Computer Networks 56 (2012) 99–111

Contents lists available at SciVerse ScienceDirect

Computer Networks

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

Network protection codes: Providing self-healing in autonomic networks using network coding ☆

Salah A. Aly [a], Ahmed E. Kamal [b,*], Osameh M. Al-Kofahi [c]

[a] Dept. of Electrical Eng., Princeton University, Princeton, NJ 08544, United States
[b] Dept. of Electrical & Computer Eng., Iowa State University, Ames, IA 50011, United States
[c] Dept. of Computer Eng., Yarmouk University, Irbid, Jordan

ARTICLE INFO

Article history:
Received 16 August 2010
Received in revised form 2 June 2011
Accepted 25 August 2011
Available online 5 September 2011

Keywords:
Autonomic networks
Network protection codes
Self-healing
Link failures
Network coding
Code construction

ABSTRACT

Agile recovery from link failures in autonomic communication networks is essential to increase robustness, accessibility, and reliability of data transmission. However, this must be done with the least amount of protection resources, while using simple management plane functionalities. Recently, network coding has been proposed as a solution to provide agile and cost efficient self-healing against link failures, in a manner that does not require data rerouting, packet retransmission, or failure localization, hence leading to simple control and management planes. To achieve this, separate paths have to be provisioned to carry encoded packets, hence requiring either the addition of extra links, or reserving some of the resources for this purpose.

In this paper we introduce self-healing strategies for autonomic networks in order to protect against link failures. The strategies are based on network coding and reduced capacity, which is a technique that we call *network protection codes* (NPC). In these strategies, an autonomic network is able to provide self-healing from various network failures affecting network operation. Also, *network protection codes* are extended to provide self-healing from multiple link failures in autonomic networks. Although this leads to reducing the network capacity, the network capacity reduction is asymptotically small in most cases of practical interest. We provide implementation aspects of the proposed strategies, derive bounds and show how to construct *network protection code*. The paper also develops an Integer Linear Program formulation to evaluate the cost of provisioning connections using the proposed strategies, and uses results from this formulation to show that it is more resource efficient than 1 + 1 protection. A simulation study to evaluate the recovery times, and the buffering requirements due to network coding is also conducted using the OPNET simulator.

1. Introduction

Today's communication networks are becoming complex to the degree that the management of such networks has become a major task of network operation. Therefore, the use of network autonomy such that the management functionality and its complexity is moved to within the network has become the preferred approach, hence giving rise to what is known as autonomic networks [15]. Autonomic networks are self-managed, and they are efficient, resilient, evolvable, through self-protection, self-organization, self-configuration, self-healing and self-optimization (see for example [6,8,16] and the references therein). Therefore an autonomic network promotes the autonomy of network operation with minimum human involvement. However, it is also important not to overload the management plane

☆ This paper was presented in part at the IEEE Globecom 2008 Conference, New Orleans, LA, December 1–4, 2008 [2].

* Corresponding author.
E-mail addresses: salah@princeton.edu (S.A. Aly), kamal@iastate.edu (A.E. Kamal), osameh@yu.edu.jo (O.M. Al-Kofahi).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.08.013

of autonomic networks to the degree that the management functionality consumes significant amounts of computing and communication resources. This paper addresses the self-healing functionality in autonomic networks. Self-healing has been traditionally implemented using techniques such 1 + 1 protection, 1:N protection, and dynamic restoration [19]. 1 + 1 Protection is a proactive, agile, and a rather expensive technique, in which each working path is protected using another backup path on which a second copy of the signal is transmitted, hence providing instantaneous recovery from working path failures. 1:N is a less expensive protection technique, in which one protection circuit is shared between multiple working connections. Connections which are jointly protected must be routed on link disjoint paths. The failure of any working connection, once detected, will result in rerouting its traffic on the backup circuit. Although this technique is less expensive than 1 + 1, it is slower since it involves failure detection and localization, and data rerouting. Dynamic restoration does not reserve any protection resources, and if a failure occurs, network resources must be discovered, and then used for rerouting traffic from failed connections. This is the slowest of the three approaches. Most modern self-healing schemes, such as MPLS Fast Rerouting [14], use 1:N protection.

