Journal of Advanced Computer Science & Technology, 3 (2) (2014) 214-220
©Science Publishing Corporation
www.sciencepubco.com/index.php/JACST
doi: 10.14419/jacst.v3i2.3696
Research Paper

A neural network based congestion control algorithm for content-centric networks

Parisa Bazmi *, Manijeh Keshtgary

Computer Engineering and IT Department, Shiraz University of Technology (SUTECH) Shiraz, Iran
**Corresponding author E-mail: Bazmi.Parisa@gmail.com*

Abstract

Communication across the Internet has transformed over the years, generated primarily by changes in the importance of content distribution. In the twenty-first century, people are more concerned with the content rather than the location of the information. Content-Centric Networking (CCN) is a new Internet architecture, which aims to access content by a name rather than the IP address of a host. Having the content, CCN which is natively pull-based functions based on the requests received from customers. It is also combined with the availability of in-network chaching. Because of the availability of in-network caching in CCN, chunks may be served by multiple sources. This multi-path transfer in CCN makes TCP-based congestion control mechanisms inefficient for CCN. In this paper a new congestion control algorithm is proposed, which is based on Neural Network prediction over content-centric networks. The designed NN is implemented in each router to predict adaptively the existence of the congestion on link given the current status of the network. The results demonstrate that the proposed congestion control algorithm can effectively improve throughput by 85.53%. This improvement is done by preventing queue overflow from happening, which will result in reductions in packet drop in the network.

Keywords: *Content-Centric Network, Congestion Control, Drop Prediction, Named Data Networking, Neural Network.*

1. Introduction

In 1960s, Internet architecture was being developed that was based on establishing connections between two remote hosts for resource sharing. However, today while the consumer's concern is the content, Internet is still working based on where the resources are located. This transmission of what to where has made the Internet architecture inefficient and therefore unable to serve the increasing needs and requirements of companies and individuals. As a result, a new architecture has been proposed in order to access content by a name, rather than the IP address of the device storing the content. Content-Centric Networking (CCN) [1] also called Named Data Networking (NDN) [2] is a recently proposed Internet architecture based on named data.

The sender is separated from the receiver by NDN/CCN that is similar to the Publish and Subscribe service model [3]. Communication in CCN is driven by the consumer. The consumer sends out an Interest packet, which carries a name that identifies the desired data. The Data packet is then sent back to the consumer via a path created by the Interest packet.

The combination of content name with CCN's per-packet content signatures allows any node in the network to cache named data. As a result, the data received by the consumer didn't necessarily come from the content publisher, as the content can be delivered by any cache in the network. Chunks may be served by different network nodes when retrieving an entire content object because contents are cached with a packet-level granularity. This multi-path transfer in CCN makes TCP-based congestion control mechanisms (Time out or three duplicate ACK) inefficient for CCN.

Nowadays, CCN congestion control, such as two methods, the end-to-end and hop-by-hop, is widely proposed in many papers [4]-[12]. The CCN infrastructure has created new challenges for the design of transmission protocols. As illustrated, in CCN the connection is driven by the receiver and the desired data are requested by them. Also in this network, in relation to caching the packets in the routers and the presence of several sources for each content, the hop-by-hop transmission is used instead of end-to-end in order to transmit the data to the receiver. Thus it is possible that one packet from the A source is sent to the receiver and next packet of the content is reached to the receiver from the B source. Therefore, the single RTT estimator, which is used in the receiver to control the congestion in the IP network and is also proposed in the [4]-[6] papers for congestion control in CCN, is not an appropriate method to control the congestion in CCN [8] [9].

The congestion prediction in packets' routing and their transmission management can play a significant role in ensuring and quarantining the quality of network application services. Avoiding congestion, and therefore packet loss, can improve the quality of network application services, especially for multimedia applications where retransmission of the discarded packets is not a feasible and appropriate method. In this paper a new method of congestion control in CCN is proposed by using the Neural Network (NN), but which does not have the complexity of the present congestion control methods. The NN can learn the complex patterns and can be used as an efficient predictor of traffic in computer networks, given their high ability of self-learning and versatility. The NN are widely used in computer networks to predict the traffic on routers and network congestion control [13]-[16].

