In-Network Guide Performance in Wireless Multi-Hop Cache Networks

Kento Ikkaku, Yukio Sakaguchi, and Miki Yamamoto
Faculty of Engineering Science,
Kansai University
Email: {k439232, yama-m}@kansai-u.ac.jp

***Abstract*—In wireless multi-hop networks, content servers are generally located outside a wireless multi-hop network and a user accesses these servers through a gateway node. So, content traffic has a tendency to be concentrated at a gateway node, which might cause throughput degradation in a whole wireless multi-hop network. Caching network is one promising way to resolve this technical problem. However, caching network has a limitation of cache availability on a default-path. For wired networks, Breadcrumbs has been proposed as in-network guide for cached contents. In this paper, we evaluate Breadcrumbs in wireless multi-hop networks. Our simulation results show that Breadcrumbs improves total throughput performance of wireless multi-hop networks. And also they surprisingly show that not only popular content throughput but also unpopular content throughput is improved.**

I. Introduction

In recent years, wireless multi-hop networks have attracted much attention as a flexible wireless access technology [1]. In wireless multi-hop networks, nodes can communicate with other nodes located outside their radio communication range by transmitting data through multiple nodes. When a client in a wireless multi-hop network would like to obtain a content, it should access a corresponding content server, generally located outside a wireless multi-hop network, through a gateway node (GW). All inbound and outbound traffic concentrates at GW, and thus heavy radio interference around GW causes frame loss at layer 2. In IEEE 802.11, a frame is discarded when its retransmission successively fails[1]. This frame loss at layer 2 poses recovery process towards link failure in layer 3. Frequently happened route recovery processes generate heavy signaling traffic, which causes significant total throughput degradation in a wireless multi-hop network [2].

One of the promising ways to resolve this technical issue of traffic concentration around GW is "cache network" [3], [4]. In wireless cache networks, each node in a wireless multi-hop network is equipped with cache storage and stores content files going through it according its caching policy, LRU (Least Recently Used) is generally applied. In cache networks, popular contents have a tendency to be stored in cache storage. When a content request encounters cached content on the way to the GW, content download is launched from this encountered node. So, wireless cache networks are expected to reduce concentration of content request and content traffic of popular contents around GW.

[1]Generally, 7 successive failures of retransmission cause frame loss

Cache networks, not only wired network but also wireless network, have a limitation only along default-path, i.e. a content request can find cached content only on the way to a content server (GW in a wireless network). One simple way to make cached contents not along default-path available is explicit cache coordination [5]. However, it requires exchange of control packets for cache coordination and this overhead might cause degradation of total throughput in a wireless multi-hop network. The other promising approach is implicit cache coordination. Breadcrumbs (BC) [6] has been proposed as an implicit cache coordination approach for a wired network.

In Breadcrumbs, a content request is routed initially towards a content server by location-ID, e.g. IP address. When a content request arrives a content server and a content file is downloaded, each router along the download path stores a pointer, called Breadcrumbs. This pointer points to the direction in which the content was sent. When a content request eventually encounters Breadcrumbs for the corresponding content along the default path, it is redirected towards Breadcrumbs direction. The search for content by Breadcrumbs pointer continues, until content is found or a content request reaches a node having no breadcrumbs (this invokes invalidation of Breadcrumbs trail).

The paper evaluates Breadcrumbs in wireless cache networks. When Breadcrumbs is applied for a wireless multi-hop network, breadcrumbs pointer might have a tendency to direct outbound from GW because content download is generally originated from GW. Our simulation results show that Breadcrumbs brings significant throughput improvement when compared with a cache network because outbound redirection of content requests distributes content traffic towards regions away from GW. Our simulation results also show that throughput of low-popular contents can be surprisingly improved as well because reduction of content traffic around GW stimulates content download of low-popular contents from the GW.

The rest of this paper is organized as follows. In Section 2, we explain contents distribution in wireless multi-hop networks. Section 3 discusses the BC technology and how it is applied to wireless multi-hop cache networks. In Section 4, the performance of wireless multi-hop cache networks with BCs. Finally, we conclude the paper in Section 5.

