An AI solution to BGP Convergence Problem

Bin Gui, Yang Yushuai

Abstract

Today internet has become one of the most important element of the society, from everyday life to commercial activity, internet has huge impact on their efficiency and quality. This impact is increasingly growing with the growth of the technology and the scale of the internet, and that is why when the internet starts having glitches, costs also significantly increase.

BGP, as the internet packet delivery system, responsible for all those glitches that relate to the routing. As the scale of the internet expand, it becomes more and more important to keep the BGP system stable. That is why it is essential to solve BGP convergence problem which cause the latency and other problems in internet.

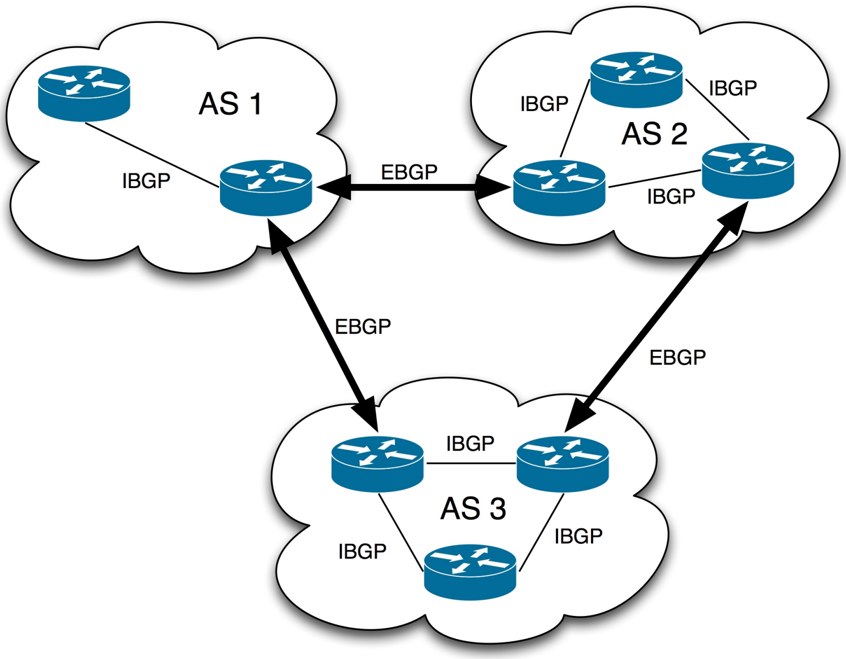
AI is a technology that has a strong capacity of representing hypothesis, being able to capture underlying regularities inside data sets and therefore building internal representations from the data. It is capable of dealing massive amount of data generated by the BGP system and potentially find pattern of the network and predict its behaviour.

Those are reasons why our group would like to make a project that focus on solving BGP convergence problem using an AI solution. Our goal is to find how to reduce the massive effect of the BGP convergence problem by implementing AI on the BGP system, and see if it does have an effect on the internet.

BGP

Our internet is divided into network groups called Autonomous Systems, ASes. AS is a collection of connected Internet Protocol routing prefixes, for example All device under Temple University domain form an AS, to visit any devices or resources like website of Temple a visitor need to connect to Temple AS first, then visit desired resources.

It needs to be pointed out that not all AS can be directly connected from another AS, it is very often that to connect an AS from another AS, it has to pass through several ASes in the process, just like if a man want to visit a city from another city, he has to pass through cities between those two cities.



A demonstration of BGP system

If considering ASes as vertices, then BGP is the edge connecting those vertices. To be more specific, it is the EBGP that connect different ASes while there exist IBGP that connect device under one AS. A stable connection between AS is the key to keep the internet healthy, however it is common that an AS starts having trouble, and affect all connection that pass through this AS. To reduce the impact of it, BGP has a mechanism that use backup route when its current route stop working. However the switches from current route to backup route could take very time to happen, sometimes backup route may also not working due to policy conflicts or other issues. Those are BGP problems that need to be solved by AI.