The designed NN in this paper predicts adaptively the existence of the congestion on link given the current status of the network. The NN has been implemented in routers and acts in the strategy layer. Before each Interest is transmitted on the router's outgoing interface, the congestion presence on the link is predicted and is avoided.

The reminder of this paper is organized as follows. Section 2 describes the overall features of the CCN node model, while the related work on congestion control in CCN is described in Section 3. A detailed description of NN based congestion control is given in Section 4. The NN design and evaluation is presented in Section 5 and 6. Section 7 presents performance evaluation of proposed algorithm, and concluding remarks are given in Section 8.

2. CCN node model

The main function of a CCN node that is introduced in [1] is very similar to a TCP/IP node: the Interest packet is received on a router interface and a longest prefix match is done on its name. Interest is sent depending on an interface specified in the search. Data follows the exact same path (in the reverse direction) as the Interest that solicited it. Each CCN node is composed of three components: Forwarding Information Base (FIB)، Pending Interest Table (PIT) and Content Store (CS).

FIB is used to forward Interests toward the data source. PIT follows the forwarded Interests and keeps track of them so that the returned chunks can be transmitted to its requestor. CS is basically the router's buffer memory that caches the data packets.

3. Relate works

In this section, the proposed methods for congestion control in CCN are presented in two groups: end-to-end and hop-by-hop.

End-to-end congestion control: In this method the congestion control is performed by the receiver by controlling the transmission of interest packets. In [4] and [6] the Interest Control Protocol (ICP) and Information Centric Transport Protocol (ICTP) are introduced. In these protocols, the congestion window is placed in the receiver side to control the number of transmitted Interest packets. This window specifies the maximum number of Interest packets that can be exited from the receiver. In ICTP, the TCP same algorithm (including slow start, fast retransmission and fast recovery) is proposed in the receiver side in order to transmit the Interest packets which has the least adaptation with the CCN infrastructure. The congestion window also acts as Additive Increase Multiplicative Decrease (AIMD) in ICP. In these two protocols, only a single RTT estimator is used in the receiver to predict the network status. Due to the possibility of the multiple-path for each flow in CCN, a single RTT is not an appropriate method for this type of infrastructure.

The authors of [7] consider separate time out and the congestion window for each flow, but for all the sources/caches that are placed in a channel single time out is considered. In [8] and [9], separate RTT is calculated for each route in any of the receivers. Although using multiple RTT is appropriate for CCN infrastructure, as explained in [12], it adds much complexity to the receiver.

Hop-by-hop congestion control: Transmitting one data packet in response to each Interest creates a flow balance in CCN. This one-to-one flow balance can be used to control the congestion effectively. Substantially, the congestion caused by data packets on the downstream direction of the link is prevented by controlling the Interest packet transmission in reverse path [11].

PIT load can also be used to control the congestion in CCN. The number of present entries in PIT indicates the amount of loads on the routers. By limiting the numbers of the PIT entries in each router the need for congestion control by end nodes can be eliminated [5] [10]. This method is designed in order to shape the Interest packets transmission in the form of hop-by-hop in the routers. Because the amount of packets on the link is less than the link capacity in each link, then the congestion on the link is avoided.

Limiting the rate of Interest packets transmission is another mechanism for congestion control in strategy layer. Congestion caused by data packets can be controlled by pacing the rate of Interest packets transmission in the reverse path. In [11] each node forwards the Interest packets on the link with a lower rate of the allowed limit. Congestion on the link can be avoided by this method because the packets transmission rate on each link can be regulated and adjusted in proportion to the network total capacity.

