

2011

A Dynamic Local Method for Bandwidth Adaptation in Bundle Links to Conserve Energy in Core Networks

Lin Liu

Computer Sciences, linliu@cse.unl.edu

Byrav Ramamurthy

University of Nebraska at Lincoln, bramamurthy2@unl.edu

Follow this and additional works at: <http://digitalcommons.unl.edu/cseconfwork>



Part of the [Computer Sciences Commons](#)

Liu, Lin and Ramamurthy, Byrav, "A Dynamic Local Method for Bandwidth Adaptation in Bundle Links to Conserve Energy in Core Networks" (2011). *CSE Conference and Workshop Papers*. 193.

<http://digitalcommons.unl.edu/cseconfwork/193>

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Conference and Workshop Papers by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

A Dynamic Local Method for Bandwidth Adaptation in Bundle Links to Conserve Energy in Core Networks

Lin Liu and Byrav Ramamurthy
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, Nebraska 68588-0115
Email: {linliu, byrav}@cse.unl.edu

Abstract—Energy savings in bundle links of the core network has been investigated recently. The bundle link technique is widely used in current core networks to provide higher bandwidth and more resilience. Basically a bundle link is composed of several high-speed physical sublinks which could be SONET connections, Ethernet circuits, etc. in order to make them work together as a virtual connection. In current network operations, all of the sublinks are activated if the bundle link is powered on although the sublinks could be shut down or brought up independently. Smartly and dynamically shutting down and bringing up sublinks and their attached ports according to the traffic demand or estimation could greatly increase the link utilization and save a large amount of energy. In previous work, we proposed a local heuristic threshold-based method to explore the potential energy savings in core networks by adjusting the number of active sublinks in the bundle links. In this paper, we further explore the possibilities to reach a better tradeoff between energy-saving performance and congestion risk by setting different value combinations of two parameters: the utilization threshold and sublink-adding strategy. From that, we propose a dynamic and hybrid local heuristic threshold-based algorithm, which we call HDLHT algorithm, to achieve a better tradeoff between energy-savings and congestion risk. In HDLHT algorithm, the bundle links are given different combinations of these two parameters according to the burstiness of their traffic. A simulation experiment deploying HDLHT algorithm was conducted on a synthetic network and the results show that HDLHT can greatly reduce the number of congestion occurrences with limited decrease in energy savings.

I. INTRODUCTION

In recent years, the energy-shortage problem and the related carbon emission problem have received considerable attention. Power awareness has become an important and necessary consideration in the design and implementation of most industries. Rated as one of the least energy-efficient industries in the world, the telecommunication industry suffers from many inefficiencies such as pretty low link utilization, redundant devices or links, etc. With rapid increase in Internet users and various end devices, and the rise of multimedia, mobile, P2P and social network applications, the Internet traffic is growing exponentially. In 2008, Cisco expected that the global IP traffic would double every two years through 2012 [1]. The rapidly increasing network traffic demands mean the faster increase of network energy consumption. With more application servers

coming online, in many legacy sites, the limit for air cooling system has almost been reached and people have to struggle to deploy more expensive cooling systems or construct a new site, thus expediting the rate of infrastructure upgrades by ISPs. In conclusion, the power consumption of networks have increased rapidly and the situation has become worse. Research on energy-aware network design or green networks has become popular in recent years. Realizing that current networks lack power awareness, all kinds of energy-saving research attempts have been conducted such as hardware design, network protocol design, traffic scheduling, etc.

Among these, one of the major concerns to be addressed is the pretty low link utilization of current networks. Data shows that current link utilization of Internet Service Providers (ISPs) is estimated at 30% - 40% [2] and the link utilizations of research primary networks are even lower. This means that a large amount of energy consumed in network connections is wasted. The low link utilization could be due to the overestimation of future traffic demands at the initial network design phase. A more plausible reason comes from the coarse scalability of network bandwidth upgrades. For example, upgrading 1 Gbps Ethernet connection to the next scale (10 Gbps) possibly would produce a low link utilization since the real traffic demand normally will not increase so quickly to ten times. In fact, the most important reason for low link utilization is due to the unpredictable and bursty nature of Internet traffic. Internet traffic varies greatly according to the users' communication activities. The gap between the peak and the trough of the link traffic is huge within a day. As Internet works on a best-effort basis, usually the bandwidth of a specific link is set to be at least beyond the maximum expectation of future link traffic to reduce the possibilities of overflow or congestion.

