Differentiated Services in Named-Data Networking

Yusung Kim
Dept. of Computer Engineering
Sungkyunkwan University
Suwon, South Korea
yskim525@gmail.com

Younghoon Kim
Dept. of Computer Science
KAIST
Daejeon, South Korea
kyhoon@gmail.com

Ikjun Yeom
Dept. of Computer Engineering
Sungkyunkwan University
Suwon, South Korea
ikjun@skku.edu

***Abstract*—Named Data Networking (NDN) is an emerging communication paradigm to resolve a traffic explosion problem due to repeated and duplicated delivery of large multimedia content. To make NDN being useful more widely, 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 IP diffserv model. Traffic classification and packet marking are performed at edge routers, and class based service differentiation is provided by core routers. The proposed model, at the same time, supports fascinating features of NDN for efficient content delivery such as interest aggregation and content caching. To demonstrate the proposed model more realistically, we implement and evaluate it on a CCNx testbed.**

I. Introduction

Named Data Networking (NDN) has been proposed to resolve a traffic explosion problem due to repeated and duplicated delivery of large multimedia content [1]. In NDN, instead of host addressing, request (interest) and data packets have their own content names, and they can be cached in NDN routers. With in-network caching, NDN provides a location-independent communication model. When a user wants to retrieve content, it can be downloaded from any node which is either an original provider or a router holding the requested data in its cache. Since requests can be served by intermediate routers, it is expected to reduce bandwidth consumption and service burden of content providers and network providers. In the user's point of view, we can also expect fast response time. One other advantage of NDN is that it can handle mobile users effectively. Since data packets are delivered via the reverse path of interest packets without address specific routing, upon handover, simple retransmission of an interest packet is enough for a mobile user to keep connection without complex mobility management such as in Mobile IP.

Most researches on NDN so far have focused on designing and improving the basic functional components such as naming [2], routing [4][5], cache management [6][7], congestion control [8][9], and mobility [10][11]. The researches seem to achieve their own purposes successfully. In this paper, we extend our attention for accommodating 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 of different traffic classes such as voice traffic, multimedia streaming traffic, and web traffic. For that, the differentiated services (diffserv) model in IP networks [14] can be a good starting point. To provide various QoS with a scalable manner, per-hop behaviors (PHB) are employed in the diffserv model instead of per-flow state, and this is well-matched with the connectionless communication model of NDN.

Due to different characteristics of NDN and IP networks, however, the diffserv model for IP networks cannot be directly applicable to NDN. Here, we address two distinctive properties of NDN for adopting the diffserv model as follows: (a) IP networks are basically sender-driven, and packet marking for service differentiation is 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 of an interest packet. Hence, to meet the various requirements for the same content from individual users, we need to consider to mark on interest packets; and (b) once a data packet is entered to a NDN network, it can be cached and served for other interest packets. Hence, it may not be efficient to directly mark the service class on a data packet.

In this paper, we propose a diffserv model for NDN to deal with the above properties. The proposed model is basically similar to the diffserv model in IP networks to secure scalability such that 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 interest packets instead of data packets since a data packet can be delivered to different users with different service classes depending on individual users' subscriptions; (b) for interest packet marking, we measure the incoming rate of data packets instead of the sending rate of interest packets since the actual service differentiation is realized by different treatment for data packets, and the data rate is hard to be estimated from the interest rate without knowing the exact data size per interest; and (c) we introduce a new field for service classes in Pending Interest Table (PIT). When we pend an interest into PIT, we record its service class as well so that we can provide service differentiation to data packets without marking on them. In Fig. 1, we depict our diffserv model for NDN.

It should be noted that marking on interest packets and utilizing PIT for service differentiation is critical to realize service differentiation without compromising the architectural advantage of NDN on mobile users. Since PHB for a data packet is solely determined by its corresponding interest, we do not need any special treatments for mobile users to provide service differentiation, and the users can maintain their connections with subscribed classes by (re)transmitting interests while moving.

