Dynamics of Load-Sensitive Adaptive Routing

Hao Wang
Carleton University
Ottawa, ON, Canada
nhwang@sce.carleton.ca

Mabo R. Ito
University of British Columbia
Vancouver, BC, Canada
mito@ece.ubc.ca

Abstract—Shortest Path First (SPF) routing protocols, such as OSPF and IS-IS, are currently the dominant intra-domain IP routing protocols and are widely used in the ISP backbones. Although the traffic on the Internet is highly dynamic, OSPF and IS-IS are not adaptive to the changing traffic, because the shortest path generated by these protocols are based on the link weights, which are fixed and usually can not be changed during network operation. This paper investigates a way of changing the weights in OSPF/IS-IS adaptively according to the changing traffic load on the links. The feedback effect and the stability issue of adaptive routing are analyzed from a control system point of view. The paper shows why Minimal-Delay Adaptive Routing, such as the routings in the early ARPANET, is not stable, and proposes some techniques to make our Load-Sensitive Adaptive Routing (LSAR) stable. Finally, the performance of LSAR is evaluated by simulation. The result shows that LSAR can significantly improve QoS of the network by increasing network throughput and reducing packet drop ratio.

Index terms—Adaptive Routing, OSPF, IS-IS, Shortest Path First (SPF) routing, Traffic Engineering, Load Balancing, QoS

I. INTRODUCTION

Shortest Path First (SPF) routing protocols such as OSPF [14] and IS-IS [15] are the dominant intra-domain Internet routing protocols today, and are widely used in the Internet Service Provider (ISP) backbone networks. The weights, or the lengths, of the links in an OSPF/IS-IS network are set by the network operator and usually are not changed during the network operation. The traffic is routed along the shortest paths from the sources to the destinations. Because the link weights are fixed, the paths between the origin and destination pairs are fixed regardless the traffic load changes.

As suggested by Cisco, a simple default weight setting for OSPF is to make the weight of a link inversely proportional to its bandwidth [13]; while for IS-IS, the default weight setting of all links are the same value of 10 [16]. It is obvious that such simple weight settings cannot always provide good routing performance in terms of avoiding congestion for various network topologies, traffic patterns and traffic loads. Some work has been done on optimizing the link weights in OSPF/IS-IS networks for some particular network topologies and traffic loads [6] [7] [8] [9] [10], and it has been shown that by properly set the weights in an OSPF/IS-IS network, significantly better network throughput and performance can be achieved compared to the simple default weight settings.

The limitation of previous research is that the optimizations were done offline based on a fixed traffic demand matrix and a fixed network topology. The weights of the links are still static during the network operation, and the routing algorithm is not adaptive. As it is well known, the traffic on the Internet is very dynamic. The traffic pattern, traffic load and network topology change from time to time. One set of weights can only work well for some particular traffic loads, but not for the others.

The objective of this paper is to investigate a way of adapting the weights in SPF routing, such as OSPF and IS-IS, in real-time according to the traffic loads on the links and to evaluate its performance over non-adaptive routing. The feedback effect and the instability issue of adaptive routing are studied, and a new routing algorithm, Load-Sensitive Adaptive Routing (LSAR) which is adaptive but without the undesirable unstable behavior, is presented. Due to the extremely complex nature of the dynamics of packet switched data networks and the lack of an effective mathematic tool to model such complex dynamic systems, the analysis in this paper is mostly qualitative rather than quantitative.

For detailed information about this work, please refer to [17].

II. LOAD-SENSITIVE ADAPTIVE ROUTING

The first adaptive routing algorithm implemented in the early ARPANET was a Minimal-Delay Adaptive Routing (MDAR) algorithm [4] [5]. But soon it was noticed that such an algorithm had a severe oscillation problem. The cause of the routing oscillation is because of the feedback effect between the route choice and the link delay [1] [2]. To be more specific, the route choice of the routing algorithm is based on the measurements or estimates of the link delay, and the link delay is affected by the route choice in return. To avoid unstable routing, which severely degrades the performance of the network, later routing protocols abandoned the idea of making the routing adaptive to the traffic load.