4. NN based congestion control algorithm

Congestion would occur when the total amount of incoming packets to the network exceed the network capacity. During the period of congestion in the network, queues in the routers are filled. As a consequence the packets which are passing from these routers suffer many delays. If the congestion continues, the queue is overflowed. The packets which cannot fit in the queue are discarded by the router and must be transmitted again by the source [17] [18].

As a result, dropping packets by router indicates the queue overflow. This means the load on the link has exceeded the link capacity and will result in congestion in the network. In IP network, the packet dropping is used as a signal for congestion detection and avoidance [19]. Using the NN is one way of predicting the drop on the network [20] [21]. The authors in this paper, attempt to use the drop occurrence prediction in link as a signal to detect the congestion and to avoid it. The Multilayer Perceptron (MLP) NN is used to dynamically predict the occurrence of drop on each link in the routers. Therefore, each router in the network collects statistics information from the available traffic amount on each of the link connected to it. The pre-trained NN in each router is implemented as an intelligent agent that predicts the occurrence of the next drop on the outgoing link based on the gained information and trained data. The congestion control algorithm operates as follows:

1) Before transmitting the Interest packet on the outgoing interface, the implemented NN on the router predicts the drop occurrence in the next time for the link on interface based on the present status of the network in the strategy layer.
2) If the NN detects that the likelihood of dropping packets on the link is low, then the Interest is transmitted in accordance with previous strategy. If the router predicts a drop on the link (the load on the link is close to the link maximum capacity) then it uses another link to forward the Interest.
3) If the congestion is predicted for all the paths (available load on all the outgoing paths is close to its capacity) an alarm will be sent to the consumers to alert them to reduce the Interest transmission rate (the next Interest will be sent with a delay).

5. Neural network design

The designed NN in the routers is a feed forward, supervised two-layer NN. According to the proposed theory in [22] it has been proved that one hidden layer will be sufficient to estimate any continuous function. Therefore, a two-layer NN including one hidden layer and one output layer is designed using the NN toolbox from Matlab software.

To ensure the best results, selecting the appropriate parameters to set the NN is one of the most important parts in the NN implementation. Therefore, during the NN training, different settings and configurations to drop prediction in the next time are done. Input parameters and the number of hidden layer neurons were evaluated differently. The Mean Square Error (MSE) is the performance criterion which is used in this design to compare different models of NN, and also to evaluate the accuracy of designed NN. Beside the MSE the regression curve is also used to evaluate the NN performance. The NN is executed and repeated between 10 to 50 times for each value, so other parameters are kept consonant and the NN is evaluated several times by means of this parameter in order to achieve the best results.

The considered input parameters include parameters that represent the amount of load on the path. Additionally parameters which are related to the path topology based on the congestion definition and the parameters which influence the congestion [18][19]. Several different scenarios containing congestion were implemented in the simulator and were evaluated in order to obtain the correct input parameters.

By comparing the MSE graphs and the analysis of the regression curve in the different designs, NN with 8 input neurons, 10 hidden neurons with the sigmoid activation function and one neuron with a linear function in the output layer, have achieved the best prediction with the least error for the desired model.

It should be noted that the cache is one of the influential parameters in CCN, which in this simulation cache is considered a constant parameter, therefore it is not included in the input parameters of the NN. The number of senders and the Interest packet transmission rate of all of the consumers are considered constant in addition to the cache. Fig. 1 shows the designed NN.

