Journal of Network and Computer Applications 60 (2016) 155–169

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Differentiated forwarding and caching in named-data networking

Yusung Kim [a], Younghoon Kim [a], Jun Bi [b], Ikjun Yeom [a,*]

[a] Department of Computer Engineering, Sungkyunkwan University, Suwon, South Korea
[b] Department of Computer Science and Technology, Tsinghua University, Beijing, China

ARTICLE INFO

Article history:
Received 29 October 2014
Received in revised form
19 June 2015
Accepted 11 September 2015
Available online 4 December 2015

Keywords:
Named data networking
Differentiated services

ABSTRACT

Named-data networking (NDN) is designed to provide scalable and efficient content delivery, particularly for reducing redundant data transmission. To make the usefulness of NDN more widespread, however, it should support various types of traffic and their quality of Service (QoS) requirements. In this paper, we propose a differentiated services (diffserv) model for NDN. For scalability, the proposed diffserv model is designed to follow the guidelines from the diffserv model in the current Internet. Traffic classification and packet marking are performed at edge routers, and class-based service differentiation is provided by core routers. The proposed model is also considered to take the advantages of NDN unique features such as interest aggregation and in-network caching. We implement our model on the NDN module in ns3, and show the effectiveness of NDN diffserv through extensive simulations.

1. Introduction

Named-data networking (NDN) has been proposed to resolve a traffic explosion problem resulting from the repeated delivery of a large amount of multimedia content (Jacobson et al., 2009). In NDN, instead of host addressing, a content name is used for content retrieval and delivery. As using the name-based communication, NDN can provide unique features such as in-network caching and request (interest) aggregation. Both the features can efficiently reduce redundant data transmission. A user sends an interest packet to retrieve content, and the interest packet can be served from any node that is either an original content provider or an intermediate router holding the requested data in its cache. When a router receives multiple interest packets for the same content, the interest packets can be aggregated. Only one (the first arrived) interest packet is forwarded, and the responded data packet will be copied to all requesters. The benefit of NDN is expected to reduce the bandwidth consumption and service burden of content providers. From the user's point of view, a fast response time can be also expected.

Most research on NDN so far has focused on designing and improving the basic functional components such as naming (Wang et al., 2012), routing (Eum et al., 2013), forwarding (Yi et al., 2013), cache management (Koong Chai et al., 2013; Kim and Yeom, 2013), congestion control (Carofiglio et al., 2012; Denn, 2013), and mobility (Tyson et al., 2012; Kim et al., 2012). The research seems to achieve its own purposes successfully, and recently NDN starts to be integrated to real applications such as a web browser (Qiao et al., 2014) and a building management system (Shang et al., 2014). In this paper, we extend our attention to accommodate various traffic characteristics in NDN. Even though NDN was initially proposed for efficient content delivery, for successful deployment, it should be able to satisfy different requirements for different traffic classes such as voice traffic, multimedia streaming traffic, and web traffic. To do this, the differentiated service (diffserv) model in the current Internet Protocol (IP) networks (Blake et al., 1998) can be a good starting point. In the IP diffserv model, edge routers conduct complex jobs such as traffic classification and conditioning, and core routers run class-based per-hop behaviors(PHBs) in a scalable manner.

Due to the different characteristics of NDN and IP networks, however, the diffserv model for IP networks cannot be directly applicable to NDN. For example, IP networks are basically sender-driven, and traffic classification and conditioning are performed when a data packet is entered from an edge router. However, NDN is receiver-driven, and a data packet is transmitted as a response to an interest packet. Once a data packet is entered into an NDN network, it can be cached and served for other interest packets. Here, four distinctive properties of NDN for adopting the diffserv model are addressed as follows.

- There is no end-to-end connection in NDN, and data packets of content can be partially retrieved from multiple sources which are either a content provider or in-network caches.

* Corresponding author.
E-mail addresses: yskim525@gmail.com (Y. Kim), kyhoon@gmail.com (Y. Kim), junbi@cernet.edu.cn (J. Bi), ikjun@skku.edu (I. Yeom).

http://dx.doi.org/10.1016/j.jnca.2015.09.011
1084-8045/

- Multiple interest packets for the same data are aggregated, and only one interest packet arrives at a content producer-side edge router. The edge router cannot know how many interest packets are aggregate, and it is difficult to perform per-flow (or per-user) management at the content producer-side.
- When retrieving data packets from caches in core routers, we need a way to conduct traffic classification and conditioning without harming the scalability of diffserv.
- The IP diffserv model focuses on differentiated forwarding only, but differentiated caching needs to be considered since in-network caching is a fundamental component in NDN.