II. Content Distribution in the Wireless Multi-Hop Network

A. Concentrated Traffic at the GW

In wireless multi-hop networks, the hidden terminal problem degrades the overall throughput performance. The hidden terminal can be resolved by the RTS/CTS (request to send/clear to send) mechanism in IEEE 802.11. The exposed terminal problem is also another technical issue for degradation of the overall throughput. We evaluated the overall throughput of wireless multi-hop networks under uniformly distributed traffic and showed that the overall throughput increases with the offered load until a certain level and then moderately decreases [7]. Decrease of the overall throughput after it reaches the largest value, is due to overhead traffic for link and route recovery phase in AODV (ad hoc on-demand distance vector).

Under our focused situation, i.e. content distribution in wireless multi-hop networks, content request traffic and their response traffic of content file transfer concentrates at the GW. This heterogeneous traffic condition degrades overall throughput performance more significantly.

B. Cache Approach for Wireless Multi-hop Networks

One of the promising solutions for concentrated traffic at the GW is cache network. In wireless multi-hop cash networks, content file is temporally cached at each node along the download path as in wired cache networks. When a content request eventually encounters cached content on the way to the content server through the GW, content is downloaded from the cache.

Cache networks generally have a limitation that only the caches on the way to the server can be searched. In a wireless multi-hop network, only the caches on the way to the GW, i.e. caches on the default-path, are searched. In cache networks, nodes close to the content server have a tendency to store more contents than other nodes [8]. This means contents stored at nodes around the content servers are frequently replaced, i.e. their lifetime is generally rather short. At these caches, not only popular contents but also unpopular contents are temporally stored. Replacement caused by storing these unpopular content might throw out popular contents. This is the case for a wireless multi-hop network. In a wireless multi-hop network, nodes around the GW have a tendency to store more contents and their lifetime is rather short. Therefore, the closer to the GW a content request reaches, the smaller its cache hit probability is. Lee et.al [8] also shows that the further a node is located from the content server (the GW in a wireless multi-hop network), it has the higher cache hit probability. In order to enhance the overall throughput of cache networks, a new approach which efficiently and actively utilizes caches based on these reported characteristics of cached contents is required.

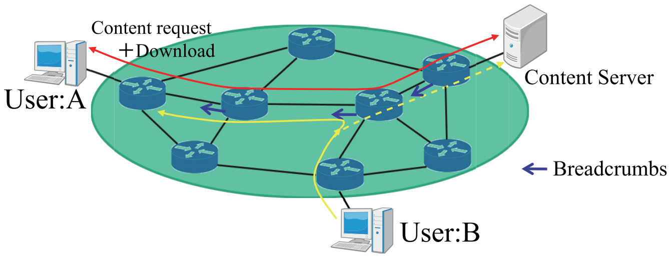

Fig. 1: Overview of Breadcrumbs.

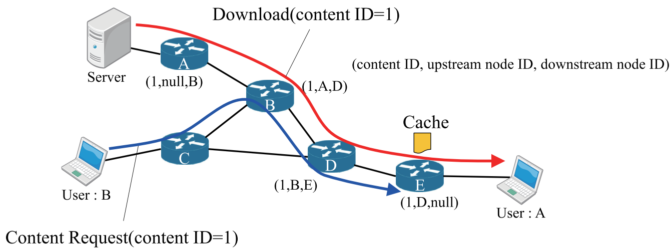

Fig. 2: Behavior of the Guidance by Breadcrumbs

III. Breadcrumbs

A. Overview of Breadcrumbs

The BC aims at efficiently and actively utilizing content caches located outside of the default path. Fig. 1 shows an overview of BC behavior. When user A downloads a content from a server, a copy of the content is stored in the cache of each intermediate node (router) on the route. Furthermore, each intermediate node stores a pointer indicating the download direction called BC. When user B sends a content request to the content server, this content request is initially routed by its location-based ID, e.g. IP address. When it eventually encounters one of the BCs, it is routed by its content ID, i.e. a content request is forwarded towards a pointer direction of BC. And it is transmitted in a hop-by-hop manner by its content ID. When the content request finds the cached content along the BC trail forwarding, content file is downloaded to user B from the cache. This cache locates outside of the default path for user B, so BC enables efficient use of network caches.

