Performance Analysis of CCN on Arbitrary Network Topology

Ryo Nakamura and Hiroyuki Ohsaki
Graduate School of Science and Technology
Kwansei Gakuin University
Hyogo 669-1337, Japan
Email: {r-nakamura,ohsaki}@kwansei.ac.jp

***Abstract*—In this paper, by utilizing the MCA (Multi-Cache Approximation) algorithm, which is an approximate algorithm for numerically solving cache hit rates in a multi-cache network, we analytically obtain performance metrics of CCN (Content-Centric Networking). Specifically, our analytic model consists of multiple routers, multiple repositories (e.g., storage servers), and multiple entities (e.g., clients). We obtain three performance metrics — content delivery delay (i.e., the average time required for an entity to retrieve a content through its neighbor router), throughput (i.e., the number of contents delivered from an entity per a unit time), and availability (i.e., the probability that an entity can successfully retrieve a content from the network). Through several numerical examples, we investigate how the network topology affects the performance of CCN. Our findings include that the closer an entity is to the requesting repository, the more beneficial the contents caching is in terms of content delivery time and availability, and that the farther an entity is from the repository, the more beneficial the content caching is in terms of throughput.**

I. Introduction

In the recent years, Content-Centric Networking (CCN) [1] has been under the spotlight as one of the networks mainly focusing on the contents that are transmitted and received (data-centric networks) instead on the hosts that transmit and receive the contents (host-centric networks).

CCN is expected to deliver high availability since multiple repositories maintain copies of an identical content while also allowing reduction in traffic volume by caching the content relayed by network routers. With Internet Protocol (IP), one must communicate directly with the host that maintains the content in order to obtain a certain content. CCN does not require identification of the host that maintains the content, and the content can be obtained from anywhere as long as they exist in the network. As a consequence, CCN is expected to address reduction in content delivery delays, reduction in traffic volume transferred through the network, and improvement in availability because of capability to disseminate content copies within the network in a natural way.

In the literature, there have been many studies that investigated the effectiveness of CCN through simulation experiments. For example, in [2], the effectiveness of CCN in video streaming is investigated through simulation. Their results show that the performance of CCN is not greatly affected by the topology and that the effectiveness of CCN depends largely on the distribution of content requested by users. In [3], the power consumption for video streaming is examined when either IP or CCN is used. Their results show that CCN can reduce the overall power consumption because of reduced traffic volume, which is resulted from caching, although the power consumption in network devices is increased for maintaining a large volume of cache.

There have also been mathematical analyses of CCN [4]–[8]. For example, the authors of [4] developed a Markovian model for the cache in only a single CCN router, and examined the performance of CCN in a tree topology by complementing their Markovian model with simulation results. The caching performance of a single CCN router with Interest packet aggregation was analyzed in [5]. The authors of [6] calculated the CCN throughput and content delivery delay on cascade-type and binary tree-type network topologies. As a result, it is shown that the CCN throughput and content delivery delay depend on the cache size, content size and content popularity distribution. In addition, authors of [7] calculated the content delivery delay, throughput and download time when an entity performed multipath access from multiple leaf routers in a tree network by extending analysis in [6]. However, these analytical studies were limited to simple network topologies, and the effectiveness of CCN in a more general network topology has yet to be figured out.

The pioneering work in performance analysis of a multi-cache network is [9], which proposed the Multi-Cache Approximation (MCA) algorithm for analytically calculating cache hit rates of intermediate nodes. The authors of [9], mostly focus on the link-level performance (i.e., cache hit rates) rather than on the network-level performance (e.g., delivery delay, throughput, and availability).

In [8], the performance of CCN in a general network topology was analyzed. In this study, the cache hit rates at routers and content delivery delay (referred to as Virtual Round-Trip Time in [8]) in a general network topology were obtained by utilizing the MCA algorithm [9]. While the current paper analyzes the CCN performance using the MCA algorithm in a similar fashion to [8], it also analytically calculates the throughput and availability assuming link failures in addition to content delivery delay.

