Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), January Edition, 2011

A Novel Virtual Network Mapping Algorithm for Cost Minimizing

ZHANG Shun-li, QIU Xue-song
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing 100876, China, E-mail: zhangshunli@bupt.edu.cn

***Abstract*-Resources assigned to virtual network are not optimal resources, which are caused by some scarce resources. To solve this problem, this paper proposes a novel virtual network mapping algorithm that can realize mapping cost minimizing, called CMVNMA, based on two characteristics of network virtualization environment, that is virtual network has lifecycle and substrate network resources are increased or decreased periodically. CMVNM includes virtual network mapping sub-algorithm (VNMsA) that can label virtual nodes and virtual links which are not allocated optimal resources, and heuristic migration sub-algorithm (HMsA) that can realize saved substrate resources maximization and migration cost minimization. Simulation results show that CMVNMA can save around 15% substrate network resources, and HMsA can use litter time to save the most substrate network resources than greedy migration algorithm (GMA) and random migration algorithm (RMA).**

***Keywords*-migration, network virtualization, virtual network mapping, virtual network resource allocation.**

1 INTRODUCTION

Network virtualization is an important method to solve rigid issue of the current network [1,2]. Under the network virtualization environment, Network Provider (NP) is divined into Substrate Network Provider (SNP) and Virtual Network Provider (VNP) [2,3]. The duty of SNP is to build the Substrate Network (SN). The duty of VNP is to create Virtual Network (VN) through rent SN resources and provide professional services for users.

VN needs to share the substrate node resources and substrate link resources to achieve the communication under the network virtualization environment. So, virtual network resource allocation is one of the key problems and the latest research achievements include [4-8]. (In this paper, mapping, allocation and assignment are used alternately.) But, substrate resource utilization rate is still not high. The main reason is that some virtual network resources do not obtain optimal substrate network resources, because parts of substrate network resources are scarce.

Fortunately, network virtualization environment has two important characteristics: (1) Virtual network will release occupancy resources when its lifecycle is end; (2) SNP increases or decreases substrate network resources periodically according to the resource utilization rate of substrate network. Based on these two characteristics, this paper presents a novel virtual network mapping algorithm that can realize mapping cost minimizing, called CMVNMA. CMVNM includes virtual network mapping sub-algorithm (VNMsA) that can label virtual nodes and virtual links which do not obtain optimal resources, and heuristic migration sub-algorithm (HMsA) that can realize saved resources maximization and migration cost minimization. Simulation results show that CMVNMA can save around 15% resources of substrate nodes and substrate links than existing virtual network mapping algorithm, and HMsA can use litter time to save the most substrate network resources than greedy migration algorithm (GMA) and random migration algorithm (RMA).

2 RELATED WORK

About virtual network resource allocation, there have been many researches and the typical latest researches include [4-8]. The path splitting and migration methods were used to improve the success rate of virtual network mapping in [4]. In order to coordinate two phases between the node mapping and link mapping, [5] formulated the VN embedding problem as a mixed integer program through SN augmentation and simulation experiments show that the proposed algorithms increase the acceptance ratio and the revenue while decreasing the cost than algorithm in [4]. [6] proposed distributed autonomic resources management framework and migrated virtual node between adjacent substrate node to deduce substrate link pressure and save substrate link resource. [7] addressed the problem of optimally redeploying the existing virtual network infrastructure as the substrate network resources evolves and focused on minimizing the upgrading cost, with satisfying node resource constraint and path delay constraint. [8] put forward a reconfiguration method and could realize network resources load equilibrium and improve the network benefits.

To sum up, the current researches do not consider the problem that virtual network resources are not assigned optimal resources, because parts of substrate resources are scarce. Failing to design optimization algorithm to make virtual nodes and virtual links obtain the optimal substrate network resources when substrate network environment changes. In view of the above question, this paper designs a novel virtual network mapping algorithm to optimize virtual network resources and implement saved resources maximization and migration cost minimization.