BGP Convergence

So what is exactly the BCP convergence? In general, when a node v find an available path to its destination and all its routing information remain unchanged until next change in the network that influence it happened. We called that node converged.

When a change in topology happens, network convergence delay happens since switching to backup path takes time to happen. I will explain later why it is necessary to take some time to switch path. Sometimes there is no guarantee that a BGP system will eventually converge due to policy conflicts, this situation is called unsolvable BGP system, it is also a type of BGP convergence problem. There are other types of convergence problem like ghost information, which is not important in this project since the AI solution we focus on is to reducing delay and solve unsolvable BGP system.

Why it is necessary for BGP system to take time to switching route? When an AS X change its path connecting to other ASes, other ASes that pass through X to connect other AS should notice that the cost of using AS X to reach other ASes has changed, or even become unavailable. It is nature for AS X to inform its neighbors that there is an update in AS X happens. So a change in one AS could cause changes in many others. If an AS send too many updates information in very short time period, the whole network could be flood by those update information. Plus, the processing power of those information accepter, which is router, is limited, with limited storage power of routers too many updates information could cause packet lost in routers.

That is why routers has a MRAI timer, which controls how often a router will send updates information. This timer has a default setting for all router which is 30s, however in different situation 30s may be too long or too short, which would delay the convergence. Is it possible for router to adapt different MRAI depending on the different situation of the network? Within the power of AI, it is fairly possible.

An AI solusion

BGP system is a distributed system, which means a node cannot know the whole topology of the internet, and information received by the node may already outdated when reached. So router cannot make decision based on information it acquired now, to decide what MRAI value it wants to use to reduce convergence delay it has to predict the topology of the internet. The classic algorithm is not good at predicting the topology with that scale, that is why using machine learning algorithm could be the solution of this problem.

There are some difficulties applying machine learning on BGP system, however.

First is the collection of training data. Since ASes are owned by organization or some individuals, most of those AS data are not public. There is a project called RIPE RIS that constantly collection BGP updates and routing tables which could be used for the training purpose.

The second question is how to monitor the internet to collect real-time input and measure the change on internet due to the implementation of AI. To do so researchers could use PEERING project which provide information of the real internet routes. Its use require approval so our group is not able to test our implementation on real internet.

The third problem is how to implement the AI. The problem with the current BGP system is that all routers are vendor defined, which mean it only contain the functionality provide by vendors. To make the implementation of the AI possible, you could either purchase specialized routers with the AI implemented by vendors, by a huge price, or, using System Defined Network, which is a logically centralized controller that rules a group of switches using a standard interface to achieve the functionality of ordinary operation system, that could implement an AI on routers directly.

The next question is what algorithm should be used. Though most common machine learning algorithm can be used to build the prediction model, we prefer Long short-term memory algorithm in this case. That is because ordinary RNN suffers from the vanish gradient problem. Vanish gradient problem occur when training sequences are long, it makes the training more and more difficult since the neural networks gradient get close to 0 when the training process is long. That is to say, the model will gradually forget what it has learned. LSTM could address this issue

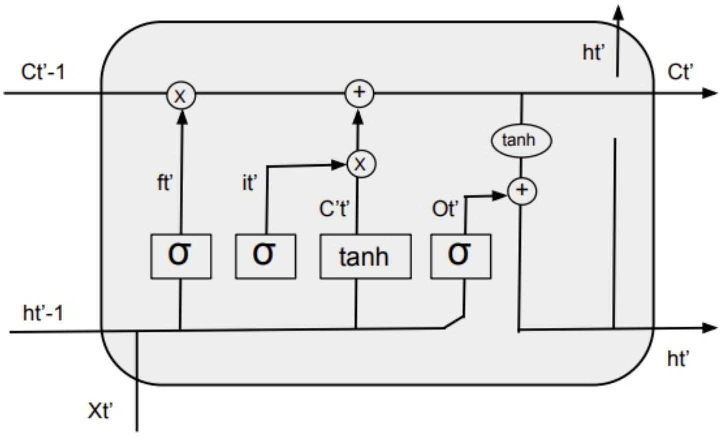


Figure 2.5: Anatomy of a LSTM network [2]

