuse threshold*, the route can be used again.

The figure of merits decays exponentially over time. Exponential decay has several advantages : it can be implemented very efficiently using precomputed *decay arrays*. Also, using exponential decay, the figure of merit keeps trace of previous instabilities for a fairly long time : old instabilities become less and less important over time, while newer ones have more weight.

Network administrators have lots of freedom in choosing the behavior of the penalty term : they can control the half-life of the penalty term, its maximum value (thus controlling the maximum time a route can be damped) and both the reuse and cut thresholds.

Here is an example showing how the figure of merit evolve over time. The route flapped four times before exceeding the *cut threshold*. It then remained stable.

The RFC proposes several optimizations to decrease processing time, at the cost of a slightly bigger memory footprint. *E.g. reuse lists* are used to not have to recompute the figure of merit of all routes : damped routes with similar penalty terms are grouped together in a same reuse list. Penalty terms of the routes in a given reuse list are then re-computed only when necessary.

III. IMPLEMENTATION

this part is kind of straightforward -> just explain how we did it with bird. cite coder’s doc + github rep. of code

- A. Data structures
- B. Processing withdrawals
- C. Processing route advertisements
- D. Configuration parameters
- E. Timers
- F. Miscellaneous

IV. EVALUATION

- A. Small scale
 - 1 router with ca. 10 neighbors -> topology 3
- B. 30 BGP routers in NSL cluster
 - what we’re going to need :
 - number of route damped for each bgp proto vs. number of diff. routes advertised/withdrawn => show there are less updates and unnecessary traffic when damping = activated
 - do it with different parameters ?
 - convergence time not affected ?

V. CONCLUSION

show importance of stability

VI. FUTURE WORK

possible extensions

real scale tests

VII. ACKNOWLEDGEMENT

REFERENCES

- [1] The RFC 2439, BGP Route Flap Damping, <http://www.ietf.org/rfc/rfc2439.txt>
- [2] RIPE Recommendations On Route-flap Damping, <http://www.ripe.net/ripe/docs/ripe-378>
- [3] Bird Routing Project, <http://bird.network.cz>
- [4] Our publicly available repository, <https://github.com/alexchap/Albatros-Project>