The proposed model is evaluated by experiments in a

978-1-4799-3088-3/14/$31.00

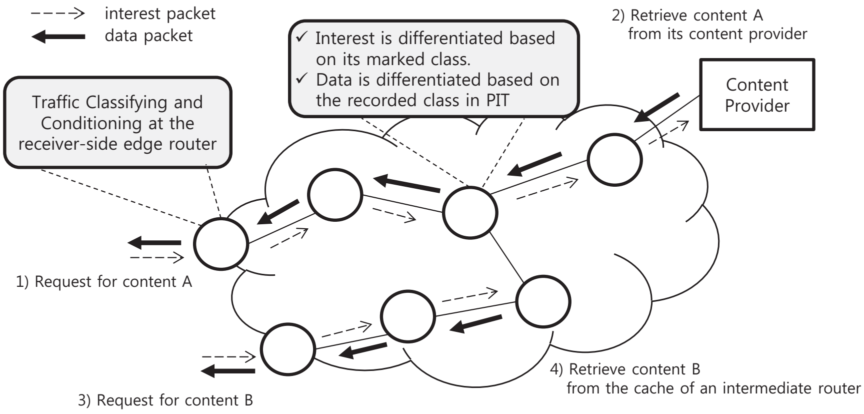


Fig. 1. Receiver-driven and class-based differentiated services

testbed. For that, we have implemented a prototype of our model on CCNx [12], and set up the testbed with a set of virtual machines. The experimental results show that our proposed model can effectively provide service differentiation for NDN.

II. Background and Motivation

A. Introduction to DiffServ

As more services have emerged in the Internet such as voice over IP, interactive games, and video streaming, the best-effort (BE) service becomes not enough to support all these services, and the demand for various QoS requirements increases. Consequently, the Internet Engineering Task Force (IETF) has proposed two representative service models to satisfy the demand for QoS. The first model is the integrated service (intserv) which is characterized by resource reservation [13]. To utilize intserv, each application sets up an end-to-end path and reserves network resources. Reservation-based intserv is successful to provide required QoS guarantee, but also causes a scalability issue.

To overcome the deployment difficulty of IntServ, the differentiated services (diffserv) architecure has been proposed in [14]. To achieve high scalability, the diffserv provides coarse-grained class-based service differentiation rather than service guarantee. In a diffserv network, depending on the target application, a user subscribes to one of the service classes such as expedited forwarding (EF) for interactive voice and video services [15] and assured forwarding (AF) for real-time streaming services [16]. Best-effort (BE) is still available for users who do not want to pay additional cost. Here, note that the subscription is contracted with the maximum data rate to protect the minimum service quality. Then, packets from a user are marked according to the subscription contract at an edge router when they are injected to a diffserv network. It is also noted that marking behavior is differentiated by the service classes. For EF class, the amount of traffic exceeding the contracted rate is unconditionally discarded for providing high service quality whereas, for AF class, the exceeding amount of traffic could be marked as BE for achieving high throughput. Even though, it can cause out of order data delivery, in NDN, reordering data packets is inevitable because data can be served from multiple sources. Since the complex operations such as classification and marking are performed at the network edge, simple priority queueing is enough to provide service differentiation in the network core.

B. Overview of NDN

In Named Data Networking (NDN) (initially proposed as Content Centric Networking) [1], 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 in 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, a 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 to reduce bandwidth consumption avoiding redundant data retrievals from the original content provider.

NDN has a multicast nature by 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 toward 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, which prevents a router from redundantly forwarding multiple interest packets toward a provider. When a data packet returns, it is replicated and forwarded to all the interfaces recorded in the PIT entry.

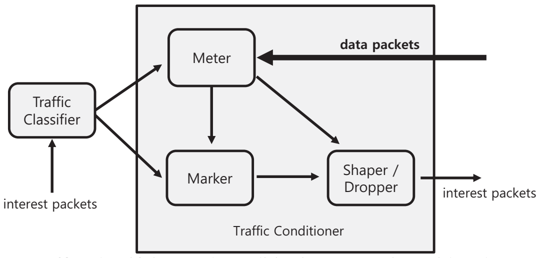


Fig. 2. Traffic classifying and conditioning at receiver-side edge networks

III. A Receiver-Driven DiffServ Model in NDN

A. Basic Operations

We propose a receiver-driven and class-based diffserv model in NDN. In our model, interest packets are assigned into a few number of service classes. When a receiver generates an interest packet, the receiver-side edge router marks the service class information on the interest packet, and controls the interest marking rate. The interest packet is differentiated according to its service class when the interest packet is forwarded to a content provider.