How is that work? Unlike RNN, each repeating model in LSTM has 4 layers instead of 1, each layer represented by a small rectangle in the figure. Xt' is the input, the X and + are operations used to process the output of those layers. The most important feature of LSTM is cell state and different gates. Cell state is represented by C in the figure, forwards or throw away information according to the decision made by the forget gate (ft), input gate will decide which information should be stored in cell state, and output gate will decide which information should be sent out from this model as output.

With all of those set, a router with complete functionality of applying this AI is ready to be put into the field. The implemented models are able to predict the BGP source convergence time for announcements and can offer researchers and network operators useful insights on the BGP dynamic behavior. The accuracy of the prediction is quite amazing, as the average of 78%. With the ability to predict the topology of the internet a dynamic value of MRAI can be estimated by using this prediction and thus reduce the impact of BGP convergence delay.

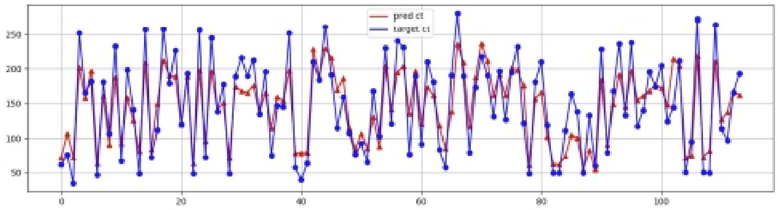
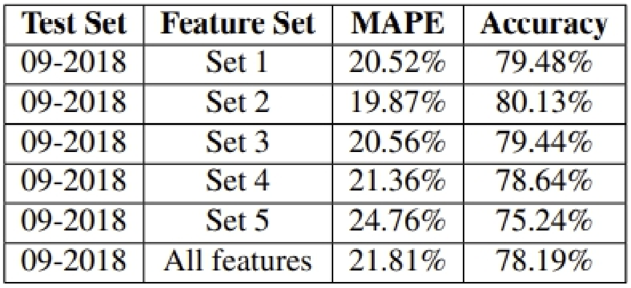


Figure 4.11: BGP Convergence prediction- Month 09

Table 4.5: Performance of Model with training set 02 and Test Set 09-2018



Test Set	Feature Set	MAPE	Accuracy
09-2018	Set 1	20.52%	79.48%
09-2018	Set 2	19.87%	80.13%
09-2018	Set 3	20.56%	79.44%
09-2018	Set 4	21.36%	78.64%
09-2018	Set 5	24.76%	75.24%
09-2018	All features	21.81%	78.19%

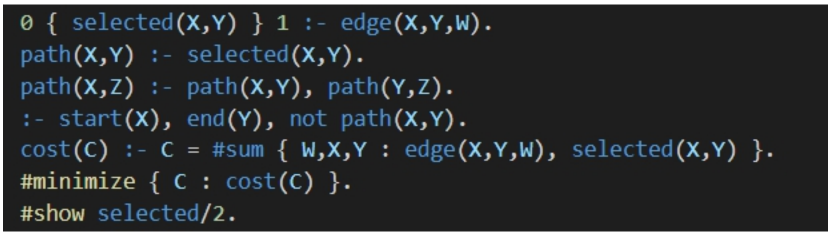
ASP Solution

Answer set programming (ASP) is a form of declarative programming oriented towards difficult (primarily NP-hard) search problems. It is based on the stable model (answer set) semantics of logic programming. In ASP, search problems are reduced to computing stable models, and answer set solvers—programs for generating stable models—are used to perform search. The computational process employed in the design of many answer set solvers is an enhancement of the DPLL algorithm and, in principle, it always terminates (unlike Prolog query evaluation, which may lead to an infinite loop).