To increase the flexibility of bandwidth upgrades, the *bundle link* or *link aggregation* technique is widely used in most core networks. The basic idea is to bundle several sublinks together to serve as a virtual link. For example, two or more 1Gbps Ethernet links can be bundled together to satisfy current traffic demand. That is, instead of upgrading to the expensive next higher rate link, we can gradually add more

low-bandwidth sublinks to better follow the growth of rising traffic. In this way, the network utilization could be increased and the network capital and operational costs could be greatly reduced. A bundle link also helps to increase the reliability in case of link failure and the capability of disaster tolerance of a core network. Now, the sublinks within the bundle link can be shut down or brought up independently and if we can smartly shut down some idle sublinks when traffic is low, a great amount of energy could be saved.

In previous work, we designed a local heuristic threshold-based algorithm to allow each core router to direct its port operations (shutting down or bringing up) according to the link utilization of the previous time slot. A simulation experiment deploying our algorithm was conducted on a synthetic network which uses the network topology and traffic data from the corresponding links of Internet2. There were two major fixed settings in the previous simulation experiment: an empirical value of 90% was set as the fixed utilization threshold for all bundle links of the synthetic network and a fixed low aggressive sublink-adding strategy was employed, that is, each time the link utilization of the latest activated sublink exceeds the threshold, only one new sublink was activated. Using this Fixed Local Heuristic threshold (FLHT) method, the simulation experiment results (using the modified Internet2 topology) show that within almost one month, which is, from July 1, 2010 to July 30, 2010, 86% energy could be saved in unit of port*5minutes while keeping the risk of traffic overflow low. One of the advantages of the local threshold-based algorithm is that it always keeps at least one sublink active for each bundle link no matter how low the traffic is, thus, the network topology was kept unchanged to avoid unnecessary and unpredictable impact to the higher layer applications. Most importantly, since core routers can direct decision-making of their ports' up and down operations by checking their own link utilizations during the last time slot, the local heuristic threshold-based algorithm can realize fast-response to varying traffic without scalability problem. In contrast an Integer Linear Programming (ILP) based global optimization algorithm (such as in [2]) suffers from many problems such as being slow-acting, not scalable and resulting in a changing topology (see section II for additional details on Related Work).

In this paper, we propose a dynamic method to achieve better network performance while achieving considerable energy savings. As we can see, the objective of a heuristic threshold-based algorithm is to save energy by rightsizing the bundle links' capacities and by increasing their link utilizations. We cannot simply pursue as much energy savings as possible and ignore the possible impact from too frequent overflows; also since the network impact from small overflows could be negligible, we might accept them without losing the chance of greater energy savings. The objective of this study is to look for good tradeoffs in this case. As we can see, two parameters mostly contribute to the control of the height of dynamic roof and the compression rate of headroom: the link utilization threshold and the sublink-adding strategy, which

is, how aggressive is the strategy to add new sublinks when more bandwidth is needed. First, we separately evaluate the influence of these two controlling parameters on the energy savings and overflow risk by some simulation experiments. Then we examine the total effect of several possible value combinations of these two parameters. Based on that, we would propose a new algorithm to combine the selection of parameter combinations based on the traffic burstiness of bundle links to reach better network performance optimization. Since bundle link concept is used for all kinds of links (SONET connections, Ethernet circuits, wavelength, etc). we use an abstract "port" to represent its physical connector and related links. Also to simplify calculation, we only calculate energy savings of two end router ports and prefer using "port*time" as the energy unit instead of watt. Finally, although we only investigate shutting the pair of router ports for a sublink, in an actual network, the energy consumption due to other devices such as router line cards, transponders and amplifiers, can also be reduced.

although shutting down the pair of ports, energy consumption of the possible amplifiers and multiplexers and other devices between them can also reduced.

