Computer Communications 35 (2012) 1535–1548

Contents lists available at SciVerse ScienceDirect

Computer Communications

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

On the queueing behavior of inter-flow asynchronous network coding

Y. Yuan [a,*], K. Wu [b], W. Jia [c], Y. Peng [a]

[a] National Lab for Parallel and Distributed Processing, National University of Defense Technology, Changsha, China
[b] Dept. of Computer Science, University of Victoria, B.C., Canada
[c] Dept. of Computer Science, City University of Hong Kong, Hong Kong SAR, China

ARTICLE INFO

Article history:
Received 17 March 2011
Received in revised form 30 March 2012
Accepted 24 April 2012
Available online 3 May 2012

Keywords:
Performance evaluation
Network coding
Queueing analysis
Congestion control

ABSTRACT

Despite the substantial research efforts on network coding, its real-world implementation is mainly over wireless networks or peer-to-peer networks. The deployment of network coding in the Internet core still largely lags behind. Among the many challenges, one difficulty is the selection of routers to perform network coding, which relies on the understanding of the queueing behavior of network coding. Unfortunately, the intricate queueing behavior of network coding, even for a single node case, is still unclear. In this paper, we build a generic queueing model to answer many fundamental questions, including for example, under what condition is the system stable? How many packets could be possibly coded when multiple stochastic traffic flows pass through a coding node? What is the quantitative relationship among the traffic arrival rate, the service rate, and the coding opportunities under a general network configuration? Based on our analytical results, we propose a self-adjustable delay-based coding mechanism for better congestion control. Our work provides network researchers and engineers with insights on the queueing behavior of network coding, which are helpful in future applications of network coding in the Internet core.

1. Introduction

The concept of network information flow was introduced in 2000 [1], and since then network coding has triggered enormous research and development activities. In the existing Internet architecture, network devices such as routers and switches mainly function as store-and-forward relay devices. Network coding technique changes the above fundamental design principle and allows the algebraic combination of data packets at intermediate network nodes by using, for example, a bit-by-bit XOR operation. Destination nodes perform decoding operations to recover the original data packets.

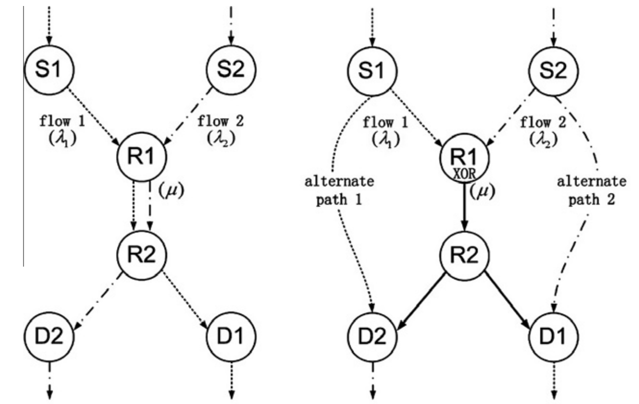
In theory, it has been proved [1] that network coding can achieve the maximum information flows for multicast, which are otherwise impossible for traditional store-and-forward networks. In practice, network coding has been implemented and tested in different systems, particularly in wireless networks where radio transmission is broadcast in nature and a node can overhear neighbors' transmissions [2]. Network coding has also found its success in peer-to-peer networks for content distribution [3]. Extra benefits of network coding include reducing energy consumption for wireless networks, enhancing network security and reliability, reducing bandwidth cost for content distribution over P2P networks, and much more.

With the practical success of network coding in wireless networks, it becomes a natural and important question how network coding can be deployed in the Internet core. This question cannot be simply answered because network coding requires substantial changes on current router architecture and existing protocols for traffic control. Among the many challenges, one difficulty relies on the subgraph selection in large networks, i.e., the selection of routers to perform network coding. To illustrate this, an example using the broadly cited "butterfly" network in the Internet core is depicted as follows:

Example 1. As illustrated in Fig. 1, flow 1 ($S1 \rightarrow D1$) and flow 2 ($S2 \rightarrow D2$) share the link from R1 to R2. Suppose every link is lossless. Due to the heavy traffics from S1 and S2, router R1 may eventually get congested (Fig. 1(a)). To reduce the traffic load, R1 first notifies S1(S2) to route its packets to D2(D1) via redundant path 1(2), then XORs the buffered packet pairs from each flow and transmits the coded packets to router R2 (Fig. 1(b)). Note that D1 or D2 can decode the coded packets by calculating the XOR with its corresponding original packets from the redundant path. However, since the two flows arriving at R1 may not be well synchronized, R1 may only buffer packets from one of the two flows, such that the coding opportunities among the two flows may not always exist.

* Corresponding author. Tel.: +86 13875853342.
E-mail addresses: yuanyuan@nudt.edu.cn, kenyuan44@gmail.com (Y. Yuan), wkui@cs.uvic.ca (K. Wu), wei.jia@cityu.edu.hk (W. Jia), pengyuxing@nudt.edu.cn (Y. Peng).