Figure 2 shows detailed behavior of Breadcrumbs. Content download of content 1 by user A leaves a pointer, BC, at each router along the path (router A, B, D and E). The CN of each node consists of the following 5 elements:

- content ID,
- upstream node ID,
- downstream node ID,
- the latest time stamp when the content is transmitted to the node, and
- the latest time stamp when the node receives the query for the content.

Due to space limitation, BC pointer in this example is just depicted 3-tuple; content ID, upstream node ID and downstream node ID. A content request for content 1 sent from user B is initially routed towards the content server by location-based ID. And, in this example, this content request encounters the corresponding BC, i.e. BC for content 1, at router B. After that, each router forwards a content request by content-ID based forwarding. For example, at router B, a content request is forwarded to router D as its BC indicates router D for downstream node.

Figure 2 shows successful example for BC trail. However, BC trail has possibility to be failed. This failure of BC trail happens when cached contents on BC trail eventually are replaced and removed. This invalid BC trail causes inefficient content request forwarding. So, BC has the following two mechanisms for removing BCs, time out and trail invalidation.

- Time Out
  In cases where there are no accesses directed to downstream nodes in a certain period, intended contents are not likely to be stored in downstream nodes. Time Out is used to avoid such cases. For each BC, each node manages Time Out by using latest time stamp T_f when a content is transmitted to the node and the latest time stamp T_q when the node receives the query for the content. For time stamps T_f and T_q, we prepare lifetimes (threshold) L_f and L_q, respectively. If the current time exceeds $T_f + L_f$ (resp. $T_q + L_q$), the BC is removed from the node.
- Trail Invalidation
  Trail Invalidation removes a sequence of BCs along the direction opposite to the direction guided by the BCs (i.e., upstream direction). Trail Invalidation occurs when queries directed by a BC do not find its intended content on a new route. In this case, the starting point is the node at the end of the sequence of BCs along the new route. Also, Trail Invalidation occurs when a node receives a query from a downstream node. This is because downstream nodes do not likely to store the intended content. In this case, the starting point is the node receiving the query.

B. BC in wireless multi-hop cache networks

In this paper, we apply BC to wireless multi-hop cache networks. Although BC is originally proposed for wired networks, it can work in wireless multi-hop cache networks because it basically uses IP routing based on location IDs. Note that after a content request finds a BC on its route, it is transmitted in hop-by-hop manner based on content IDs. In wireless multi-hop networks, the content request can be transmitted to the next node by specifying the IP address of the next node based on information in the BC pointer. Therefore, BC can be easily applied to wireless multi-hop cache networks.

In wireless multi-hop cache networks with BCs, download of content file from content servers is generally originated from the GW. Direction of BC pointer is along download direction, so it basically indicates a direction of going away from the GW. Therefore, when content requests sent by clients are guided by BCs, they are generally forwarded in the direction outbound of the GW. When content requests on the BC trail occasionally find the corresponding contents, its download is routed by location ID. When AODV is applied for layer 3 routing protocol, content files are transmitted along the shortest paths (exactly, they might not be the shortest path, but they are close to the shortest one). Therefore, introduction of BCs is expected to distribute lots of downloads away from GWs and reduce traffic concentration around the GW. In this paper, we evaluate these effects of BC in detail.

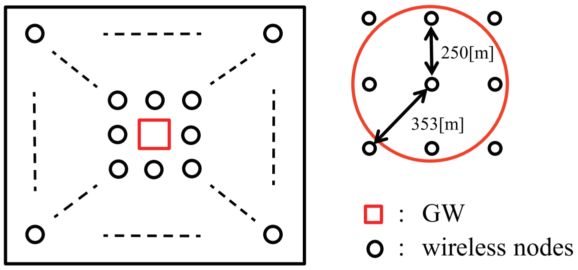

Fig. 3: Network model.

TABLE I: System parameters.

