Opportunistic Bandwidth Sharing for Virtual Network Mapping

Sheng Zhang†, Zhuzhong Qian†, Bin Tang†, Jie Wu‡, and Sanglu Lu†
† State Key Lab. for Novel Software Technology, Nanjing University, China
‡ Department of Computer and Information Sciences, Temple University, USA
† {zhangsheng,tb}@dislab.nju.edu.cn, {qzz,sanglu}@nju.edu.cn, ‡ jiewu@temple.edu

***Abstract*—Network virtualization has emerged as a powerful way to fend off the current ossification of the Internet. A major challenge is virtual network mapping, which is to assign substrate resources to virtual networks (VNs) such that some predefined constraints are satisfied and substrate resources are utilized in an effective and efficient manner. Due to the NP-completeness of this problem, a variety of heuristic algorithms have been proposed. However, existing solutions rarely consider the inefficient utilization of bandwidth resources due to the network traffic fluctuation. In this paper, we study the opportunistic bandwidth sharing in a single physical link among multiple virtual links from different VNs. We formulate the problem of assigning time slots to dispensable sub-flows with constraints on the performance guarantee and the objective of minimizing the number of time slots used, as an optimization problem. Two heuristic algorithms *HA-I* and *HA-II*, which consider the problem from different perspectives, are presented. Extensive simulations are conducted to evaluate the effectiveness and efficiency of our algorithms.**

***Index Terms*—virtual network embedding; opportunistic bandwidth sharing; Chernoff bound; NP-complete**

I. Introduction

Due to the competing policies and interests of its stakeholders, the deployment of new technologies in the Internet is painfully slow. *Network virtualization* [1–4] has emerged as a promising approach to overcome this problem. In network virtualization, traditional *Internet service providers* (ISPs) are divided into : *infrastructure providers* (InPs) who maintain *physical/substrate networks* (SNs), and *service providers* (SPs) who purchase resources from multiple individual InPs to build logical service networks, known as *virtual networks* (VNs), on top of substrate networks and offer customized value-added services to end users. This decoupling not only brings flexibility of deployment of new architectures, but also provides diversity of services in a competitive environment.

One of the fundamental problems in network virtualization is the *virtual network embedding/mapping* (VNE) problem, which maps each virtual node/link to a physical node/path such that (i) some predefined constraints are satisfied and (ii) SN resources are utilized in an effective and efficient manner. As the VNE problem is proven NP-complete [5], many heuristic algorithms [6–11] have been proposed.

Conventional VNE algorithms allocate dedicated bandwidth resources to virtual links, however, the allocated bandwidth is not fully utilized due to the network traffic fluctuation, as shown in Auckland Data Trace [12]. In this case, the SP wastes the purchased resources, and the InP loses potential customers. Opportunistic spectrum access [13], which is proven to be an effective way of making full use of the frequency spectrum, gives us some inspirations: idle bandwidth in one virtual link can be utilized by other virtual links. That is, we could allocate bandwidth according to traffic fluctuations in virtual links and allow several virtual links share certain common bandwidth to achieve efficient utilization. We use *opportunistic bandwidth sharing* to denote this concept; this is different from the traditional bandwidth sharing where bandwidth is shared among concurrent flows [14], i.e., multiplexing.

In this paper, we study the opportunistic bandwidth sharing in a single physical link among multiple virtual links from different VNs. We formulate the optimization problem called the *collision restricted assignment* (CRA), which assigns time slots to dispensable sub-flows to minimize the number of time slots used. Two heuristic algorithms *HA-I* and *HA-II* are presented to address this problem. *HA-I* adopts a greedy approach, which is similar to first-fit [15]. In *HA-II*, we relax CRA by the Chernoff bound [16], then prove the NP-completeness of the converted problem, called the *expectation restricted assignment* (ERA) problem, and develop heuristics based on relaxation and first-fit. Simulations are conducted to evaluate the effectiveness and efficiency of our algorithms.

Our contributions are summarized as follows: (i) To the best of our knowledge, this is the first attempt to apply opportunistic bandwidth sharing in VNE and provide the formulation. (ii) We propose two heuristic algorithms *HA-I* and *HA-II* to address CRA from two different perspectives.

