Computer Networks 83 (2015) 199–216

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Bandwidth-efficiency-oriented topology optimization for integrated switching systems based on circulant graphs

Dexiang Wang

Juniper Networks, Inc., Sunnyvale, CA 94089, USA

ARTICLE INFO

Article history:
Received 19 September 2014
Received in revised form 9 January 2015
Accepted 16 March 2015
Available online 23 March 2015

Keywords:
Bandwidth efficiency
Circulant graph
Dynamic programming
Integrated switching system
Topology optimization

ABSTRACT

This work studies a topology optimization problem by leveraging a regular network topology–circulant graph. The result of the topology optimization is applied to the design of the integration network for modern integrated switching systems (ISSs), which integrate individual switching devices together to boost switching capacity and to reduce network management complexity. The target of the optimization is to maximize bandwidth efficiency on the integration links and hence to maximize the ISS capacity to accommodate traffic. This eventually translates to minimizing the average inter-node hop distance within the ISS.

First, the problem is formulated into an integer linear program (ILP) and it is solved to its optimality for moderate-size networks. Then, by comparing with the Moore bound and unconstrained topology optimization, we show the circulant graph can be a qualified candidate with good bandwidth efficiency. Finally, due to the scaling limitation of the ILP approach, a dynamic programming (polynomial-time) algorithm is proposed and results show that it performs very closely to the optimal solutions induced from the ILP solver.

1. Introduction

Many applications of modern packet switching systems, such as for enterprise networks [1] and data center networks [2], demand advanced features like large scale, high capacity, low latency, and ease of management. Those features are not affordable by traditional standalone switching devices. Integrated switching systems (ISSs) have become a widely accepted solution to address the above challenges.[1] They provide flexible integration of individual switching devices with the help of newly developed control and management protocols. A network operator can easily manage switching resources (ports, VLANs, etc.) across all the switching devices within an ISS. An illustrative ISS is abstracted in Fig. 1, which consists of four switching devices (numbered 1, 2, 3, and 4). The four devices are connected via six internal integration links (1A–3A, 1B–4B, 1C–2C, 2A–4A, 2B–3B, and 3C–4C), which form an integration network with a full-mesh topology. The external network ports (usually a few dozen of them on each device) are configurable to admit traffic from (or to switch traffic to) the outside of the ISS. Network ports *a*, *b*, *c*, and *d* are shown as examples of external ports in Fig. 1 on each device. Three packet flows (flow 1, flow 2, and flow 3) are shown in Fig. 1. Flow 1 is admitted from port *a* of device 1 (named 1*a*) and is destined to egress from port *a* of device 3 (named 3*a*), while flows 2 and 3 are sourced from ports 1*c* and 1*d*, and are destined to ports 4*d* and 2*d*, respectively. The ISS calculates the shortest path within the integration network for each flow of packets. Hence, packets of flow 1 take integration link 1A–3A to be forwarded. Packets of flows 2 and 3 take integration links 1B–4B and 1C–2C to be forwarded respectively.

E-mail address: camelwdx@gmail.com

[1] Branded examples of ISSs include Juniper Networks' virtual chassis technology [3], Cisco Systems' switching stacks [4], etc.

http://dx.doi.org/10.1016/j.comnet.2015.03.012
1389-1286/

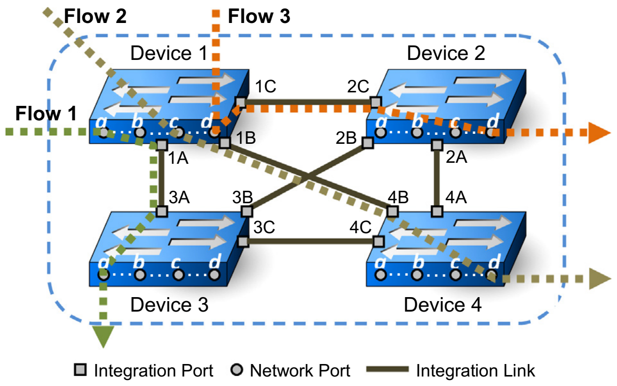

Fig. 1. An integrated switching system (ISS) with four switching devices and three packet flows traveling through the integration network.

1.1. Application of the ISS to data center networks

Besides the management and capacity advantages of ISSs that are traditionally desirable for enterprise and data center networks, the ISS offers additional potential to the data center networks. Current data center networks prevailingly appear in tiered architectures [5], as shown in Fig. 2(a). The whole data center network is topologically organized in a tree-like structure. The access tier is composed of top-of-rack (TOR) switches that connect the servers to the data center network fabric. The aggregation tier consists of high-capacity switches (usually modular switches) that interconnect the TOR switches, and the core tier is deployed to connect the data center to the wide area network (Internet). Depending on the size of the data center, the aggregation and the core tier may collapse into a single tier [6].

There are different types of communications during a data center's operation. For example, in a data center running MapReduce type applications on top of Hadoop, there are communications between servers in the same rack as well as between servers across different racks for the data exchanges generated during the procedures of Map and Reduce computation. The former does not introduce cross-rack traffic while the latter does and it consumes aggregation/core tier bandwidth, as illustrated by the traffic flow depicted in Fig. 2(a). There are also communications between servers and WAN due to the need of internet-facing applications. This also introduces traffic at the aggregation/core tiers. According to the reports in [6–9], the network bandwidth at aggregation/core tiers are much more heavily utilized than that at the access tier in the current real-world data centers. (The bandwidth utilization at the core tier can be 10–100 times higher than that at the access tier. 20% of the links at the core tier can be "hot spots" at least 50% of the time, while only 3% of the links at the access tier appear as "hot spots" for 0.1% of the time.)

In addition to the normal operations, data centers run routine operations for data reparation due to server failures. On a normal day, 20 server failures are typically expected in a data center of around 3000 servers [10]. Since a typical Hadoop fault tolerance strategy is to store 3 copies for each data block across the data center for resiliency, the reconstruction of damaged data copies needs to move the data copies across the racks, which adds an additional burden to the aggregation/core tiers [10]. Moreover, the recent research on applications of erasure codes to distributed storage systems suggests that network bandwidth can be traded for better storage efficiency [11]. For example, a modified (14, 10) Reed-Solomon code, called Xorbas, can achieve 1.6× storage efficiency compared with 3.0× storage efficiency led by the traditional 3-replica fault tolerance strategy [10]. However, this is at a cost of introducing 5-time traffic load to the network for the same amount of data reparation, because the reconstruction of damaged data blocks needs to move and aggregate a much higher number of data blocks for parity calculation than the simple 3-replica strategy needs to move. A calculation shows that 30% of daily network traffic can come from data reparation even if only 8% of data is protected by the (14, 10) Reed-Solomon code in a real-world data center [10]. Since most of the reparation traffic comes from inter-rack communications due to the requirement of better fault tolerance, the projected application of such code will further increase the bandwidth utilization at the aggregation/core tiers that have already been heavily loaded.

The ISS can be deployed as a good solution at the access tier to solve the problem, since it provides direct integration connections between TOR switches and hence the communications between racks do not need to go through the aggregation/core tiers. Fig. 2(b) shows an example where the TOR switches are connected simply through a ring and the cross-rack communications now can be carried at the access tier only. Note that the bandwidth efficiency of the integration network is determined by proper integration link investment and integration topology design, which is the major problem studied in this paper.

Virtualization has become a trend of evolution for data centers, from server virtualization to network virtualization. The same physical server can host multiple virtual servers that may virtually connect to each other through an internal virtual network. The virtual networks are then integrated into the physical network as above discussed via network interface virtualization (NIV) [12]. In such a