Parameter	Value
Field size	1500 × 1500 [m^2]
Number of wireless nodes	80
Number of contents	300
Packet size	512 [byte]
Content size	10 [packets]
Cache size	3 [contents]
Effective communication range	300[m]

IV. Performance Evaluation of Wireless Multi-Hop Cache Networks with Breadcrumbs

To evaluate the performance of wireless multi-hop cache networks with BCs, we conduct simulation experiments. In this evaluation, we use AODV as a routing protocol.

A. Simulation model

We use a static network model shown in Fig. 3 in which wireless nodes do not move in order to evaluate the basic characteristics of introduction of BCs to wireless multi-hop cache networks. The wireless nodes are arranged to form a grid and the GW is located at the center of the network. The distance between any two adjacent wireless nodes is 250 [m] and the communication range of each wireless node is 300 [m]. Thus, each wireless node can communicate with four wireless nodes as shown in Fig. 3. We use a QualNet 3.9.5 [9] as a simulation tool. We assume that each wireless node generates queries for contents whose destination is the GW according to a Poisson process. The popularity of the contents follows Zipf distribution [10]. The total number of content requests generated in each simulation trial is set to be 10000. The system parameters are listed in Table I.

In the simulation experiments, we compare caching by BCs (we refer to it as Cache(BC), hereafter) with the following two methods. The first method does not use caching. Each wireless node always sends content requests to the GW and all contents are downloaded from the GW. The second method uses caching, but does not use BCs. Specifically, it only

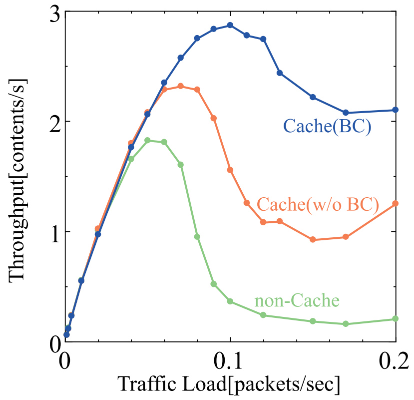

Fig. 4: Overall throughput.

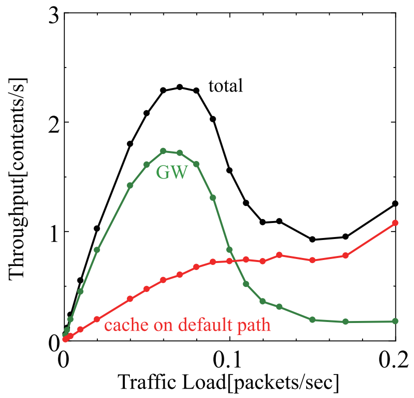

Fig. 5: Throughput properties in Cache(w/o BC).

uses caches on default-paths. We refer to the first scheme and the second scheme as non-Cache and Cache(w/o BC), respectively.

B. Overall throughput characteristics

Fig. 4 shows the overall throughput as a function of the traffic load. The overall throughput is defined as the ratio of the total number of contents downloaded by all wireless nodes to time spent downloading all contents in the simulation trial. Traffic load is defined as generation rate of content requests at each wireless node. In simulation experiments, we do not consider background traffic. Specifically, traffic transmitted in the network are only content requests and contents. As shown in Fig. 4, the overall throughput of non-Cache significantly decreases after the traffic load exceeds 0.06. Although the similar result was obtained in [7] which evaluates the throughput of wireless multi-hop networks where traffic is distributed uniformly, the throughput degradation in this paper is much larger than in [7]. This is because in the network model in this paper, traffic concentrates at the GW. Our simulation results also show that the overall throughput of Cache(BC) and Cache(w/o BC) is larger than non-Cache because traffic is distributed by using cashes in wireless nodes. Note that Cache(BC) improve the overall throughput more efficiently than Cache(w/o BC).

We then examine the reason of this difference. Figs. 5 and 6 show the details of throughput performance of Cache(w/o BC) and Cache(BC), respectively, as a function of the traffic load. In these figure, we plot the throughput of contents downloaded from the GW (labeled with “GW”). Also, we plot the throughput of contents downloaded from caches of wireless nodes on default paths (labeled with “cache on default-path”) and the throughput of contents downloaded from caches of wireless nodes with BCs (labeled with “cache on BCs”). Performance curve labeled “total” shows the overall throughput.

