Journal of Network and Computer Applications 56 (2015) 1–18

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

ACCPndn: Adaptive Congestion Control Protocol in Named Data Networking by learning capacities using optimized Time-Lagged Feedforward Neural Network

Amin Karami *

Computer Architecture Department (DAC), Universitat Politècnica de Catalunya (UPC), Campus Nord, Jordi Girona 1-3, 08034 Barcelona, Spain

A R T I C L E I N F O

Article history:
Received 10 May 2014
Received in revised form
30 April 2015
Accepted 19 May 2015
Available online 2 July 2015

Keywords:
Named data networking
Congestion control
Time-lagged feedforward network
Particle swarm optimization
Genetic algorithm
Fuzzy set

A B S T R A C T

Named Data Networking (NDN) is a promising network architecture being considered as a possible replacement for the current IP-based Internet infrastructure. However, NDN is subject to congestion when the number of data packets that reach one or various routers in a certain period of time is so high than its queue gets overflowed. To address this problem many congestion control protocols have been proposed in the literature which, however, they are highly sensitive to their control parameters as well as unable to predict congestion traffic well enough in advance. This paper develops an Adaptive Congestion Control Protocol in NDN (ACCPndn) by learning capacities in two phases to control congestion traffics before they start impacting the network performance. In the first phase – adaptive training – we propose a Time-Lagged Feedforward Network (TLFN) optimized by hybridization of particle swarm optimization and genetic algorithm to predict the source of congestion together with the amount of congestion. In the second phase -fuzzy avoidance- we employ a non-linear fuzzy logic-based control system to make a proactive decision based on the outcomes of first phase in each router per interface to control and/or prevent packet drop well enough in advance. Extensive simulations and results show that ACCPndn sufficiently satisfies the applied performance metrics and outperforms two previous proposals such as NACK and HoBHIS in terms of the minimal packet drop and high-utilization (retrying alternative paths) in bottleneck links to mitigate congestion traffics.

1. Introduction

Information-Centric Networking (ICN) (Jacobson et al., 2009; Ahlgren et al., 2012; Dannewitz et al., 2013) has been proposed as a solution for a viable and vital replacement for the current IP-based Internet due to the fundamental limitations of the Internet in supporting today's content-oriented services (Conti et al., 2013; Lee and Nakao, 2013; Rossini and Rossi, 2013; Li et al., 2012). Named Data Networking (NDN) (Jiang and Bi, 2013) is a prominent example and ongoing research effort of ICN design. The main goal of NDN is to support the dissemination of named content rather than the current host-centric (end-to-end) delivery of content to a named host. In NDN, a consumer asks for a *Content* (*Data*) by sending an *Interest* request using name prefix (content identifier) instead of today's IP prefix (content location). An Interest packet is routed towards the location of the content's origin where it has been published. Any router (intermediate node) on the way checks its cache for matching copies of the requested content. The requested content is returned by any node that holds a copy of the content in its cache. On the way back, all the intermediate nodes store a copy of the content in their caches to satisfy subsequent users interested in that content (i.e., in-network caching). Congestion takes place in NDN routers when the number of arrival data packets is higher than the queue's capacity which causes an overflow in the routers' buffer (Rozhnova and Fdida, 2012; Saino et al., 2013; Karami and Guerrero-Zapata, 2015a). When this happens a high data packet loss and increase in the end-to-end delay occur affecting negatively on the performance, stability and robustness of the network (Qian et al., 2008; Lestas et al., 2007). This leads to under-utilization of the available resources and degradation of throughput and quality of service (Xu and Sun, 2014; Li et al., 2014).

This difficulty has recently motivated researchers to explore ways of congestion control in NDN. Some of the relevant contributions are (Rozhnova and Fdida, 2012; Saino et al., 2013; Fu et al., 2012; Saucez et al., 2012; Carofiglio et al., 2013; Xia et al., 2013; Yi et al., 2013). The main weak points of the proposed methods are: a too high sensitivity to their control parameters as well as the inability to predict congestion traffic well enough in advance. This will often bring unfair bandwidth sharing, network instability, packet loss, additional delay and so on (Barabas et al., 2011a; Bonald et al., 2000). The first goal of any method against congestion can be the early detection (ideally long

* Tel.: +34 934011638.
E-mail address: amin@ac.upc.edu

http://dx.doi.org/10.1016/j.jnca.2015.05.017
1084-8045/

before the problematic traffic builds up) of its existence. If the congestion problem can be recognized in advance, changing network parameters can possibly prevent such costly network breakdowns. Network traffic prediction plays an important role in guaranteeing quality of service in computer networks (Cortez et al., 2012). The prediction of network traffic parameters is feasible due to a strong correlation between chronologically ordered values (Barabas et al., 2011a). Their predictability is mainly determined by their statistical characteristics including self-similarity, multi-scalarity, long-range dependence (LRD) and a highly non-linear nature (Dharmadhikari and Gavade, 2010). Prediction algorithms can be embedded into network communications to improve the global performance of the network by anomaly detection, proactive congestion detection (or avoidance), and allow a better quality of service by a balanced utilization of the resources (Cortez et al., 2012; Li et al., 2009; Alarcon-Aquino and Barria, 2006). Contributions from the areas of operational research, statistics and computer science have lead to forecasting methods. In particular, the field of Time Series Forecasting (TSF) deals with the prediction of a chronologically ordered values (Makridakis et al., 1997; Claveria and Torra, 2014). The goal of TSF is to model a complex system as a black-box in order to predict the systems behavior based on the historical data (Barabas et al., 2011a; Cortez et al., 2006).

