

A NEURAL NETWORK BASED TRAFFIC-AWARE FORWARDING STRATEGY IN NAMED DATA NETWORKING

PARISA BAZMI AND MANIJEH KESHTGARI

*Department of Computer and Information Technology,
Shiraz University of Technology, Shiraz, Iran.*

bazmi.parisa@gmail.com; keshtgari@sutech.ac.ir

(Received: 29th Jan. 2016; Accepted: 3rd May. 2016; Published on-line: 30th Nov. 2016)

ABSTRACT: Named Data Networking (NDN) is a new Internet architecture which has been proposed to eliminate TCP/IP Internet architecture restrictions. This architecture is abstracting away the notions of both hosting and operation based on naming datagrams. However, one of the major challenges of NDN is supporting a QoS-aware forwarding strategy so as to forward Interest packets intelligently over multiple paths based on the current network condition. In this paper, Neural Network (NN) based Traffic-aware Forwarding strategy (NNTF) is introduced in order to determine an optimal path for Interest forwarding. NN is embedded in NDN routers to dynamically select the next hop based on the path overload probability achieved from the NN. This solution is characterized by load balancing and QoS-awareness in terms of delay and packet drop via monitoring the available path and forwarding data on the traffic-aware shortest path. The performance of NNTF is evaluated using ndnSIM which is a NS-3 module that implements the NDN communication model. The simulation results show the efficiency of this scheme in terms of network QoS improvement of 17.5% and 72% reduction in network delay and packet drop respectively.

ABSTRAK: Data Rangkaian Bernama (NDN) adalah seni bina Internet baru yang telah dicadangkan untuk menghapuskan sekatan seni bina Internet TCP / IP. Seni bina ini mengabstrakkan konsep hos dan dijalankan berdasarkan menamakan datagram. Bagaimanapun salah satu cabaran utama NDN adalah menyokong strategi penghantar sedar QoS supaya paket Interest dikirim dengan bijak melalui beberapa laluan berdasarkan kepada keadaan rangkaian semasa. Dalam kertas ini, Neural Network (NN) berdasarkan strategi penghantaran Traffic-Aware (NTTF) diperkenalkan untuk menentukan laluan yang optimum untuk penghantar Interest. NN tertanam dalam router NDN untuk memilih hop seterusnya secara dinamik berdasarkan kebarangkalian jalan beban dicapai dari NN. Penyelesaian ini mempunyai ciri-ciri pengimbangan beban dan kesedaran QoS dari segi kelewatan dan paket drop melalui pemantauan jalan yang ada dan penghantaran data di atas jalan Traffic-Aware yang tersingkat. Prestasi NTTF dinilai dengan menggunakan ndnSIM yang merupakan modul NS-3 yang melaksanakan model komunikasi NDN. Keputusan simulasi menunjukkan kecekapan skim ini dari segi peningkatan rangkaian QoS adalah sebanyak 17.5% dan pengurangan kelewatan rangkaian dan drop paket adalah sebanyak 72%.

KEYWORDS: *named data networking; content-centric networking; forwarding strategy; neural network*

1. INTRODUCTION

Current Internet architecture is based on a TCP/IP solution and has focused on point-to-point communication between two entities. Today, the content from services dominates the marketplace and Users care about content, while the Internet is still working based on where the content is [1]. Therefore, it is unable to serve these new applications and the increasing needs of companies and individuals [2]. A new architecture has been proposed originally in [1] to remove a TCP/IP Internet architecture restriction which is the use of the naming datagram instead of IP addresses. Content-Centric Networking (CCN), also called Named Data Networking (NDN), is the latest proposed Internet architecture based on naming datagrams. This architecture is abstracting away the notion of hosting [1].

In NDN, communication begins with the consumer. First, the consumer requests its desired content by sending an Interest packet which contains its desired content name. The requested data is then transmitted to the consumer in the reverse path from Interest packet receiver. The NDN router has three components: Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). When an Interest packet is received on the router's interface, the CS (buffer memory which caches the data packets) is checked to find whether the received Interest name is matched with any CS entry. If the name is matched, the cached data packet is sent back on the incoming interface. Otherwise, it checks the PIT table which keeps track of forwarded Interest packets to find whether the desired Interest has already been sent or not. If the Interest has been sent, the interface of the router which Interest packet was received through it is added to PIT to transmit data packets in the reverse path. If the Interest name is not matched with any PIT or CS entry, the longest prefix match is done for Interest packet routing in the FIB table to decide which router's interface to send an Interest out to [1].