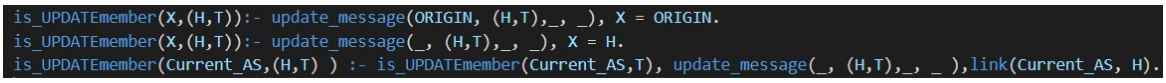
Answer Set Prolog - a language for knowledge representation and reasoning based on the answer set/stable model semantics of logic programs. The language has roots in declarative programming, the syntax and semantics of standard Prolog, disjunctive databases and non-monotonic logic. Unlike “standard” Prolog it allows us to express disjunction and “classical” or “strong” negation.

$$r: \quad \underbrace{b_1 \textit{ or } \dots \textit{ or } b_m}_{head(r)} \leftarrow \underbrace{a_1, \dots, a_n, \textit{ not } a_{n+1}, \dots, \textit{ not } a_{n+k}}_{body(r)},$$

And Clingo is an ASP system to ground and solve logic programs. Here is an example we use clingo to find the shortest path. We used Clingo in our ASP-BGP implementation.

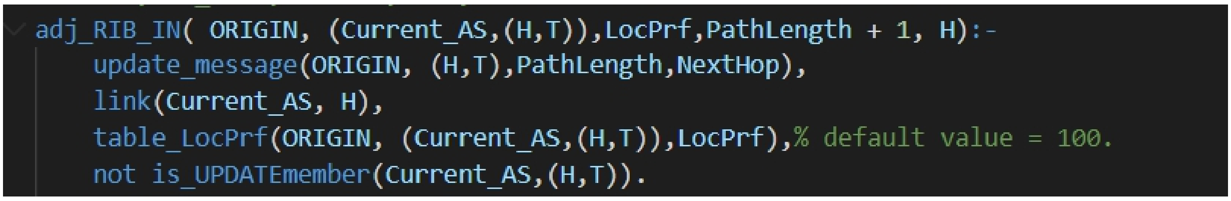


```
0 { selected(X,Y) } 1 :- edge(X,Y,W).
path(X,Y) :- selected(X,Y).
path(X,Z) :- path(X,Y), path(Y,Z).
:- start(X), end(Y), not path(X,Y).
cost(C) :- C = #sum { W,X,Y : edge(X,Y,W), selected(X,Y) }.
#minimize { C : cost(C) }.
#show selected/2.
```



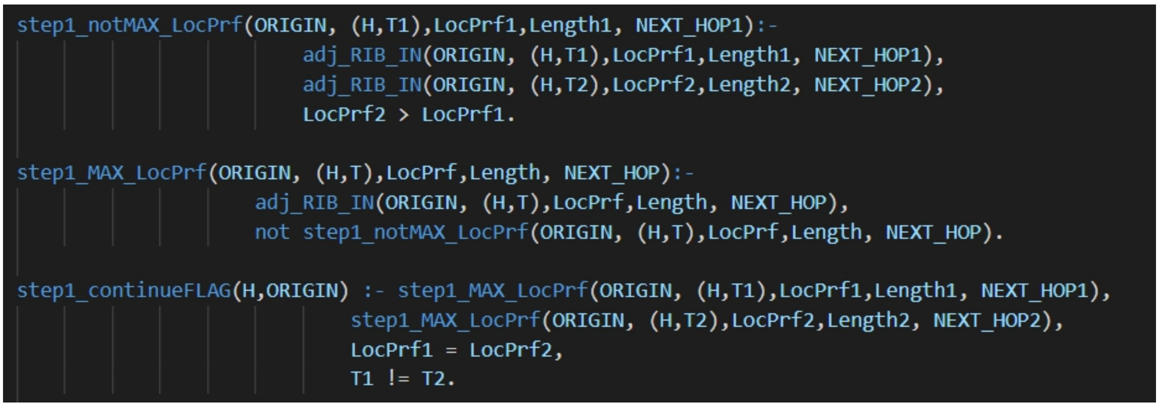
```
is_UPDATEmember(X,(H,T)):- update_message(ORIGIN, (H,T),_, _), X = ORIGIN.
is_UPDATEmember(X,(H,T)):- update_message(_, (H,T),_, _), X = H.
is_UPDATEmember(Current_AS,(H,T) ) :- is_UPDATEmember(Current_AS,T), update_message(_, (H,T),_, _ ),link(Current_AS, H).
```