0140-3664/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2012.04.020



(a) Using traditional forwarding, router R1 gets congested when $\lambda_1 + \lambda_2 \geq \mu$

(b) Using network coding, the congestion at R1 is alleviated

Fig. 1. Example 1: Traffic control with network coding in the Internet core. Suppose that λ_1 and λ_2 denote the mean flow arrival rates, and μ denotes the mean service rate).

Thus, to increase the coding opportunities, a delay is introduced into the coding mechanism of R1. Specifically, this delay-based coding mechanism can be implemented as follows:

- R1 maintains a separate queue for each flow.
- When R1 becomes idle, it checks whether the queue for flow 1 and the queue for flow 2 both have packets.
- If both have packets, R1 XORs the packets at the head of each queue and sends out the coded packet.
- If not, R1 waits for a preset maximum delay to increase the coding opportunities. During this period, if both queues become non-empty, R1 performs network coding and transmits the coded packet as soon as possible. Otherwise, R1 transmits the head packet in the non-empty queue without network coding. We call the waiting delay as *opportunistic delay*, and the preset maximum delay as the *maximum opportunistic delay*.

However, this simple scenario leaves us many interesting problems about network coding in the Internet core: Is the router competent for the traffic control? How should the router operate its network coding mechanism? To answer these questions, we need a better understanding of the intricate queueing behavior at coding nodes to quantitatively disclose the relationship among traffic patterns, service pattern, coding opportunities, and coding gain.[1] This analysis will provide important guidance on the protocol design and the route selection over the Internet core.

Unfortunately, people's understanding on the queueing behavior of network coding is still very limited. The challenge comes from the complex correlation among the queueing packets. Specifically, as for Example 1, When the maximum opportunistic delay is set to be infinity, a coding node holds its coding operation until all the "required" packets from each flow arrive. This is referred as "synchronous network coding", and is equivalent to an assembly-like queueing system, which has been comprehensively studied by the queueing system community. It has been proved that with more than two input Poisson input flows, the state space will explode and the problem becomes intractable to analyze [4]. The synchronous coding also has been proved to be unstable with unbounded buffer size, unless the arrivals are deterministic and fully synchronized [5]. When a coding node allows partial "required" packets coded together or perform no coding during a bounded maximum opportunistic delay, this is referred as"asynchronous network coding". However, the queueing behavior of asynchronous network coding is more complex than synchronous one, since the number of packets that can be coded is random. Many efforts have been devoted to the performance analysis of asynchronous network coding [6–8], but simple assumptions (e.g. memoryless traffics) are generally posed and none deals with a generic queueing model. We stress that this high-level classification of synchronous and asynchronous network coding is generic enough and divides existing network coding strategies into different domains with respect to queueing behavior.

Unlike previous network coding research focusing on code design [9] and optimal scheduling strategies [10], in this paper, we constrain ourselves to the queueing behavior analysis of asynchronous network coding and its applications. In this paper, a generic queueing model with arbitrary stochastic traffic arrivals and service capacity is built to describe the network coding system of a single router. By analyzing the coding opportunities, we construct a matrix to decouple the correlation among the queueing packets. According to the coding opportunity matrix, we simplify our analysis by reducing the DoF (degree of freedom) among the preset variables. Then based on the queue decomposition technique, we obtain an approximation solution for the queueing model. We further discuss the stability condition and the coding gain for this system. Finally, a self-adjustable delay-based coding mechanism is proposed based on our analytical results. This method controls the congestion effectively and achieves a nearly optimal coding gain.

To the best of our knowledge, existing queueing modeling and analysis on asynchronous network coding all assumed the memoryless network configurations. In this paper, we are thus motivated to answer the following questions:

- What is the quantitative relationship between the average packet delay, the traffic arrival rate, the service rate, and the maximum opportunistic delay, when the traffic arrivals and the service times are both stochastic and follow **general** distributions?
- Under what condition the system is stable in the sense that the router's buffer will not eventually overflow as sure?
- In a stable system, what is the maximum coding gain at the coding node?
- How can we use network coding for better congestion control?

2. Related work

Network coding was first introduced by Ahlswede et al. in 2000 [1]. Since then, substantial research has been devoted to studying the performance of network coding under various application contexts. Various network coding schemes have been developed. They differ from each other in scheduling strategies and/or network stacks where network coding is implemented. According to the code generation scheme, network coding could be implemented deterministically [11] or randomly [12]; according to the network stacks where network coding is implemented, it could be divided into application-layer network coding [13], network-layer network coding [14], MAC-layer network coding [15], or physical-layer network coding [16]; according to the operations on flows, network coding could be classified as inter-flow [17] and intra-flow network coding [18]. In this paper, we are interested in the queueing analysis of network coding and group our discussion into synchronous network coding and asynchronous network coding from the viewpoint of queueing behavior.

[1] It is defined as the ratio of the number of transmissions required by forwarding, to the number of transmissions used by network coding to deliver the same set of packets.