NDN's loop-free data forwarding permits NDN routers to select its outgoing interface or interfaces adaptively from multiple choices. However, supporting intelligent forwarding strategy is one of the major challenges of NDN implementation [1, 3, 4].

On the other hand, NDN offers obvious advantages over TCP/IP, including adaptive forwarding provided by the state maintained in the router. Since, in NDN, the router PIT keeps track of Interest packets and receiving data packets can be observed for each Interest, routers can adaptively measure throughput, delivery performance, and packet loss. This NDN characteristic gives NDN routers an opportunity to forward packets intelligently and adaptively considering the current network status [5].

In this paper, this beneficial behavior of NDN is taken into account to design a new intelligent load-balancing forwarding strategy which is called Neural Network based Traffic-aware Forwarding strategy (NNTF) for NDN. The Neural Network (NN) is designed to select the next hop dynamically, based on the current network status achieved from the feedback of data and Interest packets. This solution is enacted by load balancing and QoS-awareness via monitoring -not only available path resources but also the Interest and data packets.

The remainder of this paper is organized as follows. Section 2 describes the related work on the NDN forwarding strategy. A detailed description of NNFT is given in Section 3. Section 4 presents the performance evaluation of the proposed algorithm, and Section 5 concludes the paper.

2. RELATED WORKS

The original NDN forwarding strategy is flooding in which Interest packets are broadcast on all possible router outgoing interfaces [1]. Since the data packet is forwarded via the reverse path, the data packet, which was received sooner, is considered and the other received data packets from other interfaces are discarded. This strategy makes use of the fastest path, and as a result the path with the lowest delay due to the existence of a wide range of data and Interest packets in the network, the performance of the network in terms of network delay, bandwidth consumption, and energy efficiency is reduced dramatically.

The other possible strategy which can be used in NDN is to unicast Interest packets to the best outgoing interface. However, this strategy is challengeable in finding the best outgoing interface dynamically and it is resilient against network failure. Some probabilistic forwarding strategies are proposed to find the best next hop to transmit the Interest packet [3, 4, 7].

The initial design of the NDN forwarding strategy is described in [5, 7]. Using PIT and two-way Interest and data exchange gives the NDN forwarding strategy the opportunity to immediately detect network problems and decide on the outgoing interface using the current network condition. The Interest NACK packet is introduced in these papers to notify downstream nodes about a problem in a specific path. Interface ranking is described in this paper based on the forwarding policy. Possible outgoing interfaces for each name in FIB are ranked based on the network policy and then an Interest is sent to the interface with the highest rank.

Probability-based Adaptive Forwarding (PAF) strategy is based on the ant colony algorithm optimization is presented in [3]. An Ant colony is used to calculate the outgoing interface selection probability. It uses the feedback from data and Interest packets to probe the performance of each interface in terms of delay. As a result, the selection probability is computed adaptively based on the network condition and optimizing a performance metric (delay). The load is balanced between interfaces of the router.

A Greedy Ant Colony Forwarding (GACF) algorithm is proposed in [4] as an NDN forwarding strategy. It is a QoS-aware forwarding strategy which uses the Ant Colony algorithm to optimize the possible paths and select the next hop probabilistically based on the current network state.

Using the Ant Colony algorithm in PAF and GACF adds some complexity to NDN as distinguished Interest packets are needed for probing only. In this paper, the best next-hop is investigated by implementing a preconfigured Neural Network (NN) in each router to decide which path should be selected to send the Interest based on the current network condition. The preconfigured NN in the routers is very simple and does not need a separate Interest packet for probing like Ant Colony does.