In this paper, the content delivery delay, throughput, and availability in CCN are analytically calculated for an arbitrary network topology. A CCN network comprising of multiple routers and multiple repositories is modeled. Performance when multiple entities request contents stored in repositories is analyzed. The time required until an entity obtains the content after making a request (content delivery delay), throughput for content acquisition, and probability for an entity to successfully obtain the content under probabilistic link failures are also analytically calculated. Furthermore, the effect of network

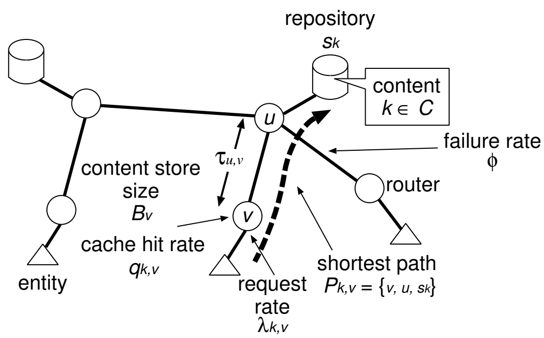


Fig. 1. Analytic model

topology on the effectiveness of CCN is also studied through several numerical examples.

This paper is constructed as follows: First, Section II describes the analytic model used in this paper. Section III analyzes the content delivery delay, throughput and availability in CCN for an arbitrary network topology. Section IV looks into the effects of network topology on the effectiveness of CCN using several numerical examples. Section V examines the validity of our approximate analysis by comparing our analytic results with simulation ones. Finally, Section VI provides the summary of this paper and future challenges to be addressed.

II. Analytic model

The topology for CCN network comprised of multiple routers (CCN routers) and multiple repositories is expressed as an undirected graph $G = (V, E)$ (Fig. 1). Hereafter these routers and repositories are collectively referred to as *nodes*.

C represents the collection of all contents present in the network. To simplify, it is supposed that all contents have the same size. The Content Store size for router v is expressed as B_v, the communication delay for the link between node u and node v (i.e., the propagation delay plus all processing delays) as $\tau_{u,v}$. The failure rate for each link is set equally to ϕ.

It is supposed that each content exists in a single repository, and that the Forwarding Information Base for each router is properly set up by the routing protocol based on the shortest path.

The shortest path from node $v \in V$ to the repository which stores content $k \in C$ is expressed as $P_k^v = (v, \ldots, s_k)$. Here, s_k indicates the repository which stores content k. The n-th node in the shortest path P_k^v is expressed as $P_k^v[n]$. Therefore, $P_k^v[1] = v$ and also $P_k^v[|P_k^v|] = s_k$.

It is considered that each entity is connected directly to a router, that the bandwidth between an entity and a router is sufficiently large, and that the communication delay between an entity and a router is small enough to be neglected.

The arrival rate of Interest packets for content k received by node v from directly connected subordinate entities is expressed as $\lambda_{k,v}$. In addition, the cache hit probability for content k at node v is expressed as $q_{k,v}$. For the repository s_k that stores content k, we define $q_{k,s_k} = 1$.

III. Analysis

A. Content delivery delay

First, the content delivery delay (i.e., the expected time required until the requested content is obtained after an entity requests for it) in CCN when there is no link failure (i.e., $\phi = 0$) is obtained.

Since the router caches contents within the network in CCN, the content delivery delay is reduced, and also reduction of traffic volume transferred within the network can be expected. Since each content is distinguished by its unique identifier in CCN, contents are recycled at the network level. A router returns the Data packet without further relaying the Interest packet if it has the Data packet corresponding to the Interest packet in its Content Store.

When an entity sends out an Interest packet to the network, the router forwards the Interest packet to the nearest repository according to the routing table called the Forwarding Information Base. If the content corresponding to the Interest packet is cached in a router on the path, the router returns the corresponding content to the entity as a Data packet. If the content corresponding to the Interest packet is not cached in a router on the path, the Interest packet arrives at the repository and the repository returns the corresponding content to the entity as a Data packet.

The cache hit probability $q_{k,v}$ for content k at router v can be approximately obtained using Multi-Cache Approximation (MCA) algorithm [9] or Multi-Cache with Aggregation Approximation (MCAA) algorithm [8] for partial networks comprised only with routers.

