

The Pennsylvania State University



Computer Networks
CSE 514

Application of Named Data Networking Project Report

Project by:

*Tingyao Hsu, Wenliang Sun, Yanbo Sun
Dec. 5, 2017

1. Introduction

Nowadays, many applications are provided on the Internet, and various kinds of transmitted messages are also transmitted to every corner of the world via the Internet. The transmission technologies and services for these messages need to be continuously researched and innovated. Many Internet services such as HTML (Hypertext Markup Language), Voice over IP (VoIP), Web 2.0, and numerous instant messaging services all need to rely on the Internet to make these technologies widely available and to benefit people's lives. With the development of internet, there are endless problems with the Internet itself. For example, when a user queries or shares data, he or she is accustomed to using the query name of the data to describe it instead of a series of meaningless numbers such as an IPv4 or IPv6 address. In order to solve the above problem, some people have proposed Domain Name System (DNS) service. However, the traditional centralized architecture can easily cause single-point errors and system overload problems. Therefore, when popular information needs to be repeatedly and quickly delivered to each user, the technology of the content core network [1] (CCN) can fully satisfy the requirements of the Internet development.

Named-data networking (NDN) is based on the CNN architecture. NDN is one of the improved technologies that make the Internet better. The future network traffic will be much faster than the speed of network construction [2], and the NDN architecture that can address the increased network traffic problem. In the NDN architecture, one of the methods is to add a cache mechanism to achieve more efficient bandwidth usage. In addition, in order to ensure the security and reliability of data contents, it has been introduced the cryptographic digital signatures and encryption mechanisms [3]. The above several characteristics that we just mentioned, including addressing mode, routing mechanism, security model, etc. will be discussed in this paper.

2. Classification

In the recent years, distributing and manipulating named information is a major application in the Internet. Many sorts of distribution technologies such as P2P and CDN have been widely implemented in addition to the basic web-based content distribution. They have promoted a new communication model of accessing data by name rather than their origin IP address.

In response to the current traffic growth in applications such as mobile video and cloud computing over the Internet, a diverse set of technologies and distributed services are deployed that cater for different forms of caching, replication and content distribution. These methods are currently deployed in different silos where different CDN providers and P2P applications rely on proprietary allocation techniques. Independence of distribution channels is not possible to satisfy the distribution of uniquely and safely recognize named information and different distribution methods are usually implemented as overlays which result in unnecessary inefficiencies.

The term of Information-Centric Networking first appeared around 2010, which is inspired by Van Jacobson's Google Tech Talk. After the talk which points out the new direction of moving the Internet toward a content distribution architecture, several different architectures under the concept of ICN have been constructed in recent years.

ICN represents a broad research direction of content/information/data centric approach to network architecture. We are going to describe the overview of various part of ICN architecture and its contents such as data-oriented networking architecture, content-centric networking, named-data networking, publish or subscribe, and network of information. The core idea behind ICN architectures is that who is communicating is less significant than what data are required [11].

We show the classification of ICN as in the figure 1. The first part of ICN concept was initially proposed in TRIAD and in 2006 the data-oriented network architecture(DONA)[9] project was released, which improved the security and architecture of TRIAD. Then the continuation of the Publish Subscribe Internet Routing Paradigm project, which is known as the Publish Subscribe Internet Technology project has proposed a publish and subscribe protocol stack that replace the IP protocol stack. In the next part, the Network of Information(NetInf) project was proposed[10]. Another essential part of ICN is CCN, which was proposed by Van Jacobson and the current work being performed to enhance the CCN architecture is the role of Named Data Networks(NDN).