3. NEURAL-NETWORK BASED TRAFFIC-AWARE FORWARDING STRATEGY

In this section, the use of NN is described to determine an optimal next-hop selection in NDN forwarding strategy. The forecasting algorithm using the neural network is embedded in each router to predict the load on its outgoing interfaces. In the NDN strategy layer, the router transmits Interest packets on a traffic-aware shortest path. Using a

predicting algorithm in NDN, forwarding strategy can play a pivotal role in guaranteeing QoS and minimizing expected delay and loss in NDN.

3.1 Neural Network Design

The Usage of NN as a tool for traffic prediction has been prevalent in the computer network due to its ability to learn complicated patterns [9-12]. NN can be used for next-hop selection in NDN forwarding strategy [6]. The summary of the designed NN and its design motivations are described in this section.

In the designed NN network, packet drop is considered a warning that the load on the link is more than that link capacity. In other words, dropping packets occurs when the link is overloaded and the router's queue has overflowed [13]. On the other hand, NN is a great way of packet drop prediction as it was used widely in TCP/IP networking [14, 15].

The designed NN predicts the probability of drop occurrence which is considered a link overload probability. The pre-trained multilayer perception (MLP) NN is implemented in each router and works in the router's strategy layer to dynamically predict the overload probability based on the received feedback from that link and its training data. Figure 1 illustrates the designed NN. As can be seen in Fig. 1, a feed-forward, supervised two-layered neural network is designed using Matlab software's NN toolbox to predict overload probability on each link. A two-layer NN including one hidden layer and one output layer is selected based on the proposed theory in [16] that proved that one hidden-layer is enough to estimate any continuous function.

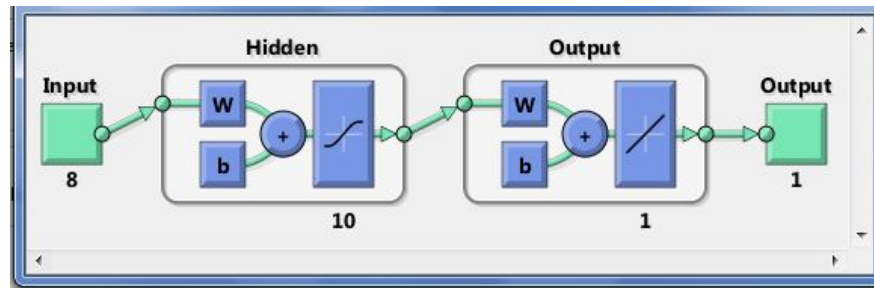


Fig. 1: Designed NN diagram for the forwarding strategy.

To reach the best NN design, different input parameters and different numbers of hidden layer neurons were evaluated. The Mean Square Error (MSE) is used to compare different models of NN implementation. The NN is run and repeated between 10 to 50 times for each value of a single parameter, while other parameters are kept constant and the NN is examined several times by means of this parameter so as to select the best NN design.

As was discussed in [6], the mentioned input parameters include the factors which have an impact on the amount of load on the path in addition to the topology based parameters that influence congestion occurrence and link overload. Different scenarios were implemented to obtain the best collection of input parameters. After few trials, between 10 to 50 times for each NN parameter, the neural network with 8 input neurons and 10 hidden layers with the sigmoid activation function and one neuron with a linear function in the output layer is selected to predict overload probability based on the analysis of MSE and regression curve in the different designs. The 8 NN inputs comprise network topology input parameters and load parameters for each link as follows [6]:

1. Number of incoming Interest packets (Byte)
2. Number of outgoing Interest packets (Byte)
3. Number of incoming data packets from the link (Byte)
4. Number of outgoing data packets on the link (Byte)
5. Link bandwidth
6. Link delay (s)
7. Current queue size (Byte)
8. Queue capacity (Byte)

The data from different network topologies with different loads is used as a training data in the designed NN. The NN dataset is divided into three sets including 70% for training, 15% for validation, and 15% for testing data, in order to evaluate the accuracy of the designed NN. Figure 2 shows the errors for each set of data. The final MSE (0.00004) is too small, which illustrates the correctness of NN design. Additionally, there is no evidence of over-fitting by iteration 703. Therefore, the NN evaluation results show the correct implementation of NN in addition to the accurate dependency between input and output parameters. Table 1 gives the summary of the implemented NN algorithm.