MCA is an approximation algorithm to analytically calculate the cache hit probability in a multi-cache network [9]. MCA uses Single-Cache Approximation (SCA) [10] to calculate the cache hit probability at a single node whose buffer size is finite and whose cache replacement algorithm is FIFO or Least-Recently Used (LRU). It repeatedly applies SCA to each node in the network and calculates the cache hit probability at each node.

MCA algorithm calculates $m_{k,v}$ that satisfies the following equations through repeated calculations [9].

$$r_{k,v} = \lambda_{k,v} + \sum_{v' : k \in R(v', v)} m_{k,v'} \quad (1)$$

$$p_{k,v} = \frac{r_{k,v}}{\sum_{i=1}^{|C|} r_{i,v}} \quad (2)$$

$$\vec{q_v} = contents(\vec{p_v}, B_v) \quad (3)$$

$$m_{k,v} = r_{k,v}\,(1 - q_{k,v}) \quad (4)$$

Here, $m_{k,v}$ indicates the rate of misses (i.e., the number of misses occurred per unit time) at node v for content k. $r_{k,v}$ indicates the request rate (i.e., the total of the request rate flowing in from upstream nodes and the request rate received from entities directly connected to the node) for content k at node v. $R(v, v')$ is the collection of contents for which node v is the next hop for node v' on the shortest path. For example, if $R(v, v') = \{k\}$, it means that node v is located at the next hop in the shortest path for content k at node v'. Also, $p_{k,v}$ and $q_{k,v}$ are the relative request rate and the cache hit probability for content k at node v, respectively. $\vec{p_v}$ and $\vec{q_v}$ are vectors consisting of $p_{k,v}$'s and $q_{k,v}$'s for all contents, respectively.

In [8], the MCAA algorithm, which extended the MCA algorithm and modeled the aggregation of Interest packets at a router, was proposed.

By utilizing either of these MCA and MCAA algorithms, the cache hit probability $q_{k,v}$ for content k at router v can be calculated.

The probability for the Data packet corresponding to an Interest packet requested by node v for content k to be returned from the n-th node on the shortest path P_k^v (i.e., hitting the cache at the n-th node or the n-th node being the repository) is expressed as $\eta_{k,n}^v$. Since content k is always returned from one of the nodes on the path P_k^v if there is no link failure, $\sum_n \eta_{k,n}^v = 1$.

Since the cache hit probability at the n-th node is $q_{k,P_k^v[n]}$, $\eta_{k,n}^v$ is given by

$$\eta_{k,n}^v = q_{k,P_k^v[n]} \prod_{i=1}^{n-1} (1 - q_{k,P_k^v[i]}). \tag{5}$$

Therefore, the expectation of content delivery delays for the Interest packet received by node v for content k is given by

$$D_k^v = \sum_{n=2}^{|P_k^v|} \left(\eta_{k,n}^v \sum_{m=1}^{n-1} 2\, \tau_{P_k^v[m],P_k^v[m+1]} \right). \tag{6}$$

Since the arrival rate of Interest packets received by node v from entities, which are directly connected, for content k is $\lambda_{k,v}$, the expectation of content delivery delays, D^v, at node v for all contents is given by

$$D^v = \sum_{k \in C} \frac{\lambda_{k,v}}{\sum_{k' \in C} \lambda_{k',v}} D_k^v. \tag{7}$$

B. Throughput

Next, the throughput for content retrieval in CCN when link failure does not occur (i.e., $\phi = 0$) is obtained.

In general, the size of an Interest packet is expected to be much smaller than the size of a Data packet. Thus it is supposed that the traffic for Interest packets is small enough to be neglected. The size of the Data packet is expressed as S.

The rate at which the Data packet is returned by node v for content k is expressed as $x_{k,v}$. The rate at which node v received Interest packets for content k is $r_{k,v}$ and the cache hit probability is $q_{k,v}$, so we have

$$x_{k,v} = r_{k,v}\, q_{k,v}\, S + \sum_{v' : k \in R(v',v)} \xi_{k,v}(v'), \tag{8}$$

where $\xi_{k,v}(v')$ indicates the reception rate for Data packets for content k that flows from node v' into node v.