The following are of the optimum and used input parameters:

1) Number of received Interest packets from the link (in bytes)
2) Number of sent Interest packets on the link (in bytes)
3) Number of received data packets from the link (in bytes)
4) Number of sent data packets on the link (in bytes)
5) Link bandwidth(in bytes)
6) Link delay(in seconds)
7) Number of packets in queue(in bytes)
8) Queue Size(queue capacity in bytes)

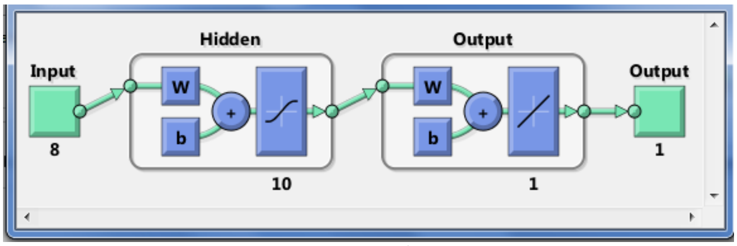

Fig. 1: Designed NN Diagram for Congestion Control

6. Neural network evaluation

To build an NN model, the dataset is divided into three sets (70% for training, 15% for validation and 15% for testing). Fig. 2 illustrates training errors, validation errors, and test errors for the designed NN. As shown in Fig. 2 the result is reasonable, because of the following considerations:

- The final mean-square error is 0.00004 that is so small and prove that NN works accurately.
- The test and validation set errors have similar characteristics. If the test curve increases significantly before the validation curve, then some over fitting might have occurred [23]. Therefore, No over fitting has occurred by iteration 703 (where the best validation performance occurs).

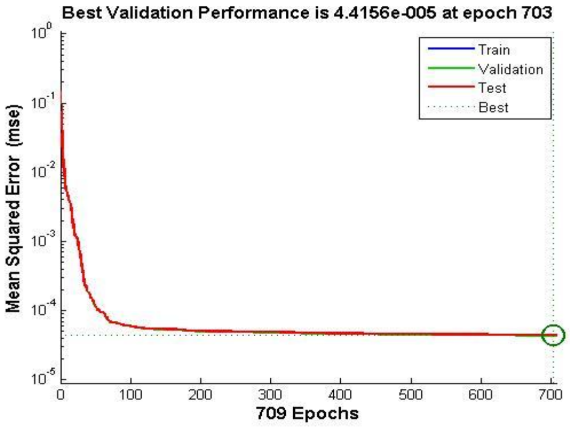

Fig. 2: NN Performance Curve Based On MSE for Training, Test and Validation Set.

Analyzing the regression is one of the parameters to evaluate the performance of the NN, which shows the relationship between the predicted and the target values. The 45 degree sloped line shows that the predicted value in the NN (NN output) is exactly equal to the target values which have been given to the NN. R=1 indicates that there is an exact linear relationship between output values and the target values. If R value is close to zero, it means that there is no linear relationship between the NN output and the target value, and also the NN is not working correctly [23]. Fig.3 demonstrates the regression curves for the training, testing and validation set. This figure shows that a very precise and exact prediction is done in this network (R>0/995 that is the very close value to one).

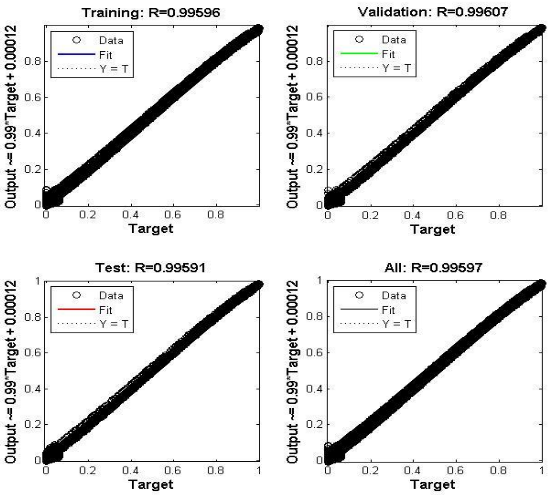

Fig. 3: NN Performance Curve Based on Regression for Training, Test and Validation Set.