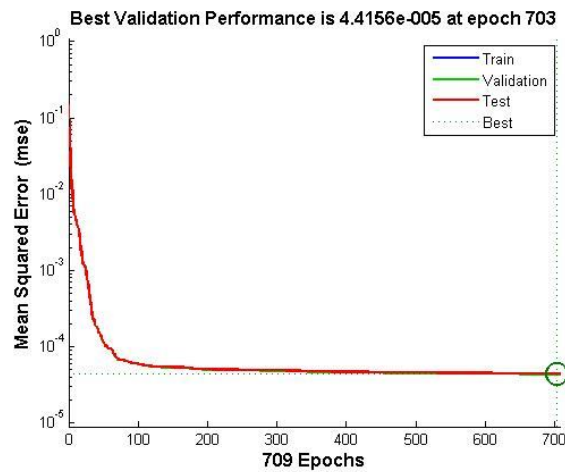


Fig. 2: NN performance curve based on MSE for training, test, and validation set.

Table 1: Summary of the implemented NN algorithm

Parameter	Value
number of records for NN training	2465625
dataset division (training, validation, testing)	random
learning algorithm	scaled conjugate gradient
performance criterion	MSE

3.2 Interface Ranking

As described in [7], the available outgoing interfaces for each name in the FIB entries, which is obtained from routing policy, are ranked to make forwarding strategy more convenient. This ranking is based on a forwarding policy that NDN considered for sending Interest packets. In this paper, the traffic-aware shortest path in terms of delay is

used as a next-hop for sending Interest packets. As a result, the path with the shortest propagation delay is assigned the highest interface rank.

3.3 Forwarding Strategy Algorithm

When a new Interest is received in the router and it does not match any entry in the Content Store and PIT, a new PIT entry will be created. The new Interest is then forwarded to the highest-ranked interface if NN overload prediction for this face is lower than 50%. Otherwise, the next highest-ranked interface, whose overload prediction is less than 50%, is selected as a next-hop. In the situation where there is no available interface with an overload probability below 50%, the face with the smallest overload probability (lowest load) is opted for to forward the Interest packet. The overload probability threshold (50%) was obtained and optimized from doing various simulations. The figure in which the performance of the network was higher is considered as a threshold to change the current working path. Figure 3 illustrates NNTF algorithm. In this algorithm, NeuralNetwork_output (interface) is a method that implements trained NN for overload prediction of specified interface in the router. NNTF is implemented in all routers' strategy layer in order to select next-hop for sending Interest.

```

Function Forward (Interest, PitEntry)

  if FibEntry ← FIB.Find (Interest.Name ) then
    for each interface in FibEntry by rank do
      if interface ∉ PitEntry.Outgoing then
        if NeuralNetwork_output(interface) < 0.5 then
          Set PitEntry.RetryTimer
          Transmit(interface, Interest)
          Add interface to PitEntry.Outgoing
          Is_sent ← True
          Stop processing
        end if
      end if
    end for
    if Is_sent=false
      interface← Min(NeuralNetwork_output(interface))
      Transmit(interface, Interest)
    end if
  end if
end function

```

Fig. 3: Forwarding strategy algorithm.

NNTF works based on the information provided by the NN regarding the current network status. By fast rerouting, this scheme maintains load balancing and congestion avoidance. As a result, the overall performance of the network, including delay and packet loss, is improved. One of the other major features of this scheme is that it avoids instability by avoiding frequent vacillation of paths, unless it is predicted that the path is nearly overloaded.

4. EVALUATION OF PROPOSED ALGORITHM

Due to the fast rerouting and load balancing based on NN output, the proposed forwarding strategy would improve the QoS parameters and network performance. To evaluate the proposed forwarding strategy, a scenario is simulated in an open-source ndnSIM [17] package, which implements the NDN protocol stack for the NS-3 simulator. The GEANT real topology [18] with 22 routers, 1 provider, and 20 consumers is implemented in ndnSIM. GEANT is a real core network in Europe. This topology has been used in various evaluations of the NDN network [6, 19-21]. Table 2 gives information on the implemented scenario.