If the interest packet meets any NDN diffserv node which has the requested data, the node uses the class information marked in the interest packet. All NDN diffserv nodes along the reverse path of the interest packet can differentiate the data packet by forwarding it following the service class recorded in PIT.

B. Traffic Classification and Conditioning

Service levels in the NDN diffserv model are determined between service providers and customers in the form of a Service Level Agreement (SLA). The SLA specifies the supported service classes and the amount of traffic allowed in each class. Interest packets are classified, policed, and shaped at the ingress routers of the receiver-side edge networks.

In NDN, each interest packet is mapped to a content chunk. As we control the marking rate of interest packets, we can control the receiving rate of data packets. However, when the size of every content chunk is unknown, it may be difficult to decide the sending rate of interest packets. Since the sizes of the content chunks are different according to applications or even variable in one application, we suggest that Traffic Conditioner measures the receiving rate of data packets and controls the marking rate of interest packets. The process at the receiver-side edge router is presented in Fig. 2.

In a sender-driven diffserv model, it is straightforward to measure and shape the data sending rate. In our model, the current measured data rate is resulted by the previous interest marking rate. To reduce the oscillation of the data rate, we need a smooth average of the data rate over a period of time [17]. In Algorithm 1, we present a simple scheme to provide a static interest marking rate. In this scheme, the edge router marks a service class information on interest packets in order to retrieve assured data packets as much as the contract rate. The procedure in Algorithm 1 is conducted every interval which is the product of the average round trip-time (RTT) and a constant value. The average of RTT can be measured since NDN already monitors interest packets' expiration time in PIT.

Algorithm 1 Control of interest marking rate

1: Doing the following procedure every $interval$
2: **procedure** INTEREST MARKING RATE CONTROL
3:     update avg_rtt
4:     update $data\_receiving\_rate$
5:     update S : avg_chunk_size
6:     update N : $avg\_number\_of\_marked\_intersts$
7:     $R = (N \times S)/interval$
8:     **if** $contract\_rate < R$ **then**
9:         decrease $interest\_marking\_rate$
10:     **else**
11:         increase $interest\_marking\_rate$
12:     **end if**
13:     $interval = C \times avg\_rtt$ $\triangleright$ C is a constant
14: **end procedure**

In the line seven, the estimated data rate, R, is calculated by multiplying the average number of marked interest packets and the average chunk size. As comparing R with the contract rate, the router decreases or increases the interest marking rate.

As we mentioned above, if the incoming interest packet rate is higher than the interest marking rate, the excessive interest packets can be discarded or be sent out without marking based on policies or SLAs. In our prototype implementation, we allow the edge router sends excessive interest packets (for AF classes) without marking. The unmarked interest packets are treated as a BE class. If some of them survive and retrieve data packets, AF users can receive more throughput than their contract rates.

When AF users receive additional throughput by unmarked interest packets, their total data receiving rates can be higher than their contract rates. Network operators or users may want to limit the total data receiving rate to the contract rate. To achieve the goal, we provide a dynamic scheme to limit the data receiving rate at the edge router. The line seven in Algorithm 1, we set R to be the average data receiving rate. If the average data receiving rate is higher than the contract rate, the router decreases the interest marking rate until the data rate matches with the contract rate. If network condition changes and the average data receiving rate is lower than the contract rate, the router will increase the interest marking rate.

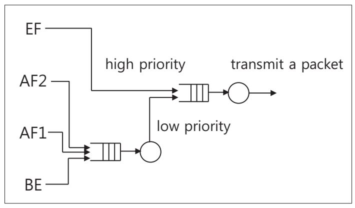
C. Marking on Interest only (Data without Marking)

In NDN, there are two types of packets; interest and data. To provide QoS correctly, both interest and data packets needs to be assured. These two packet types may carry the information of the service class in their headers. In our NDN diffserv model, we mark the service class on interest packets only at the receiver-side networks.