We use ASP to model selection procedure in BGP. Here are some key steps, Firstly , is a recursive rule to check if the current AS is a receiver of the UPDATE message. If so we can just drop it, if not we store it in the local adjacent routing information base.



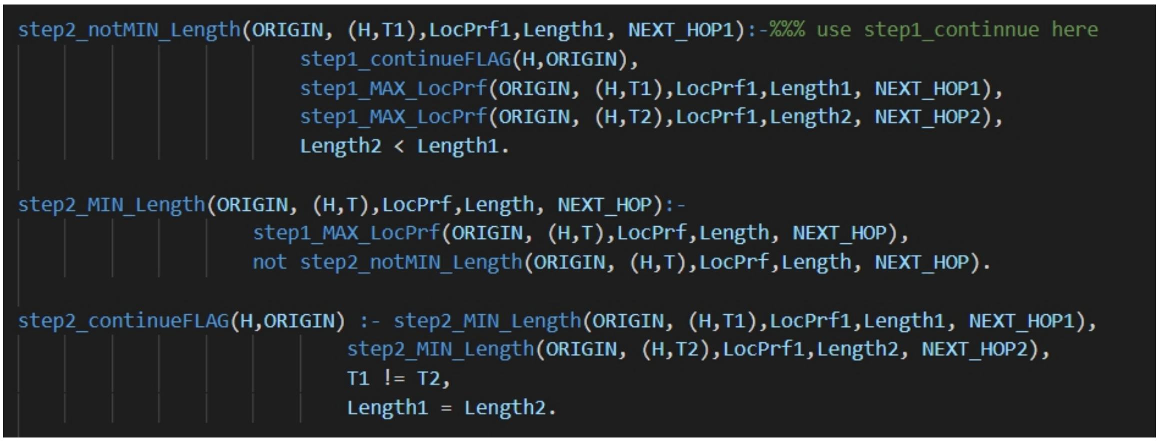
```
adj_RIB_IN( ORIGIN, (Current_AS,(H,T)),LocPrf,PathLength + 1, H):-
    update_message(ORIGIN, (H,T),PathLength,NextHop),
    link(Current_AS, H),
    table_LocPrf(ORIGIN, (Current_AS,(H,T)),LocPrf),% default value = 100.
    not is_UPDATEmember(Current_AS,(H,T)).
```

The Adj-RIBs-In stores routing information learned from inbound UPDATE messages that were received from other BGP speakers. Their contents represent routes that are available as input to the Decision Process.



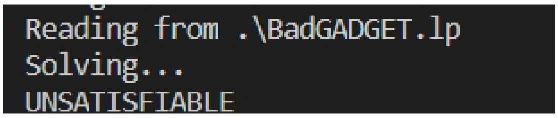
```
step1_notMAX_LocPrf(ORIGIN, (H,T1),LocPrf1,Length1, NEXT_HOP1):-
                        adj_RIB_IN(ORIGIN, (H,T1),LocPrf1,Length1, NEXT_HOP1),
                        adj_RIB_IN(ORIGIN, (H,T2),LocPrf2,Length2, NEXT_HOP2),
                        LocPrf2 > LocPrf1.

step1_MAX_LocPrf(ORIGIN, (H,T),LocPrf,Length, NEXT_HOP):-
                    adj_RIB_IN(ORIGIN, (H,T),LocPrf,Length, NEXT_HOP),
                    not step1_notMAX_LocPrf(ORIGIN, (H,T),LocPrf,Length, NEXT_HOP).

step1_continueFLAG(H,ORIGIN) :- step1_MAX_LocPrf(ORIGIN, (H,T1),LocPrf1,Length1, NEXT_HOP1),
                            step1_MAX_LocPrf(ORIGIN, (H,T2),LocPrf2,Length2, NEXT_HOP2),
                            LocPrf1 = LocPrf2,
                            T1 != T2.
```

Then we use the first attribute -- local preference to select the best path .If there are more than one path left, the second step in decision process is select path with minimal path length.