The remainder of this paper is organized as follows. Section II provides the basic model and problem formulation. An optimal solution is given in Section III. *HA-I* and *HA-II* are presented and analyzed in Sections IV and V. Our simulation results are presented in Section VI. We overview the related work in Section VII and conclude this paper in Section VIII.

II. Preliminaries

A. The Model

We consider a substrate link SL shared among multiple virtual links VL_i. Each virtual link VL_i is associated with a flow f_i, and flows on different virtual links are assumed to be independent of each other. To leverage the benefit of opportunistic bandwidth sharing, we assume that the bandwidth sharing in the substrate network is based on *time division multiplexing* (TDM), where the whole time is partitioned into

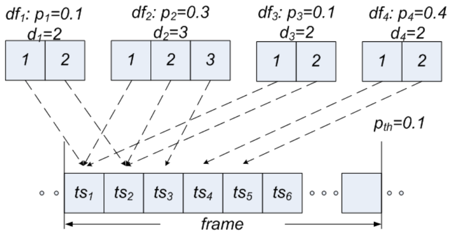


Fig. 1. An illustration of assignment of time slots to dispensable sub-flows

multiple frames with equal length, and each frame is further divided into L equal time slots, $ts_1, ts_2, \ldots, ts_L$.

As it is difficult to capture the characteristics of network traffic fluctuation, we use a simplified model: each flow f_i is composed of a basic sub-flow bf_i, which exists all the time, and a dispensable sub-flow df_i, which occurs with a probability p_i. We denote b_i and d_i as the number of time slots in each frame required by bf_i and df_i, respectively.

B. Problem Formulation

Now we consider the time slot allocation by an InP for opportunistic bandwidth sharing in a substrate link. Generally, in order to obtain more profit, the InP wants the substrate link to be shared by as many flows as possible. To achieve this, multiple dispensable sub-flows are allowed to share a common time slot with opportunities, while for each basic sub-flow, the InP has no choice but to allocate the required number of slots. So we will only consider dispensable sub-flows in the sequel.

To save time slots for upcoming flows, we prefer that each time slot can be assigned to as many dispensable sub-flows as possible, thereby, increasing its utilization. However, when more than one dispensable sub-flows occur at the same slot, a *collision* happens, which would bring down transmission performance. Let X_i indicate whether dispensable sub-flow df_i occurs, i.e., $Pr[X_i = 1] = p_i$. For each time slot k, let D_k denote the set of dispensable sub-flows it is assigned to, and let $Y_k = \sum_{i \in D_k} X_i$. Then, the probability of a collision happening at slot k, denoted by $PC(D_k)$, is:

$$\begin{aligned} PC(D_k) &= Pr[Y_k \geq 1] \\ &= 1 - \prod_{i \in D_k}(1 - p_i) - \sum_{i \in D_k}(p_i \prod_{j \in D_k, j \neq i}(1 - p_j)) \end{aligned} \quad (1)$$

To break the utilization-collision tradeoff, we assume that the probability of a collision happening at each time slot can never go beyond a given threshold p_{th}. Our objective is to minimize the number of time slots used by all of the dispensable sub-flows.

Problem 1: (Collision Restricted Assignment, CRA) Given a set of n dispensable sub-flows df_i, $i = 1, 2, \ldots, n$, each requires d_i time slots with probability p_i, and a threshold p_{th}. Find an assignment of time slots to these dispensable sub-flows to minimize the number of time slots used, such that: 1) for each dispensable sub-flow df_i, the number of time slots assigned to it is at least d_i; 2) for each assigned time slot, the collision probability at that time slot is no more than p_{th}.

Fig. 1 shows a feasible assignment. ts_1 is assigned to three sub-flows, df_1, df_2 and df_3, because they collide with a probability 0.064 (by Eq. (1)), which is less then $p_{th} = 0.1$. ts_3 can not be assigned to df_2 and df_4 simultaneously, because the collision probability is $0.3 \cdot 0.4 = 0.12 > p_{th}$.