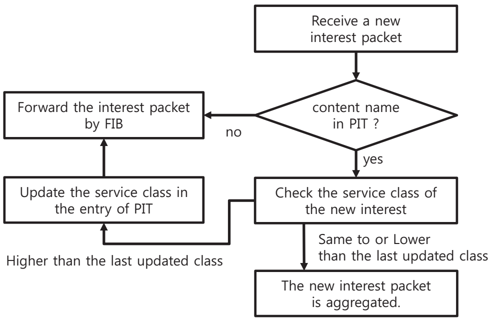
When the interest packet traverse networks, the information of the interest packet is recorded in PIT at NDN routers. We add a field to store the information of the service class in PIT as shown in Fig. 3(a). The requested data packet follows the reverse path of the interest packet. By consuming the information of PIT, a NDN router knows how to differentiate the data packet and where to transit as described in Fig. 3(d). Therefore, there is no necessary to conduct complex traffic classification and conditioning for data packets, and a data

Pending Interest Table		
Content Name	Service Class	Face ID
/xxx/voice1	EF	0
/yyy/b.patch	AF_1	1
	BE	0, 2
/zzz/video1	AF_2	2
…	…	…

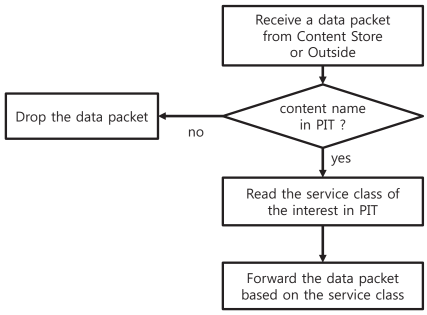
(a) Modification of PIT



(b) Per-hop behavior



(c) Interest handling



(d) Data handling

Fig. 3. Interest and data handling

packet does not need to carry the information of its service class. It enables to take the advantages of NDN such as in-network caching and location-independent data retrieval. In the view of mobility, the receiver-driven diffserv model complies with the principle of NDN mobility support. A receiver continuously sends a new interest packet after moving to a different network. The interest packet is classified at the new ingress edge router. The data packet can be retrieved by following the reverse path of the new interest packet according to the information of the service class in PIT.

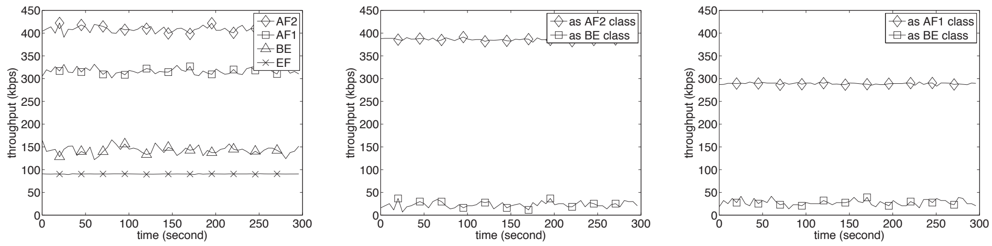
A collection of (both interest and data) packets with the same service class is referred to as a behavior aggregate (BA) [14]. At the core of the network, each BA is associated with a particular forwarding treatment known as per-hop behavior (PHB). In Fig. 3(b), there is an example of PHB implementations; the expedited forwarding (EF) PHB, the assured forwarding (AF) PHB, and the default best-effort (BE) PHB. After mapping each packet to PHB based on the service class information, PHBs provide resource assurances and different levels of services by allocating forwarding resources (bandwidth and buffer) to each BA. Since the concept of PHB in our model is same with the IP diffserv model, our model secures the simpleness and scalability of the core of networks.

D. Free Upgrading

When two different users want to retrieve the same content with different service classes, in IP diffserv, a sender needs to transmit the content twice as marking the different service classes on packets. In NDN diffserv, if users send interest packets for the same content with different service classes, receiver-side edge routers mark the different service classes on the interest packets. Data packets will be retrieved based on the service class information recorded in PIT. However, when multiple interest packets for the same content arrive at a NDN router, the interest packets are aggregated in PIT. If an interest packet with a lower service class was added in PIT, a new interest packet with a higher service class cannot be forwarded to a content provider. Therefore, the requested data will be retrieved as a lower service class. We solve this issue by modifying the process of the interest aggregation in PIT.

