
Shan GAO*, Taku KIHARA*, Sho SHIMIZU*, Yutaka ARAKAWA*, Naoaki YAMANAKA*, Akifumi WATANABE†
*Department of Information and Computer Science, Faculty of Science and Technology, Keio University, 3–14–1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522, Japan
Email: gao@yamanaka.ics.keio.ac.jp
†IPFlex Inc. Kamiosaki 2–27–1, Sun felista Meguro 6F, Shinagawa-ku, Tokyo, 141–0021 Japan

Abstract—This paper proposes a novel traffic engineering method using on-chip diorama network that consists of virtual nodes and virtual links. The diorama network is implemented on reconfigurable processor DAPDNA-2. In these years, traffic engineering has widely researched to guarantee QoS (Quality of Service). The proposal is an experimental solution with the on-chip diorama network, where virtual links and virtual nodes are constructed by some PEs (processing elements). We obtain the realistic traffic fluctuation through the behavior of virtual packets exchanged on the on-chip diorama network. In this paper, as first trial to achieve our final goal, we implemented diorama network and confirmed basic path calculation, where both functions are an essential function of our algorithm. The diorama network traffic engineering can realize more sophisticated network design like adaptive traffic balancing or multi-metric design.

I. INTRODUCTION

In recent years, multimedia services have been developed in Internet, such as video meeting, VoIP (Voice over Internet Protocol) and Internet radio, etc. Therefore, the next generation network inquires the more high-speed transfer capability, high reliability and QoS [1]. Traffic engineering [2]–[4] has become an important issue to secure the high reliability of network.

Traffic engineering aims to control traffic flow for optimizing network resources. There are several core issues in traffic engineering such as TE routing [5] and load balancing [6], [7], etc. The goal of TE is to select an optimal route for each traffic demand and to distribute traffic. In conventional routing algorithm, the shortest path [8] which has minimum cost is always selected to transfer data between a source node and a destination node. However, because the shortest path is always selected, the traffic load will increase, hence, congestions will occur. Therefore, even though the shortest path is chosen based on efficiency, the link will end up congested because of the heavy traffic load. It may lead to a significant deterioration of QoS.

In TE approaches, finding an optimal path to avoid congestion for each traffic with considering several metrics rather than only one metric. TE has been extensively studied, there are some related proposal shown in below. Open Shortest Path First (OSPF) is a link-state routing protocol, which was standardized by IETF 1998 [10]. In this protocol, each link has four metrics of different types, such as delay, bandwidth, reliability, and load, which all can be considered for path calculation. However, only one metric can be used at a time for path search. To obtain the optimal path, the four metrics should be considered simultaneously.

Another TE approach which is an Adaptive Multi-Path (AMP) algorithm was proposed in [9]. This algorithm calculates multiple paths and optimizes the traffic splitting ratios at each router to reach near-optimal load balancing. However, AMP only uses network information in a local area. To reach optimal load balancing, it is necessary that all nodes get global information. However, link state flooding to get global information can cause a large signaling overhead and require more memory capacity for storing all paths information.

To reduce the impact of flooding mechanism on the real network and to consider several metrics simultaneously, we propose an experimental TE method which monitors virtual packet flows for optimal path searching. However, it is difficult to monitoring packet appropriately in real network. Therefore, to construct an on-chip diorama network based on a real network, we define the format of virtual packet and emulate the data transfer for monitoring the virtual packet flow. Since flooding mechanism operate in the diorama network, each virtual node can keep global information and do not impact on the real network. Moreover, three metrics such as delay, bandwidth and link utilization rate, can be considered in our TE method. By considering these three metrics simultaneously, the optimal path can be chosen for each traffic demand. It is the final goal of our research.

Rest of this paper is organized as follows. In Section II, we describe several important steps to achieve our goal. In Section III, we explain our diorama network construction method basic path search algorithms. Simulation results are provided to evaluate the performance of the proposed algorithm is Section
IV. Finally, Section V concludes this paper.

