Scalable and Energy-efficient Optical Tree-based Greedy Router

Sahel Sahhaf, Abhishek Dixit, Wouter Tavernier, Didier Colle, Mario Pickavet, Piet Demeester

Internet-Based Communication Networks and Services (IBCN) research group of the Department of Information Technology (INTEC), Ghent University-iMinds

ABSTRACT

The IP prefix lookup as a core function of today's Internet routers is becoming a bottleneck due to ever increasing throughput demands. The increase in the size of the routing tables limits the scalability of current routing architecture. Greedy routing as an alternative to IP routing does not require lookups in order to find the next hop, and neither requires storing a forwarding table in every node. In this routing, network nodes are assigned 'coordinates' and the incoming packets are forwarded to the neighbours which are closer to the packets destinations. In this paper, a novel all-optical greedy router has been designed. In order to perform the greedy routing, each node is assigned an ID which determines the location of the node in the spanning tree of the network. The router is constructed through the interconnection of SOAs and directional couplers. The circuit provides high data rate and is more scalable and more energy-efficient than conventional IP routers.

Keywords: All-optical logic gates, all-optical flip flop, greedy routing, spanning tree

1. INTRODUCTION

Internet traffic keeps growing, the routing table sizes are increasing and higher speed links are demanded. In traditional IP routing, forwarding decisions are based on longest-prefix matching in a forwarding table. Therefore, the scalability of current routing protocols is limited by the size of the routing tables. [1] reports more than 400K RIB entries in current BGP routers. Greedy routing has been proposed as a solution to the scalability issue of IP routing.

In greedy routing, every node in the network is assigned a coordinate. Using these coordinates, each node forwards the incoming packets to the neighbour which is closer to the intended destination. In this scheme, only local knowledge is required in order to find the next hop of a packet. Each node stores only the coordinates of its neighbours which make the scheme more memory-efficient than traditional IP routers. Network coordinates can be based on the physical location of the nodes e.g. GPS coordinates, which leads to the term geographic routing [2] or can be based on virtual coordinates in a special metric space leading to geometric routing [3].

A problem in this scheme is that packets might reach local minima. This means that the current node is closer to the destination than any other neighbour. Greedy embeddings [4] avoid this situation by ensuring that coordinates are mapped to the nodes in a way that for every node, there is always a neighbour which decreases the distance toward the destination. Greedy routing based on this embedding never gives rise to local minima.

In this work, we use a greedy embedding based on the spanning tree of the network [5]. The embedding is a numbering which reflects the location of the nodes in the spanning tree. Greedy routing based on this numbering and the tree-distance between the nodes guarantees a 100% successful delivery to every destination.

Due to ever-increasing throughput demands, high data rate is becoming a significant requirement in the future Internet architectures. With more data in the network, the energy consumption of the communication components such as routers, servers, etc increases significantly. Due to the advances in optical technology, optical networks were found to provide energy-efficient solutions for high data rates [6]. Therefore, we implement our proposed greedy router using this technology. In comparison with a conventional IP router, the optical greedy router is more energy-efficient, more scalable in terms of the routing table size and provides high data rate.

The rest of the paper is organized as follows. In Section 2, the greedy embedding based on the spanning tree of the network is explained. Section 3 describes the architecture of the proposed circuit for the optical greedy router and its operation. Section 4 analyzes the design and its power consumption and Section 5 concludes the paper.

2. GREEDY EMBEDDING BASED ON THE SPANNING TREE OF THE NETWORK

Greedy embeddings are used to avoid local minima in greedy routing. In [5], we proposed a simple numbering of the nodes based on the spanning tree of the network to guarantee a successful delivery to every destination in the network. This numbering determines the location of each node in the tree. Tree-distances are used in order to find the next hop of a packet. The tree-distance between 2 nodes is the hop count on the tree between them. In order to calculate the numbering, first a spanning tree of the network is constructed. The children of each node are numbered from 1 to d (d = number of children). Finally, each node can calculate the coordinates for its children by adding the number assigned to each child after the last non-zero coordinate in its own coordinate set. All the coordinates of the root node are zero. Figure 1 depicts a sample numbering of the network nodes.

In order to calculate the tree-distance between 2 nodes, first their closest common parent should be found. Then the hop count on the tree from each node to the common parent is calculated. The sum of these two hop

counts indicates the tree-distance between them. This calculation can be performed using the coordinates of the nodes. Consider nodes '1,1,0' and '1,2,0' in Fig. 1 as a case study. The first common coordinates in the coordinate sets of the 2 nodes indicate the coordinate set of the closest common parent (first common coordinate in the two nodes is '1' and therefore '1,0,0' is the closest common parent). The number of non-zero coordinates after the common ones in each coordinate set indicates the hop count of the node to that common parent. In this example, both nodes have only one non-zero coordinate. The sum of these two hop counts determines the tree-distance between them which is 2 in this example.

Figure 1. Network nodes numbering

The tree-based greedy routing is different from tree routing, because the shortcut links (links which are not in the spanning tree of the network) can also be used.