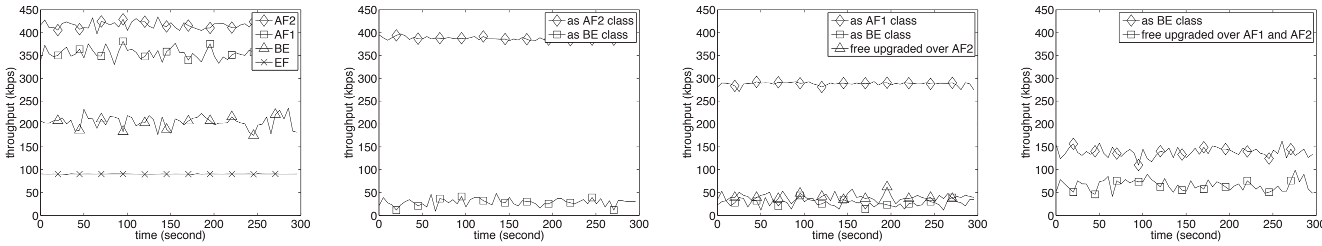
We allow that a NDN router re-sends a new interest packet if the interest packet has a higher service class than the highest service class of previous interest packets for the same content. The router adds (or updates) the new highest service class into the related PIT entry. The requested data packet will be retrieved as the last updated (highest) service class. For example, in Fig. 3(a), interest packets are arrived from 0, 1, and 2 faces (or network interfaces) for the content name of ”/yyy/b.patch”. If all interest packets have the same service class such as BE, the router forwards only one interest packet to its content provider. When a new interest packet with AF1 arrives at the router, the router will forward the interest packet toward the content provider to notify the service class update. Finally, the related PIT entry has two service classes for BE and AF1, and each class includes the list of incoming interfaces. Fig. 3(c) and Fig. 3(d) shows the processes of handing interest and data packets.

With the modification of interest aggregation, a NDN diffserv node gives a data packet the highest service class aggregated in PIT. Users with lower service classes can experience that their service classes are upgraded for free. We call it free upgrading of the service class. When an AF1 user receives a data packet by free upgrading over AF2 where the priority of the AF2 (assured forwarding) class is higher than that of the AF1 class, the data packet should be retrieved as the AF2 class. It means that the AF1 user can have a room to retrieve another data packet as the AF1 class at the time. Therefore, free upgrading increases the throughput of the AF1 user. Due to free upgrading, a user may receive the higher



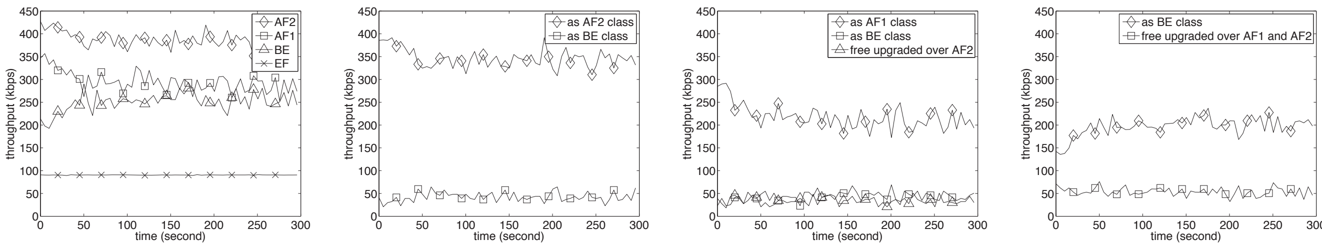
(a) Total throughput (b) Throughput detail of AF2 (c) Throughput detail of AF1

Fig. 4. Static interest marking scheme without a free upgrading case



(a) Total throughput (b) Throughput detail of AF2 (c) Throughput detail of AF1 (d) Throughput detail of BE

Fig. 5. Static interest marking scheme with a free upgrading case



(a) Total throughput (b) Throughput detail of AF2 (c) Throughput detail of AF1 (d) Throughput detail of BE

Fig. 6. Dynamic data rate limiting scheme with a free upgrading case

data rate than the user's contract rate. If the user charges per byte for assured forwarding, the user may want to reduce the total data rate to the contract rate. In our model, the receiver-side edge router can dynamically limit the total data rate by controlling the interest marking rate.