3 PROBLEM FORMULATION

3.1 Substrate Network (SN)

A weighted undirected graph $G^S = (N^S, E^S, A_N^S, A_L^S)$ is used to denote SN, where N^S is the set of substrate nodes and E^S is the set of the substrate links. Each substrate node $n^S \in N^S$ has attribute $a_N^S \in A_N^S$, which include CPU capacity weight value $c(n^S)$ and geographic location $loc(n^S)$. Each substrate link $e_{ij}^S = link(n_i^S, n_j^S) \in E^S$ has attribute $a_L^S \in A_L^S$, which include bandwidth capacity weight value $b(e^S)$. The substrate path is denoted by P^S, which is the set of substrate link. For example, the path of source node s to the destination node t is expressed by $P^S(s,t)$. In the right of Figure 1, there is a substrate network, in which the numbers over the links represent available bandwidths and the numbers in rectangles represent available CPU resources.

Some substrate network resources become scarce after SNP assigns substrate network resources to virtual network request over time. In order to continue to provide resources for the new VN request, SNP will increase substrate network resources periodically. On the contrary, if one substrate network resource utilization rate is too low, this is closed.

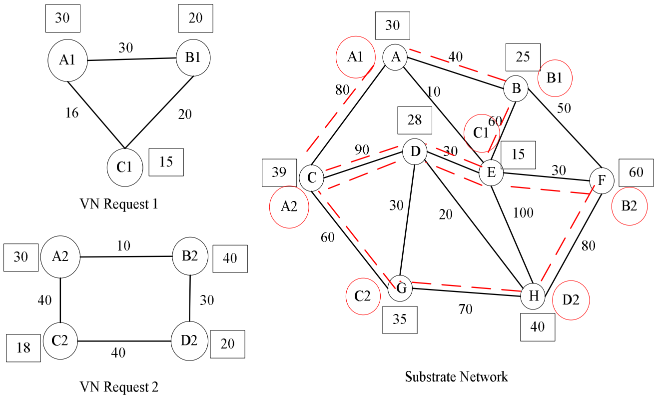


Fig.1. virtual network resource allocation

3.2 Virtual Network Request (VNR)

Similar to the SN, a weighted undirected graph $G^V = (N^V, E^V, A_N^V, A_L^V)$ is used to denote VNR. We express the VNR of virtual nodes and virtual links in terms of the attributes of the nodes and links of the substrate network. Each attribute $a_N^V \in A_N^V$ of virtual node $n^V \in N^V$ has an associated non-negative value $D^V \in a_N^V$ expressing how far a virtual node can be placed from the location specified by $loc(n^V)$. For example, there are two VNR in the left of Figure 1.

As the description in [2, 5], each VN has a lifecycle. In this paper, the arrivals of VNR are modeled by a Poisson process, and the duration of the requests follows an exponential distribution.

3.3 Virtual Network Resource Allocation (VNRA)

When a VNR arrives, SN must to decide whether to allocate resources to VNR or not. The allocation of VNR to SN can be decomposed into two major components, which are node allocation and link allocation.

1) Node allocation

Each virtual node from the same VNR must be allocated to a different substrate node by a mapping $F_N : N^V \to N^S$ from virtual node to substrate node, which is defined as:

$$F_N(n_i^V) \in N^S, F_N(n_i^V) = F_N(n_j^V) \qquad (1)$$

$$\text{if } n_i^V = n_j^V, n_i^V, n_j^V \in N^V$$

Subject to:

$$c(n_i^V) \le c(F_N(n_i^V)), F_N(n_i^V)=n_k^S \in N^S \qquad (1a)$$

$$dis(loc(n_i^V), loc(n_k^S)) \le d_i^V, d_i^V \in D_i^V \qquad (1b)$$

Where $dis(i,j)$ measures the distance between the locations of two nodes *i* and *j*. Constraint condition (1a) denotes that the CUP capacity of substrate nodes allocated to virtual node must bigger than CUP capacity of virtual node request. Constraint condition (1b) denotes that the distance between substrate node and virtual node must be the scope of virtual node request. In Figure 1, the first VNR has the node mapping {A1 → A, B1 → B, C1 → E}. The second VNR has the node mapping {A2 → C, B2 → F, C2 → G, D2 → H}.