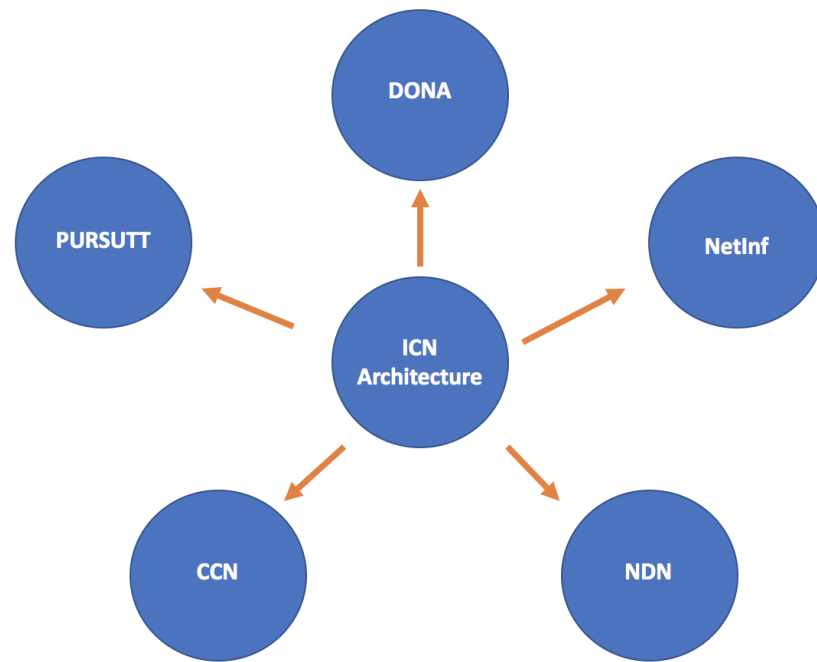


Figure.1 Classification of ICN

3. Details of Classes

3.1 CCN/NDN

CCN (the content-centric network) is a PARC project that proposed by Van Jacobson. The basic idea of the CCN is that interest packet including the user's desired data are broadcast by a consumer node, and routing protocols are used to distribute the location of content based on the name using the LPM (longest prefix matching). The hierarchical naming scheme lead to the routing aggregation leveraged. In this architecture, the data producer routes the user's desired data, along with additional authentication and data-integrity information, along the interests reverse route. Additionally, the CCN architecture includes some details such as naming, security, caching, name resolution and routing, transport, and mobility.

NDN (Named-data networking) is based on the CNN architecture, they are similar in some aspects. The NDN also obey the interest/data packet rules to transmit data. But there are also some differences between CNN and NDN. NDN reduce the interest and data search time and the looping issue.

3.2 DONA

In DONA, the provider is responsible for publishing the content in the network. To provide data, the node must use the parsing infrastructure for authorization. The name resolution uses a route-by-name paradigm. Instead of using DNS servers, DONA is implemented based on the network entities called resolution handlers(RH). Request (FIND) packets are forwarded through multiple RHs to a node with a copy of the content. Content / data can be obtained in two ways: (1) sending packets of interest over the same path and enabling caching on each encountered RH, or (2) sending back directly to the consumer. The source also has the option of registering their principal with the RH so that the request packet can be sent directly to them. However, registration must be regularly updated. RH uses a tiered routing approach to find the closest content provider.

Any name resolution used in DONA provides support for network middleware (such as firewalls, network address translators, proxies, etc.) by providing a separate mechanism for path discovery.

3.3 NetInf

The NetInf and DONA architectures have some similarities. For example, the NetInf also uses flat namespace. That means a public key infrastructure is not required. The NetInf architecture is based on the widely used multipurpose internet mail extensions standard. NetInf also contains which search primitives provide links between the object name and search item. There are 2 objects-retrieval methods in NetInf, they are name resolution and name-based routing. Furthermore, the source/NetInf node can either register with the name resolution service to publish the content or use a routing protocol to announce the routing information. When we using the name-based routing, client could send the NDO GET request directly, it means forward to the source. The data are sent to the client when a copy of the NDO is arrived. However, in some scenario the hybrid scheme are better. Because this scheme allows NetInf to adapt and scale itself to different requirements in the network. Depending on its scalability, the NetInf could be used as an extra layer on the top of the conventional network infrastructure.

3.4 PURSUIT

The Publish Subscribe Internet Technology(PURSUIT) [14] provides an enhancement of Publish Subscribe Internet Routing Paradigm(PSIRP). In PURSUIT, sources publish the content into the network and receivers can subscribe to the published contents through rendezvous systems. The convergence system helps locate the scope and publications in the network. Each posted

content belongs to a specific naming range. The subscription request contains a scope identifier (SI) and a rendezvous identifier (RI), which together identify a particular desired content. Using these identifiers during matching will generate a forwarding identifier (FI), which is used by the source to forward the data. A bloom filter[13] is specified in FI, which the intermediate router uses to select the interface for forwarding content. This alleviates the ongoing state of router maintenance. However, a bloom filter produces some false-positive result, resulting in forwarding on an interface that does not have a receiver. In addition, PURSUIT uses a flat namespace consisting of two types of names, RI and SI. These identifiers together construct the name of the content. RI helps mapping the content between publishers and subscribers. Moreover, routers use forwarding identifiers to identify the path from publishers to subscribers.