II. STEPS TO ACHIEVE OUR GOAL

For our traffic engineering approach in on-chip diorama network, some implement steps are required to achieve our final goal. The detail of these steps is shown below.
1. How to construct the virtual node and virtual link
   The node and the link are two main components of real network. To emulate a node and link behavior in a real network, we use several PEs in DAPDNA-2 to construct the virtual node and the virtual link. We add some original functions in virtual node and virtual link for implement our traffic engineering approach. The detail of this step is described in Section III-B.
2. How to construct the diorama network
   Connect several diorama network and virtual link appropriately to construct diorama network based on real network. The detail of this step is described in Section III-B.
3. How to collect network flow with virtual packet
   Decide the data format of virtual packet which is sent on the chip for recording the route information. The detail of this step is described in Section III-B.
4. How to create diorama network based on the real network automatically
   To trace the topology change of real network flexibly, to automate the function that rearrange the link and node is needed. We make the program which automatically generates the description file of configuration, to realize automation of the PE arrangement. The detail of this step is described in Section III-D.
5. How to assign Hardware resource
   5.1 How to divide network
      To construct larger network, the network division method is required. We divide the large network to several sub-networks and construct diorama network base on each smaller sub-network.
   5.2 How to control sub-network
      After dividing the large network, we have to control the packet flow between each sub-network. The virtual sub-network which was constructed on processor should change at proper timing and transmits the virtual packets to another sub-network. The dynamic reconfiguration technique and multichip technique of DAPDNA-2 were used to control changing sub-network.
      The detail of this step is described in Section III-C.

In this paper, we describe about step 1, 2, 3 as the first trial. We also explain an example of step 5 which divide a 10 nodes network to 3 sub-networks. Moreover, we implement a basic routing algorithm, the shortest path algorithm. At last we show the effectiveness of our experimental path search method implemented on on-chip diorama network.

III. PROPOSED SCHEME

In this section, we show the basic algorithm of proposed TE approach and also explain a simple shortest path search based on diorama network which is the first step of our final goal.

A. Algorithm Summary

1) Basic Method: Our method uses flooding mechanism to exchange global information between each virtual node and considers three metrics at a time. The goal of our algorithm is optimal path search and traffic distribution.

Two types of virtual packet are defined to exchange global information and to search path. The first type of virtual packet is called virtual flooding packet (VFP). A VFP has three main fields: first field is to recording source node number; second field is to record bandwidth of traffic demand from the source node and third filed is to record link number. According to sent VFPs in diorama network, every virtual node can get global information. The second type of virtual packet is called virtual path search packet (VPSP). A VPSP has two main fields: bandwidth recording field and link number recording field. The former field is used to record the smallest bandwidth along the path. The later field is used to record number of all passing links.

Three metrics, such as delay, link utilization and bandwidth, are considered in our TE method and the method operated in an on-chip diorama network. Therefore, initializing parameters of those three metrics in a diorama network based on current network is the first step of our TE method.

For considering the link utilization, the link cost will be changed dynamically according the number of passed packets. For example, packet counter function is added in virtual link to count the number of passing packets. The link cost will be increased when the numbers of passing packets surpass a threshold amount (designed by network administrator). Therefore, this link will not be chosen in the next time of searching the shortest path. The bandwidth is a parameter of a virtual link and both VFPs and VPSPs have a field to record the bandwidth of link. Each virtual link generates several clocks delay according the real link delay.

Figure 1 shows an example of parameter initializing which uses VFPs. In this example, two VFPs pass through virtual link No.2. (In this example, the third field is abbreviated.) One VFP is from virtual node X and uses 80M bandwidth. Another VFP is from virtual node Y and uses 30M. Therefore, the bandwidth of virtual link No.2 remains 50M. The virtual counter is changed from 0 to 2. If the threshold amount was decided as two, the virtual link delay is increased from 8 to 10 (The increase is also decided by network administrator).

![Fig. 1. Parameter deterministic method on virtual link](image-url)

To search the optimal path, a simple parallel shortest path search was used in our TE method. When new traffic demand
comes, a VPSP is broadcasted from the source node to each branch. The bandwidth of passing link will be recorded in the bandwidth field of VPSP. When the VPSP arrived at a new node, VPSP will be broadcasted again. If a VPSP arrived at a new link which has a smaller bandwidth than the bandwidth which is recorded in VPSP, the smaller bandwidth value is rewritten into the bandwidth field. The passing link number is also recorded in VPSPs. Finally, VPSPs are collected in the destination node. The path which has smaller delay is shown by the faster arrived VPSP. Since the smallest bandwidth along the path is recorded in VPSP, we can choose the optimal path which enough bandwidth and small delay for each traffic demand.