2) Link allocation

According to the location of substrate nodes which hosted the source node and destination node of virtual link, each virtual link is mapping to a substrate path. It is defined as $F_E : E^V \to P^S$ from virtual links to substrate path for all $e_{ij}^V = link(n_i^V, n_j^V) \in E^V$.

$$F_E(n_i^V, n_j^V) \subseteq P^S(F_N(n_i^V), F_N(n_j^V)) \qquad (2)$$

Subject to:

$$b(P) \ge b(e^V), \forall P \in F_E(e^V) \qquad (2a)$$

Constraint condition (2a) denotes that the bandwidth capacity of substrate path must bigger than the bandwidth capacity requested by virtual network. In Figure 1, the first VNR has been assigned the link mapping {(A1, B1) → {(A,B) } , (B1, C1) → {(B,E)} , (A1, C1) → {(A,C), (C,D), (D,E)} }, and the second VNR has the link mapping{(A2, B2) → {(C,D), (D,E) , (E,F)} , (A2, C2) → {(C,G)} , (C2, D2) → {(G,H)} , (D2, B2) → {(H,F)}}.

3) Mapping revenue

Substrate network can obtain revenue through assignment resource to VNR. Similar to the previous work in [4,5], when mapping a VNR $G^V(t)$ at time t, this paper defines the mapping revenue as:

$$R_{map}(G^V(t)) = \sum_{e^V \in E^V} b(e^V) + \rho \sum_{n^V \in N^V} c(n^V) \qquad (3)$$

where coefficient ρ is used to tradeoff the revenue between bandwidth and CPU.

4) Mapping cost

When substrate network hosted a VNR, some substrate network resources were consumed by virtual nodes and virtual links. When mapping a VNR $G^V(t)$ at time t, this paper defines the cost of mapping a VNR as:

$$C_{map}(G^V(t)) = \sum_{e^V \in E^V} \sum_{e^S \in E^S} b_{e^S}^{e^V} + \sum_{n^V \in N^V} c(n^V) \qquad (4)$$

where $b_{e^S}^{e^V}$ denote the sum of substrate link e^S bandwidth consumed by e^V.

5) Mapping objective

The mapping objective is that more VNRs can be accepted by SN over time T. This paper defines the mapping objective as:

$$\max(\lim_{T\to\infty} \sum_{t=0}^{T} R_{map}(G^V(t))/T - \lim_{T\to\infty} \sum_{t=0}^{T} C_{map}(G^V(t))/T) \qquad (5)$$

3.4 Virtual Network Migration (VNM)

In order to make the virtual resources which do not obtain optimal resources get the optimal resources, VNM should include node migration and link migration.

1) node migration

The node migration may happen when the location of virtual node is not satisfied or the link distance between substrate node hosted virtual node and substrate node hosted adjacent virtual node is not the shortest link. This paper uses $M_N : N^S \to N^S$ to denote that virtual node $n_i^V \in N^V$ migrates from n_i^S to n_j^S, that is:

$$M_N((n_i^S) \to (n_j^S)), F_N(n_i^V) = n_i^S \in N^S, F_N(n_i^V) = n_j^S \in N^S \qquad (6)$$

Subject to:

$$c(n_i^V) \le c(n_i^S), c(n_i^V) \le c(n_j^S) \qquad (6a)$$

$$dis(loc(n_i^V), loc(n_j^S)) \le d_i^V \le dis(loc(n_i^V), loc(n_i^S)) \le d_i^V + \lambda,$$

$$d_i^V \in D_i^V \tag{6b}$$

Or

$$\sum_{n_j^V \in N_i^V} long(P^S(n_j^S, F_N(n_j^V))) \leq \sum_{n_j^V \in N_i^V} long(P^S(n_i^S, F_N(n_j^V))) \tag{6c}$$

Constraint condition (6a) denotes CPU capacity of substrate node, whether migration before or migration after, must meet the CPU capacity constraint of virtual node request. Constraint condition (6b) denotes the location of substrate node after migration can meet the location constraint of virtual node request. Constraint condition (6c) denotes there are bandwidth resources which can be saved by node migration. N_i^V is used to denote the set of virtual nodes adjacent to virtual node n_i^V. λ is used to denote location toleration value of SLA (Service Lever Agreement),which is set 10 in this paper[9]. $long(P^S(n_i^S, n_j^S))$ represents the sum of bandwidth between n_i^S and n_j^S used by virtual link, which is defined in section of link migration. The objective of node migration is:

