HTS: A Hierarchical Method for Load Balancing in Autonomous Networks

MohammadReza HeidariNezhad, Zuriati Ahmad Zukarnain, Nur Izura Udzir and Mohamed Othman

Faculty of Computer Science & Information Technology, University Putra Malaysia, 43400 Serdong, Malaysia

Summary
The load balancing is the most important issues to reduce congestion in Autonomous computer networks. The inter-domain traffic control methods have been addressed by researchers. In this paper, we are proposed HTS; a BGP-based hierarchical network traffic reconfiguration method to redistribute traffic on overloaded link for IP-based network. Proposed algorithm fulfills at two steps. Firstly overload detection function through an explorer algorithm to construct overloaded link table due to disruption link. Finally HTS scheme scattering traffic on overloaded path to exterior level. The performed experiments have been shown efficiency of proposed method in load balancing and good performance when the offered traffic loads change randomly

Key words:
Autonomous Networks, Load Balancing, Hierarchical Method, Traffic Management.

1. Introduction

Autonomous configurable networks are a type of Autonomous Systems (ASes) that typically made from a network of homogeneous or heterogeneous reconfigure-able modules or agents. They can autonomously change their physical or logical connections and rearrange their configurations [1]. In telecommunications domain, an AS is often called access network. The objects that create network can seem as individual agents that communicate together. The Internet is a most famous AS that collected of different autonomous subnet connected together by the core (backbone) network as shown in Fig 1.
Routing messages in networks is an essential component, as each IP packet in the Internet must be passed through each network (or AS) that it must traverse from its source to destination. Inside the AS, the routers compute path with run a distributed algorithm. Indeed, current interior routing protocols, such as OSPF, RIP, and IEGP are based on methods, as are many exterior protocols, such as BGP [2] and EGP [3]. The Border Gateway Protocol (BGP) is the current de facto standard for inter-domain routing protocol [4]. In BGP terminology, a domain is called AS.

Manuscript received June 5, 2008.
Manuscript revised June 20, 2008.

Hence in this paper we address a BGP-base network reconfiguration method, regardless of topology or type of protocols to scatter traffic at the constrain links. In the especial case, we investigate intermediate component configuration in an IP based composite networks that each router act as an autonomous agent.

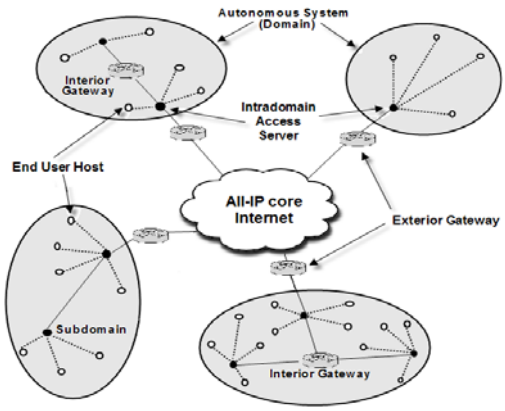


Fig.1. Internet Architecture as AS

The remainder of this paper is organised as follows. In section two, we explain the inter-domain traffic control as a brief background, and the required modifications to BGP. In section three, we describe in detail proposed algorithm. We evaluate the performance of our method by present simulations results in section four. We compare our approach with related work in section five. The paper is concluded in section six.

2. Inter-domain Traffic Control

Internet routing in IP-based networks is handled by intra-domain (inside boundaries) and inter-domain (between boundaries) protocols. Inside a single domain, traffic is controlled based on network topology and organization policies. Across domains, the inter-domain routing protocol is used to distribute reachability information and is only aware of the interconnections between distinct ASes due to scalability. Actually BGP is a path-vector protocol that works by sending route advertisements [5].

The BGP routers use an input filter to select the acceptable advertisements and output filters to select the best routes in the BGP routing tables [6]. But the default BGP routing standard acquiesces of some deficiency. To address this effect, we introduce an extension rather than slightly modify the BGP routing protocol.

3. Hierarchical Traffic Model

Link fault is one of the main causes for the link overloading. This event influences performance of whole network due to delay in packet delivery. If we consider ASes links as a hierarchical structure, congestion of one peer cause overload traffic in exterior peers. To understand this point, consider the simple network shown in Fig. 2. In order to reconfiguration of links, a novel approach called Hierarchical Traffic Scattering (HTS) is proposed. Actually HTS is a cooperative method that provide best path configuration for ASes router's table.