Figure 2 shows an example of bandwidth recording. In this example, a VPSP arrived at the virtual link No.2. The bandwidth which recorded in the VPSP is 80 and it is bigger the bandwidth of virtual link No.2. Then, the bandwidth value of virtual link No.2 is rewritten into the VPSP. The bandwidth value of virtual link No.3 is 90, and it is bigger than the bandwidth which recording in the VPSP. Therefore, When the VPSP arrived at the virtual link No.3, the bandwidth value which recording in the VPSP will not change.

![Fig. 2. Link bandwidth recording method using virtual link](image)

2) Parallel Shortest Path Search: As an important step of our method, we proposed a simple parallel shortest path search algorithm using on-chip diorama network. We send a VPSP and let this VPSP passes through each path between a source node and a destination node. The packet which passed the shortest path will arrive at destination node first. The algorithm summary is shown below.

- **step 1** Assign a link number and a node number to each link and node.
- **step 2** Send a virtual packet from source node, and broadcast this packet into each branch of the source node.
- **step 3** When the virtual packet pass through the link, the link number is recorded in the virtual packet. When the virtual packet arrived at a new branch node, the virtual packet will be broadcasted.
- **step 4** Repeat step 3 until a virtual packet arrived at destination node. The information of first arrived virtual packet shows the shortest path.

**B. Implementation on DAPDNA-2**

We construct a diorama network in dynamically reconfigurable processor DAPDNA-2 [11] developed by IPFlex Inc [12]. DAPDNA-2 consists of DAP(Digital Application Processor), a high-performance RISC core DNA(Digital Network Architecture). The DNA is embedded in an array of 376 PE(Processing Elements), which are comprised of computation units, memory, synchronizers and counters. The DNA has several memory banks to store configurations. There are 3 banks memory in background memory and a bank in foreground. These 4 banks store 4 configurations, but just the foreground memory is active. DNA can change configuration by loading another configuration among 3 background memories.

![Fig. 3. Diorama network example consists of virtual link and virtual node using processor elements (PEs)](image)

We construct a diorama network by configuring several PEs in DNA. We set parameters of each PE to emulate various functions which real nodes and real links have. For searching the route from source node to destination node, we send a virtual packet which transmits in the processor. Figure 3 shows an example of our diorama network construction. There are 6 nodes and 10 links in real network, so we construct 6 virtual nodes and 10 virtual links by using several PEs in parallel processor, and connect them. Then we read a 32-bit data as a VPSP from the main memory, and transmit this VPSP in the processor. The virtual packets are broadcasted from the source node. Finally, we collect VPSP at destination node and write it into the memory. When we want to use this VPSP, we can read it from the memory.

Our proposed algorithm is implemented on evaluation board DPADNA-EB4. DPADNA-EB4 is Full-size PCI board and plugged into a PCI slot of the PC as shown in Figure 4. There are two DAPDNA-2 processors on a DPADNA-EB4.

![Fig. 4. The evaluation board DPADNA-EB4](image)

1) Implementation of Shortest Path Search: In this implementation of shortest path search, we define the bitmap of the VPSP and the link delay is fixed. This time, bandwidth is not considered in the implementation. The design of the virtual node and virtual link is also described in this subsection.

We first describe the bitmap of the VPSP. Figure 5 shows an example of the bitmap of VPSP.

In this example, a 32 bits VPSP is divided into two fields. Lower 26 bits of VPSP is a field where the link number is
recorded. The upper 6 bits is a field that the bandwidth of the passed route is recorded. However, this field is not used in this time.

The bitmap of a VPSP in this example can be used in a network which has 26 links. If network expand, we should use more VPSP to collect path information. For example, if we want to implement our proposed algorithm on a network which has 52 links, we can use two 32 bits VPSPs to collect path information. (We define this two packets is a one set.) At every branch point, we broadcast the set of virtual packets.

a) Virtual Node: Basic virtual node functions are shown below. Figure 6 shows an example of node that degree is three. The virtual node has three input ports and three output ports. One output port and one input port is a set of a virtual link. Many functions of the virtual node are shown below.