$$\min_{n_i^S} \sum_{n_j^V \in N_i^V} long(P^S(n_i^S, F_N(n_j^V))) \tag{7}$$

In Figure 1, substrate node C and D also meet the location constraint of virtual node A2. A2 is mapping on C because D do not meet the CPU capacity constraint. After network environment is changed, CPU capacity of node D is increased. A2 can be migrated from C to D, so as to save 10 units of bandwidth. Because virtual link A2-B2 need consume 3*10 units of bandwidth before migration and only 2*10 after migration. Analogously, the migration B2 from F to E can save 10 units bandwidth.

2) link migration

This paper uses $M_E : P^S \to P^S$ to denote that virtual link $e^V = (n_i^V, n_j^V) \in E^V$ migrates from path consuming bandwidth more to path consuming bandwidth less, which is defined as:

$$M_E(P_{old}^S \to P_{new}^S) \tag{8}$$

$$b(P_{old}^S) \geq b(e^V), b(P_{new}^S) \geq b(e^V),$$

$$p_{old}^S, p_{new}^S \subseteq P^S(F_N(n_i^V), F_N(n_j^V))$$

Subject to:

$$long(P_{old}^S) \geq long(P_{new}^S) \tag{8a}$$

In which

$$long(p_{ij}^S) = long(P^S(F_N(n_i^V), F_N(n_j^V))) = \sum_{l_i \in P_{ij}^S} b(l_i)$$

, p_{ij}^S denotes the substrate path between virtual node n_i^V and n_j^V. P_{ij}^S denotes all substrate paths between substrate nodes which hosted n_i^V and n_j^V. Constraint condition (8a) denotes virtual link bandwidth consuming migration after must less than that of migration before. For example, virtual link A1-C1 of VN1 is mapped on substrate path A-C-D-E. So,

$$long(P_{A1C1}^S) = long(P^S(n_A^S, n_E^S)) = b(l_{AC}) + b(l_{CD}) + b(l_{DE}) = 16 + 16 + 16 = 48.$$

The objective of link migration is

$$\min_{p_{ij}^S \in P_{ij}^S} long(p_{ij}^S) \tag{9}$$

In Figure 1, virtual link A1-C1 can be mapped on substrate path A-C-D-E or A-E. Because A-E has not adequate bandwidth requested by A1-C1, A-E is mapped on A-C-D-E. When A-E has sufficient bandwidth to meet the constraint of A1-C1, A-E can be migrated to A-E and can save 2*16 units of bandwidth.

3) migration cost

Migration can consume network resource and affect performance of VN and SN, so, the times of node migration and link migration should less. The migration cost is defined as :

$$C_{migrate} = \omega_1 \cdot C_{node} + \omega_2 \cdot C_{path} \tag{10}$$

In which, C_{node} and C_{path} denote the times of node migration and link migration. Coefficients ω_1, ω_2 are used to tradeoff the cost of node migration and link migration. In order to compute convenience, the migration cost can be denoted as bandwidth consuming.

Frequent migration will greatly increase migration cost and also cannot guarantee migration success. Therefore, migration time is important problem for migration, which will be our next work. This paper set the migration time as the time of substrate network increases resources or the lifecycle of multiple VNs is end.

4) migration revenue

Migration revenue includes two parts: (1) VN performance is enhanced through allocate virtual node location more precise and virtual link much short. (2) Some link bandwidth resources can be saved through node migration and link migration. When virtual node is mapped on substrate node that can meet location request, this paper sets revenue as δ and δ is equal to 10 units of bandwidth. So, the migration revenue is defined as:

$$R_{migrate} = \alpha_1 \cdot R_{node} + \alpha_2 \cdot R_{path} + \alpha_3 \cdot T_{node} \cdot \delta \qquad (11)$$

In which, R_{node} denotes the link bandwidth saved by node migration, R_{path} denotes the link bandwidth saved by link migration. T_{node} denotes the times of node migration.