III. Optimal Solution

Inspired by the *cutting stock* problem[1], we first present an alternate formulation which can remove all of the collision constraints. A *pattern*, is defined as a set of dispensable sub-flows, such that even when a single time slot is assigned to all of these dispensable sub-flows simultaneously, the probability of a collision does not go beyond the threshold p_{th}. For each possible pattern j, let x_j represent the times that pattern j appears in a feasible assignment. Then, CRA can be formulated as the following integer linear programming (ILP):

$$\begin{aligned} \min \quad & \sum_{j=1}^{h} x_j \\ \text{s.t.} \quad & \sum_{j=1}^{h} a_{ji} x_j \geq d_i, \forall i : 1, 2, ..., n \\ & x_j, \text{ nonnegative integer}, \forall j : 1, 2, ..., h \end{aligned} \quad (2)$$

In the above, h is the number of all possible patterns, and a_{ji} indicates whether df_i is included in pattern j.

Generally, Eq. (2) can be optimally solved using intelligent exhaustive search approaches, such as backtracking and branch-and-bound [18]. However, when every p_i is very small, the number of possible patterns can be exponentially large, which will cost time to construct these patterns and to solve the ILP. Thus, the ILP-based approach is difficult to be applied in practice. In the next two sections, we develop two practical heuristic algorithms.

IV. Heuristic I: Collision Restricted First-Fit

When each dispensable sub-flow requires one single time slot, i.e., $d_i = 1$ for all i, we note that CRA is very similar to *bin packing*[2]. The only difference is that, in bin packing, the occupied size in each bin is the sum of sizes of all packed items; in CRA however, the collision probability in a time slot is not linear (nor multiplicative) of the occurring probabilities of those dispensable sub-flows the time slot is assigned to.

First-fit is a straightforward heuristic algorithm with an approximation factor of 2 for bin packing. Items are considered in an arbitrary order, and for each item, first-fit attempts to place the item in the first bin that can accommodate the item. If not, the item is put into a new bin. As first-fit can be executed online and has a low time complexity, we use the core idea of first-fit to heuristically solve CRA due to their similarities. The heuristic algorithm *HA-I* is shown in Algorithm 1.

[1]Cutting stock problem [17]: Given a number of rolls of paper of fixed width waiting to be cut, yet different customers want different numbers of rolls of various-sized widths, find a cutting method to minimize the waste.

[2]Bin packing [15]: Given n items with sizes $s_1, s_2, ..., s_n \in (0, 1]$, find a packing method in unit-sized bins that minimizes the number of bins used.

Here, $Collision(ts_{pos}, df_i)$ is a function that returns the probability of collision at ts_{pos} if ts_{pos} is assigned to df_i. To calculate the collision probability efficiently, we adopt the following approach. Let D_k be the set of dispensable sub-flows that the time slot k is currently assigned to. Also let:

$$A(D_k) = \prod_{i \in D_k}(1 - p_i) \text{ and } B(D_k) = \sum_{i \in D_k}(p_i \prod_{j \in D_k,\ j \neq i}(1 - p_j)),$$

then the collision probability $PC(D_k) = 1 - A(D_k) - B(D_k)$. When slot k is to be assigned to a new sub-flow df_l, then:

$$\begin{aligned} A(D_k \cup \{df_l\}) &= A(D_k)(1 - p_l) \\ B(D_k \cup \{df_l\}) &= B(D_k)(1 - p_l) + A(D_k)p_l \end{aligned} \quad (3)$$

which can be used to calculate the new collision probability.

Algorithm 1 The 1st Heuristic Algorithm for CRA: *HA-I*

INPUTS: d_1, p_1, d_2, p_2, ..., d_n, p_n, and p_{th}
For each dispensable sub-flow df_i: let $count \leftarrow 0$
  While($count < d_i$)
    Let $pos \leftarrow 0$
    While($Collision(ts_{pos}, df_i) > p_{th}$)
      $pos \leftarrow pos + 1$
    Assign ts_{pos} to df_i
    $count \leftarrow count + 1$

V. Heuristic II: Expectation Restricted First-Fit

As there are many substrate links that need to handle bandwidth sharing, we should keep the time slot assignment algorithm as simple as possible. Although the calculation of collision probability is much faster by Eq. (3) than Eq. (1), it still requires five addition and three multiplication operations. Hence, we propose another heuristic algorithm based on *Chernoff bound* [16], which only uses one addition operation.

***Theorem** 1:* let D_k denote the set of dispensable sub-flows time slot k is assigned to, the probability of collision will be no more than p_{th} if $\mu e^{1-\mu} \leq p_{th}$, where $\mu = E[Y] = E[\sum_{i \in D_k} X_i]$, X_i indicates df_i, and e is the exponential constant.