The rate of Data packet transmission for content k sent from node v' to node v is expressed as $\zeta_{k,v'}(v)$. It should be noted that $\zeta_{k,v'}(v)$ is the rate at which transmission is made from node v' to node v, and $\xi_{k,v}(v')$ the rate of reception at node v from node v'.

If the bandwidth between node v and node v' is $\mu_{v,v'}$, Data packets exceeding bandwidth $\mu_{v,v'}$ is discarded at transmission from node v'. Therefore, $\xi_{k,v}(v') \le \zeta_{k,v'}(v)$.

The rate at which node v' returns Data packets for content k is $x_{k,v'}$ and only a fraction $m_{k,v}/r_{k,v'}$ of that is transmitted to node v. So, we have

$$\zeta_{k,v'}(v) = x_{k,v'} \frac{m_{k,v}}{r_{k,v'}}. \tag{9}$$

Assuming that fair queuing is conducted at all routers and the rate of loss for Data packets is proportional to the transmission rate for Interest packets, $\xi_{k,v}(v')$ is given by

$$\xi_{k,v}(v') = \min(\mu_{v,v'}, \sum_{i \in C} \zeta_{i,v'}(v)) \frac{\zeta_{k,v'}(v)}{\sum_{i \in C} \zeta_{i,v'}(v)}. \tag{10}$$

Therefore, the rate of transmission for Data packets for content k returned by node v to entities connected immediately below, T_k^v, (i.e., throughput) is given by

$$T_k^v = \frac{\lambda_{k,v}}{r_{k,v}} x_{k,v}. \tag{11}$$

The total throughput T^v for Data packets for node v to return to the entities connected immediately below is given by the sum of throughputs for all contents k.

$$T^v = \sum_{k \in C} T_k^v \tag{12}$$

C. Availability

Finally, the content availability (i.e., the probability that an entity can successfully obtain the requested content) in CCN is derived.

Since the router caches contents in CCN, it is possible to obtain a content as long as all links to the router caching the content is functioning properly even when other links in the network temporarily fail.

The availability for content k at node v is expressed as A_k^v. That is, A_k^v is the probability that the Data packet corresponding to an Interest packet can be obtained properly when the Interest packet is sent from node v to request for content k.

If a cache is hit at the n-th node on the shortest path P_k^v from node v to repository s_k which maintains content k, the content can be properly obtained if the $2n$ links on the path are properly functioning. Since the cache hit probability at the n-th node is $q_{k,P_k^v[n]}$ and the failure rate for each link is ϕ equally, we have

$$A_k^v = \sum_{n=1}^{|P_k^v|} \left(\eta_{k,n}^v (1-\phi)^{2(n-1)} \right). \tag{13}$$

Since the rate of arrival for Interest packets received by node v from entities directly connected below for content k is $\lambda_{k,v}$, the availability A^v for all contents at node v is obtained as

$$A^v = \sum_{k \in C} \frac{\lambda_{k,v}}{\sum_{k' \in C} \lambda_{k',v}} A_k^v. \tag{14}$$

IV. Numerical examples

First, content delivery delay for content k at router v (Eq. (6)) in a linear network topology, in which five routers and one repository are connected in serial (see Fig. 2(a)), is shown in Fig. 2(b). Repository (node 6) stores 500 of contents $C = \{1, \ldots, 500\}$. The arrival rate of Interest packets for content k at router $v\,(1 \leq v \leq 5)$ from directly-connected entities, $\lambda_{k,v}$, is given by a Zipf distribution with the mean of 20 [request/s] and the exponent parameter of 1.0. Therefore, the arrival rate of Interest packets at a specific router for every content is heavy-tailed. For instance, content 1 is the least popular content, and content 500 is the most popular content. Furthermore, the content store size B_v is set equally to 50 [content] for all routers, and communication delay $\tau_{u,v}$ equally to 1 [ms] for all links. Link failure rates at all links are equally set to $\phi = 0$ unless stated otherwise. The packet size S of Data packets is 8 [Kbyte], and the bandwidth $\mu_{v,v'}$ between nodes (i.e., routers and repositories) is equally set to 100 [Mbit/s]. The bandwidth between an entity and its neighbor router is infinity; i.e., links at network edges never become the performance bottleneck.