5) migration objective

The migration objective is to minimize the migration cost and maximize the migration revenue, which is defined as:

$$\max(R_{migrate} - C_{migrate}) \qquad (12)$$

4 VN RESOURCE ALLOCATION ALGORITHM

From section 3 description, we can see that VN mapping is a NP problem. In this paper, CMVNM includes virtual network mapping sub-algorithm (VNMsA) and heuristic migration sub-algorithm (HMsA). VNMsA can label virtual nodes and virtual links which do not obtain optimal resources. HMsA can realize saved resources maximization and migration cost minimization. In this section, we descript the VNMsA. HMsA will be introduced in section 5.

VNMsA has two steps, which are node mapping and link mapping. In the step of node mapping, we present greedy nodes mapping algorithm (GNMA) for each of virtual node and substrate node set which can satisfy the constraint of VNR is found for each virtual node. During the link mapping step, link mapping algorithm based on the k shortest paths is used to find substrate path which can meet the bandwidth constraint and bandwidth cost minimizing.

4.1 Greedy Nodes Mapping Algorithm (GNMA)

According to the location and CPU constraint of each virtual node, GNMA finds substrate node sets for them. If there is not substrate node can meet constraint of virtual node n_i^V, the location scope of virtual node can be expanded λ according to the toleration of SLA, and virtual node is marked as need migration node. Equation (13) is used to order the substrate nodes in substrate node set.

$$AR(n_i^S) = c(n_i^S) \cdot \sum e^S \in E(n_i^S)^{b(e^S)} \qquad (13)$$

Where $e^S \in E(n_i^S)$ denotes link that pass the node n_i^S.

Algorithm 1 Greedy Nodes Mapping Algorithm (GNMA)

```
Step 1. Get out virtual node $n_i^V$;
Step 2. Find substrate nodes which has location $loc(n_i^V)$ and put
        them in set $S_i^V$;
Step 3. Delete the substrate nodes in $S_i^V$ which do not meet
        constraint c($n_i^V$); if $S_i^V$ is not null, GOTO Step 5;
Step 4. If current virtual node has marked, return mapping fault, and
        GOTO Step 7; else mark current virtual node as need
        migration node, and expand $loc(n_i^V)$ to $loc(n_i^V)+\lambda$,
        and GOTO Step 2;
Step 5. Descending order the substrate node in $S_i^V$ using Equation
        (13);
Step 6. If virtual node set is null, return mapping success, and GOTO
        Step 7; else GOTO Step 1;
Step 7. Finish;
```

4.2 Link Mapping Algorithm based on K Shortest Paths (LMAoKS)

When mapping each virtual link, the K shortest paths algorithm [10] is used to find the optimal substrate path, which can meet the bandwidth constraint and consume the bandwidth minimization. The objective of LMAoKS is to find the substrate path which has the least k value.

Algorithm 2 Link Mapping Algorithm based on K Shortest Paths (LMAoKS)

```
Step 1. Get out virtual link $e_{ij}^V$;
Step 2. Find virtual nodes $n_i^V$ and $n_j^V$ which are the end nodes of
        virtual link $e_{ij}^V$;
Step 3. Find the shortest path set $E_{ij}^V$,each one of which has source
        node in $\boldsymbol{S}_i^V$ and destination node in $\boldsymbol{S}_j^V$, using k the
        shortest paths algorithm;
Step 4. Set $num_{i,j}$ equal to the number of links of path which is
        the shortest path in $E_{ij}^V$;
Step 5. Delete paths of $E_{ij}^V$ whose available bandwidth capacity is
        smaller than $b(e_{ij}^V)$;
Step 6. If $E_{ij}^V$ is null, return link mapping failure, GOTO Step 11;
Step 7. Allocate path which has the smallest k value to virtual link;
Step 8. If $num_{i,j} = k$, GOTO Step 10;
Step 9. If $num_{i,j} < k$, mark current virtual link $e_{ij}^V$ as need
        migration link;
Step 10. Get out the next virtual link, if all virtual links have been
         mapped success, return link mapping success;
Step 11. Finish;
```

5 HEURISTIC MIGRATION SUB-ALGORITHM (HMsA)

In section 4, we can see that there are some virtual nodes and virtual links which are not allocated optimal resources. In this section, HMsA is used to migration these virtual resources to get optimal resources. Migration can affect SN and VN performance and consume the energy of SN and VN. In order to solve this NP problem, we first give three migration principles, and then HMsA is proposed.