A content provider may not want to allow the free upgrading situation. In the case, the content provider can explicitly notify both network providers and users of a specific service class information, which should be assigned to content, in service-level agreements.

IV. Implementation and Evaluation

A. Building a Testbed

To demonstrate our NDN diffserv model, we build a testbed composed of 11 virtual machines as a dumbbell topology. The topology consists of four receivers, four senders, two edge routers, and one core router. The prototype of NDN (called CCNx) is deployed at receivers and senders. The core router uses IP differentiated service with a tc tool including hierarchical token bucket (HTB) queueing discipline in Linux. The link capacity of the core router is limited to 1 Mbps. The bandwidth is divided into the 4 service classes; EF, AF2, AF1, and BE. The assured bandwidth of EF, AF2, and AF1 is 100 kbps, 400 kbps, and 300 kbps. BE has the rest (200 kbps) of bandwidth. Due to the performance issue of virtual machines with CCNx in a single computer, we used the relatively low bandwidth, but we believe the bandwidth is enough to demonstrate the prototype of our NDN diffserv model.

Since CCNx is operated over IP, we mark the service class information on both a interest header and an IP header. The receiver-side edge router marks the service class on interest headers, and also marks diffserv code point (DSCP) on IP headers. When the core router receives interest packets, the router differentiates the packets based on their DSCP in IP headers. When data packets are retrieved from senders, the senders read the service class information of interest headers, and mark DSCP on IP headers for data packets. The data packets traverse the core router, and the packets can be differentiated based on DSCP.

B. Evaluation Scenario

Each receiver is dedicated to create interest packets for each service class (EF, AF2, AF1, and BE). All interest packets are generated with constant-bit rates. The EF receiver retrieves 95 kbps of the data rate slightly less than the assured 100 kbps. The AF2, AF1, and BE receivers create interest packets to retrieve data rates; 500 kbps, 400 kbps, and 300 kbps. Since the data rates exceed their assured rates, packet losses are inevitable.

We have three experiment scenarios; the first scenario is a static interest marking scheme without a free upgrading case. The second scenario is a static interest marking scheme with a free upgrading case. 10 percent of AF2 interest packets have the same content name with those of AF1 and BE receivers at the same time. The BE receiver also has the same content name with 10 percent of AF1 interest packets. It means that the AF1 receiver can experience free upgrading over AF2, and the BE receiver also experiences free upgrading over AF2 and AF1. The last scenario shows a dynamic scheme to limit the data receiving rate with a free upgrading case.

C. Evaluation Result

Fig. 4 shows the result of the first scenario. In Fig. 4(a), total throughput is presented for each receiver with a different service class. As we expected, the EF receiver retrieves 95 kbps without oscillation. The AF2 and AF1 receivers have a little more throughput than their contract rates(400 kbps and 300 kbps), while the BE receiver has less throughput than 200 kbps. The reason is shown in Fig. 4(b) and Fig. 4(c). The AF2 and AF1 receivers retrieved about 400 kbps and 300 kbps by marked interest packets and took additionally some data packets gained by unmarked interest packets (as a BE class). The data packets of the BE receiver should compete with data packets which are retrieved by the unmarked interest packets of the AF2 and AF1 receivers. Therefore, the throughput of the BE receiver was reduced as much as the AF2 and AF2 receivers took as a BE class.

Fig. 5 presents the result of the second scenario. Since the AF2 receiver does not experience free upgrading, its throughput is same with the result of Fig. 4. The throughputs of AF1 and BE receivers increased compared to Fig. 4. The total throughput of the AF1 receiver is the average 350 kbps. Its 300 kbps is retrieved by marking the AF1 class on interest packets. The additional 50 kbps is gained by both unmarked interest packets and free upgrading. The BE receiver also gained additional 60 kbps from free upgrading over AF2 and AF1.