II. RELATED WORK

Gupta and Singh [3] proposed the idea of putting the subcomponents or ports of network devices to sleep to save energy. They discussed the impacts on the current Internet protocols. Further, they design and evaluate a Dynamic Ethernet Link Shutdown (DELS) algorithm to make sleeping decisions [4]. Gupta and Singh's ideas were evaluated by simulation traffic from a synthetic traffic generator and a wide range of real-time traffic traces collected at a campus high-speed backbone switch. The simulation experimental results show that with power awareness design, significant energy savings can be achieved with little impact on packet loss or delay [5]. Even for highly bursty traffic, the Ethernet interfaces at both ends can be put in extremely low power modes anywhere from 40%-98% of the time observed. Chabarek [6] discussed the energy consumption of various hardware configurations of widely used core and edge routers. Based on that, they discovered the potential energy savings achievable in some example networks and the results show that power consumption can vary by as much as an order of magnitude. Nedevschi [7] discussed two forms of power management on interfaces of network devices: sleep or rate adaptation. Andrews et al. [8] study and compare the performance of the routing and scheduling problem putting each network element into zero-rate mode or full-rate mode with given routing. Ananthanarayanan et al. [9] propose schemes for power reduction in network switches. Their schemes can dynamically tune parameters to adapt to changing traffic patterns to guarantee a bounded and specified increase in latency. The simulation experiments deploying their schemes on packet traces collected from an enterprise network show that simple power reduction schemes can achieve up to 32% energy savings with minimal increase in latency or packet-loss while

with appropriate hardware support such as Wake-on-Packet, shadow ports and fast transitioning of the ports between its high and low power states, 90% of the maximal theoretical savings can be reached. Jin et al. [10] consider the problems of existing flow level bandwidth provisioning solutions such as high implementation complexity, poor performance guarantees and inefficiency to process variable length packets and propose the Flow LEvel Bandwidth pRovisioning (FEBR) algorithm which has a number of implementation advantages, such as no speedup requirement, bounded crosspoint buffer sizes, distributed scheduling, and low time complexity. FEBR was implemented in the OpenFlow software switch to provide a practical flow level bandwidth provisioning solution.

Fisher et al. [2] first investigated the problem of energy savings in bundle links and formulated this problem as an NP-complete problem and they proposed three heuristic algorithms to solve the integer linear programming (ILP) formulation. Their method results in a changing network topology and requires knowledge of traffic demands for each bundle link. Also the method is not scalable and takes a long time (might be more than half an hour) to complete the calculation. In [11] we proposed a fixed local heuristic threshold-based (FLHT) algorithm, which is local-decision-making, autonomous, scalable, fast-acting and topology-invariant. Also we conduct simulation experiments on a synthetic network which is based on the network topology and link traffic data from Internet2 network. The simulation experiment results show that a large amount of energy in bundle links could be saved by deploying FLHT algorithm at the cost of a small possibility of overflow. While the FLHT algorithm uses fixed utilization threshold and fixed sublink-adding strategy we would like to design a more dynamic local heuristic threshold-based algorithm which can dynamically adjust the parameter values with the varying traffic to further reduce the possibilities of overflow.

III. PROBLEM STATEMENT

The problem in this paper can be formalized as follows: for each bundle link, given its link traffic in the last time slot (T) and its number of sublinks (N), find good parameter settings including utilization threshold (R) and sublink-adding strategy (S) to reach a better tradeoff (D) between energy savings (E) and overflow risk (O). As we discussed in previous work, network traffic demand is pretty hard to predict, so we use traffic in last time slot as input parameter and combine it with threshold to trigger the router port operations.

As mentioned earlier, the FLHT algorithm could generate an undesirable byproduct: buffer overflows. Actually overflow happens sporadically and cannot be 100% guaranteed to be removed in all networks due to the unpredictability and bursty nature of network traffic. The overflow could bring the problems of data loss or affect the performance of higher layer applications. Some methods and techniques has already been applied to reduce or remove the impact of overflow. One of them is the congestion control mechanism in TCP/IP. Another common method in ISP networks is deploying some rerouting mechanisms to redistribute the excessive traffic through load