5.1 Migration Principles

(A) migration time: time that substrate network resources are increased and decreased; time that multiple VNs lifecycle is end.

(B) migration scope: migration nodes and migration links which have the remainder time t_{remain} to lifecycle end is more than the remainder time threshold TH_t; migration links which have the migration bandwidth bw_{used} is more than TH_{bw}. After confirming the migration resources, migration nodes and migration links are ordered decreasing which are described as follows:

(a) Sorting method of migration nodes uses the Equation (14), bw_{saved} is used to present saved bandwidth after node migration or link migration;

(b) Sorting method of migration links uses the Equation (15). The coefficients $\alpha 1, \alpha 2, \alpha 3, \alpha 4, \alpha 5$ are used to tradeoff among different variable.

$$\alpha 1 \times t_{remain} + \alpha 2 \times dis(loc(n_i^V) - loc(F_N(n_i^V))) + \alpha 3 \times bw_{saved} \quad (14)$$

$$\alpha 4 \times t_{remain} + \alpha 5 \times bw_{saved} \quad (15)$$

(C) migrating nodes first and migrating links second: Node migration can involve the link change, so old links related to migration node need to be migrated and obtain optimal substrate paths.

5.2 Heuristic Migration sub-Algorithm (HMsA)

Because migration is a NP problem, HMsA is proposed according to migration principles mentioned above. Migration includes node migration and link migration two steps. During the link migration, link migration sorting algorithm based on preorder relationship (LMSAoPR) is presented, which can solve the issues that there are conflict and dependence relationship among migration links (such as in Table 1).

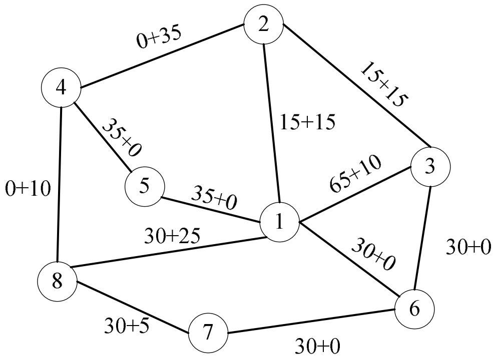
Algorithm 3 Heuristic Migration sub-Algorithm (HMsA)

```
Step 1. If the migration time arrive using migration principle (A);
Step 2. Determine the migration resources scope using migration
        principle (B), get migration node set $M_N$ and migration
        link set $M_E$;
Step 3. Node migration:
        (a) Sort nodes in $M_N$ using Equation (14);
        (b) Delete nodes which can not be met CPU capacity
            constraint to migration;
        (c) Migrate nodes in order and updating the substrate path
            according new location of migration nodes;
Step 4. Link migration:
        (a) Sort links in $M_E$ using Equation (15);
        (b) Delete links which can not be met bandwidth capacity
            constraint to migration;
        (c) Sort links of $M_E$ using LMSAoPR;
        (d) Migrate links in order;
Step 5. Finish;
```

Algorithm 4 Link Migration Sorting Algorithm based on Preorder Relationship (LMSAoPR)

```
Step 1. Solve substrate link resources needed for each virtual link to
        migration;
Step 2. Solve substrate link resources released for each virtual link
        after migration;
Step 3. Find preorder set for each virtual link;
Step 4. Merge all preorder sets to obtain the order of migration
        links;
Step 5. Return the order of migration links;
```

Now we give the detail description to LMSAoPR. From Figure 2, we can see that there are 4 virtual links want to be migrated which are shown in Table 1. But, these migrations are failed all because of the scarce of links resources. Assume that SNP increases the bandwidth 10 to substrate link 1-3, and increases the bandwidth 10 to substrate link 2-3. Now A,B,C,D can be migrated.



X+Y : used bandwidth + idle bandwidth

Fig.2. link migration

Table 1. Link migration analysis

sequence number	A	B	C	D
virtual link	1-3	4-3	1-3	7-3
substrate path	1-2-3	4-5-1-3	1-6-3	7-8-1-3
optimal objective	1-3	4-2-3	1-3	7-6-3
link resources Needed	1-3	4-2;2-3	1-3	7-6;6-3
link resources Released	1-2;2-3	4-5;5-1;1-3	1-6;6-3	7-8;8-1;1-3
conflict	C		A	
dependence	A,B	A,B;	C,D	C,D
preorder set	B,D	A	B,D	C

We use link migration in Figure 2 as an example. Link A and link B are dependence on each other because resources released by link A can be used by link B, and vice versa. There is a conflict between Link A and link C because substrate link 1-3 is needed by link A and link C simultaneously.