In this paper, we develop ACCPndn (Adaptive Congestion Control Protocol in Named Data Networking) which is a new congestion control protocol with learning capacities. The ACCPndn focuses on two phases for congestion control before building up in NDN. *The first phase – adaptive training* – learns from the past breakdowns to how to detect the problem beforehand. This phase allows to identify the source of the congestion together with the amount of congestion. This phase uses time-lagged feedforward neural network (TLFN) approach. The TLFN adopts a multilayer perceptron ANN (Artificial Neural Network) and time series forecasting (TSF) (Peralta Donate et al., 2011; Gomes and Ludermir, 2013). The major advantages of neural networks in time series forecasting are their flexible nonlinear modeling capability that there is no need to specify a particular model form and high data error tolerance (Ren et al., 2014; Khashei et al., 2008). A Back-Propagation is a most popular NN algorithm (BPNN) to determine and adjust network parameters, weights and biases. Despite the advantages of BPNN, it has some drawbacks that the most important one being their poor trainability. It might fall to local minima and cause overfitting and failure of the network training (Das et al., 2014; Wong et al., 2010). There is a recent trend to train BPNN with bio-inspired optimization algorithms for different applications (Yogi et al., 2010; Malviya and Pratihar, 2011; Lee and Lee, 2012). In this paper, in order to improve the performance of BPNN, a new combined algorithm namely particle swarm optimization (PSO) and genetic algorithm (GA) is presented to optimize the weights and the biases of network, and to prevent trapping in local minima. The results show that our proposed combination of PSO/GA with TLFN (TLFN+PSO-GA) performs better than the GA/PSO, PSO and GA in terms of the applied performance criteria.

When the source(s) and the amount of congestion are identified, they are sent to the second phase for congestion control before building up. *The second phase – fuzzy avoidance* – performs a fuzzy decision-making approach to proactively respond to network congestion rather than simply wait for a congested queue to overflow and the tail drop all subsequently arriving data packets. The application of fuzzy decision-making techniques to the problem of congestion control is suitable due to the difficulties in obtaining a precise mathematical (or a formal analytical) model, while some intuitive understanding of congestion control is available (Chrysostomou and Pitsillides, 2003; Chrysostomou et al., 2009). Its use allows to regulate effectively the incoming Interest packets in each routers' interface.

The main objective of the proposed protocol is to enable a stable equilibrium and satisfy some basic requirements which are characterized by the utilization of multiple paths and few packet drops. The second objective is to present a scalable and fast convergence properties with respect to varying bandwidths, traffic patterns, and number of users at different times utilizing the network. The evaluation through simulations shows that ACCPndn can quickly and effectively respond against congestion problems in a timely manner and performs successfully even in the large networks as compared to two recent congestion control mechanisms namely NACK (Yi et al., 2013) and HoBHIS (Rozhnova and Fdida, 2012) in terms of the applied performance metrics.

The rest of the paper is organized as follows. Section 2 describes the background materials of NDN. Section 3 gives an overview of the time series forecasting approach. Sections 4 Sections 5 describe the PSO and the GA, respectively. In Section 6, multilayer perceptron neural network is described in detail. Section 7 provides a overview of fuzzy set theory. The proposed method (ACCPndn) is completely presented in Section 8. Section 9 presents the evaluation setup. The results of simulation are also presented in Section 10. Section 11 discusses the results of this paper. Finally, Section 12 draws conclusions.

2. Named Data Networking (NDN)

NDN communication protocol is receiver-driven and data-centric. All communication in NDN is performed using two distinct types of packets: `Interest` and `Data`. Both types of packets carry a `name`, which uniquely identifies a piece of data that can be carried in one data packet (Afanasyev et al., 2013; Ran et al., 2013). Data names in NDN are hierarchically structured, e.g., eight fragment of a youtube video would be named `/youtube/videos/A45tR7Kg5/8`. In addition to the data name, each Interest packet also carries a random `nonce` generated by the consumer. A router checks both the name and nonce of each received Interest packet. If a newly arrived Interest packet carrying the same name as a previously received Interest packet from a different consumer, or a previously forwarded Interest looped back, the router drops the Interest packet. Therefore Interest packets cannot loop. Each NDN router maintains three major data structures (Karami and Guerrero-Zapata, 2015b,c):

1. The Pending Interest Table (PIT) holds all not yet satisfied Interest packets that were sent upstream towards potential data sources. Each PIT entry holds one or multiple incoming physical interfaces and their corresponding Interest packets.
2. The Forwarding Information Base (FIB) forwards Interest packets to one or multiple physical network interfaces based on the forwarding strategies. The strategy module makes forwarding decisions for each Interest packet.
3. The Content Store (CS) or buffer memory temporarily buffers data packets for data retrieval efficiency.

When a NDN router receives an Interest packet, it first checks its CS (cache). If there is no copy of the requested content, it looks up its PIT table. If the same name is already in the PIT and the arrival interface of the present Interest is already in the set of arrival interface of the corresponding PIT entry, the Interest is discarded. If a PIT entry for the same name exists, the router updates the PIT entry by adding a new arrival interface to the set. The Interest is not forwarded further. Otherwise, the router creates a new PIT entry and forwards the present Interest using its FIB. When an Interest packet is satisfied by the content's origin where it was published, on the way back, all the intermediate nodes store a copy of content in their caches to answer to probable same Interest requests from subsequent requester (Muscariello et al., 2011; Karami and Guerrero-Zapata, 2015b).