A. Weight Mapping in LSAR

When treated as a feedback system, the system block diagram of MDAR implemented in the early ARPANET can be drawn as shown in Fig. 1. The input to this system is the changing traffic load, which changes the utilization of the links in the network. The routing algorithm changes the weights of the links according to the mapping function between the link utilization and the link weight. In MDAR, the weight is set proportional to the link delay, so the weight mapping curve has the shape of a queuing delay curve. The output of the weight mapping is the updated weights of the links, which become the input of the Shortest Path First (SPF) algorithm, which calculates the shortest path for each origin destination pair in the network and updates the routing tables of the routers. The routers route the traffic using the updated route tables, therefore the traffic load on each link of the network changes because of the route update. The weight mapping, the SPF algorithm and the route update blocks form a feedback loop of the system, which, from control theory, determines the dynamics of the system.
The weight mapping of MDAR can be expressed as
\[ W = W_0 + D = W_0 + d(u) \] (1)
where \( W \) is the output weight, \( W_0 \) is a constant, and \( D \) is the link delay which is a function of link utilization \( u \).

Bertsekas concluded in [1] that the reason for the unstable oscillation of MDAR is that the value of the derivative \( d' \) is too large when the network is heavily loaded, in other words when \( u \) approaches 1. To put the same conclusion in the words of control theory, the reason for the instability of the system is because the gain of the feedback loop is too high when \( u \to 1 \), since \( W' = d' \) and \( W' \) represents the gain of the weight mapping block in the block diagram.

The function \( d(u) \) is a nonlinear function, because of the queuing nature of link delay. The derivative \( d' \) is relatively small when the network is lightly loaded but increases dramatically when the load is heavy. A small gain in the feedback loop usually means less sensitivity and more stability for the system; a large gain in the feedback loop usually makes the system more sensitive but can also lead the system to instability if it is too large. This matches the behavior of MDAR very well. When the network is lightly loaded, the routing is stable, but when it is heavily loaded, routing oscillation appears. Given the above analysis, it is clear that the nonlinear weight mapping function in MDAR is the cause of instability. Therefore, in our proposed LSAR, linear mapping functions are used, which means that the weight mapping block has a constant gain.

Fig. 2 shows a weight mapping function of LSAR, with maximal value of 4. The flat section of the mapping function from link utilization value 0 to 0.5 is called the non-adaptive area, within which the weight remains constant. The reason for having this non-adaptive area is that when the link is lightly utilized, the link performance in terms of delay and packet loss rate is good enough, so it is not necessary to change the weight. By doing so, it can also save routing overhead and router CPU loading. The section after the non-adaptive area is a straight line increasing proportionally to the link utilization. The maximal value of the weight mapping function is usually set to be smaller than 4, because in networks that are rich with alternate paths, on average, the weight of 4 is big enough to shed all the traffic off a link [3]. The larger the value, the more powerful the routing is in balancing traffic. But there is a trade-off between the traffic balancing power and the stability of the routing algorithm. In the simulations shown later in this paper, the maximal weight is set to be 3.

-changing the weight mapping function in the feedback loop does help improve the stability of the system, but it is not enough. In the next section, the dynamics of adaptive routing systems is further analyzed by studying the dynamic behavior graphs.

B Damping by Averaging

For the sake of simplicity, let’s consider an adaptive routing network where there is only one link changing its weight, while the rest of the links keep their weights constant. It is also assumed that the traffic flows over the network is static. Fig. 3 shows the dynamic behavior of the weight-changing link in such a network.

The horizontal axis in Fig. 3 denotes the utilization of the link. The vertical axis denotes the normalized weight assigned on this link. The curve increasing with link utilization represents the weight mapping function, which can be written as \( w = f(u) \), where \( u \) is the link utilization and \( w \) is the recalculated weight based on the link utilization of \( u \). The step-shape decreasing curve represents the resulting link utilization corresponding to an assigned weight, and it is referred to as \( u = g(w) \). The shape of \( g(w) \) is determined by the discrete nature of network flows. The shape of \( f(u) \) does not affect our analysis, as long as it is a monotonically increasing function with \( d(0) \geq 0 \). The point where \( f(u) \) and \( g(w) \) intersect is called the equilibrium point \( P \).

In Fig. 3, the system starts with initial weight \( w_0 = 2 \). From the load curve, it can be deduced that the link utilization resulting from this weight assignment \( u_1 = g(w_0) \) is about 0.7. The routing algorithm calculates the weight of the link by checking the weight mapping curve, so \( w_1 = f(u_1) = 4.3 \). Then the link weight of 4.3 is updated, and the load over the link changes accordingly, \( u_2 = g(w_1) = 0.15 \). Consequently, the weight of the link changes again, \( w_2 = f(u_2) = 1.2 \). The routing algorithm sets the weight to 1.2 in the next update, and results the change of link utilization \( u_3 = g(w_2) = 0.8 \). Then the link utilization of 0.8 leads to a link weight update, \( w_3 = f(u_3) = 5 \).