Preorder set of virtual link consists of virtual links which can release substrate link resources and these substrate link resources released are needed by current virtual link. In table 1, virtual link A needs substrate link resource of 1-2,2-3 which can be released by virtual B and virtual D. So, preorder set of virtual link A consists of virtual link B and D.

Now we explain the methods of merging all preorder sets to obtain the order of migration links. First getting out the preorder set s_p which has the most number of elements, and then merging the other preorder sets into s_p. In table 1,for example, preorder sets include $\{$ B,D,A $\}$, $\{$ A,B $\}$, $\{$ B,D,C $\}$, $\{$ C,D $\}$. $\{$ B,D,A $\}$has the most number of elements and is in front of others, so B,D are put into preorder set, that is $s_p = \{B, D\}$. When merging { A,B }, because virtual link A is in front of virtual link B, so, we put link A into s_p and in front of B, that is $s_p = \{A, B, D\}$. Lastly, we can get $s_p = \{A, B, C, D\}$.

5.3 Time Complexity Analysis

Assuming there are n virtual links which want to be migrated, HMsA needs two steps to migrate links. First, each virtual link is analysis such as in table 1, which consumes time $o(n)$. Second, n virtual links are sorted, which needs $o(n^2)$. Therefore, the total time is $o(n^3)$

However, if using greedy migration algorithm (GMA) which migrates virtual links continuously until all virtual links migration failure, which needs time $o(n!)$. If using random migration algorithm (RMA) which migrates virtual links using random order and each virtual link is migrated only once. So, RMA needs time $o(n)$.

6 SIMULATIONS

In this section, we first describe experiment environment, and then present some results. There are not existing researches which are related with problem solved in this paper. We compare mapping algorithm CMVNMA and CMVNMA-no-HMsA, where CMVNMA-no-HMsA denotes that HMsA is not executed, and is used to simulate existing virtual network mapping algorithms.

6.1 Simulation Environment

Similar to [4,5], this paper uses GT-ITM[11] to create virtual network and substrate network. The number of substrate nodes changes from 50 to 200. Each pair of substrate nodes are connected with probability $pr = 0.2$. The CPU capacity of substrate nodes and bandwidth capacity of substrate links distribute in the range (10,100). The number of virtual nodes of VNR is uniform distribute in (2,10). Probability that each pair of virtual nodes is connected is 0.2. The CPU capacity of virtual nodes and bandwidth capacity of virtual links distribute in the range (2,10). Virtual nodes (or substrate nodes) are connected by virtual nodes (or substrate nodes) selected randomly, if which is connected by none of virtual link (or substrate link).

6.2 Simulation Results

(1) Comparison of Saved Bandwidth

In order to validate the result of migration, VNRs arrive continuously until the VNRA fails. Then migration algorithm HMsA is used to migration virtual nodes and virtual links.

Figure 3 shows the saved bandwidth resources through using CMVNMA. We can see that CMVNMA saves more bandwidth resources than CMVNMA-no-HMsA. In addition, saved bandwidth resources increase gradually when substrate network size becomes bigger. This situation indicates that there are more virtual nodes and virtual links which can not be allocated optimal resources.

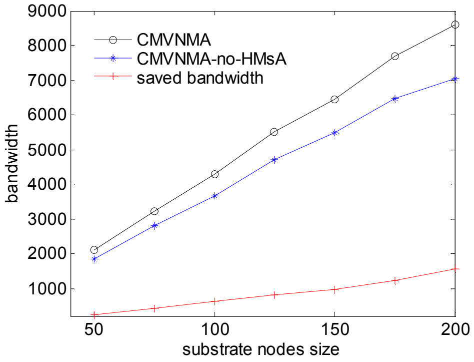


Fig.3. comparison of saved bandwidth

(2) Comparison of Link Migration Algorithms

Link migration algorithms include LMSAoPR, GMA and RMA, which have been described in section 5.3. Figure 4 represents migration revenue of three algorithms, namely saved bandwidth capacity.