7. Simulation results and analysis

The proposed algorithm is implemented in the ndnSIM [24] simulator. In this paper the scenario has been implemented by using the GEANT [25] topology. GEANT is the core network containing 22 routers, which have connected several research centers and universities together in Europe. This topology has been used in various assessments for the NDN network, including [26] - [30]. There are 20 consumers and one content producer in the network. Separated nodes are considered for the consumers, which are connected to the routers directly. The producer is considered one of the 22 core routers in the GEANT topology. Therefore the total number of the network nodes is equal to 42. Table 1 shows the summary of the implemented scenario.

Table 1: Network Topology

Parameter	Value
Topology	GEANT with 22 router
Link capacity	1, 100 Mbps
Number of consumers	20
Number of producer	1
Chunk size	1400 Bytes
Interest Size	32 Byte
Simulation time	30 S

NN based congestion control algorithm predicts packet drop in order to control the router queue size, and thus prevents the occurrence of queue overflow. Fig. 4 compares the evolution of the queue length over time between the CCN routers with and without NN based congestion control. As Fig. 4 shows, CCN has lower queues size under congestion control algorithm. Each queue has the capacity of 20 packets. As shown in Fig. 4, queue size under congestion control doesn't exceed 18 packets (in core 18). Therefore, it can effectively prevent queue overflow. With NN based congestion control mean queue size in network reduced by 93.91% in the network.

Fig. 5 shows packet drop rate in the network routers over the time with and without congestion control in CCN. As can be seen from Fig. 5, NN based congestion control effectively controls congestion so that the packet drop rate in the network (due to queue overflow) is reduced. Mean of drop in all network routers over the simulation time is 569.5764 Kbits/s for CCN without congestion control that reduced by 82.417 Kbits/s under congestion control. Therefore by reducing packet drop, the throughput of network improves by 85.53%.