Table 2: Network scenario

Parameter	Value
Topology	GEANT with 22 router
Link capacity	1, 10 Mbps
Number of consumers	20
Number of producers	1
Chunk size	1400 Bytes
Interest Size	32 Byte
Cache size	0
Consumer Requested file size	12600 Kbyte
Simulation time	100 S

YouTube traffic [22] is considered in this paper to evaluate the video QoS improvement by implementing the proposed NN. Video traffic with standard quality of 360 p, 1000 Kbit/s transmission rate and 30 frames per second is simulated for this traffic. Delay and packet drop are two common QoS parameters in multimedia for evaluating network efficiency which are used in this paper. Two scenarios, including the shortest path forwarding strategy and traffic-aware shortest path with NN implementation, are simulated to evaluate the efficiency of NNTF.

Table 3 shows the delay comparison in 22 consumers between these two scenarios. The calculated delay is the time between transmitting an Interest packet and receiving the desired data [23, 24] which consists of queuing, propagation, and transmission delay. By fast rerouting, based on the path overload probability, traffic-aware forwarding strategy forwards Interest packets on the shortest and at the same time, not crowded path. As can be seen in Table 3, the total network delay is reduced by 9%. It is obvious that under larger traffic load, this algorithm would act more effectively. To show this, some background traffic is added to the network to increase the probability of path overload. Table 4 gives the results of this simulation. As shown in Table 4, total network delay with NN prediction under larger traffic reduced by about 17.5 percent which is evidence of the efficiency of proposed algorithm. Therefore, NNTF works appropriately under heavy traffic, and it is an efficient forwarding strategy under heavy network load.

To evaluate the network performance, the comparison of packet drops in the core routers of the network is calculated. Figure 4 shows the packet drop rate in 10 routers of the network for both shortest path and NNTF under heavy traffic. The details of the packet drop in all routers are presented in Table 5. As can be seen in Table 5, NNTF reduces packet drop in network routers by 72.11% due to the fast rerouting that results in congestion avoidance and queue overflow prevention.

Table 3: Comparison of the delay (s) in network consumer between the Normal Shortest Path algorithm and NNTF

Consumers	Mean Delay in 100 S (S)	
	Shortest Path	NNTF
C1	27.9489	26.5963
C2	4.53689	4.091
C3	28.6975	26.471
C4	9.68352	7.67606
C5	12.7325	11.9117
C6	28.5499	25.1191
C7	26.8557	21.0264
C8	29.6673	27.28
C9	28.7097	28.706
C10	28.2589	28.289
C11	6.86975	6.534
C12	24.2168	22.768
C13	26.4765	23.8981
C14	4.85138	4.813
C15	27.4695	23.8017
C16	14.6193	14.599
C17	23.038	20.8679
C18	26.4403	22.8733
C19	27.069	26.5185
C20	16.8177	11.5951
Overall Network's Mean Delay in 100 S	21.17545	19.26975
Improvement percentage	8.996%	

Table 4: Comparison of delay(s) in network consumer between the Normal Shortest Path algorithm and NNTF under heavy traffic

Consumers	Mean Delay in 100 S (S)	
	Shortest Path	NNTF
C1	28.70056	23.507802
C2	25.18629	6.042782
C3	29.58026	27.94695
C4	29.29922	27.43917
C5	30.27463	25.54593
C6	26.6662	27.02151
C7	22.54109	12.92972
C8	27.57811	27.93101
C9	30.16761	25.57313
C10	28.80371	27.93101
C11	28.99133	22.70501
C12	28.7824	24.00278
C13	26.6225	24.5023
C14	29.44756	22.43217
C15	27.15052	21.4878
C16	25.18447	21.59808
C17	26.47577	23.8936
C18	28.94153	23.27276
C19	29.50166	24.47638
C20	19.98871	14.81094
Overall Network's Mean Delay in 100 S	27.49429	22.6872
Improvement percentage	17.4835%	