4. Performance Comparison

Our experiment use the environment as blow: the OS is Ubuntu 16.04.3, simulation tool is ndnSim2.4 and the development language is C++. In the project, we use a specific method of routing. All forwarding strategies require knowledge of where Interests can be forwarded. Unlike IP routing, the methodology of routing in NDN may not be ambiguous. But without such knowledge forwarding strategies the network cannot make decisions and it would probably drop any Interests packets.

By default, all nodes have empty FIB (Forwarding Information Base) so we need to either manually configure routes or use global routing controller. The first part of routing is manually routing, in which routes can be configured manually using the function `AddRoute` in

StackHelper and static method of StackHelper. Another method to manage FIB in large topologies is to use a global routing controller.

The initially inspiration of the first application is to implement a simple producer consumer named data network. The first example shows basic characteristics of NDN. In the first simulated topology there are 3 nodes, one producer, one consumer and one router connected with point-to-point links. Consumer is simulated using ConsumerCbr reference application and generates Interests towards the producer with frequency of 10 Interests per second. Producer is simulated using Producer class, which is used to satisfy all incoming Interests with virtual payload data which is 1024 bytes. And FIB on every node is populated using default routes.

This scenario simulates a very simple network topology in which consumer requests data from producer with frequency 10 interests per second and interests contain constantly increasing sequence number. For every received interest, producer replies with a data packet, containing 1024 bytes of virtual payload. From figure 2 we can see that we ran the experiment of 3 nodes and from the experimental results, the rate of sending Interest packet is 2.88 kbt/s and the rate of sending data is 85.6 kbt/s.

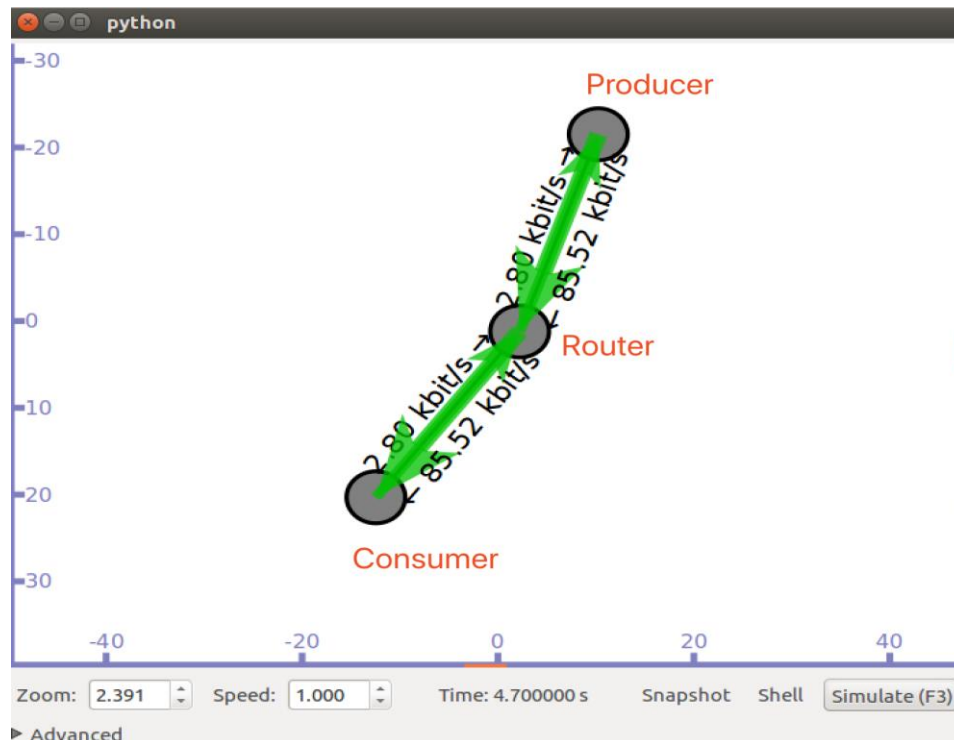


Figure.2 Performance of Simple Consumer-Producer Network

The second implementation is to add another producer and a consumer. From figure 3 we can see that under this scenario, we are able to speed up the transition rate and get the following result. The possible reason for this rate change is that when we add a producer, the overall transition rate increase significant. From the picture we can see that transition rates to consumer are enhanced from 85.6 to 88.48 kbt/s.