One can find from Fig. 2(b) that content delivery delay becomes smaller as it is more frequently accessed (i.e., k is larger). It can also be found that the content delivery delay is larger for routers farther away from the repository (i.e., smaller v). There are five hops from the router on the left end (node 1) to repository (node 6), and the content delivery delay when there is no content caching (when it is directly obtained from the repository) is $1 \times 5 \times 2 = 10$ [ms]. From Fig. 2(b), it is found that the content delivery delay is about 9 [ms] at maximum for $k = 1$ and nearly zero at minimum for $k = 500$ as content caching is done.

Second, throughput for content k at router v, T_k^v, is shown in Fig. 2(c). Note that the y-axis is plotted in a logarithmic scale. This figure shows that the throughput is significantly different for every content since the arrival rate of Interest packets is given by a Zipf distribution in our numerical examples. The throughput for popular contents (i.e., k is larger) exceeds 10 [Mbit/s], but that for unpopular contents (i.e., k is smaller) is less than 0.1 [Mbit/s]. One can find from this figure that, different from the content delivery delay in Fig. 2(b), routers far away from the repository (i.e., smaller v) achieves *higher* throughput than those close to the repository (i.e., larger v). This phenomenon can be explained by *filtering effect* in a multi-stage caching. Namely, in a multi-stage caching, popular contents are likely to be hit at an earlier stage. Hence, popular contents are less likely to be accessed at a later stage. The link bandwidth at a later stage is competed with requests for different contents. However, because of the filtering effect, popular contents are less likely to be accessed so that requests for unpopular contents are likely to gain higher throughput.

The availability for content k at router v (Eq. (13)) when link failure rate is set to $\phi = 0.1$ is shown in Fig. 2(d). This figure shows that the content availability is improved dramatically by content caching. Since the link failure rate between router 5 and repository (node 6) is also ϕ, the availability for router 5 is $(1-\phi)^2 = 0.81$. As it is shown from Fig. 2(d), availability exceeds 0.5 except for contents with low popularity even for routers that are far away from the repository (such as routers 1 and 2). It is possible to obtain the content in CCN if one of the routers on the path has cached the content (and all links to the router are functioning properly).

Next, content delivery delay for content k at router v in a rather simple network topology shown in Fig. 3(a) where five routers and two repositories are connected, is shown in Fig. 3(b). Three entities are connected to router 1, router 2 and router 3, respectively. Similarly to Fig. 2(b), there are 500 contents $C = \{1, \ldots, 500\}$ in the network. One repository (node 6) has contents $1, \ldots, 250$ and the other (node 7) has contents $251, \ldots, 500$. The other conditions are the same as Fig. 2(b). Hereafter, the network topology shown in Fig. 3(a) is called *simple network topology*. Note that in Fig. 3(b), content delivery delays at router 2 and router 3 are indistinguishable.

Figure 3(b) shows that the content delivery delay is smaller as the requesting router is closer to the repository holding the content. Such a small content delivery delay is caused by higher cache hit rates at routers around the repository, as well as the smaller number of hops from the requesting router to the corresponding repository.

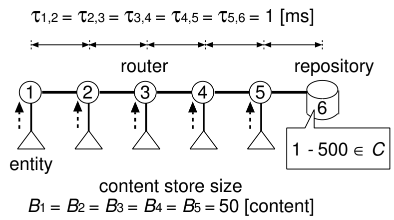
Throughput for content k at router v, T_v^k, in the simple network topology is shown in Fig. 3(c). Again, this figure shows that the throughput is significantly different for every content. Namely, throughput for popular contents (i.e., large k) is quite large although the throughput for unpopular contents (i.e., small k) is very small. However, such a difference is caused by the difference in Zipf-distributed request rates. A notable difference from Fig. 2(c) is that throughputs at routers 1, 2, and 3 are almost the same in Fig. 3(c) even for unpopular contents. This phenomenon can also be explained by (non-existence of) filtering effect in a multi-stage caching. The number of hops in the simple network topology is either 2 or 3 so that strong filtering effect is not likely to happen. Thus, unpopular contents are not likely to benefit higher throughput caused by the filtering effect.