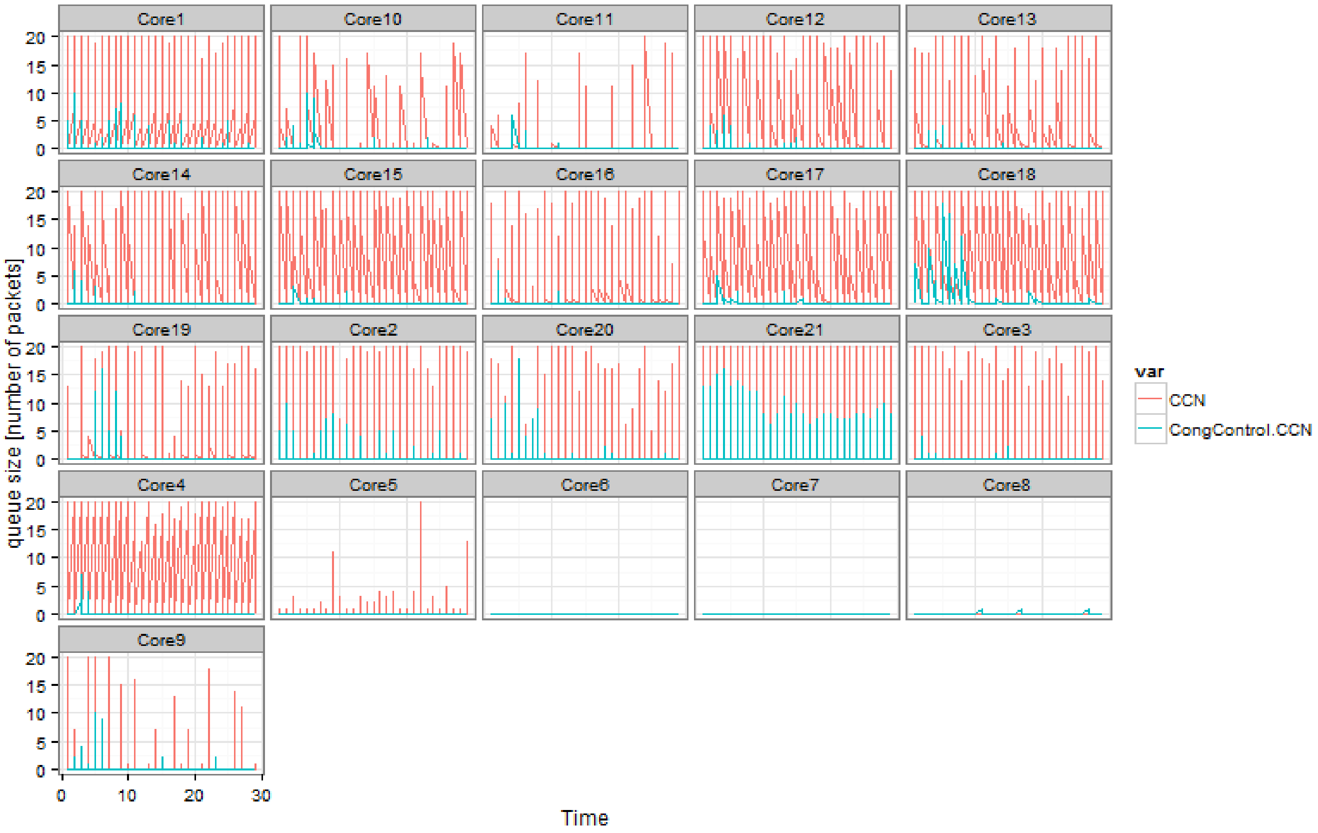

Fig. 4: Comparison of Queue Size Between CCN with and Without NN Based Congestion Control in 21 Routers in Network