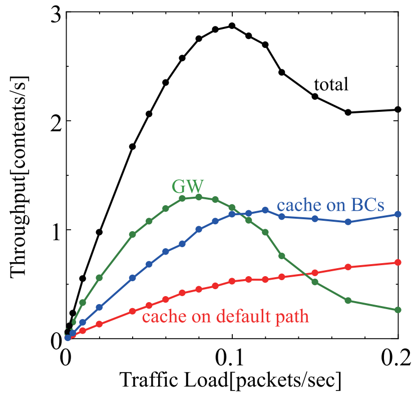

Fig. 6: Throughput properties in Cache(BC).

As we can see from Fig. 5, the throughput of contents downloaded from the GW in Cache(w/o BC) is almost the same as the overall throughput in non-Cache shown in Fig. 4. Meanwhile, the throughput of contents downloaded from caches of wireless nodes on default paths increases with the input load. Therefore, the overall throughput in Cache(w/o BC) is larger than that in non-Cache. These results simply show that caching is effective for wireless multi-hop networks.

As shown in Fig. 6, the throughput of contents downloaded form the GW in Cache(BC) is smaller than that in non-Cache and Cache(w/o BC). This is due to the increase in content downloads from caches and thus queries arriving at the GW is just reduced. The throughput of content downloads from the cache on BC trail is especially high. Therefore, the overall throughput in Cache(BC) is also improved significantly. In the following two subsections, we discuss the reason for the throughput improvement in detail.

C. Impact of the popularity of contents

Figs. 7 and 8 show the number of downloads for each content which is downloaded from caches and the GW, respectively, as a function of the content ID, where the input traffic load is 0.1. Also Fig. 9 shows the total number of downloads for each content as a function of the ID of the content, where the input load is 0.1. Note that the content IDs are sorted in descending order of popularity. Specifically, the content with ID 1 has the highest popularity. In Figs. 7, 8 and 9, we only show characteristics for top 100 popular contents (in simulation experiments, 10000 contents are generated).

As shown in Fig. 7, content download from caches in BC is improved over the whole content ID when compared with

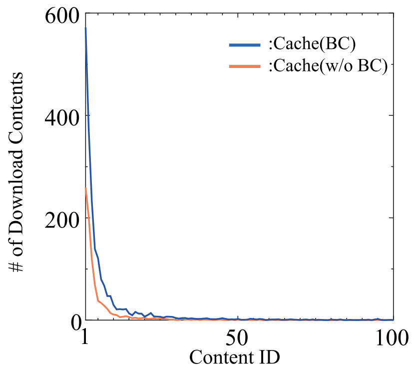

Fig. 7: Number of contents downloaded from Caches.

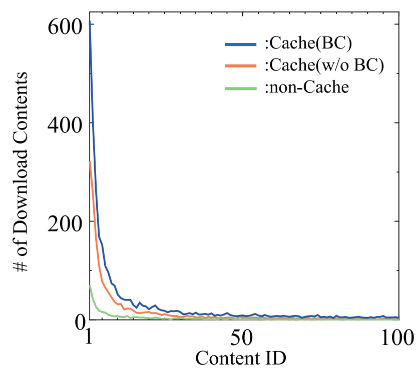

Fig. 9: Total number of downloaded contents.

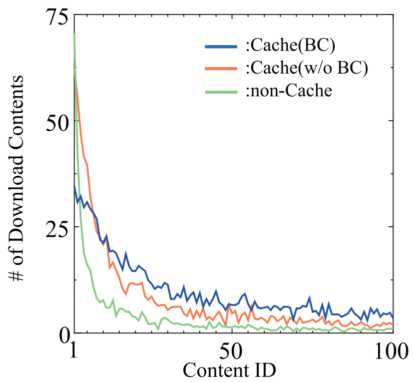

Fig. 8: Number of contents downloaded from the GW.