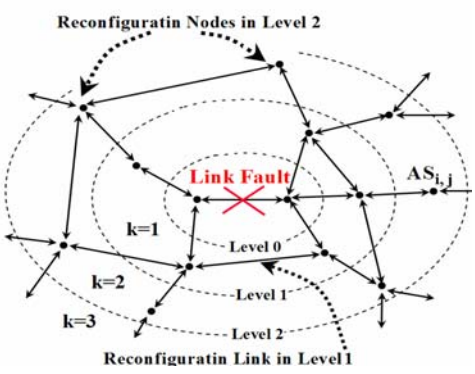


Fig. 2. Levels in HTS

Routers in different ASes use BGP routing protocol to exchange update messages about how to reach different destination. The main idea behind this scheme is that by starting an overloaded path; try to find adjacent path in exterior layers in order to packet rerouting (Level 0) and continue to upper level. Rerouting is not necessarily performed at the constraint node of the congested link, as all agents lying on routes that pass through the congested link shall be considered by the algorithm for a possible change of their routing tables. A router sends an announcement to notify its neighbor of a new route to the peer destination. We complete this process with broadcasting an identification message to surveillance the route when time traveling packets lapsed. Each advertisement includes additional attributes about the route tables, including the list of ASes along the path to the destination. The exterior level, called k traced by this advertisement. Before accepting an advertisement, the receiving router investigate message for the presence of duplicated AS number in the AS path to detect and remove routing loops. We assumed inter-domain traffic as an assignment problem in combinatorial optimization. This problem consists in assigning the traffic t_{ij} towards each destination i through provider j to minimize the total value of the assignment $\sum_{i=1}^{k}\sum_{j=1}^{k} C_{ij}t_{ij}$, where C_{ij} represent the total cost of sending traffic through agent destination i through agent j. We used maximum of this value to calculate link's threshold [7]. The k indicate number of layers need to continue that determine by overload detection phase. We assumed the cost of sending traffic per links is determinable before examination through cost vector. Note that type of relationship between domains, i.e. customer-provider and peer-peer does not affect on our method. Dynamic traffic for analytical model and link failures created through stochastic pattern. Proposed method has two basic steps as follow:

3.1 Overload Detection

This method is an iterative approach to visit all links at given sub network(Alg.1). Overload detection is performed by token-based broadcasting UDP message that traveled from initial node to next hop through the adjacent links.

Alg. 1: Overload Detection Phase

```
1.  { Input :
    Initial Network (AdjacentTable[n][m], CongestionTable[n][m])}
2.  { Output : OverloadRouteTable[n][m] , k}
3.  Initialization : k ←1
4.  Overload Detection Function
5.   for i ∈ [1..n] do
6.    for j ∈ [1..m] do
7.      for each NextHop ∈ AdjacentTable[i][j]
8.        [i..k][j..k] ← Link of max value in CongestionTable[n][m]
9.        Broadcast (TokenMessage[i..k][j..k], NextHop)
10.       DelayEcho[i..k][j..k] ← Time of
                      ReceivingMessage(CurrentHop, NextHop)
11.       Threshold [k..i][k..j] ←
          max Σ_{i=1}^{k} Σ_{j=1}^{k} Congestion Table[k..i][k..j]
    {Threshold[i][j] has been extracted from
CongestionTable[n][m]}
12.       CurrentHop ← NextHop
13.       NextHop ← New NextHop by min value in AdjacentTable
14.     if DelayEcho[i..k][j..k] is greater than Threshold[i..k][j..k]
            then
                          {Overload occurred}
15.       Update(OverloadRouteTable[i][j] with
            DelayEcho[i..k][j..k])
16.     end if
17.       Update (CongestionTable[n][m] with Threshold[i..k][j..k])
18.     Increment(k)
19.     end for
20.    end for
21.  end for
22. return (OverloadRouteTable[n][m] , k)
23. end                          {Function }
```

The needed time to pass this packet, used as criterion to overload occurrence whereas links abnormal traffic. *AdjacentTable* is containing of link load between each peer and *CongestionTable* stores the congested links in the search space. Maximum value in *CongestionTable* candidates root link to calculate *OverloadRouteTable* (line 13-16).