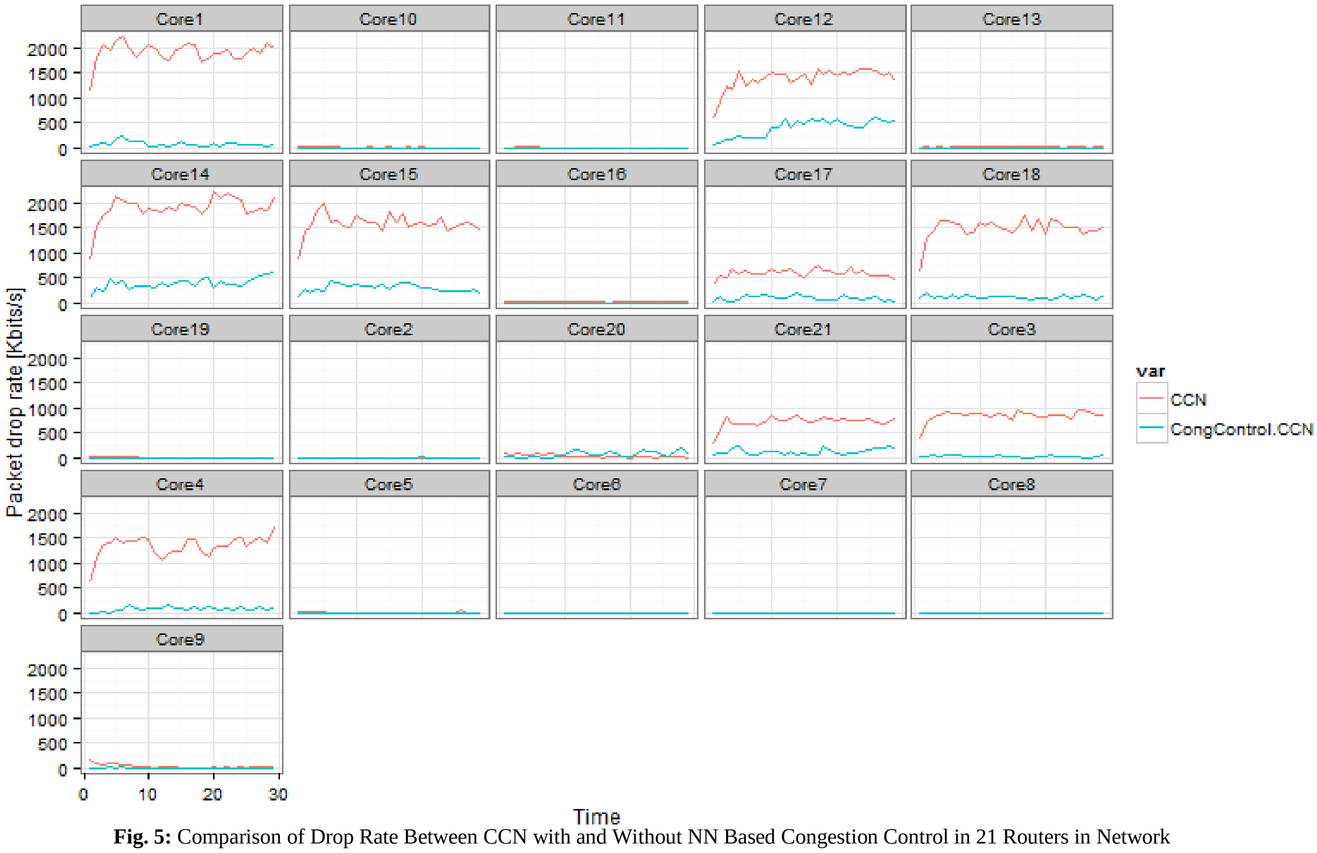

Fig. 5: Comparison of Drop Rate Between CCN with and Without NN Based Congestion Control in 21 Routers in Network

8. Conclusion

The presence of multiple sources for each of the content makes CCN congestion control complex and challengeable. In this paper, a new congestion control algorithm in CCN has been proposed based on NN. The designed NN predicts the existence of the congestion on link based on the current network status adaptively. The NN has been implemented in each router and acts in the strategy layer. Before each Interest is transmitted on the router’s outgoing interface, the congestion presence on the link is predicted and prevents it from happening. The performance of NN based congestion control has been evaluated with GEANT topology using ndnSIM simulator. The results show that the proposed congestion control algorithm can effectively control congestion and improve network throughput by 85.53% in the simulated network.

References

[1] *V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in Proceedings of the 5th international conference on Emerging networking experiments and technologies, 2009, pp. 1-12. http://dx.doi.org/10.1145/1658939.1658941.*

[2] *Named data networking project (NDN),”* Available: http://named-data.org.

[3] M. Särelä, T. Rinta-aho, and S. Tarkoma, "RTFM: Publish/subscribe internetworking architecture," in *ICT-MobileSummit Conference Proceedings*, 2008.

[4] G. Carofiglio, M. Gallo, and L. Muscariello, "ICP: Design and evaluation of an interest control protocol for content-centric networking," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, 2012, pp. 304-309.

[5] G. Carofiglio, M. Gallo, and L. Muscariello, "Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks," in Proceedings of the second edition of the ICN workshop on Information-centric networking, 2012, pp. 37-42. http://dx.doi.org/10.1145/2342488.2342497.

[6] S. Salsano, A. Detti, M. Cancellieri, M. Pomposini, and N. Blefari-Melazzi, "Transport-layer issues in information centric networks," in Proceedings of the second edition of the ICN workshop on Information-centric networking, 2012, pp. 19-24. http://dx.doi.org/10.1145/2342488.2342493.

[7] S. Arianfar, P. Nikander, L. Eggert, and J. Ott, "Contug: A receiver-driven transport protocol for content-centric networks", in *IEEE ICNP 2010 (Poster session).*

[8] Y. Wang, T. Muto, Z. Su, J. Katto, and S. Awiphan, "A Consideration on Congestion Control for CCN," *Packet Video Workshop, Poster Session*, 2013.

[9] G. Carofiglio, M. Gallo, L. Muscariello, and M. Papalini, "Multipath congestion control in content-centric networks," *IEEE NOMEN*, vol. 13, 2013.