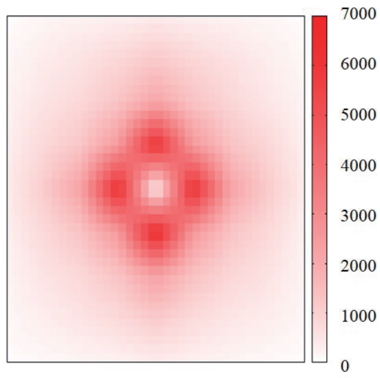

Fig. 10: Traffic distribution in non-Cache.

Cache only (Cache without BC). Especially, great improvement is obtained for high popular contents. For example, the number of downloads from caches in BC is almost twice as large as that in Cache only. As explained in Section III-B, frequent content replacements reduces cache hit ratio of popular contents at nodes around the GW. And nodes located away from the GW can obtain high cache hit ratio for popular content. Thus, BC which guides content requests to distant region from the GW, can obtain high cache hit ratio for popular contents. This is the reason for great improvement of BC shown in Fig. 7.

Simulation results in Fig. 8 show that download from the GW in BC is smaller than other approaches. This is due to filtering effect of caches, i.e. improved cache hit rate for popular contents consequently decreases content requests arrived at the GW. Fig.8 also shows that downloads for non-popular contents in BC are larger than other approaches. In BC, popular contents have a tendency to be transmitted in distant region from the GW and thus traffic concentration towards the GW is significantly improved. This traffic distribution creates room of content transmission of non-popular contents from the GW.

These two improvements brought by BC, download from caches for popular contents and download from the GW for non-popular contents, achieve improvement of content download in the whole range of popularity (Fig. 9).

D. Spatial Traffic Distribution

In order to confirm that downloads of popular contents are distributed to wireless nodes located away from the GW by BCs, we visualize the distribution of the number of packets passing through each wireless node. Figs. 10, 11, and 12 shows the distribution in non-Cache, Cache(w/o BC), and Cache(BC), respectively. In these figures, high densities mean that the number of packets is large. As shown in Figs. 10 and 11, there is much traffic around the GW in non-Cache and Cache(w/o BC). On the other hand, in Cache(BC), densities around the GW are low and densities of areas away from the GW are rather high compared with non-Cache and Cache(w/o BC). These results confirm that BC distributes content traffic to distant region from the GW.

Table II shows the number of hops for content request and content download of non-Cache, Cache(w/o BC) and Cache(BC). In these results, Cache(w/o BC) has larger content download and this seems a little strange because cache download on the default path reduces download path length. This is because in non-Cache content requests generated at a node away from the GW can hardly arrive the GW due to heavy traffic concentration around the GW. So, in non-Cache only short length download is available. Even though BC requires redirection of content request on BC trail, the number of content request hops is rather short than non-Cache. So, BC trail length is not so long in a wireless multi-hop network.

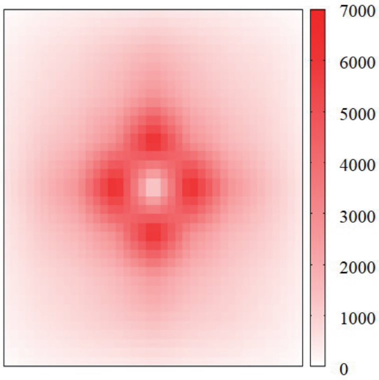

Fig. 11: Traffic distribution in Cache(w/o BC).

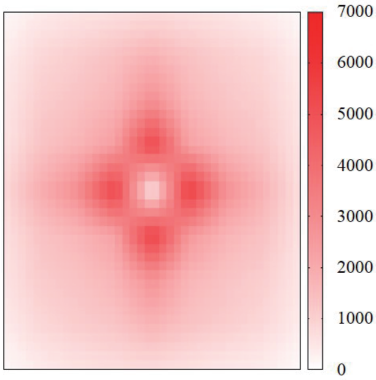

Fig. 12: Traffic distribution in Cache(BC).