In this work, we assume that the explained coordinates are calculated and assigned to the network nodes. Also, each node knows the coordinates of its neighbours. The focus of the work is on the greedy forwarder.

3. OPTICAL GREEDY ROUTER ARCHITECTURE AND OPERATION PRINCIPLES

In this section, first the general architecture of the proposed greedy router/forwarder is described and then the implementation of different components such as all-optical logic gates and all-optical flip flop are explained.

3.1 Greedy router architecture

A greedy router/forwarder is composed of two main components: i) a tree-distance calculator and ii) a comparator. The first component calculates the tree-distance between every neighbour and the destination of the incoming packet and the comparator finds the minimum value among the calculated distances (next hop). In this paper, we propose a circuit for the tree-distance calculator and the comparator can be based on existing all-optical designs [7]. For simplicity and directed focus on the functionality, we explain the tree-distance calculator design for only one neighbour. In a greedy forwarder, such a distance calculator is required for every neighbour of the node in the network. Therefore, the proposed circuit scales with the degree of the nodes in the network.

The general idea of this design is that given two coordinate sets, the first uncommon coordinates are located and the counting of non-zero coordinates starts from that location. As this counting is performed for both of the coordinate sets, the final value indicates the sum of the two hop counts and therefore, the tree-distance between them. Figure 2 depicts the architecture of the proposed design. This circuit is composed of logic gates such as AND, OR and XOR, a flip flop a counter and a multiplexer. All of these components are based on SOAs and directional couplers.

Figure 2. Architecture of tree-distance calculator for one neighbour

The inputs of the circuit are trains of bits. The coordinate sets of the destination (D) and a neighbour (N) are applied to a XOR gate to find the first uncommon bits in the two coordinate sets. Applying the output of the XOR to an OR gate together with a feedback from a flip flop results into a signal which is set to '1' from the moment the first uncommon bit is detected and it remains '1' up to last bit of the coordinate set. Using this signal, we know when the counting of the non-zero coordinates should be started. As we need to count the non-zero coordinates in both coordinate sets, we need to replicate this signal. Using a multiplexer and a fiber delay element with duration equal to the length of a coordinate set, we generate the same signal twice consecutively. Therefore, the length of the multiplexer's output is twice the length of a coordinate set.

Although we are working with bits, we need to consider that in tree-distance calculation, coordinates in a coordinate set should be counted and not the bits. Therefore, we use a pattern which determines the last bit of each coordinate in a set with '1'. Applying this pattern together with the output of the multiplexer to an AND gate results in to a signal which is '1' only at the last bit of the coordinates and only if the first uncommon bit in

the two coordinate sets is detected. If this signal is applied to the enabler of a counter, all the coordinates of the two sets from the location of the first uncommon coordinate are counted up to the end of the coordinate sets. However, only non-zero coordinates should be considered.

The second part of the circuit guarantees that only non-zero coordinates are counted. The two coordinate sets are applied to the inputs of an OR gate consecutively (ND). Looking at Fig. 2, one of the inputs is applied to the AOOR2 through a fiber delay with the duration of one bit. Considering that the coordinates of a set consist of two bits, if there is at least one bit equal to '1' in each coordinate, the output of the AOOR2 would be '1' in the location of the last bit of that coordinate. The required number of OR gates and the delay elements is dependent on the number of bits in a coordinate. This means that for N-bit coordinates, N-1 single-bit fiber delays and one N-way OR gate (N-1 2-input OR gates) are required which replace the AOOR2 and the delay element in Fig. 2.

Finally, feeding the outputs of the AOAND1 and AOOR2 to the AOAND2 results into a signal which is '1' only at the location of the last bit of a coordinate if the first uncommon coordinate in the two sets is detected and the coordinate is non-zero as well. Using this signal as the enabler of a counter results into counting the hop count of both nodes to their closest common parent. The counter can be based on existing all-optical designs [8]. In [9], other functionalities of an all-optical router such as clock recovery and header separation are discussed.

3.2 All-optical components

In this section, the architecture of different all-optical components used in the design of the tree-distance calculator is explained briefly. Figure 3.A illustrates a XOR gate. The part in the dash line determines a SOA-MZI configuration. The two inputs are fed to the arms of SOA-MZI. If both inputs are '1', due to the phase shifts through the couplers, the output signal would be destructive. The output is '1' only if one of the inputs is '1'.

Figure 3. All-optical components based on SOAs and directional couplers

In Fig. 3.B an AND gate is emulated by using 2 inverters. The selected parts work as inverters. The output is '1' only if the NOT of in2 is '0' and the in1 is '1'. The next gate is the OR gate. The two inverters (selected parts) are used to smoothen the output and to create only 2-level signals. Figure 3.D illustrates a multiplexer. The selected part shows a SOA-MZI configuration which works as a bar and cross state switch. In case the select is '1', the in2 is in the output and in case of '0' the in1 goes to the output.

The final figure depicts a flip flop which is level triggered. The selected part on top determines a SOA-MZI configuration which is used to put the input data to the output in case the clock is '1'. The selected part in the bottom is a feedback loop which maintains the output without any change in case there is no clock.