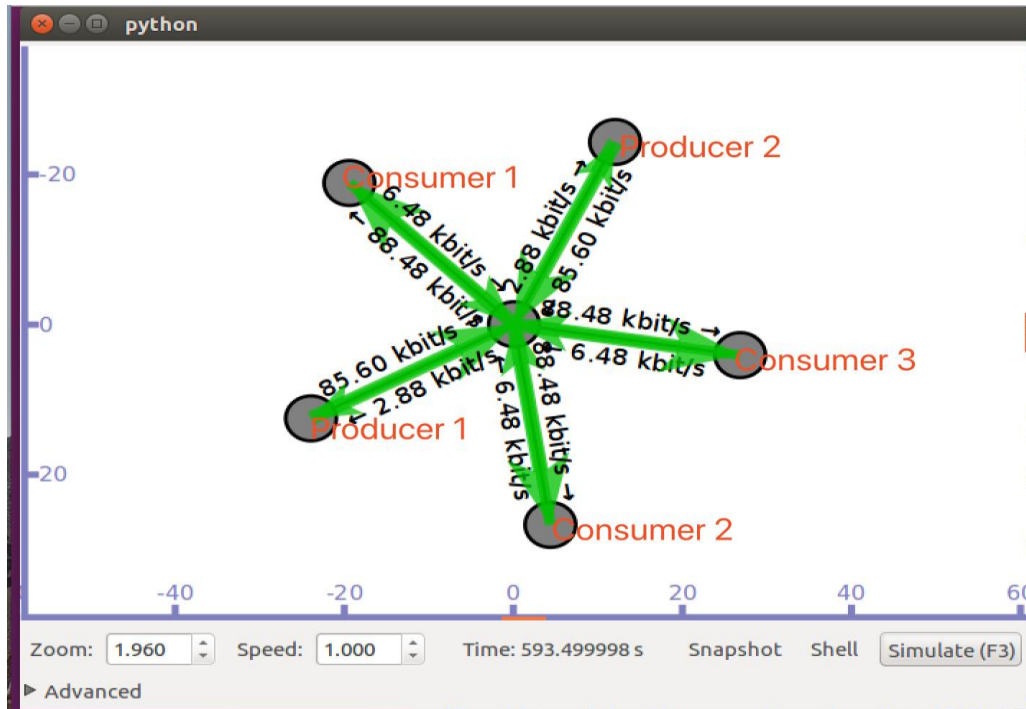


Figure.3 Performance of 2-producer 3-consumer Network

The following experiment is a 9-node grid experiment. From figure 4 we can see that this scenario simulates a grid topology using PointToPointGrid module. All links are 1Mbps with propagation 10ms delay. FIB is populated using NdnGlobalRoutingHelper. In this experiment, consumer requests data from producer with frequency 100 interests per second and interests packets contain constantly increasing sequence number. For every received interest, producer replies with a data packet, containing 1024 bytes of virtual payload.