- Copy the VPSP from the input port.
- Send the VPSP to other output ports except the output ports which in a same virtual link. In our algorithm, the virtual packet will be broadcasted in each branch node. So, the VPSP should be sent to each output port.
- Conflict avoidance function. This function is to prevent conflicting of VPSPs when two or more VPSPs arrived in a virtual node in same time. We prepare two paths in one port, one path has larger delay than another one. When two VPSPs arrive at this port at same time, one VPSP will be sent to the path which has smaller delay, and another one is sent to a bigger path.

b) Virtual Link: Basic virtual link functions are shown below.

- Record the link number into the arrived virtual packet. Figure 7(a) shows an example of our virtual link design. In this example, we show the lower 8 bits of virtual packet only. When the virtual packet 00000100 pass through the link No.1, rightmost bit will be changed to one. Then, the output virtual packet is 00000101. If a virtual packet is passing the link No.2, then the second bit from right side will be changed to one.

- Prevent loop. Check the bitmap of the virtual packet. For example, check a bitmap of a arrived virtual packet in link No.1, if the rightmost bit is one, so the packet has passed this link. Figure 7(b) shows more detail of this function. When virtual packet 01100001 came into link No.1, we first calculate AND operation of 01100001 and 00000001 to record the link number. Then, we compare the result to zero. If the result equals zero, we send forward 01100001 to next node. However, in this example, the result is one, the virtual packet will be deleted.
- Generate delay. For example, if the delay of link in real network is three, we generate three clocks delay in a virtual link. So, the packet passes through different path, will take different clocks. The virtual packet passed minimum costs path will first arrive at the destination node.

The network shown in Figure 8 has six nodes. At first, we assign link numbers to all links and node numbers to all nodes. In this example, the network has 8 links, so the link number should be defined from No.1 to No.8. The VPSP is initialized by 0x00000000 and the all 32 bits of VPSP is used to record the link information.

Figure 8 shows an example of our shortest path search.
step 1 At the source node, we broadcast virtual packets 00000000 (in this example, we only show the lower 8 bits). The virtual packet will be sent forward to link 1 and link 2.

step 2 Virtual packets pass through the link No.1 and No.2, and the link number is recorded to the virtual packet. The output of link 1 and link 2 are 00000001 and 00000010 respectively. Then, we focus on node 1. Node 1 has 2 branches, so we send 00000001 and 00000100 to link No. 3 and link No. 4 in the same way. The output virtual packets of link No. 3 and link No. 4 will be 00001001 and 00010001.

step 3 When the VPSP arrived at other nodes, step 2 will be repeated until the virtual packet arrive at the destination node.

step 4 Finally, the information of the first arrived packet shows the shortest path.

In the network shown in figure 8, the first VPSP which arrived at the destination node is 0x01000101. The information of this VPSP shows the shortest path. The VPSP passed through link No.1, link No.3 and link No.7. So, the shortest path is 0-1-2-5.

Our proposed approach also can collect all route information between a source node and a destination node that not only the shortest path.

2) Implementation of Our TE Method: As the first trial, we implement the proposed shortest path search. This shortest path search can be used in our TE method. However, several functions should be added in virtual nodes and virtual links.

- Add a filed to record the bandwidth of passed virtual link in the bitmap of a VPSP.
- Add a function in virtual links to record the remaining bandwidth of the virtual link.
- Add bandwidth rewriting function in each virtual link.

We also can implement our TE method based on the implementation of the shortest path search. More works is required. For example, several works is shown below.

- Define the bitmap of VFPs.
- Link delay changing function should be added in virtual links.
- Implement of diorama network initializing function.

In this paper, we did not mention these works. These works are our future work.

C. Network Division

DAPDNA-2 has 376 PEs. We use a lot of PEs to emulate some complicated functions. When the network expands, the number of PE becomes insufficient to construct diorama network. Therefore, we use dynamic reconfiguration technique to solve this problem. Figure 9 shows the example of path search in a 10 nodes network. In this example, we divide the real network into three parts at first. We use three memory banks to store the diorama network configurations based on each network block. Then we decide how to change memory bank that use event rotation.