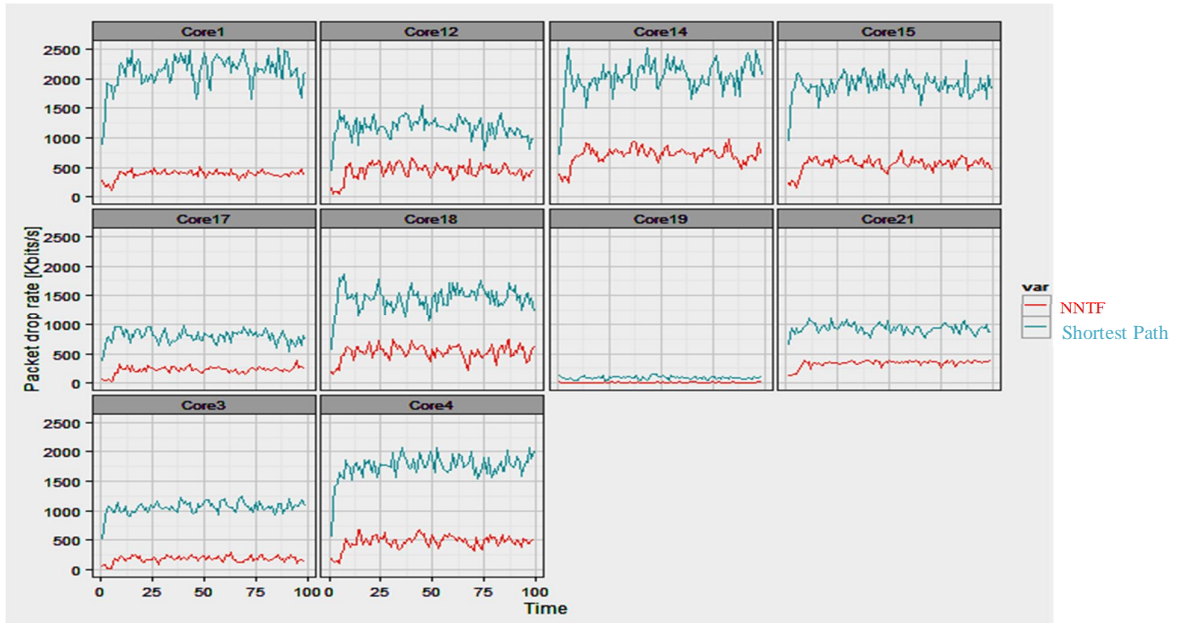


Fig. 4: The Comparison between the Shortest Path and NNTF of the packet drop rate of ten routers, in Kbit.

Table 5: Comparison of packet drop in the routers between the Shortest Path and NNTF, in Kbit.

Network Routers	Mean Packet Drop in 100 S (Kbit)	
	Shortest Path	NNTF
Core1	2129.455	384.97
Core2	35.9596	0.16162
Core3	1059.879	181.98
Core4	1752.646	473.535
Core5	45.1772	4.84849
Core6	0	0
Core7	0	0
Core8	0	0
Core9	173.8182	22.6263
Core10	51.9596	13.0101
Core11	87.91919	5.73737
Core12	1171.071	440.081
Core13	110.7879	10.5051
Core14	2034.909	710.141
Core15	1884.606	558.222
Core16	92.92929	12.0404
Core17	794.3434	224.647
Core18	1452.364	519.758
Core19	90.50505	6.62626
Core20	204.9293	26.5859
Core21	927.8384	337.212
Overall Network's Mean Packet Drop (Kbit)	671.4806	187.271
Improvement percentage	72.11%	

5. CONCLUSION

NDN has been introduced as a new Internet architecture in order to remove current Internet architecture restrictions by using naming data rather than the host IP address. One of the main challenges of NDN is to provide an intelligent forwarding strategy in routers, as is addressed in this paper. Here, NNTF has proposed as a traffic-aware shortest path forwarding strategy that works based on NN prediction. A NN is designed and embedded in each NDN router to predict overload probability of each path, which is then used in the path selection for Interest forwarding. The performance of NNTF is evaluated by simulation using ndnSIM, showing the efficiency of this scheme under heavy traffic by reducing 17.5% and 72% of network delay and packet drop respectively.