This paper introduces a technique to provide self-healing that results in simplifying the management plane, as well as the control plane. The technique uses reduced capacities and network coding.

Network coding is a powerful tool that has been used to increase the throughput, capacity, and performance of communication networks [1,5]. It offers benefits in terms of energy efficiency, additional security, and reduced delay. Network coding allows the intermediate nodes not only to forward packets, but also encode/decode them using algebraic primitive operations. We illustrate the principles of network coding using the example shown in Fig. 1, in which two sources, A and B, would like to deliver their transmitted data units, a and b, respectively, to both of two destination nodes, T_1 and T_2. A and B can deliver their data units to T_1 and T_2, respectively, on the outer links. However, since the data units a and b, which are to be delivered to T_2 and T_1, respectively, on the inner links, will collide on the link from C to D, network coding is employed. In this case the sum $a + b$ is formed, where the addition is over the binary field, and is delivered to both T_1 and T_2, as shown in the figure. T_1 and T_2 can recover b and a by adding $a + b$ to the data units, a and b, which are received on the outer links, respectively. Without network coding, the link C to D has to alternate between carrying a and b, hence resulting in a lower network throughput.

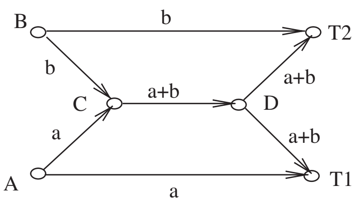

Fig. 1. An example to illustrate network coding.

One application of network coding that has been proposed recently is to jointly protect a number of link disjoint connections against link failures [10–12]. This is achieved by transmitting combinations of data units from multiple (link disjoint) connections on a backup circuit in a manner that enables each receiver node to recover a copy of the data transmitted on the working path. In case a single working path fails, the receiver can use this copy for the purpose of data recovery. This recovery is achieved without failure localization, and without data rerouting. It is also achieved by sharing a common protection circuit, which takes the form of a p-Cycle [10], or a tree [12], among a number of connections, hence achieving the protection functionality in a cost efficient manner. This, however, requires the provisioning of extra protection circuits over which the combined data units are transmitted. Such circuits may require the addition of links to the network under the Separate Capacity Provisioning strategy (SCP), or that paths be provisioned using existing links if the Joint Capacity Provisioning strategy (JCP) is used, hence reducing the network traffic carrying capacity.

Certain networks can allow extra transmissions and the addition of bandwidth, e.g., the addition of wavelength channels on existing fibers, but they do not allow the addition of new transmission lines, e.g., the addition of new fibers. In this paper, we propose an approach in which we use network coding to provide agile, and resource efficient protection against link failures, and without adding extra paths. The approach is based on combining data units from a number of sources, and then transmitting the encoded data units alternately on the different connections, hence using a small fraction of the bandwidth allocated to the connections in a fair manner. This disposes of the requirement of having extra protection circuits. In this scenario, once a path fails, the receiver can recover the lost packets easily from the neighbors by initiating simple queries.

The contributions in this paper can be stated as follows:

(i) We introduce a self-healing strategy against single link failures using network coding and a reduced capacity strategy, rather than using dedicated protection circuits. The developed protection strategy is achieved over the binary field, hence the encoding and decoding operations are done using bit-wise XOR operation.
(ii) Although single link failures are the most common type of failures in networks, multiple link failures may also occur, albeit with a smaller probability. We therefore extend the above scheme to protect against multiple link failures.
(iii) We develop a theoretical foundation of *protection codes*, in which the receivers are able to recover data sent over t failed links out of n primary links.

This paper is organized as follows. In Section 2 we present the network model and problem definition. Sections 3 and 4 discuss single and multiple link failures and how to protect these link failures using reduced capacity and network coding. Section 5 presents code constructions and an example of a *network protection code*. In Section 6 we present an Integer Linear Program to find the optimal