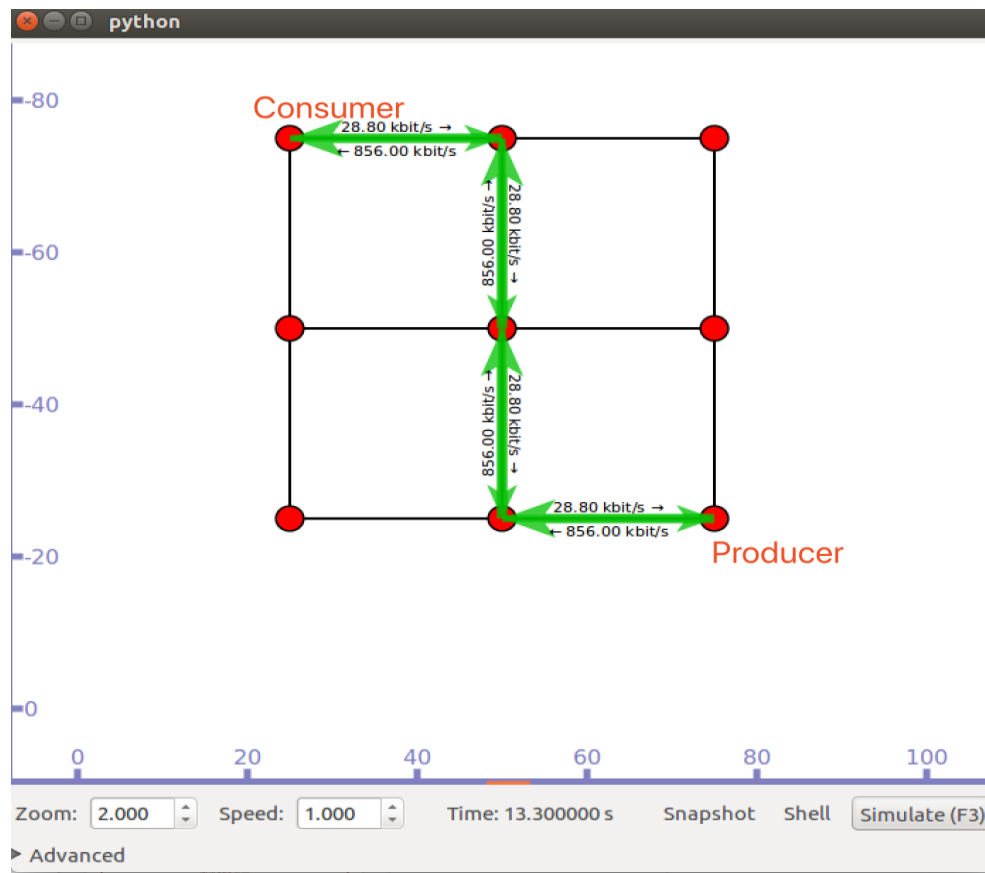


Figure.4 Performance of the 9-node grid Network

Another essential experiment of NDN is the 6 node bottleneck topology experiment. From figure 5 we can see that in this scenario, two sources sends Interest packets and destinations send back Data packets. However, there is a bottleneck link between two producers and two customers. The bandwidth between source and router is 10 Mbps while the bandwidth between destination and router is the same. The bottleneck is set between two routers, with a bandwidth of 1Mbps.