REFERENCES

- [1] Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL. (2009) Networking named content. ACM Proc. 5th Inter. Conf. Emerging Networking Expts. Tech., 1-12.
- [2] Future Content Networks Group. Why do we need a Content-Centric Future Internet? Proposals towards Content-Centric Internet Architectures, European Commission, Information Society and Media. Available at: <http://www.future-Internet.eu/>. Retrieved July, 2014.
- [3] Qian H, Ravindran R, Wang GQ, Medhi D. (2013) Probability-based adaptive forwarding strategy in named data networking. IFIP/IEEE International Symposium on Integrated Network Management, 1094-1101.
- [4] Chengming LI, Wenjing LIU, Okamura K. (2013) A greedy ant colony forwarding algorithm for Named Data Networking. Proc. Asia-Pacific Advanced Network, 34:17-26.
- [5] Yi C, Afanasyev A, Wang L, Zhang B, Zhang L. (2012) Adaptive forwarding in named data networking. ACM SIGCOMM Comp. Comm. Rev., 42(3):62-67.
- [6] Bazmi P, Keshtgari M. (2014) A neural network based congestion control algorithm for content-centric networks. J. Adv. Comp. Sci. Tech., 3(2):214-220.
- [7] Yi C, Afanasyev A, Moiseenko I, Wang L, Zhang B, Zhang L. (2013) A case for stateful forwarding plane. Comp. Comm., 36(7):779-791.
- [8] Floyd RW. (1962) Algorithm 97: Shortest path. Communications of the ACM, 5(6):345.
- [9] Barabas M, Boanea G, Dobrota V. (2011) Multipath routing management using neural networks-based traffic prediction. Third Int. Conf. Emerging Network Intelligence, 118-124.
- [10] Rouhani M, Tanhatalab MR, Shokohi-Rostami A. (2010) Nonlinear neural network congestion control based on genetic algorithm for TCP/IP networks. IEEE Second Int. Conf. Comp. Intel. Comm. Syst. Netw. (CICSyN), 1-6.
- [11] Cho HC, Fadali MS, Lee H. (2005) Neural network control for TCP network congestion. IEEE Proc. American Control Conference, 3480-3485.
- [12] Rovithakis G, Houmkozis CN. (2005) A neural network congestion control algorithm for the internet. IEEE Proc. Inter. Symp. Intel. Control, Mediterranean Conference on Control and Automation, 450-455.
- [13] Jacobson V. (1988) Congestion avoidance and control. ACM SIGCOMM Comp. Comm. Rev., 18(4):314-329.
- [14] Mehrvar HR, Soleymani MR. (2001) Packet loss rate prediction using a universal indicator of traffic. IEEE Inter. Conf. Communications, 3:647-653.
- [15] Barve YD. (2009) Neural network approach to the prediction of percentage data packet loss for wireless sensor networks. IEEE 41st Southeastern Symposium on System Theory, 103-106.
- [16] Kurkova V. (1992) Kolmogorov's theorem and multilayer neural networks. Neural networks, 5:501-506.
- [17] Afanasyev A, Moiseenko I, Zhang L. (2012) ndnSIM: NDN simulator for NS-3. University of California, Los Angeles, Tech. Rep.

- [18] Geant project website. Available: <http://www.geant.net>. Retrieved July, 2014.
- [19] Tortelli M, Grieco LA, Boggia G. (2013) Performance assessment of routing strategies in Named Data Networking. IEEE ICNP, 2013.
- [20] Rossi D, Rossini G. (2012) On sizing CCN content stores by exploiting topological information. INFOCOM Workshops, 280-285.
- [21] Ciancaglini V, Piro G, Loti R, Grieco LA, Liquori L. (2013) CCN-TV: a data-centric approach to real-time video services. 27th Inter. Conf. Advanced Information Networking and Applications Workshops (WAINA), 982-989.
- [22] Youtube Sample Rate: <https://support.google.com/youtube/answer/1722171?hl=en>. Retrieved July, 2014
- [23] Almishari M, Gasti P, Nathan N, Tsudik G. (2013) Optimizing bi-directional low-latency communication in named data networking. ACM SIGCOMM Comp. Comm. Rev., 44(1):13-19.
- [24] Jiang X, Bi J. (2013) Interest set mechanism to improve the transport of named data networking. ACM SIGCOMM Comp. Comm. Rev., 43(4):515-516.