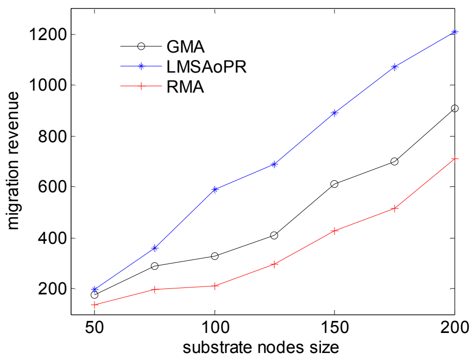


Fig.4. comparison of link migration algorithms

LMSAoPR obtains the maximal migration revenue, GMA secondly and RMA obtains the least migration revenue. So, LMSAoPR uses less time and gets more migration revenue than GMA. GMA gets more revenue than RMA, but GMA consumes more time than RMA. To illustrate these situations, GMA and RMA do not solve the issues of link migration dependence and link migration conflict. In addition, three link migration algorithms all obtain more migration revenue along with substrate network size becomes bigger, which are the result that more and more virtual nodes and virtual links are not assigned optimal resources in the big substrate network.

7 CONCLUSION AND FUTURE WORK

To save substrate resources maximization and migration cost minimizing, this paper puts forward a new resource allocation algorithm which can realize mapping cost minimization, called CMVNMA. CMVNMA is very good solution to the problem that virtual nodes and virtual links do not obtain optimal resources, caused by some substrate network resources scarce.

Although algorithm CMVNMA can save more resources than others algorithms, migration time is needed to research further. In addition, autonomic computing in resource management is a research hotspot. In next work, we will try to use non-cooperative games to realize resource independent optimization migration.

ACKNOWLEDGMENT

This work was partly supported by 973 project of China (2007CB310703), Funds for Creative Research Groups of China (60821001) and National Natural Science Foundation of China (60973108，60902050).

REFERENCE

[1] J. Turner, D. Taylor, “Diversifying the Internet”, in Proceedings of the IEEE Telecommunications Conference (GLOBECOM’05), 2005.,vol. 2.

[2] N. Feamster, L. Gao, and J. Rexford, “How to lease the Internet in your spare time,” SIGCOMM Computer Communication Review, 2007, vol. 37, no. 1, pp. 61–64.

[3] N.M.Chowdhury, R.Boutaba, “Network Virtualization:The Past,The Present,and The Future”, IEEE Communications Magazine,July,2009.

[4] M. Yu, Y. Yi, J. Rexford, M. Chiang, “Rethinking virtual network embedding: Substrate support for path splitting and migration”.ACM SIGCOMM CCR. 2008,38(2), 17–29.

[5] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, “Virtual Network embedding with coordinated node and link mapping,” in Proc. of IEEE INFOCOM, 2009.

[6] Marquezan, C.C., Granville, L.Z.,Nunzi, G., Brunner, M., “Distributed autonomic resource management for network virtualization”.IEEE Network Operations and Allocation Symposium (NOMS).2010.463 – 470.

[7] Zhiping Cai, Fang Liu, Nong Xiao, “Virtual Network Embedding For Evolving Networks”, in Proceedings of the IEEE Telecommunications Conference (GLOBECOM), 2010

[8] Zhu Y, Amm ar M. “Algorithms for assigning substrate network resources to virtual network components”. Proceedings of the IEEE INFOCOM. Barcelona, Catalunya, Spain,2006. 1-12

[9] “AT&T Managed Internet Service (MIS),”http://new.serviceguide.att.com/mis.htm, 2009.

[10] D. Eppstein, “Finding the k shortest paths”. In Proc. IEEE Symposium on Foundations of Computer Science, 1994.

[11] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, “How to model an internetwork”. In Proc. IEEE INFOCOM, 1996.

First Author: ZHANG Shun-li, born in 1981, Ph.D. candidate. His research interests include next generation network and network management.

Second Author: QIU Xue-song, born in 1973, professor, Ph.D. supervisor. His research interests include next generation network and network management.