Proof:

$$\begin{aligned} PC(D_k) &= Pr[Y > 1] \leq Pr[Y \geq 1] \\ &= Pr[Y \geq (1 + \delta)\mu] \quad (\text{Let } \delta = \frac{1}{\mu} - 1 > 0) \\ &\leq (\frac{e^{\delta}}{(1 + \delta)^{1+\delta}})^{\mu} \quad (\text{Chernoff bound}) \\ &= \mu e^{1-\mu} \end{aligned}$$

The theorem follows immediately. ∎

Theorem 1 reveals a relationship between μ_{th} and p_{th}, which we use to modify the CRA problem by changing the constraint on collision to capacity at each frame, and later show the new problem is NP-complete.

***Problem** 2:* (Expectation Restricted Assignment, ERA) Given a set of n dispensable sub-flows df_i, $i = 1, 2, \ldots, n$, each requires d_i time slots with probability p_i, and a threshold p_{th}. Find an assignment of time slots to these dispensable sub-flows to minimize the number of time slots used, such that: 1) for each dispensable sub-flow df_i, the number of time slots assigned to it is at least d_i; 2) for each ts_k, the expectation of number of sub-flows ts_k is assigned to is no more than μ_{th}.

***Theorem** 2:* The ERA problem is NP-complete.

Proof: Given an instance of bin packing, we construct a corresponding instance of ERA by letting μ_{th} be the bin size, $p_1, p_2, ..., p_n$ be the sizes of n items, respectively, and $d_i = 1$ for i. In doing so, we reduce bin backing to a special case of ERA and prove ERA to be NP-hard. It is also easy to see that ERA is in NP, and therefore, ERA is NP-complete. ∎

Up to now, the original CRA problem has been transformed into the ERA problem and we have proven it to be NP-complete. From the proof, we immediately obtain a solution to CRA by solving ERA with first-fit, as shown in Algorithm 2. The calculation of expectation is linear, which takes only one addition operation when a new dispensable sub-flow comes. Hence, *HA-II* runs faster than *HA-I*.

Algorithm 2 The 2nd Heuristic Algorithm for CRA: *HA-II*

INPUTS: d_1, p_1, d_2, p_2, ..., d_n, p_n, and μ_{th}
For each dispensable sub-flow df_i: let $count \leftarrow 0$
  While($count < d_i$)
    Let $pos \leftarrow 0$
    While($Expectation(ts_{pos}, df_i) > \mu_{th}$)
      $pos \leftarrow pos + 1$
    Assign ts_{pos} to df_i
    $count \leftarrow count + 1$

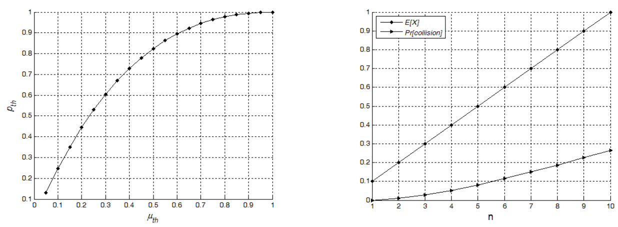
On the other hand, the relaxation may decrease the number of dispensable sub-flows one time slot can be assigned to, which can lead to a performance degradation. To characterize the relaxation gap, we depict the relationship between μ_{th} and p_{th} in Fig. 2(a). We notice that μ_{th} is smaller than p_{th}, which indicates that the relaxation by Theorem 1 may not be tight, i.e., bandwidth can not be fully utilized if we do nothing more than using first-fit to solve the ERA problem.

We use the following example for illustration: suppose ts_k is assigned to n independent sub-flows, each occurring with the same probability p, then the probability of collision is:

$$Pr[collision] = 1 - (1 - p)^n - np(1 - p)^{n-1} \quad (4)$$

The expectation of the number of sub-flows is $E[X] = np$. Fig. 2(b) shows the values of $Pr[collision]$ and $E[X]$ for $n = 1, 2, ..., 10$, from which we note that $E[X]$ is much larger than $Pr[collision]$ for the same n.