balancing. With these efforts, the impact of overflow could be greatly reduced or removed if the overflow happens in a low amplitude or lasts for a short period. Otherwise, the situation could become even disastrous if the overflow lasts for a long time and in a high amplitude, especially when that link is a high speed link. Local threshold-based algorithms could increase the possibilities of overflow since the headroom of bundle links are greatly compressed and the bursty traffic can more easily break through the roof. When overflow happens, the response time of the local heuristic threshold-based algorithms depends on two factors: how fast it detects overflow and how quickly it calculates the solution. The calculation time of local heuristic threshold-based algorithms is almost negligible since it uses simplified local-decision-making strategy and does not need to solve the ILP-based NP-complete problems. Actually the maximum overflow detection time in FLHT algorithms almost equals the traffic sampling interval. Since most network management systems sample link traffic in frequency of each 5 minutes, although the finest sampling frequency of Internet2 traffic is every 10 seconds, in our previous simulation experiments we consolidate the data into 5 minute intervals. The FLHT algorithm could make local decisions at the same rate of the sampling frequency and its response time could be greatly reduced with higher sampling frequency. In previous work, we used fixed utilization threshold (90%) and fixed sublink-adding strategy (each time we bring up only one sublink) for port operations. Now we are considering to make these two parameters dynamic to reach a better tradeoff between energy savings and overflow risk. Intuitively, a more aggressive strategy (activating more sublinks every time) could reduce the possibilities of overflow at the cost of lower energy savings since now the headroom for bundle links is greater. Also a lower utilization threshold strategy works along the same way: reducing the energy-saving benefit to decrease the overflow risk. We would like to separately evaluate their performances in different situations. A further step is to evaluate different parameter value combinations to find out appropriate combinations that could provide better tradeoff optimization. We refer to our new algorithms introducing dynamic parameters in previous FLHT algorithm as dynamic local heuristic threshold-based algorithms (DLHT).

We consider three types of DLHT algorithms with different sublink-adding strategies: very aggressive, medium aggressive and low aggressive. We separately call them DLHT-High, DLHT-Medium and DLHT-Low algorithms. The differences among them are: DLHT-High algorithm always activates three sublinks when the current link utilization exceeds the threshold; while DLHT-Medium activates two sublinks when necessary and DLHT-Low just activates one sublink. The similarity among them is that: they all shut down only one link each time the utilization of latest activated sublink is low enough (actually for the low utilization threshold, we set it as zero). The reason for shutting down only one sublink when necessary is that we observed that traffic often increases quickly and drops quickly and the strategy which reduces the number

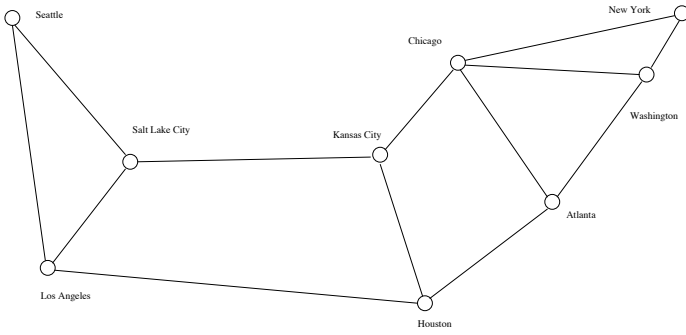


Fig. 1. Topology of Internet2

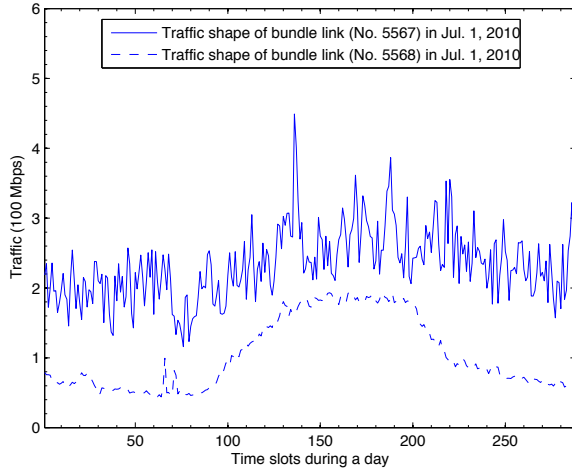


Fig. 2. Traffic shape of bursty link and smooth link in Internet2

of sublinks aggressively leads to frequent port operations which not only consume considerable energy but also to add time delays, thus, offsetting the benefits. Actually DLHT-Low algorithm is identical to our previous FLHT algorithm. Another reason to design DLHT-High and DLHT-Medium algorithms is that in the previous simulation experiments we found that there are cases of very bursty link traffic, for example sometimes some bundle link is so bursty that its link traffic increases to more than double or triple of that of the previous time slot. That kind of overflow we call a *super overflow* and obviously activating one sublink at a time is definitely not enough in that case.

However, super overflow does not happen in every bundle link. The bundle links in Internet2 network can be classified into two groups: very bursty link group and normal bursty link group. Observing the long-term traffic shapes of the two bundle links, we found that they are very different: one of them is always very bursty while another one is much smoother. Fig. 2 shows one example which presents the traffic shape of two links (No. 5567 and No. 5568) between Chicago and Kansas City. What we found is that the super overflow events all happened in the very bursty link group and the number of normal overflows in the very bursty link group is much

bigger than that of the normal bursty link group. This is reasonable as burstiness is the major reason for producing overflow. Intuitively, it is more difficult to use rerouting or congestion control mechanisms to reduce or remove the impact of super overflow.