4. ANALYSIS OF THE PROPOSED DESIGN

As explained in Section 3.2, all the all-optical components are based on SOAs and directional couplers. As couplers are passive components, SOAs dominate the total power consumption of the proposed optical greedy router/forwarder. Table 1 reports the required number of SOAs in different components of a greedy forwarder. Each level of an all-optical counter consists of 3 SOAs [8], resulting into 15 SOAs for a 5-bit counter. Authors in [7] proposed an all-optical comparator which compares 2 N-bit patterns sequentially using 6 SOAs. The required number of SOAs is independent of the pattern length (N).

As mentioned earlier, the proposed design scales with the degree of the nodes in the network. Assuming the degree of 'm' for a node, we require 'm' tree-distance calculators and one m-way comparator to find the minimum value among the 'm' calculated distances. In case of using 2-input comparator, 'm-1' comparators are required. Below, we analyze the required number of SOAs in the proposed design for a port in the router

considering Internet topology. Based on the required number of SOAs, we roughly estimate the power consumption of the proposed design.

Based on the AS-level graph, the maximum degree of an AS is around 3K. These ASes are reached through many Internet exchange points (IXP) world-wide. As there exist roughly 70 IXPs in different areas, each IXP has around 40-50 (3K/70) neighbours. Based on these numbers, we assume that the number of peers in the AS border routers is around 40-50.

Table 1. Required number of SOAs in all-optical components

All-optical Component	Number of SOAs
AND	2
OR	2
XOR	2
Multiplexer	3
Flip flop	3
5-bit Counter	15
2-input Comparator	6

In greedy routing scheme, the number of bits in each coordinate should be large enough to be able to number all the peers of an AS. Based on the above explanation, 6 bits is considered in each coordinate.

As mentioned in Section 3.1, the required number of OR gates in part of the tree-distance calculator is dependent on the number of bits in a coordinate (equals to the number of bits-1). Therefore, 5 OR gates are needed. Assuming the maximum tree-distance between two nodes is less than 32 (the maximum tree depth is considered as 16), the 5-bit counter is sufficient for the distance calculators. In the greedy forwarder, we require 64 distance-calculators and 63 2-input comparators for each port. Based on the values reported in Table 1, the required number of SOAs for the proposed circuit for one port in the greedy router is 2874.

Assuming the power consumption of 100mW for each SOA results into 287.4W for the proposed circuit (independent of the data rate). Although, this is not precise power estimation and it is only based on the number of SOAs in the greedy router, it gives an overview on the energy-efficiency of the design compared to a 100Gbps slot card with power consumption of 542W (based on Juniper T-series router [10]). Note that the power estimation for the optical router was based on the discrete design. Applying the photonic integration technology will improve the energy-efficiency of the router significantly. Also sharing the proposed circuit among the ports of a greedy router is a solution to reduce the total power consumption.

5. CONCLUSIONS

In this paper, a novel all-optical greedy router was demonstrated. The greedy routing was based on tree-coordinates and tree-distances between the network nodes. The proposed design provides high data rate and is more memory and energy efficient than conventional IP routers. The design scales with the degree of the nodes in the network. The all-optical components are implemented through interconnection of SOAs and directional couplers. The analysis of the design showed the low power consumption of the circuit, considering 64 peers for AS border routers.

REFERENCES

- [1] G. Huston, "BGP routing table reports," 2013, <http://bgp.potaroo.net/>.
- [2] B. Karp et al., "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. 6th annual international conference on Mobile computing and networking. ACM*, 2000.
- [3] R. Fonseca et al., "Beacon vector routing: Scalable point-to-point routing in wireless sensor networks," in *Proc. 2nd conference on Symposium on Networked Systems Design & Implementation*, 2005.
- [4] C. Papadimitriou et al., "On a conjecture related to geometric routing," *Theoretical Computer Science*, vol. 344, no. 1, pp. 3–14, 2005.
- [5] S. Sahhaf, et al., "Single failure resiliency in greedy routing," in *Proc. DRCN*, 2013.
- [6] A. A. M. Saleh et al., "Evolution toward the next-generation core optical network," *IEEE/OSA Journal of Lightwave Technology*, vol. 24, no. 9, pp. 3303–3321, 2006.
- [7] M. Scaffardi et al., "All-optical comparator based on cross gain modulation in semiconductor optical amplifiers," in *Proc. Optical Fiber communication/National Fiber Optic Engineers Conference*, 2008.
- [8] J. Wang et al., "All-optical counter based on optical flip-flop and optical AND gate," in *Proc. ECOC '09*, 2009.
- [9] F. Ramos et al., "IST-LASAGNE: Towards all-optical label swapping employing optical logic gates and optical flip-flops", *Journal of Lightwave Technology*, vol. 23, no. 10, pp. 2993–3011, 2005.
- [10] W. Van Heddeghem et al., "Equipment power consumption in optical multilayer networks-source data. Technical Report IBCN-12-001-01 (January 2012), available at <http://powerlib.intec.ugent.be>