For each $E[X]$, we obtain a value of p_{th} by Theorem 1. Table I gives a comparison of $E[X]$, $Pr[collision]$ (we use $Pr[c]$ in the table) and p_{th}. For instance, when $n = 2$ and $p = 0.1$, then $E[X] = 2 \cdot 0.1 = 0.2$, $Pr[collision] = 1 - (1 - 0.1)^2 - 2 \cdot 0.1 \cdot (1 - 0.1) = 0.01$, $p_{th} \geq E[X]e^{1-E[X]} = 0.445$, which means that, if we use $E[X] = 0.2$ as the bin size, we only guarantee that collision occurs with a probability no more than 0.445. In fact, this probability is about 0.01 or much smaller than 0.445.

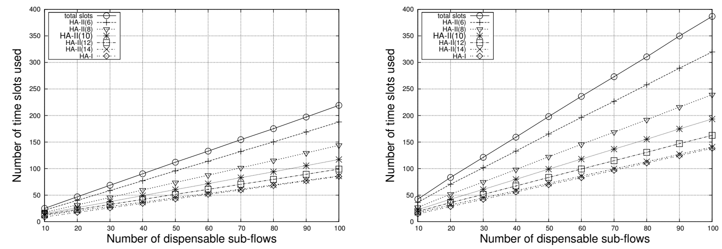


(a) $\mu_{th} \sim p_{th}$ (b) $n \sim collision$ and $n \sim expectation$

Fig. 2. (a) shows the relationship between μ_{th} and p_{th}, while (b) illustrates $Pr[collision]$ and $E[X]$

TABLE I
THE RELAXATION GAP

	$p = 0.1$			$p = 0.2$		
n	$E[X]$	$Pr[c]$	p_{th}	$E[X]$	$Pr[c]$	p_{th}
1	0.1	0	0.245	0.2	0	0.445
2	0.2	0.01	0.445	0.4	0.04	0.729
3	0.3	0.028	0.604	0.6	0.104	0.895
4	0.4	0.052	0.729	0.8	0.181	0.977
5	0.5	0.081	0.824			
9	0.9	0.225	0.994			



(a) $p_i \in (0.05, 0.10)$ (b) $p_i \in (0.05, 0.20)$

Fig. 3. Number of time slots used vs. Number of dispensable sub-flows

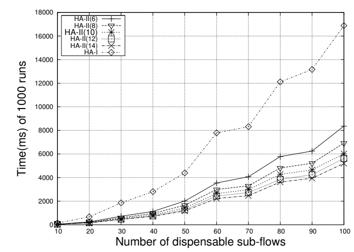


Fig. 4. Comparison of running time

The main reason behind the above scenario is, mutual independence is considered in CRA while ERA ignores it because of the linearity of expectation. To make up the relaxation gap, we replace μ_{th} by $\lambda\mu_{th}$ in *HA-II*, i.e., $Expectation(ts_{pos}, df_i) > \lambda\mu_{th}$. Here, $\lambda \geq 1$ is used to control tradeoff.

VI. Performance Evaluation

In this section, we first describe our evaluation settings, and then present the main evaluation results.

A. Evaluation Settings

Comparing our algorithms with previous work on virtual network mapping is difficult because (i) it is the first attempt to use opportunistic bandwidth sharing in virtual network mapping, and (ii) we concentrate on the scenario of a single substrate link. Therefore, our evaluation focuses primarily on quantifying the benefits of opportunistic bandwidth sharing, comparing *HA-I* with *HA-II*, and determining λ. In our simulations (For more evaluations on other combinations of inputs, please refer to our supplemental materials, available online [19]), we let the number of slots required by each dispensable sub-flow be uniformly distributed from 2 to H, i.e., H is the higher bound of d_i. We uniformly generate p_i from two intervals: 0.05 to 0.1 and 0.05 to 0.2. To guarantee the delivery of packets with a high probability, we assume that p_{th} is 0.1. Every data point is the average of 1,000 runs.