In Fig. 6, we show the impact of limiting the data receiving rate. The receiver-side edge router limits the data rates for the AF1 and AF2 receivers. The edge router measures their total data rates, and adjust the interest marking rates until their total data rates match with their contract rates. In Fig. 6(a), total throughputs of the AF2 and AF1 receivers are about 400 kbps and 300 kbps after 50 seconds. In the case of the AF2 receiver, 350 kbps is retrieved by marked interest packets and 50 kbps is served as a BE class. Similarly, the AF1 receiver could charge for only 230 kbps as an AF1 class since the rest 70 kbps is gained by free upgrading and unmarked interest packets. Limiting the data rates of AF2 and AF1 receivers impacted the throughput of the BE receiver. The increased throughput of the BE receiver was proportional to the amount decreased at the AF2 and AF1 classes.

V. Conclusion

In this paper, we have studied a differentiated service model for Named Data Networking. We have shown the design requirements of a receiver-driven and class-based diffserv model. In our model, traffic classifying and conditioning are done only at the receiver-side edge networks. Interest packets carry the service class information, but data packets use the service class information recorded in PIT. It keeps our NDN diffserv model simple and scalable. Our model can also provide more flexibility than the IP diffserv model. By modifying the process of interest aggregation, we allow that users retrieve the same content with different service classes. It results in free upgrading that a user with a lower service class can be served by a higher service class. Due to free upgrading, users may receive the higher data rates than their contract rates. To limit their data rates, the receiver-side edge router can control their interest marking rates. We have built a testbed using a set of virtual machines as a dumbbell topology. With the combination of CCNx implementation and an IP diffserv module in Linux, we have evaluated our NDN diffserv model. We also have shown the impact of free upgrading and the limiting of the data rate. We will investigate the collaboration with differentiated caching services in NDN for future work.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013024839) and the Ministry of Science, ICT & Future Planning (2013035142).

References

[1] V. Jacobson, et al., *Networking named content*, ACM CoNEXT, 2009.

[2] L. Wang, R. Wakikawa, R. Kuntz, R. Vuyyuru, and L. Zhang, *Data naming in vehicle-to-vehicle communications*, IEEE Infocom NOMEN Workshop, 2012.

[3] M. Bari, et al., *A survey of naming and routing in information-centric networks*, IEEE Communications Magazine, 2012.

[4] L. Wang, et al., *OSPFN: An OSPF based routing protocol for Named Data Networking*, Technical report, 2012.

[5] C. Yi, et al., *On the Role of Routing in Named Data Networking*, Technical report, 2013.

[6] W. Chai, D. He, I. Psaras, and G. Pavlou, *Cache Less for More in Information-Centric Networks*, IFIP Networking, 2012

[7] Y. Kim and I. Yeom, *Performance analysis of in-network caching for content-centric networking*, Computer Networks, 2013.

[8] G. Carofiglio, M. Gallo, and L. Muscariello, *Icp: Design and evaluation of an interest control protocol for content-centric networking*, IEEE Infocom NOMEN workshop, 2012.

[9] G. Carofiglio, et al., *Optimal Multipath Congestion Control and Request Forwarding in Information-Centric Networks*, IEEE ICNP, 2013

[10] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe, *A survey of mobility in information-centric networks: challenges and research directions*, ACM MobiHoc NoM workshop, 2012.

[11] D. Kim, J. Kim, Y. Kim, H. Yoon, and I. Yeom, *Mobility support in content centric networks*, ACM Sigcomm ICN workshop, 2012.

[12] CCNx, 2013, *http://www.ccnx.org/*

[13] R. Braden, D. Clark, and S. Shenker, *Integrated Services in the Internet architecture: An overview*, RFC 1633, 1994.

[14] S. Blake, F. Baker, and D. Black. *An architecture for differentiated service*, RFC 2475, 1998.

[15] B. Davie, et al., *An expedited forwarding PHB (per-hop behavior)*, RFC 3246, 2002.

[16] J. Heinanen, et al., *Assured forwarding PHB group*, RFC 2597, 1999

[17] D. D Clark, and W. Fang, *Explicit allocation of best-effort packet delivery service*, IEEE/ACM Transaction on Networking, 1998.