3.2. Hierarchical Traffic Scattering (HTS)

HTS algorithm (Alg. 2) runs after overload detection phase and is dynamic programming-based method to produce *BestLoadTable*. This matrix contains the best load distribution (lines 11 and 14) that used in the successive BGP filters in order to apply on table of routers (Figure 3). This scheme used result constructed by previous method; *OverloadRouteTable* to reconfiguration traffic hierarchically. Nested loops are used to scan all paths through *AdjacentTable*. Lines 10–15 includes the core of proposed algorithm. Line 17 replaces all old value *OverloadRoutTable* with new entry in *BestLoad* table. The delay of paths, in terms of maximum route latency, is collected into *PathDelay* table.

Alg 2: Hierarchical Traffic Scattering (HTS)

1. { **Input** : Initial Network(AdjacentTable[n][m], TimeTable[n][m], OverloadRoutTable[n][m] , k)}
2. { **Output** : BestLoadTable[n][m] for reconfiguration links from level k }
3. **Initialization** :
4. BestLoadTable[1..n][1..m] ← ∞
5. **HTS Function**
6. **for i** ∈ [1..n] **do**
7. **for j** ∈ [1..m] **do**
8. PathDelay[i..k][j..k] ← TimeTable[i..k][j..k] ⋃ AdjacentTable[i..k][j..k]
9. $W_{i,j} \leftarrow \sum_{i=1}^{k}\sum_{j=1}^{k}$ OverloadRoutTable[i..k][j..k]
10. **if** PathDelay[i..k][j..k] is greater than $W_{i,j}$ **then**
11. BestLoadTable [i+k][j]← $\min_{i=1..k,\, j=1..m}\{TimeTable[n][m]\}$
12. Reset(TimeTable([i+k][j])
13. **else**
14. BestLoadTable [i][j+k]← $\min_{i=1..n,\, j=1..k}\{TimeTable[n][m]\}$
15. Reset(TimeTable([i][j+k])
16. **end if**
17. Update(OverloadRoutTable with BestLoadTable)
18. Restore (k)
19. **end for**
20. **end for**
21. **return** (BestLoadTable[n][m])
22. **end** *{ Function }*

If the load obtained after each traffic re-calculate ($W_{i,j}$) is less than *PathDelay*, then the *BestLoadTable* is updated to the minimum set of existing TimeTable (lines 9-10); therefore overloaded traffic on congested link is scattered upon adjacent ones in exterior level through increasing K. After iteration, the PathDelay has been reinitialized.

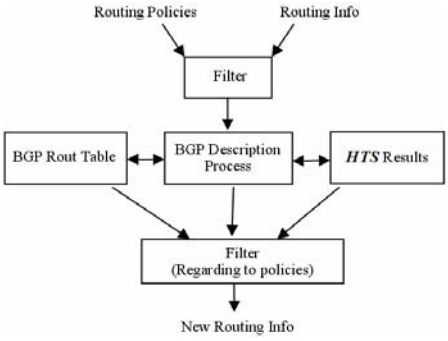


Fig. 3. Operation of HTS and BGP routing protocol

4. Computational Experiment Results

In this section, we explain result of study the behavior of our algorithm. For most realization, we use the Internet-like grid topology, real BGP routing tables and traffic matrix provided by random generator function and contains maximum 1,000 domains and 2,000 inter-domain links. At most, one link considered between peer-to-peer domains. In our experiments, we have been used traffic loaded per links as objective function. This objective function is the maximum amount of the traffic exchanged with all providers on any of the inter-domain links. The routing policy in each router is configured base on AS-path length in given topology. The represented traffic pattern is scaled from 160% down to 0% .Amount of traffic has been measured in a specific period time (900 sec). The result of our experiments is shown in figure 4 and 5. First simulation scenario started with maximum link overload for default BGP protocol. Second and third scenario is repeated for traffic destitution between layer 0 to layer k and layer k + 1 to maximum existing layer, respectively. The value k is specified by overload detection method in each scenario.