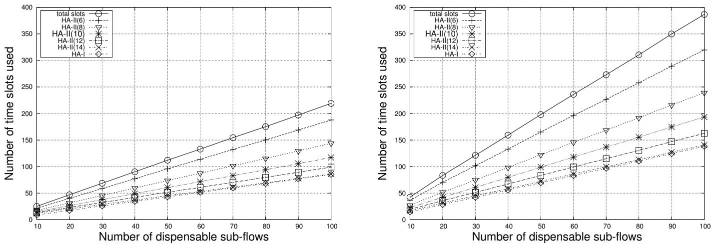
B. Evaluation Results

We first evaluate the effect of the number of dispensable sub-flows on the number of time slots used, as showed in Fig. 3. Here we let H be 10 and $HA-II(x)$ stands for $HA-II$ with $\lambda = x$. The number of dispensable sub-flows ranges from 10 to 100 with an increment of 10. The "total slots" represents the total number of slots required by all of the dispensable sub-flows, and is also the number of time slots used if there is no opportunistic bandwidth sharing. We also can see that opportunistic bandwidth sharing brings more efficient bandwidth utilization and that the larger λ is, the more bandwidth is saved. Another interesting point is that, the data points are linear in shape, which indicates that the number of slots used and the number of dispensable sub-flows have linear relationships. As we expected, a larger λ in *HA-II* leads to more possible combinations of dispensable sub-flows and further improves the effect of opportunistic bandwidth sharing; thus the slope of each line gets smaller when λ gets larger. We find that *HA-II(14)* achieves almost the same effect as *HA-I*, but *HA-II* is faster than *HA-I*.

Fig. 4 shows the time of 1,000 runs of *HA-I* and *HA-II* with $H = 30$. We notice that *HA-II(14)* is faster than other λ settings. The main reason is that, one time slot can be assigned to more dispensable sub-flows when λ grows up, so the average pos in *HA-II* becomes smaller. We also notice that the gap between *HA-I* and *HA-II* becomes larger as the number of dispensable sub-flows increases. Considering both time consumptions and results, *HA-II(14)* is better than *HA-I*.

Fig. 5 shows the effect of the higher bound of d_i, i.e. H, on the number of time slots used. The number of dispensable sub-flows is set to be 50. H ranges from 5 to 50 with an increment of 5. From these two figures, we also notice that opportunistic bandwidth sharing leads to notable improvement on bandwidth utilization and *HA-II(14)* requires nearly the same slots as *HA-I*. Furthermore, we find that more slots are required when the average of p_i becomes larger.

Next, we try to find the empirical value for λ. In our simulations, we find that the collision occurs with a probability higher than $p_{th} = 0.1$, i.e., the constraint is violated, when λ is set to 15 or larger in Figs. 3(a), 3(b), 5(a) and 5(b). Through



(a) $p_i \in (0.05, 0.10)$ (b) $p_i \in (0.05, 0.20)$

Fig. 5. Number of time slots used vs. Higher bound of d_i

more evaluations (see our supplemental materials [19]), we find the maximal allowable λ is about 10 when $p_{th} = 0.2$, and around 8 when $p_{th} = 0.3$. To explain that, we let the maximal value of p_i is p_{max} and let:

$$1 - (1 - p_{max})^x - xp_{max}(1 - p_{max})^{x-1} = p_{th}$$

Therefore, the maximal allowable λ can be calculated by $\lambda\mu_{th} = xp_{max}$. Here, λ can be used to achieve a tradeoff between bandwidth utilization and transmission performance.

To sum up, bandwidth utilization can be more efficient with opportunistic bandwidth sharing. *HA-II* is more flexible and less time-consuming than *HA-I*; the value of λ can be approximately calculated and used to achieve tradeoff between efficiency and performance.

VII. Related Work

A. Network Virtualization

In networking literature, *virtual private networks* (VPNs) and *overlay networks* share some similarities with network virtualization. They all deal with selecting nodes to construct paths. The differences are: (i) the mapping of nodes are predetermined in VPN while it is not in VNE [7, 8], (ii) only link constraints are considered in VPN while both link and node constraints are considered in VNE [7, 8], and (iii) overlays are designed in the application layer on top of IP [4].

B. Virtual Network Embedding

A significant body of research has investigated techniques for the VNE problem. Some work [7, 9] focused on the offline problem, where all VN requests are known in advance. Exclusive use of substrate nodes was considered in [6]. Attention was paid to embedding with guaranteed load balancing in [7]. A subgraph isomorphism detection based backtracking algorithm was proposed in [11]. Multi-path routing support and migration were envisioned in [10]. Linear programming and rounding were used to solve the location-based mapping problem in [8]. Some other research, such as [20], focused on distributed embedding protocols/architectures.