Availability for content k at router v in the simple network topology is shown in Fig. 3(d). The link failure rate is set to $\phi = 0.1$ similarly to Fig. 2(d). This figure shows that the availability for the corresponding content is higher as the router is closer to the repository that has the content just as it was shown in results for content delivery delay.

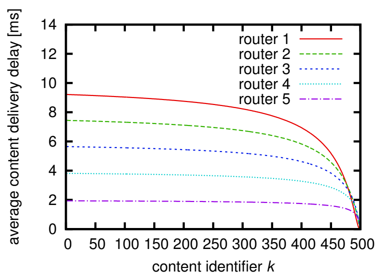
From these observations, we conclude that the benefit of performance improvement by content caching in terms of content delivery delay and availability is higher as an entity is *closer* to the repository. On the contrary, the benefit in terms of throughput is the opposite; i.e., an entity *farther* from the repository gains higher throughput.

Finally, to demonstrate the usability of our analysis, we examine the performance of CCN on a real network topology — the Abilene network topology [11] shown in Fig. 4(a). For demonstration purposes, a single repository (node 12) with 500 contents is connected to router 5. A single entity is connected to all routers. Other conditions are the same with those in cases of the linear network topology and the simple network topology . We should note that our analysis has no limitation on the number of repositories in the network and the heterogeneity in arrival rates of Interest packets at routers. We used a simple scenario since a complicated scenario makes interpretation of numerical examples rather difficult.

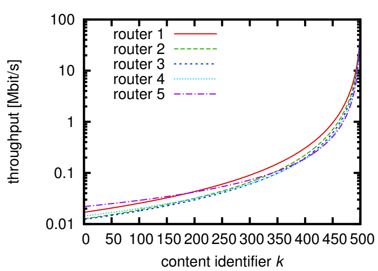
Content delivery delay, throughput, and availability for content k at router v are shown in Figs. 4(b), 4(c), and 4(d), respectively. Because of space limitation, these figures show



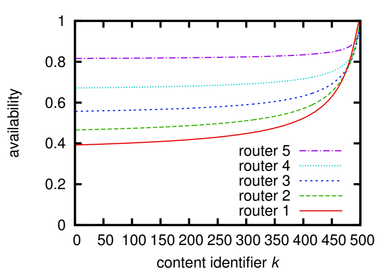
(a) five routers and one repository are connected in series



(b) content delivery delay

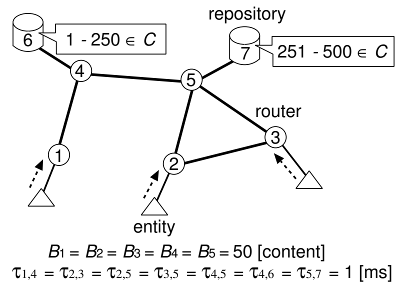


(c) throughput

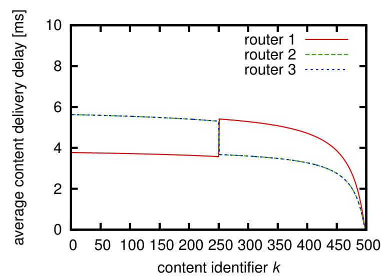


(d) availability

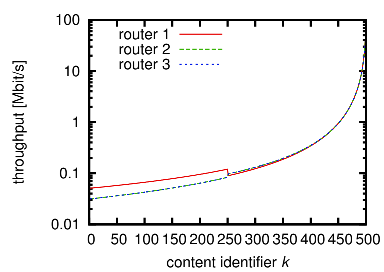
Fig. 2. Linear network topology



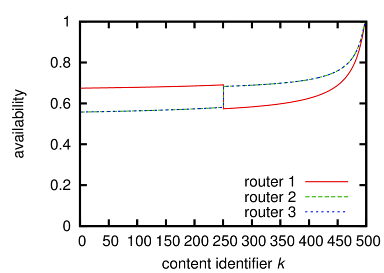
(a) five routers and two repositories are connected, with each of the two repositories keeping 250 contents