T_A and T_B represent the result of BGP fortified by HTS to redistribution of overloaded traffic. All results from simulation have been shown that there is a strong relationship between k and descent rate to achieve stable redistribution state. Also significant of packet bulk in overloaded links have been forwarded to exterior layer from L_{k+1} to L_{Max} in the simulation period. L_{Max} represent the outermost layer base on *AdjacentTable*. Simulation replication indicated that the overhead of overload detection phases due to packet traveling is not more than 10% latency of network reconfiguration.

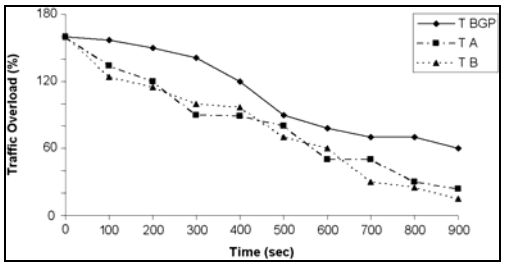


Fig. 4. Simulation Results
TBGP: Traffic distribution by default BGP protocol
TA : Traffic redistribution between L_0 to L_K
TB: Traffic redistribution between L_{K+1} to L_{MAX}

We consider number of hop-to-hop forwarding of multicast packets as a cost metric. This cost assumed as number of visited node per total node (ASes) in simulated grid network (figure 5). As the simulations with three scenario show, increasing links in broadcasting scheme to exterior layers, provide a good traffic balance among all the available ASes but relatively high cost and vice versa.

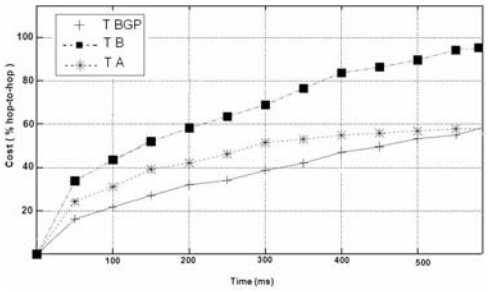


Fig. 5. Simulation for cost evaluation

5. Prior Related Work

Today, BGP is adapted to minimize the disruption of the networks due to overloaded link traffic. Several techniques proposed to control the network traffic. Most of these proposals are based on the centralised optimization algorithm and using logical path. For instance, in [8] used executed periodically method and recalculates the entire logical network using traffic statistics and predictions. The logical network is modified using these results. An alternative approach, [9] introduced inter-domain traffic control for multiple links based on genetic algorithm. This method specifies a link for prefix in AS neighbors to minimizing costs and configuration changes. On the other hand, fast restoration mechanisms have led to the use of backup paths (local, global, etc). When a fault affects a working path the traffic is then switched to the backup path. This also modifies the logical network. It is also important to perform a good spare capacity allocation, and there are schemes where the backup paths can share their bandwidth. Similarly, in attempt to meet traffic management, [10] introduced virtual peering method that is an IP tunnel between border router as source AS to destination AS router. This tunnel is established upon request from the destination AS by using backward compatible modifications to the BGP.

In [11] proposed another method to finer control on the incoming traffic but their method is difficult to use in practice due to the incomplete view of the whole Internet.

As another instance, in [12] a load-balancing system is proposed and evaluated. This system allows controlling the incoming and outgoing traffic with relies on NAT for small enterprise networks. Unfortunately do not consider this system to be applicable for large stub ASes such as broadband access providers.

6. Conclusion and Future Work

The agents in autonomous domain can adapt to change the distribution of incoming/out coming traffic by adjusting the configuration of packet forwarding. In link failure situation or increasing number of ASes, default BGP showed poor performance in load balancing due to the tie-breaking in the decision process. Furthermore, our studies have been shown that traffic balancing between ASes in the default BGP is not satisfiable in all cases.

The load balancing is a multi-mission problem that usually investigates using a dynamic based method to archive trade-off between system objectives. However we present a novel scheme; HTS, that carries out a BGP-based hierarchical traffic management method to find and scatter traffic on overloaded links in efficient manner.

The main difference between our solutions with other approach is that it does not substitute the traffic control approach but improve the BGP's performance. This method also has satisfiable potential to achieve a good scalability due to redistribution of packets. The results of our work indicate validity of this approach.

First guideline for future work in this direction is investigation additional objective functions, e.g. bandwidth limitation for archive more general results. Moreover implementing this method in a more realistic simulation environment, motivate the next studies.