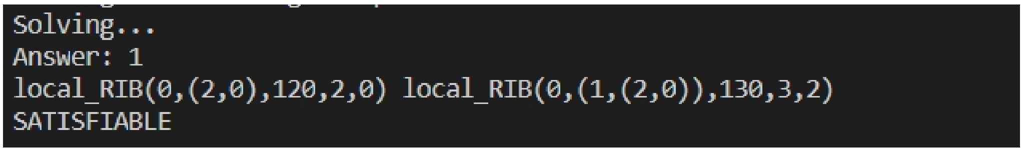


```
step2_notMIN_Length(ORIGIN, (H,T1),LocPrf1,Length1, NEXT_HOP1):-%%% use step1_continnue here
                        step1_continueFLAG(H,ORIGIN),
                        step1_MAX_LocPrf(ORIGIN, (H,T1),LocPrf1,Length1, NEXT_HOP1),
                        step1_MAX_LocPrf(ORIGIN, (H,T2),LocPrf1,Length2, NEXT_HOP2),
                        Length2 < Length1.

step2_MIN_Length(ORIGIN, (H,T),LocPrf,Length, NEXT_HOP):-
                    step1_MAX_LocPrf(ORIGIN, (H,T),LocPrf,Length, NEXT_HOP),
                    not step2_notMIN_Length(ORIGIN, (H,T),LocPrf,Length, NEXT_HOP).

step2_continueFLAG(H,ORIGIN) :- step2_MIN_Length(ORIGIN, (H,T1),LocPrf1,Length1, NEXT_HOP1),
                            step2_MIN_Length(ORIGIN, (H,T2),LocPrf1,Length2, NEXT_HOP2),
                            T1 != T2,
                            Length1 = Length2.
```

At last, we use minimal next hop as break tie. The Loc-RIB contains the local routing information the BGP speaker selected by applying its local policies to the routing information contained in its Adj-RIBs-In. These are the routes that will be used by the local BGP speaker.



```
Reading from .\BadGADGET.lp
Solving...
UNSATISFIABLE
```



```
Solving...
Answer: 1
local_RIB(0,(2,0),120,2,0) local_RIB(0,(1,(2,0)),130,3,2)
SATISFIABLE
```

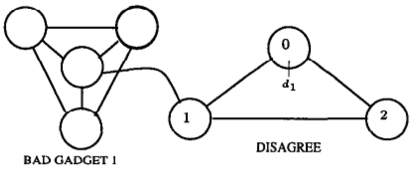
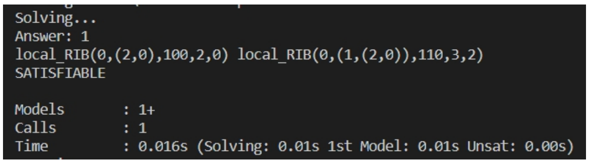