From then on, the system enters a circle where the link utilization oscillates between 0.15 and 0.8, which means route flipping for many of the origin-destination pairs in the network.
In the MDAR of early ARPANET, the weight was updated based on the link delay, so the weight mapping curve had the same shape of a queuing delay. The routing is only stable when the link is lightly utilized, because when the link utilization is small the delay curve is approximately flat. One way to increase the stability of the routing algorithm under relatively heavy utilization, as suggested in [1], is to add a large value of bias to the weight mapping function.

In our LSAR, another approach other than simply adding a bias is used. By observing the dynamic behavior of unstable adaptive routing algorithms, such as the one shown in Fig. 3, it is noticed that the unstable oscillating routing behavior is the typical behavior of a system that lacks enough damping. This suggests that the stability of the routing algorithm can be improved by adding damping factors into the system. The way of adding damping factor into the system is by averaging, which is a very common technique in control system design to introduce damping into a system. The routing algorithm still calculate the weight of the link using the weight mapping function, but instead of updating the link weight directly by this calculated value and broadcasting it to the whole network, the routing algorithm uses the average of this value and the previous updated link weights. The number of previous weights used in averaging determines the effect of damping. The more previous weights used, the more damping the system has, and the more stable the system can be. But there is a trade off between the stability and the response speed of the system. By experiments, the number of previous weights used in the averaging is set to be 3 in our simulations.

C The Minimal Weight Update

By adding damping factors into the feedback system, it is hoped that the system will gradually approaches the equilibrium point and remains there. But if take a closer look at the equilibrium point in the dynamic behavior graphs, it can be noticed that there are actually two types of equilibrium points, as shown in Fig. 4.

In Fig. 4 (a), the equilibrium point is on a horizontal section of the traffic load curve, and it is called an unstable equilibrium point. Suppose at time $t_1$, the system is at the equilibrium point, where the link utilization $u = u_e$. After a period of time at time $t$, the link updates its weight according to $u_e$ and this update results the link utilization changing to $u''$. Then at $t_2$, the weight is updated again and the utilization changes to $u'$. At $t_3$, the utilization changes back to $u''$, and so on. The oscillation starts.

The equilibrium point in Fig. 4 (b) is on a vertical section of the traffic load curve, and it is stable, because once the system is within the area between $u'$ and $u''$, it will move to the equilibrium point after one link weight update and will remain there.

It is obvious that an adaptive routing system can not be stable if the equilibrium point itself is not. But there is no way to guarantee that the equilibrium point will always be on the vertical section of the traffic load curve, unless the weight mapping curve is a flat straight line, which means no adaptive routing at all. In order to deal with the unstable equilibrium point problem, a minimal weight update limit is introduced into our LSAR algorithm.

In LSAR, the link calculates its weight using the weight mapping curve and averages this value over previous updated weights. The result after averaging is the weight value to be update at the next update time. But if the weight change between the current weight and the to-be-updated weight value is smaller than the minimal weight update limit, the weight value will not be changed at the update time, instead the link will keep its previous weight.

The routing oscillation around the unstable equilibrium point usually has small weight changes between each link updates, therefore by having a minimal weight update limit, the oscillations around the unstable equilibrium point can be prevented. It also helps to reduce the frequency of link weight updates in the network, therefore reducing routing overhead and saves the CPU resource of the router. The larger the minimal weight update limit, the more stable the system could be, but the less sensitive the routing algorithm is to the traffic changes. Considering the trade off and by experiments, the value used in our simulation is 20% of previous weight. The advantage of using a percentage instead of a fixed value is that we can have a variable sensitivity system. When the traffic load increases, the network is more likely to oscillate. Because the weight increases with the traffic load, the minimal update limit increases, as a result the routing decreases its sensitivity and increases stability.

The above analysis about the dynamics of adaptive routing shows that the stability of adaptive routing can be improved by changing the weight mapping function, adding damping factor and limiting the minimal weight update. Fig. 6 shows the system block diagram of our proposed LSAR, which includes these features.

In the next section, the performance of LSAR is evaluated by simulation. The simulation software used in our simulation is Network Simulator 2 (ns2).