Based on the above observations we propose several more aggressive methods and one of the targets is to reduce the number of super overflows. Since the possibilities of super overflow are much lower, always activating some extra idle links to prepare for the rare chances of overflow would definitely reduce the amount of energy savings. More aggressive algorithms lead to greater loss of energy savings. The value of utilization threshold also influences the amount of energy savings and the number of overflows. Lower utilization threshold will result in easier triggering of the sublink-adding operations and extra idle links will be maintained for longer periods. With more bandwidth resources becoming available, the number of overflows should drop. Thus, the two parameters might have different impacts in keeping the balance, so we would like to evaluate their joint performance.

Our previous FLHT algorithm is more favorable to the side of energy savings and DLHT algorithms are trying to reach a better balance by deliberately giving up some energy savings to produce lower number of overflow events.

IV. SIMULATION EXPERIMENTS AND RESULTS

For convenience, we still use the same synthetic network used in our previous work to conduct our simulation experiments. Basically, the synthetic network uses the same network topology based on Internet2 network as shown in Fig. 1. There are 9 sites and 26 bundle links (there are two bundle links between a pair of neighboring sites). The sampled link traffic data of Internet2 are directly deployed in its corresponding bundle links in the synthetic network and we assume each bundle link is composed of 10 sublinks, and the traffic load of a bundle link are evenly distributed to its sublinks. Also to simplify calculation, we focus on 30 days' link traffic data of Internet2 which is from July 1 2010 to July 30 2010. The original Internet2 traffic data is collected and stored in RRD files from which we can extract and aggregate the data using RRDtool. Relying on the RRDtool, the Internet2 network periodically samples the traffic data of each link and stores them in a round-robin database (circular buffer) whose archives have fixed sizes and can wrap around: the next insertion will overwrite the oldest entry. Within one RRD file, several archives might be included: archives for primary data point (PDP) and archives for several consolidated data point (CDP) derived from PDP. We can use a command similar to the following to extract the traffic data of PDP of Internet2.

```
rrdtool fetch sample.rrd AVERAGE --start
1212296400 --end 1212296400+86390 >
sample.data
```

In the above command, sample.rrd is the name of RRD file. RRDtool extracts the traffic data within the specified period (with specified start time and end time) in the sample link

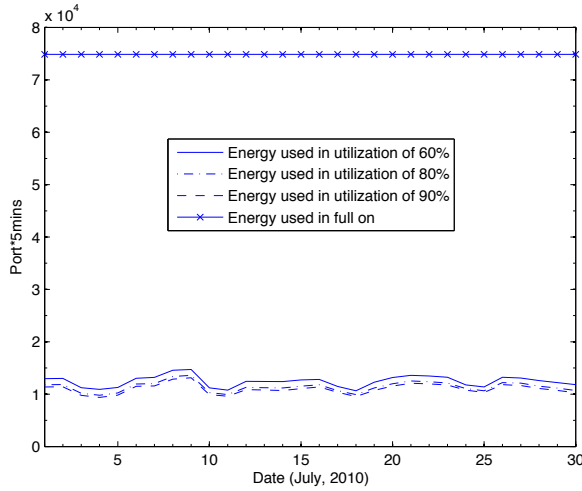


Fig. 3. Comparison of energy savings of three thresholds within 30 days deploying DLHT-Low algorithm

and stores them in the file named sample.data. The step size of Internet2 PDP is just 10 seconds, and from that, we obtain the link traffic of each 5 minutes. In fact, in each time slot (5 minutes), a specific bidirectional link will be sampled twice separately at both end sites (site A and site B) and each sampled data includes traffic data in two directions: outgoing and incoming. At a specific time slot, the outgoing traffic data of site A might not be equal to the incoming traffic data of site B; this is because both sites' sampling time might not be exactly the same. We use the maximum value of the four sample data as the link traffic data during that time slot. After completion of calculating link utilization data for all bundle links for all time slots within the 30 days, we are ready to apply our DLHT algorithms to check the energy savings and overflow risk.

Simulation experiments in previous work have shown that 86% energy savings could be achieved while causing 6111 cases of overflow by LHT algorithm. Now, we select three link utilization threshold values (90%, 80%, 60%) to compare their performance.