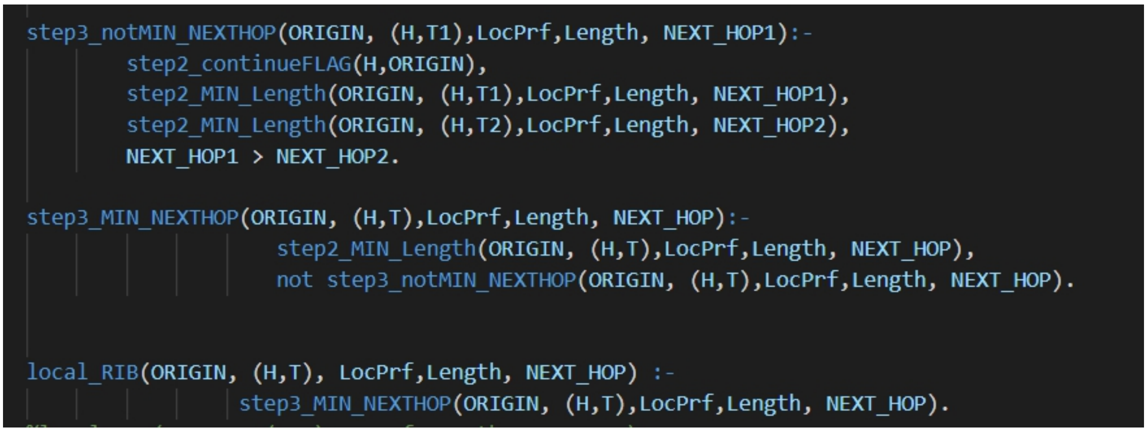
Figure 6: AS graph for PRECARIOUS.



```
Solving...
Answer: 1
local_RIB(0,(2,0),100,2,0) local_RIB(0,(1,(2,0)),110,3,2)
SATISFIABLE

Models       : 1+
Calls        : 1
Time         : 0.016s (Solving: 0.01s 1st Model: 0.01s Unsat: 0.00s)
```

We can use ASP to properly configure the routing information base for each BGP Router and select the best BGP path for them. And using this analysis, we can prevent or avoid persistent oscillations in general topologies.

For example, we can detect the Badget, because we cannot find any stable path for it. Also, for disagree situation, we randomly pick one group of paths as stable path. Precarious is tricky, a solvable BGP system have a trap in its evaluation Graph. but we still find a solution for it. A sub-graph of this system is equivalent to the system DISAGREE presented above,BAD GADGET1



```
step3_notMIN_NEXTHOP(ORIGIN, (H,T1),LocPrf,Length, NEXT_HOP1):-
        step2_continueFLAG(H,ORIGIN),
        step2_MIN_Length(ORIGIN, (H,T1),LocPrf,Length, NEXT_HOP1),
        step2_MIN_Length(ORIGIN, (H,T2),LocPrf,Length, NEXT_HOP2),
        NEXT_HOP1 > NEXT_HOP2.

step3_MIN_NEXTHOP(ORIGIN, (H,T),LocPrf,Length, NEXT_HOP):-
                    step2_MIN_Length(ORIGIN, (H,T),LocPrf,Length, NEXT_HOP),
                    not step3_notMIN_NEXTHOP(ORIGIN, (H,T),LocPrf,Length, NEXT_HOP).


local_RIB(ORIGIN, (H,T), LocPrf,Length, NEXT_HOP) :-
                 step3_MIN_NEXTHOP(ORIGIN, (H,T),LocPrf,Length, NEXT_HOP).
```

is configured to oscillate only when its center node accepts this route to d1.

Therefore, this system has only one solution: when AS 2 accepts the direct route to d1 and AS 1 accepts the route through AS2 to d1.

ASP is powerful tool for reasoning and analysis. In our implementation, we just use three BGP attributes in Decision Process, we will add more attributes in decision process in the future, and also try to apply our ASP policies on larger networking topologies.

Reference

An Analysis of BGP Convergence Properties, Timothy G. Griffin Gordon Wilfong

DeepBGP: A Machine Learning Solution to reduce BGP Routing Convergence Time by Fine-Tuning MR, Ricardo Bennesby

*Clingo = ASP + Control: Preliminary Repor*t, Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub

A User's Guide to gringo, clasp, clingo, and iclingo, Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Max Ostrowski, Torsten Schaub, Sven Thiele

Persistent Route Oscillations in Inter-Domain Routing Kannan Varadhan， Ramesh Govindan， Deborah Estrin

Answer Sets Michael Gelfond

Answer set programming https://en.wikipedia.org/wiki/Answer_set_programming