References

[1] A Distributed Autonomous-Agent Network-Intrusion Detection and Response System Available online at: http://www.cs.nps.navy.mil.

[2] Rekhter, Y. and T. Li., *A border gateway protocol 4 (bgp-4).* Internet draft, draft-ietf-idr-bgp4-17.txt, in progress, May 2002.

[3] N. Feamster, J.B., and J. Rexford, *Guidelines for interdomain traffic engineering.* ACM Computer Communication Review, October 2003. 33.

[4] B Quoitin, O.B., *A cooperative approach to interdomain traffic engineering.* Next Generation Internet Networks, 2005: p. 450- 457.

[5] L. Swinnen, S.T., S. Uhlig, B. Quoitin, and O. Bonaventure, *An Evaluation of BGP-based Traffic Engineering Techniques.* Technical Report Infonet, 2002.

[6] D. Awduche, et al., *Overview and principles of internet traffic engineering.* http://www.ietf.org/rfc/rfc3272.txt.

[7] Mohammad Reza HeidariNezhad, et al., *Load Balancing in Autonomous Networks through Hierarchical Traffic Scattering*. In *Proceeding of International Conference on Computer and Communication Engineering*. 2008. Malaysia.

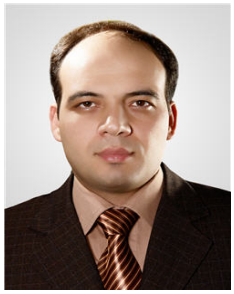
[8] B. Quoitin, S.T., S. Uhlig, and O. Bonaventure, *Interdomain Traffic Engineering with Redistribution Communities.* Computer Communications Journal (Elsevier), March 2004. 27: p. 355-363.

[9] DaDong Wang, H.W., YuHui Zhao, Yuan Gao, *Interdomain Traffic Control over Multiple Links Based on Genetic Algorithm*, in *Proceeding of 3rd International Networking and Mobile Computing Conference* 2005: China.

[10] José L. Marzo, P.V., Santiago Cots, Eusebi Calle *Distributed Architecture for Dynamic Resource Management.* China Communications Magazine, June 2005. 2(3).

[11] S. Uhlig, O.B., and B. Quoitin. *Interdomain Traffic Engineering with minimal BGP Configurations*. in *Proceeding of of the 18 International Teletraffic Congress*. September 2003. Berlin.

[12] F. Guo, J.C., W. Li, and T. Chiueh, *Experiences in Building a Multihoming Load Balancing System.* in Proceedings of IEEE INFOCOM, 2004.



MohammadReza HeidariNezhad obtained his B.Eng and M.Eng degrees in Computer Engineering field in 1996 and 1999, respectively. He is currently Ph.D. candidate in faculty of Computer Science and information technology in the University Putra Malaysia (UPM). He already published fifteen books related to computer science and network. His research interests include resource and traffic engineering, mobility management and 3G/4G wireless network.

Dr. Zuriati Ahmad Zukarnain is lecturer at the faculty of Computer Science and Information Technology, University Putra Malaysia (UPM). She received her B.S. and M.S. degrees from UPM in 1993 and 1997, respectively. She obtained her Ph.D. from university of Bradford, UK in 2006. She currently is head of department of Communication Technology and Network at the faculty. Her research interests include computer networks, quantum computing and distributed systems.

Dr. Nur Izura Udzir is a Senior lecturer at the Faculty of Computer Science and Information Technology, University Putra Malaysia (UPM). . She obtained her B.S. and M.S. in Computer Science from UPM. She obtained her Ph.D. in Computer Science from University of York, UK in 2006 She is currently leading the Information Security Group at the faculty. Her research interests include coordination models and languages, access control in open distributed systems

Dr. Mohamed Othman is an Associate Professor at the Faculty of Computer Science and Information Technology, University Putra Malaysia (UPM). He obtained his Ph.D. from National University of Malaysia in 2000. He is member of IEEE Computer Society, IEICE Communication and Engineering Science Societies, Malaysian National Computer Confederation and Malaysian Mathematical Society. He already published more than one hundred National and International journal papers and more than three hundred conference papers. His research interests include parallel and distributed algorithms, grid computing, high-speed computer network and network management (security, wireless and traffic monitoring).