(b) content delivery delay

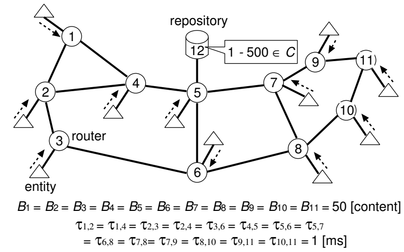


(c) throughput

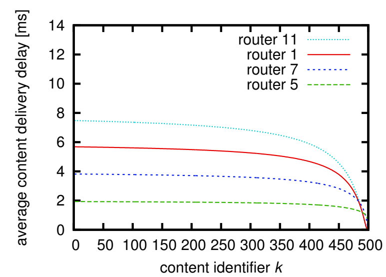


(d) availability

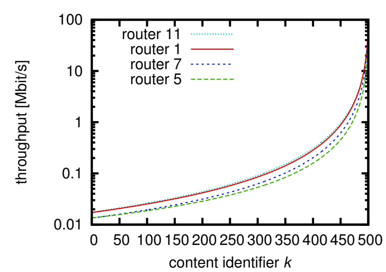
Fig. 3. Simple network topology



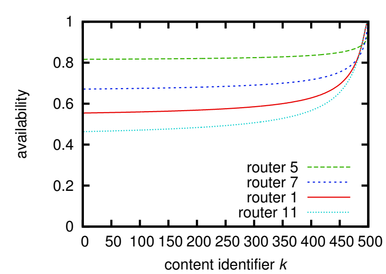
(a) Abilene network topology



(b) content delivery delay



(c) throughput



(d) availability

Fig. 4. A network topology

results only for routers 1, 5, 7 and 11. The results at routers 3 and 9 are almost the same in those at router 1. In these figures, we can see similar tendencies to those observed in the linear network topology and the simple network topology. However, because of the complexity in the network topology, Figs. 4(b), 4(c), and 4(d) exhibit more complex patterns, which imply that the performance of CCN is heavily dependent on the network topology and that performance analyst should explicitly take account of the network topology to be studied.

V. Validation

Finally, the validity of our approximate analysis is examined by comparing analytic results with simulation ones.

We developed a chunk-level CCN simulator written in Perl language, and measured content delivery delay, throughput, and availability in the linear network topology shown in Fig. 2(a). Parameters are the same with those used in Section IV. Interest packets were randomly generated from entities at a specified rate, $\lambda_{k,v}$. The queueing discipline at all routers was FIFO (First-In and First-Out). The cache replacement algorithm at all routers was LRU (Least-Recently Used). Every

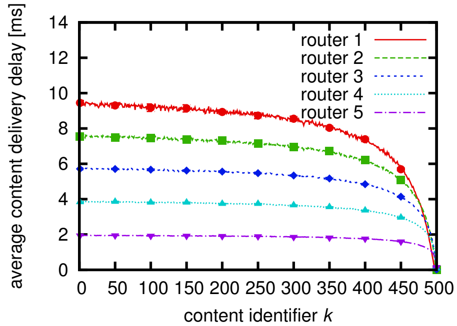


Fig. 5. Simulation results of content delivery delay for content k at router v in linear network topology

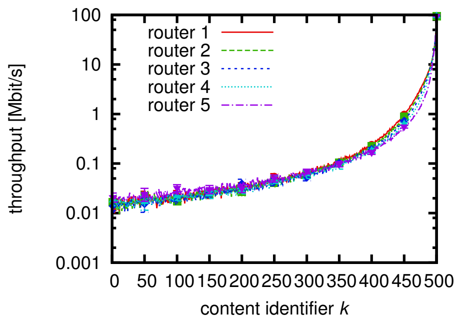


Fig. 6. Simulation results of throughput for content k at router v in linear network topology

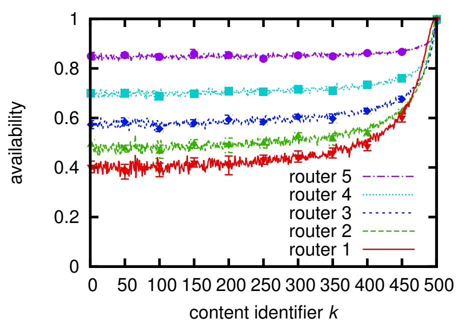


Fig. 7. Simulation results of availability for content k at router v in linear network topology

simulation run was lasted for 30 [s]. For a single parameter setting, simulations were repeated 10 times, and the average and the 95% confidence interval of all measurements were obtained. For better readability, 95% confidence intervals are shown sparsely (i.e., every 50 contents) in the following figures.