First, we deployed DLHT-Low algorithm and check how different the energy-savings are. The result is shown in Fig. 3, from which we can see that changing link utilization threshold from 90% to 60% only produces a limited decrease (around 2.3%) of energy savings. Now we are interested in how many overflow and super overflow events could be reduced in the above scenario. In this simulation experiment, we care about the two link groups' overall performance. The result is shown in Fig. 4. To make the graph clear, we only show the result of setting 60% and 90% as the utilization thresholds. The x-axis is an ordering of bundle links in which the first 13 links are from the very bursty group while the latter 13 links are from the normal bursty link group. We noticed that the very bursty link group suffers much more cases of overflow and super overflow than the normal bursty link group. Actually,

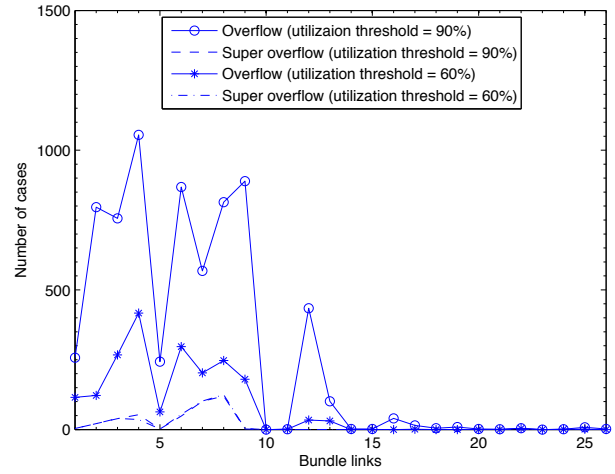


Fig. 4. Comparison of overflow and super overflow events with utilization threshold 60% and 90% over 30 days deploying DLHT-Low algorithm

the latter one has close to zero cases of overflow and super overflow. Hence there is almost no difference for normal bursty link group by setting different utilization thresholds. However in the very bursty link group, when setting 60% or 90% as the threshold, the results are very different. By setting a lower threshold, the number of overflows is greatly decreased. The number of super overflows almost remains the same under different utilization settings. This is reasonable, since for the last sublink, changing the utilization from 60% to 90% will not change much the situation when the link traffic more than doubles. It seems that the utilization setting is very effective in reducing the number of overflows of bursty links.

Next we would like to evaluate the performance of sublink-adding strategy in terms of reducing the number of super overflows. Fig. 5 shows for a utilization threshold of 60%, the number of super overflows for the algorithms: DLHT-High, DLHT-Medium and DLHT-Low across 30 days. From Fig. 5, we can see that the high aggressive strategy of DLHT algorithm could greatly reduce the cases of super overflow for the bursty link group.

From the above analysis, we know that for the normal bursty link group deploying low aggressive DLHT algorithm and setting 90% as utilization threshold is good enough, while for the very bursty link group, it is better to use more aggressive DLHT algorithms to reduce the cases of super overflow and setting lower utilization threshold to decrease the cases of overflow. It seems that the corresponding loss in energy saving is almost negligible, but in fact we are missing one consideration which is the energy consumption to conduct port operations.

Table 1 shows the number of sublink operations for several parameter combinations. It shows that more aggressive DLHT algorithms and lower utilization threshold would result in more sublink operations, consuming more energy. The increased energy consumption in sublink operations might offset the

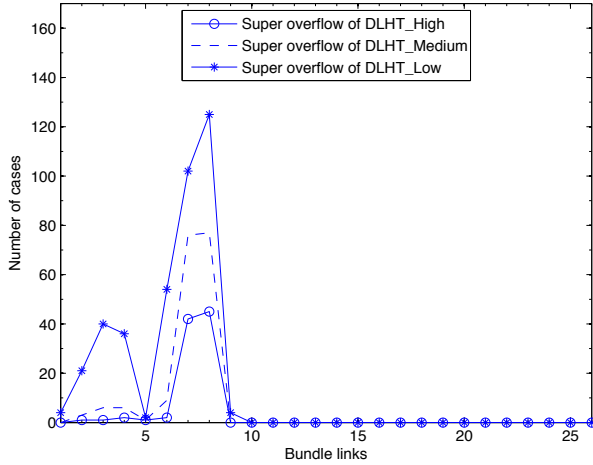


Fig. 5. Comparison of super overflow events of three DLHT algorithms with 60% utilization threshold