Content download path of BC does not increase so much, which means content request redirection caused by BC does not increase total network load(with increase of the number of hops for packet transmission, network load to wireless multi-hop networks increases even when traffic generation rate is fixed. This is because increase of transmission hops incurs radio interference.). So, we conclude that BC efficiently distribute traffic to the regions away from the GW and this distribution does not cause any harmful interference to other traffic.

V. Conclusions

In this paper, we evaluate wireless multi-hop cache networks equipped with Breadcrumbs which has been originally proposed for wired networks. Our simulation results for Breadcrumbs in wireless multi-hop networks show that Breadcrumbs brings great improvement for the overall throughput performance. Cache network also brings performance improvement, but Breadcrumbs improves more significantly in a sophisticated fashion. Our detailed investigation for performance improvement brought by Breadcrumbs clarifies that Breadcrumbs can improve throughput of not only popular contents but also non-popular contents. The reason for throughput improvement for popular contents is high cache hit ratio of content requests guided by Breadcrumbs. Breadcrumbs trail guides content requests towards the region distant from the gateway and cache hit ratio of popular contents in this region is generally higher than in the region close to the gateway. Low hit ratio of popular contents around the gateway is caused by frequent replacement even for popular contents. The reason for throughput improvement for non-popular contents is that high hit ratio of popular content reduces traffic volume around the gateway, which creates room for download of non-popular content from the gateway. From these interesting insights, Breadcrumbs is confirmed to be the promising way for throughput improvement of content traffic in wireless multi-hop networks.

TABLE II:
Number of Hops for Content Request and Download.

	non-Cache	Cache(w/o BC)	Cache(BC)
Content Request	4.632	3.768	4.356
Content Download	2.283	2.537	2.843

Acknowledgment

This research is partly supported by the National Institute of Information and Communications Technology, Japan. The authors thanks for Prof.Koji Hirata at Kansai University for his useful comments.

References

[1] I. Akyildiz, X. Wang, and W. Wang, “Wireless Mesh Networks: A Survey,” *Computer Networks*, vol. 47, no. 4, pp. 445–487, 2005.
[2] K. Tada, and M. Yamamoto, “Load-balancing Gateway Selection Methodin Multi-hop Wireless Networks,” in *Proc. IEEE Globecom*, Honolulu, HI, Nov. 2009.
[3] L Yin and G. Cao, “Supporting Cooperative Caching in Ad Hoc Networks,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 1, pp. 77–89, 2006.
[4] W. Li, E. Chan, and D. Chen, “Energy-Efficient Cache Replacement Policies for Cooperative Caching in Mobile Ad Hoc Network,” in *Proc. IEEE WCNC 2007*, Hong Kong, Mar. 2007, pp.3347–3352,
[5] H. Artail, H. Safa, K. Mershad, Z. Abou-Atme, and N. Sulieman, “COACS: A Cooperative and Adaptive Caching System for MANETs,” *IEEE Trans. on Mobile Computing*, vol. 7, no. 8, pp. 961–977. 2008.
[6] E. J. Rosensweig and J. Kurose, “Breadcrumbs: Efficient, Best-effort Content Location in Cache Networks,” in *Proc. IEEE INFOCOM* 2009.
[7] K. Yamamoto and M. Yamamoto, “Performance Improvement of Ad Hoc Networks by Deployment of Directional Antenna,” in *Proc. The Fourth International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, London, U.K., Oct. 2006.
[8] S. Lee, D. Kim, Y. Ko, J. Kim, and M. Jang, “Cache Capacity-aware CCN: Selective Caching and Cache-aware Routings,” in *Proc. IEEE Globecom 2013*, Atlanta, GA, Dec. 2013.
[9] QualNet, http://web.scalable-networks.com/content/qualnet
[10] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web Caching and Zipf-like Distributions: Evidence and Implications,” in *Proc. IEEE INFOCOM*, Mar. 1999, pp.126-134.
[11] K.Ikkaku, Y.Sakaguchi and M.Yamamoto, “Performance Evaluation of Breadcrumbs in Wireless Multi-Hop Cache Network,” IEICE Technical Report, NS2014-10, Apr. 2014(in Japanese).