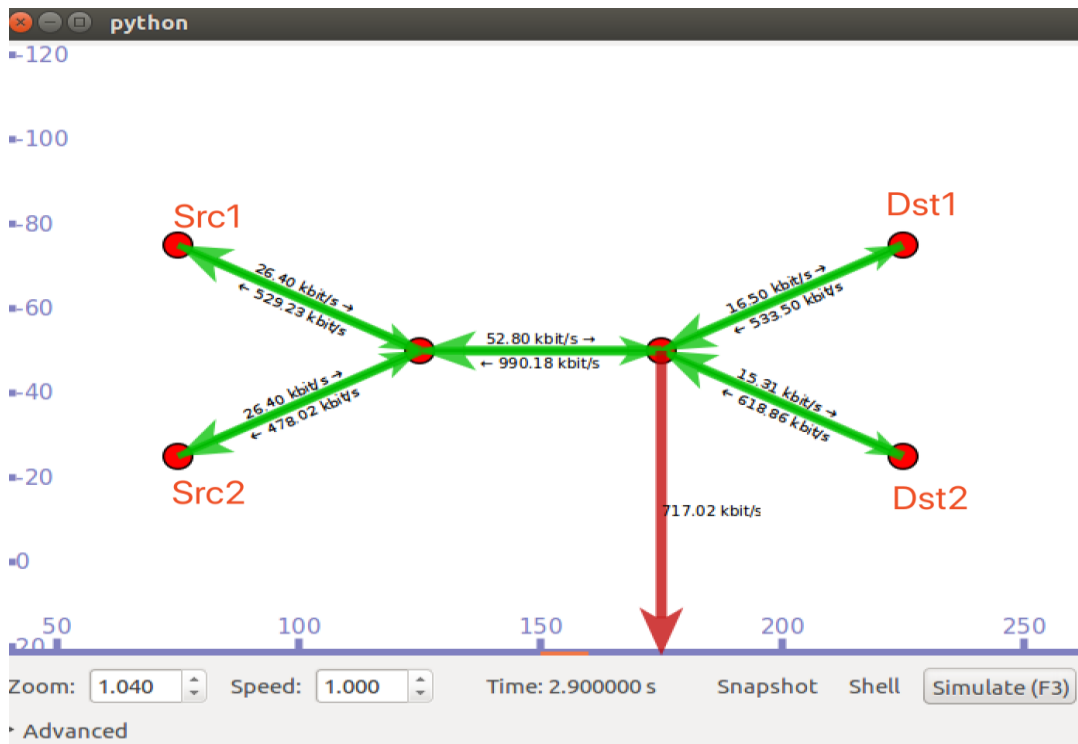


Figure.5 Performance of 6-node Bottleneck Related Network

The reason for congestion is probably because of the limitation of in-network cache. The increment of link utilization is not proportional to the increase in the number of content request. However, because of the limited router cache capacity, inappropriately cached content is cached in the upstream router along the path. As a result, the utilization of the upstream link gradually increases. Therefore, link congestion is thought to occur in this scenario.

The next simulation is a 6 node topology. From figure 6 we can see that this scenario simulates a load balancer topology. All links are 1Mbps with propagation 10ms delay. FIB is populated using NdnGlobalRoutingHelper. Consumer requests data from the two producers with frequency 10 interests per second and interests contain constantly increasing sequence number. For every received interest, a load balancing operation is performed based on a custom forwarding strategy and the selected producer replies with a data packet, containing 1024 bytes of virtual payload. Two producers generate a transition rate of approximately 400 kbps.