In this paper, we propose a diffserv model for NDN to deal with the above properties. The proposed model consists of two pieces; the first part is a consumer-driven differentiated forwarding model. Our differentiated forwarding model is similar to the diffserv model in IP networks for securing scalability. Packet marking is performed only at a network edge, and core routers provide simple class-based scheduling for service differentiation. The major differences between our model and the IP diffserv model are that (a) we mark the service class information on an interest packet instead of a data packet, and traffic classification and conditioning are conducted at consumer-side edge routers. This keeps the diffserv role of core routers simple even though data packets are retrieved from the core routers' caches. (b) In regards to marking on interest packets, data packets are decoupled from the service class information. This enables a data packet to be delivered to multiple consumers with different service classes depending on individual consumers' subscriptions. (c) We modified an interest aggregation scheme of NDN to improve bandwidth utilization more efficiently in our differentiated forwarding without unnecessary interest blocking.

In the second part, we designed a producer-driven differentiated caching model. Our differentiated caching model is also a scalable and class-based approach. A content producer decides what content is to be marked with which service class within the producer's budget. An edge router verifies the marking of data packets according to the subscription registered by the producer, and core routers simply store the packets based on their marking in network storage. Depending on the service class type, a data packet can be stored permanently during a contracted time or bypassed to avoid unnecessary caching. In another class, named as assured caching (AC), content with the same class competes for network storage space using a replacement policy such as least recently used (LRU). In this class, the proper amount of content for assured caching can be different depending on each producer's budget, the content popularity and request rate, and network conditions. For that, we provide a scheme which allows each content producer to dynamically discover the proper space in network storage when multiple producers are competing.

We have implemented our model on ndnSIM (ndnsim,) which is a prototype of NDN in network simulator (NS-3) (Ns-3,). Through extensive simulations, we show that our proposed model can effectively provide service differentiation for NDN in terms of differentiated forwarding and caching.

2. Background and motivation

2.1. Overview of NDN

In NDN, initially proposed as Content Centric Networking (Jacobson et al., 2009), a content name is constructed hierarchically to include enough information for routing. To obtain content, a user sends a request packet with the content name, called an interest packet, and this interest packet is delivered to the content provider by longest prefix matching in the forward information base (FIB), which is analogous to a routing table with the current Internet. Upon forwarding an interest packet, an NDN router adds an entry to a pending interest table (PIT) to record information on the interface that the interest packet arrived from, thereby providing a reverse path for the corresponding content delivery. Upon receiving data, an NDN router looks up the corresponding entry in PIT, and forwards data to the incoming interface of the interest packet. After forwarding data, an NDN router stores the data in a cache called Content Store. With the employment of in-network caching, an interest packet can be served from any node which has the requested data. The location-independent data retrieval is efficient for reducing bandwidth consumption, thereby avoiding redundant data retrievals from the original content provider.

NDN has a multicast nature which comes from aggregating interest packets in PIT. When a router receives an interest packet, it creates a PIT entry for the content name. The entry keeps a list of incoming interfaces from which interest packets have been received. The router forwards an interest packet to a provider only once in a certain period. If the router receives another interest packet with the same content name, then the router just adds the incoming interface information to the list. This is called interest aggregation; it prevents a router from redundantly forwarding multiple interest packets to a provider. When a data packet returns, it is replicated and forwarded to all the interfaces recorded in the PIT entry.

2.2. Differentiated services in NDN

NDN is designed to provide efficient and scalable content delivery by reducing redundant data transmissions, but NDN needs to satisfy the different requirements of various services to be successfully deployed. Without a doubt, the diffserv model of IP networks can be a good starting point. Class-based PHBs provide various QoS with a scalable manner instead of per-flow state management. This PHB is also well-matched with the connectionless communication model of NDN. Due to different characteristics of NDN and IP networks, however, the IP diffserv model cannot be directly applicable to NDN. To design the NDN diffserv model, we need to consider the distinct features of NDN such as name-based retrieval, interest aggregation, and a location-independent communication model.

In this paper, we suggest the design requirements for the NDN diffserv model. First, the NDN diffserv model should secure scalability to keep the role of core routers simple. Even in situations where core routers serve data packets from their caches, core routers do not involve complex operations such as traffic classification and conditioning. Second, the proposed model should possess the advantages of NDN features. For example, beyond the scope of IP diffserv, NDN can decouple data itself from the service class. Consumers can retrieve the same data with different service classes, and this usage can create a synergy effect with a dynamic pricing model. Last, the NDN diffserv model should consider differentiated caching since in-network caching is one of the primary features in NDN. From defining service class types, we need to design a scheme to provide differentiated caching for multiple content producers.

3. Related work

As more services have emerged on the Internet such as voice over IP, interactive games, and video streaming, the best-effort (BE) service has not been enough to support all these services, and the demand for various QoS requirements has increased.