[10] S. Oueslati, J. Roberts, and N. Sbihi, "Flow-aware traffic control for a content-centric network," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 2417-2425.

[11] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane," Computer Communications, vol. 36, pp. 779-791, 2013. http://dx.doi.org/10.1016/j.comcom.2013.01.005.

[12] F. Zhang, Y. Zhang, A. Reznik, H. Liu, C. Qian, and C. Xu, "A Transport Protocol for Content-Centric Networking with Explicit Congestion Control," in *Proceedings of The The 23rd International Conference on Computer Communications and Networks (ICCCN)*, 2014.

[13] M. Barabas, G. Boanea, and V. Dobrota, "Multipath Routing Management using Neural Networks-Based Traffic Prediction," *in The Third International Conference on Emerging Network Intelligence,* 2011.

[14] M. Rouhani, M. R. Tanhatalab, and A. Shokohi-Rostami, "Nonlinear Neural Network Congestion Control Based on Genetic Algorithm for TCP/IP Networks," in *Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on*, 2010, pp. 1-6.

[15] H. C. Cho, M. S. Fadali, and H. Lee, "Neural network control for TCP network congestion," in *American Control Conference, 2005. Proceedings of the 2005*, 2005, pp. 3480-3485.

[16] G. A. Rovithakis and C. N. Houmkozlis, "A Neural Network Congestion Control Algorithm for the Internet," in Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation, 2005, pp. 450-455.

[17] J. F. Kurose, Computer networking: a top-down approach featuring the Internet: Pearson Education India, 2005, pp. 271-293.

[18] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN systems,* vol. 17, pp. 1-14, 1989. http://dx.doi.org/10.1016/0169-7552 (89)90019-6.

[19] V. Jacobson, "Congestion avoidance and control," in *ACM SIGCOMM Computer Communication Review*, 1988, pp. 314-329. http://dx.doi.org/10.1145/52325.52356.

[20] H. R. Mehrvar and M. R. Soleymani, "Packet loss rate prediction using a universal indicator of traffic," *in Proceedings of IEEE International Conference on Communications*, vol. 3, 2001, pp. 647–653.

[21] Y. Barve, "Neural Network Approach to the Prediction of Percentage Data Packet Loss for Wireless Sensor Networks," in *Proceedings of Southeastern Symposium on System Theory*, 2009, pp.150-143 .

[22] V. Kůrková, "Kolmogorov's theorem and multilayer neural networks," *Neural networks,* vol. 5, pp. 501-506, 1992. http://dx.doi.org/10.1016/0893-6080 (92)90012-8.

[23] H. Demuth, M. Beale, and M. Hagan, "Neural network toolbox user's guide, The MathWorks," *Inc., Natrick, USA,* 2009.

[24] Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," *Tech.Rep, NDN-0005,* 2012.

[25] *Geant project website.* Available: http://www.geant.

[26] M. Tortelli, L. A. Grieco, and G. Boggia, "Performance Assessment of Routing Strategies in Named Data Networking," presented at the IEEE ICNP, 2013.

[27] D. Rossi and G. Rossini, "On sizing CCN content stores by exploiting topological information," in *INFOCOM Workshops*, 2012, pp. 280-285.

[28] V. Ciancaglini, G. Piro, R. Loti, L. A. Grieco, and L. Liquori, "CCN-TV: a data-centric approach to real-time video services," in *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, 2013, pp. 982-989.

[29] R. Chiocchetti, D. Rossi, G. Rossini, G. Carofiglio, and D. Perino, "Exploit the known or explore the unknown? Hamlet-like doubts in icn," in *Proceedings of the second edition of the ICN workshop on Information-centric networking,* 2012, pp. 7-12. http://dx.doi.org/10.1145/2342488.2342491.

[30] D. Rossi and G. Rossini, "Caching performance of content centric networks under multi-path routing (and more)," *Relatóriotécnico, Telecom ParisTech,* 2011.