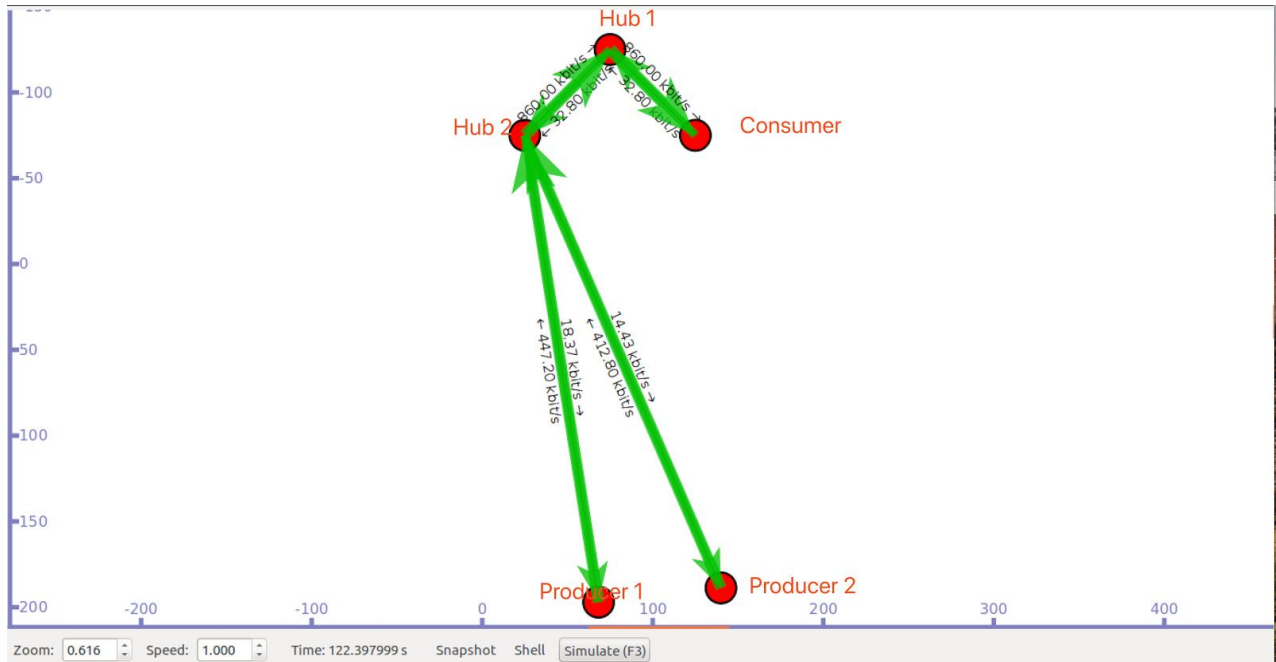


Figure.6 Performance of Load-Balancer Network

5. Conclusion

Naming data network protocols offers many advantages than other protocols, especially IP. The NDN protocol can help speed up the network data transmission and also reducing overall traffic on a given network. Ndnsm gives developers a platform for simulating different scenarios that allow us to use and learn the NDN protocol. Through the combination of the NDN protocol and the open source software ndnsim, we have shown that it is a viable alternative to traditional IP systems. Although the overall infrastructure does not currently support new protocols such as NDN, it should be considered as the Internet and its utilization rate further increased.

There are also existing some issues, we will remain them for further study. Regarding the NDN protocol survey project, we have implemented a simulation of the NDN protocols, and demonstrated its usefulness for both content distribution and point-to-point network protocols.

NDN enables the removal of application-specific middleware which is used to map between the application model and the Internet's, which will remove all its associated configuration and communication inefficiencies. In NDN, all data is secured end-to-end and the name plays an essential part in the security. Looping can also be prevented using the memory already required by the router since the data is uniquely named and can get rid of the restriction of IP routing enforces. In this scenario, it is allowed to freely use all of the connectivity of a node and distribute data and thus removes information asymmetries that unsatisfy dominant providers with the inappropriate control over routes.

In addition, the NDN protocol could fit the future faster network infrastructure and address the increased network traffic problems. Furthermore, NDN cache mechanism could improve the bandwidth usage. The characteristics of NDN such as addressing mode, forwarding strategy and security model will be implemented in the real world internet in the near future.

6. References

- [1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A Survey of Information-Centric Networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26-36, July 2012.
- [2] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu, "A Survey of Naming and Routing in Information-Centric Networks," *IEEE Communications Magazine*, vol. 50, vol. 12, pp. 44-53, December 2012.
- [3] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *Proceeding of the International Conference on Emerging*

Networking Experiments and Technologies (CoNEXT '09), Rome, Italy, December 1-4, 2009, pp. 1-12.

[4] Afanasyev A, Moiseenko I, Zhang L. ndnSIM: NDN simulator for NS-3[J]. University of California, Los Angeles, Tech. Rep, 2012, 4.

[5] Zhang L, Estrin D, Burke J, et al. Named data networking (ndn) project[J]. Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC, 2010.

[6] Jacobson V, Smetters D K, Thornton J D, et al. Networking named content[C]//Proceedings of the 5th international conference on Emerging networking experiments and technologies. ACM, 2009: 1-12.

[7] Carofiglio G, Gallo M, Muscariello L, et al. Modeling data transfer in content-centric networking[C]//Proceedings of the 23rd international teletraffic congress. International Teletraffic Congress, 2011: 111-118.

[8] Firdhous M F M. Information-Centric Networking[M]//Encyclopedia of Information Science and Technology, Fourth Edition. IGI Global, 2018: 6556-6565.

[9] Koponen, Teemu, et al. "A data-oriented (and beyond) network architecture." *ACM SIGCOMM Computer Communication Review*. Vol. 37. No. 4. ACM, 2007.

[10] Dannewitz, Christian, et al. "Network of Information (NetInf)—An information-centric networking architecture." *Computer Communications* 36.7 (2013): 721-735.

[10] Jacobson, Van. "A new way to look at networking." *Google Tech Talk* 30 (2006).

[11] Yaqub, Muhammad Azfar, et al. "Information-Centric Networks (ICN)." *Content-Centric Networks*. Springer Singapore, 2016. 19-33.

[12] Mazieres, David, et al. "Separating key management from file system security." *ACM SIGOPS Operating Systems Review*. Vol. 33. No. 5. ACM, 1999.

- [13] Bloom, Burton H. "Space/time trade-offs in hash coding with allowable errors." *Communications of the ACM* 13.7 (1970): 422-426.
- [14] Trossen, Dirk, and George Parisi. "Architecture definition, component descriptions, and requirements." *Deliverable D2 3* (2009).