III. SIMULATION

LSAR has some parameters that can be adjusted, such as the range of non-adaptive area on the weight mapping curve, the
Table 1 The parameters of LSAR.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATIC_AREA</td>
<td>0.5</td>
<td>The range of non-adaptive area on the weight mapping curve.</td>
</tr>
<tr>
<td>MAX_WEIGHT</td>
<td>3</td>
<td>The maximal weight value.</td>
</tr>
<tr>
<td>NUMBER_OF_AVERAGE</td>
<td>4</td>
<td>The number of weights used in averaging.</td>
</tr>
<tr>
<td>MIN_CHANGE</td>
<td>0.2</td>
<td>The minimal weight change between updates.</td>
</tr>
</tbody>
</table>

maximal weight value, the number of weights used in averaging, and the minimal weight update limit. By changing these parameters, the dynamic behavior of LSAR can be adjusted. Considering the trade offs mentioned earlier in the paper and after doing experiments, the parameters and their default values are set as shown in Table 1.

The topologies used in the simulations are randomly generated using the GT-ITM Topology Generator software. The performance of LSAR on both static traffic load and dynamic traffic loads are tested.

A Static Traffic Test

In static traffic simulation, the flow between each origin destination pair is implemented by a constant bit rate UDP flow. The rates of the traffic flows are determined using a method similar to the one used in [6]. By controlling the seeding of the random number generator, the traffic flows on the comparing LSAR and OSPF/IS-IS routing networks are kept exactly the same. The traffic flow patterns for a set of simulations performed on the same network topology are also the same, except the variation of the traffic load factor.

Our LSAR is tested on six different random topologies, and the simulation results are shown from Fig. 6 to Fig. 11.

For each random topology used, several simulations for different traffic loads are run. The results show that for fixed weight OSPF/IS-IS routing, the maximum link utilization in the network increases linearly as the traffic load increases, which is expected because the routing is fixed. When the maximum link utilization reaches 1, this means the traffic load over a link in the network has reached its capacity and congestion occurs.
LSAR is then used instead of OSPF/IS-IS in the same networks with the same traffic flows. It can be seen that when the traffic load is light, LSAR performs the same as OSPF/IS-IS with linearly increasing maximum link utilization. This is the effect of the non-adaptive section on the weight mapping curve. But when the traffic load becomes relatively heavy, the routing become adaptive, some traffic flows over the heavily utilized links are shed to alternative routes as the result of increased weights on these links. It is also observed in the simulations that the routings become steady after several update periods, and no continuous oscillation happens, which means the routing is stable. Under the same traffic load, the maximum link utilization with LSAR after routing become steady is significantly smaller than the one with OSPF/IS-IS. The traffic load needed to have the maximum link utilization to reach 1 becomes larger. This means that the network can accommodate more traffic load before congestion occurs, in other words the throughput of the network increases. For the simulations that were run, the average throughput increase is about 40% when LSAR is used.

B Dynamic Traffic Test

The traffic flows on real networks are mostly dynamic rather than static, so the performance of LSAR needs to be tested under dynamic traffic. Due to space limit, the details of the simulations with dynamic traffic are not included in this paper. Interested reader please refer to [17] for details.

The traffic generator used in our simulations is the one in [11], which dynamically generate TCP and UDP flows simulating FTP, WWW, Telnet and SMTP applications. This traffic generator is used to randomly generate one hour of dynamic traffic load.

Random network topologies are used. OSPF/IS-IS and LSAR are compared in each contrast pair of simulations, which have the same topology and dynamic traffic loads. The overall packet drop ratio is chosen as the metric to evaluate the routing performance. The performance of LSAR over different topologies and under various traffic loads, both light and heavy, is tested.

The simulation results show that in most cases, LSAR can improve the network performance significantly. The average packet drop rate improvement achieved from the simulations is 32.6%. Only in some exceptional cases when the network is lightly loaded and the packet drop ratio is already very low, the packet drop ratio increases, but the amount of increase is very small and the performance of the network is still very satisfying.

IV Conclusion

The dynamics of adaptive routing is investigated, and a new routing algorithm, Load-Sensitive Adaptive Routing (LSAR), is presented. The LSAR is adaptive to changing traffic load and is stable at the same time. The performance of LSAR is tested by simulation, and the results show significant improvement in network throughput and packet drop ratio compared with OSPF/IS-IS with default setting.

The performance of LSAR depends on the topology and the traffic pattern of the network. In general, the more nodes and links in the network, the more alternative routes are available from origins to destinations and the better is the LSAR performance.

REFERENCES