The summary of implementation is shown below. Step 1, step 2 and step 3 is same as the implementation of shortest path search.

step 4 To transmit VPSPs to next configuration, we use RAM element to save packets. The change of memory bank will be occurred when each RAM stored four virtual packets. We monitor all VPSPs in the diorama network. If no VPSP transmit in currently configuration, we change configuration to configuration which stored in memory bank 2.

step 5 Repeat step 3 and 4, until no virtual packets transmit in configuration which stored in memory bank 3. We collect all virtual packets which arrived at the destination node.

D. Automation of diorama network Construction

In Section II, we described several steps to achieve our goal. Step 4 is an important step to achieve our goal because design a DAPDNA-2’s configuration manually is inefficient and it is less attuned to a dynamically network change. The PEL (PE Language) is a special programming language which can describe the arrangement, parameter, and hard-wiring of PEs in the DNA. It can be used to construct diorama network automatically. The grammar of PEL is similar to C++.

The example of flow chart of automatic diorama network construction is shown in Figure 10. We write a program using C++ which generates the PEL code automatically when a network topology file was input. This program is called automation program. Then, the PEL source will be compiled by DNA compiler and execution file is created.

IV. PERFORMANCE EVALUATION

In this section, we implement the shortest path search in our constructed diorama network. We compare with Dijkstra’s algorithm, also compare all routes searching method and Breadth first search method. We execute two conventional algorithms using 3 GHz processor of Intel Pentium 4 processor and
execute our method using 166MHz reconfigurable processor of DPADNA-2.

A. Simulation Result of Dijkstra’s algorithm and our proposed

In this section, we compare the calculation clocks of Dijkstra’s algorithm [8], [13] to those of our proposed algorithm. The network topology is the same as Figure 9. We execute Dijkstra’s algorithm and our proposed algorithm 100 times and we obtain the average of execution time. The simulation result is shown in Table I.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Execution time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dijkstra’s algorithm</td>
<td>12.8</td>
</tr>
<tr>
<td>Proposed algorithm</td>
<td>0.67</td>
</tr>
</tbody>
</table>

The execution time of our algorithm is faster than the execution time of Dijkstra’s algorithm because our algorithm independent on the number of nodes. The Dijkstra’s algorithm needs labeling all nodes in network, but our algorithm only depends the total cost of the shortest path. So, our algorithm can search shortest path faster even if the network is large.

B. Simulation result of the Breath First Search algorithm and our proposed

We compare execution time of collecting all route information between the breadth first search [14] algorithm and our all path pattern search method. The simulation result is shown in Table II.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Execution time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breadth First Search algorithm</td>
<td>3500</td>
</tr>
<tr>
<td>Proposed algorithm</td>
<td>2</td>
</tr>
</tbody>
</table>

We can see that our algorithm collects all route information faster than the breadth first search algorithm because our algorithm broadcasts virtual packets and collects link information at each route at the same time.

V. Conclusion

In this paper, we proposed a novel traffic engineering approach using an on-chip diorama network. By emulating all traffics in a real network and recording path information in virtual path search packet, the proposal achieves a goal which searches the optimal path and distributes traffics. The strong point of our approach is use a global information when searching path and considering several metrics at a time. There is no deterioration of QoS because the proposed methods operated in a diorama network.

We explained several important steps to achieve our final goal and proposed a novel method to construct an on-chip diorama network in reconfigurable parallel processor DPADNA-2. For proving our proposed effectiveness of our approach, we implemented our shortest path search algorithm in our constructed diorama network. This algorithm also can search all route patterns in the network.

In performance evaluation, we showed that both of the calculation time of our shortest path search algorithm and all path pattern search algorithm faster than conventional method. Our algorithm can execute faster even when the network is large. So we proved that the experimental method based on an on-chip diorama network was effectiveness. In this paper, several basic steps of our approach was completed, the next goal is steps 4 and 6, which are described in Section II. If we complete these two steps, we can achieve the final goal of our approach that is, to distribute traffic dynamically. It will be a great progress of traffic engineering.

ACKNOWLEDGMENT

The authors thank Tomomi Sato and other staff for helping with implementation on DPADNA-2 (IPFlex Inc). This work was supported by Global COE Program “High-Level Global Cooperation for Leading-Edge Platform on Access Spaces (C12)” and by the Japan Society for the Promotion of Science’s (JSPS) Grant-in-aid for Scientific Research(B) (19360178).

REFERENCES

[12] IPFlex Inc (http://www.ipflex.com)