TABLE I
COMPARISON OF NUMBER OF SUBLINK OPERATIONS IN 9 PARAMETER COMBINATIONS

Strategy//Utilization	60%	80%	90%
DLHT-HighAggressive	126255	98859	76264
DLHT-MidiumAggressive	73202	58518	47380
DLHT-LowAggressive	30267	26600	24303

TABLE II
PERFORMANCE COMPARISON BETWEEN HDLHT AND LHT ALGORITHMS

Algorithm	Overflow	super overflow	energy savings%
HDLHT	1701	94	1855168
LHT	6874	390	1916646

benefit. However, this is an area of future work. By the above simulation experiment method, we can set different parameter combinations to different bundle links according to how bursty they are. We incorporate such hybrid parameters setting into our DLHT algorithms and refer to them as HDLHT (Hybrid DLHT) algorithms. Deploying HDLHT could reduce the cases of overflow and super overflow and reduce the loss of energy savings. To simplify the calculation, we use a simple way to evaluate our ideas: just use the parameter combination of DLHT-High algorithm and threshold of 60% for bursty bundle link group and DLHT-Low algorithm and threshold of 90% for normal bursty bundle link group. Then we compare the result of the HDLHT with results from previous work to measure the improvement. The comparison results are shown in Table 2.

Table 2 shows that with the HDLHT algorithm, the number of overflow and super overflow events are greatly reduced with very limited (3.2%) decrease of energy savings. Although there are still a lot of cases of overflow and super overflow, even deploying the HDLHT algorithm, actually most of them come from the very bursty link group which is much more

bursty than a normal commercial ISP link. For example, in our synthetic network case, the very bursty link group produces 1609 overflow and 94 super overflow events while normal bursty link group only produces 92 overflow and 0 super overflow events. So this algorithm should have much better performance when deployed in pure commercial networks.

V. CONCLUSION AND FUTURE WORK

In this paper we discover the relationship between different parameters of LHT algorithm in terms of energy savings and overflow risk. We also compare the performance of several typical parameter combinations. Finally we check the effect of hybrid parameter settings on different bursty link groups. The simulation experiment results show that HDLHT algorithm works well. It can greatly reduce the overflow risk while reducing the loss of energy savings. Several tasks are left as future work. Deploying HDLHT algorithm in pure commercial networks could demonstrate realistic performance. Also we can consider the offset effect on the energy savings from increasing router port operations. Further, more complicate parameter combinations could be evaluated. Finally, we did not consider the time effect currently. Since within a day or across weekdays and weekends, the burstiness of link could be different. So we might introduce relationship between burstiness and the time element into the algorithm design.

VI. ACKNOWLEDGEMENT

This work was supported in part by DOE (DE-SC0001277) and NSF (CNS-1040765).

REFERENCES

- [1] Cisco, "Approaching the Zettabyte Era," 2008. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481374.pdf
- [2] W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks: reducing energy consumption by shutting off cables in bundled links," in *Proceedings of the first ACM SIGCOMM workshop on Green networking*. ACM, 2010, pp. 29–34.
- [3] M. Gupta and S. Singh, "Greening of the Internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003.
- [4] —, "Dynamic ethernet link shutdown for energy conservation on ethernet links," in *IEEE ICC 2007*, june 2007, pp. 6156–6161.
- [5] —, "Using low-power modes for energy conservation in Ethernet LANs," in *IEEE INFOCOM*, 2007, pp. 2451–2455.
- [6] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *IEEE INFOCOM*, 2008.
- [7] S. Nedeveschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*. USENIX Association, 2008, pp. 323–336.
- [8] M. Andrews, A. Anta, L. Zhang, and W. Zhao, "Routing and Scheduling for Energy and Delay Mimimization in the Powerdown Model," in *IEEE INFOCOM*, 2010.
- [9] G. Ananthanarayanan and R. Katz, "Greening the switch," in *Proceedings of the 2008 conference on Power aware computing and systems*. USENIX Association, 2008, pp. 7–7.
- [10] H. Jin, D. Pan, J. Liu, and N. Pissinou, "Openflow based flow level bandwidth provisioning for CICQ switches," in *IEEE INFOCOM, 2011*, pp. 476–480.
- [11] L. Liu and B. Ramamurthy, "Rightsizing bundle link capacities for energy savings in the core network," in *IEEE GLOBECOM 2011*.