Note that we intentionally used the linear network topology rather than, for instance, the simple network topology and the Abilene network topology. The linear network topology is the simple but it is one of the most difficult network topologies to model because of its cascaded router connectivity. For instance, the performance of an entity connected at router 1 is affected by all factors such as the request pattern, content store size, bandwidth, and availability of all downstream routers and links.

Simulation results of content delivery delay, throughput, and availability are shown in Figs. 5 through 7. These figures show good agreements between analytic and simulation results in content delivery delay, throughput, and availability, which clearly show the validity of our analysis even in a cascaded network topology.

VI. Conclusion

In this paper, we have analyzed the performance of CCN on an arbitrary network topology by utilizing the MCA algorithm, which is an approximation algorithm to analytically calculate cache hit rates in a multi-cache network. The content delivery delay, throughput, and availability in a network comprising of multiple routers and multiple repositories have been analytically calculated. Through several numerical examples, we have shown that the benefits of performance improvement by content caching (i.e., reduction in content delivery delay and improvement in availability) were higher as the router was closer to the repository in CCN. In addition, we have shown the validity of our analysis through simulation experiments.

Our future challenges include verification of the effectiveness of analysis on large-scale networks with large numbers of routers, repositories, and entities, reduction of computation complexity using an approximation algorithm, and analysis of situations in which network topology is not known (such as when only the distribution of router and repository degrees are given).

Acknowledgments

We would like to thank Mr. Yoshiaki Noro for valuable discussion on the analysis in this paper.

This work was supported by JSPS KAKENHI Grant Number 25280030.

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 Fifth International Conference on emerging Networking EXperiments and Technologies (CoNEXT 2009)*, pp. 1–12, Dec. 2009.

[2] G. Rossini and D. Rossi, "A dive into the caching performance of content centric networking," in *Proceedings of the IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD 2012)*, pp. 105–109, Sept. 2012.

[3] M. R. Butt, O. Delgado, and M. Coates, "An energy-efficiency assessment of content centric networking (CCN)," in *Proceedings of the 25th IEEE Canadian Conference on Electrical & Computer Engineering (CCECE 2012)*, pp. 1–4, Apr. 2012.

[4] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou, "Modelling and evaluation of CCN-caching trees," in *Proceedings of the 10th IFIP-TC6 Networking Conference (NETWORKING 2011)*, pp. 78–91, May 2011.

[5] M. Dehghan, B. Jiang, A. Dabirmoghaddam, and D. Towsley, "On the analysis of caches with pending interest tables," in *Proceedings of the 2nd ACM Conference on Information-Centric Networking (ICN 2015)*, pp. 69–78, Sept. 2015.

[6] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *Proceedings of the 2011 23rd International Teletraffic Congress (ITC 2011)*, pp. 111–118, Sept. 2011.

[7] A. Udugama, S. Palipana, and C. Goerg, "Analytical characterisation of multi-path content delivery in content centric networks," in *Proceedings of the 2013 Conference on Future Internet Communications (CFIC)*, pp. 1–7, May 2013.

[8] W. Guoqing, H. Tao, L. Jiang, C. Jianya, and L. Yunjie, "Approximate models for CCN data transfer in general topology," *China Communications*, vol. 11, pp. 40–47, July 2014.

[9] E. J. Rosensweig, J. Kurose, and D. Towsley, "Approximate models for general cache networks," in *Proceedings of the 29th Conference on Information Communications (INFOCOM 2010)*, pp. 1–9, Mar. 2010.

[10] A. Dant and D. Towsley, "An approximate analysis of the LRU and FIFO buffer replacement schemes," in *Proceedings of ACM/SIGMETRICS 1990*, pp. 143–152, May 1990.

[11] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE Journal of Selected Areas in Comuunications*, vol. 29, pp. 1765–1775, Oct. 2011.