VIII. Conclusions

This paper focuses on how to assign bandwidth resources to multiple virtual links in a single substrate link with the help of opportunistic bandwidth sharing. To the best of our knowledge, it is the first attempt that combines virtual networking mapping with opportunistic bandwidth sharing. We formulate the problem as an optimization problem and develop two heuristic algorithms, *HA-I* and *HA-II*. The effectiveness and efficiency of our algorithms are confirmed by simulations. For future work, we plan to investigate CRA with more realistic parameters such as using continuous random variables to represent flows and combine opportunistic bandwidth sharing at the entire network level.

Acknowledgments

The authors would like to thank Yitong Yin, Daoxu Chen, and Zhuo Li for their invaluable suggestions. This work is supported in part by the National NSF of China under Grant No. 61073028 and No. 61021062; Key Project of Jiangsu Research Program under Grant No.BE2010179 under Grant No. BE2010179; the National 973 Basic Research Program of China under Grant No. 2009CB320705 and No. 2011CB302800; US NSF grants ECCS 1128209, CNS 1065444, CCF 1028167, CNS 0948184, and CCF 0830289.

References

[1] J. Turner and D. Taylor, “Diversifying the Internet,” in *IEEE GLOBECOM 2005*, vol. 2, 2-2 2005, pp. 6 pp. –760.

[2] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the Internet impasse through virtualization,” *Computer*, vol. 38, no. 4, pp. 34 – 41, april 2005.

[3] N. Feamster, L.-X. Gao, and J. Rexford, “How to lease the Internet in your spare time,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 1, 2007.

[4] N. Chowdhury and R. Boutaba, “A survey of network virtualization,” *Computer Networks*, vol. 54, no. 5, pp. 862–876, 2010.

[5] D. G. Andersen, “Theoretical approaches to node assignment,” Dec. 2002, unpublished Manuscript.

[6] R. Ricci, C. Alfeld, and J. Lepreau, “A solver for the network testbed mapping problem,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, 2003.

[7] Y. Zhu and M. Ammar, “Algorithms for assigning substrate network resources to virtual network components,” in *IEEE INFOCOM 2006*, april 2006, pp. 1 –12.

[8] N. Chowdhury, M. Rahman, and R. Boutaba, “Virtual network embedding with coordinated node and link mapping,” in *IEEE INFOCOM 2009*, 19-25 2009, pp. 783 –791.

[9] J. Lu and J. Turner, “Efficient mapping of virtual networks onto a shared substrate,” *Washington University, Tech. Rep. WUCSE-2006-35*, 2006.

[10] M. Yu, Y. Yi, J. Rexford, and M. Chiang, “Rethinking virtual network embedding: substrate support for path splitting and migration,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, 2008.

[11] J. Lischka and H. Karl, “A virtual network mapping algorithm based on subgraph isomorphism detection,” in *ACM VISA 2009*. New York, NY, USA: ACM, 2009, pp. 81–88.

[12] *Auckland Data Trace*. http://pma.nlanr.net/traces/long/auck2.html.

[13] Q. Zhao and B. Sadler, “A survey of dynamic spectrum access,” *IEEE Signal Processing Magazine,*, vol. 24, no. 3, pp. 79–89, May 2007.

[14] L. Massoulie and J. Roberts, “Bandwidth sharing: objectives and algorithms,” *IEEE/ACM TON*, vol. 10, no. 3, pp. 320 – 328, 2002.

[15] V. V. Vazirani, *Approximation Algorithms*. Berlin: Springer, 2003.

[16] H. Chernoff, “A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations,” *Annals of Mathematical Statistics*, vol. 23, no. 4, pp. 493–507, 1952.

[17] P. C. Gilmore and R. E. Gomory, “A linear programming approach to the cutting-stock problem,” *Operations Research*, vol. 9, 1961.

[18] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, September 2001.

[19] *Suppl. Materials to Opportunistic Bandwidth Sharing for Virtual Network Mapping*. http://cs.nju.edu.cn/dislab/~sheng/resource/suppl.pdf.

[20] C. Marquezan, J. Nobre, L. Granville, G. Nunzi, D. Dudkowski, and M. Brunner, “Distributed reallocation scheme for virtual network resources,” in *IEEE ICC 2009*, 14-18 2009, pp. 1 –5.