Synergic Effects among Plural Extensions of Breadcrumbs for Contents Oriented Networks

Tomohiko Yagyu
Cloud System Research Laboratories
NEC Corporation
Kawasaki-shi Kanagawa, Japan
Email: yagyu@cp.jp.nec.com

***Abstract*—Data traffic in mobile networks are rapidly growing because of the diffusion of smartphones and rich applications. It is effective to cache contents data in networks to reduce traffic load. Breadcrumbs was proposed to discover cached contents efficiently. Each router records trace information of content download, which is called Breadcrumbs, and redirects request messages along the Breadcrumbs trail. Furthermore, several extensions of Breadcrumbs, e.g., Active BC, Hop-aware BC, MSCR, BC Scoping, have been proposed. In this paper, we evaluate mutual effects among those extensions when they are working in the networks at the same time. Evalution results revealed that combination of ABC and MSCR has the synergic effects regarding improvement of cache hit ratio.**

***Keywords*-Contents Oriented Network; In-network Guide; Breadcrumb; Cache Discovery; Synergic effect**

I. INTRODUCTION

Because of the diffusion of broadband access and smartphones, it becomes ordinary to consume rich contents such as video through the networks. According to the Cisco's report [1], a smartphone generates 24 times more traffic than a non-smartphone. It also reports that traffic in mobile networks will grow 6.3EB per month by 2015. It estimates that video traffic will account for 66% of total traffic. Following the expansion of contents distribution and M2M (machine to machine) communication, the number and sort of contents in the networks will be enormous. In order to avoid congestion of network due to contents flood, it is effective to cache popular contents in the network [2]. Paul et al. proposed Cache and Forward architecture [3], which the contents are cached by routers in the networks. Furthermore, the concept of Data Centric Networking are proposed [4][5]. It is convenient for users to be able to get desired contents without knowing their locations but with using content IDs. For the cache networks, because of the complication to manage all the locations of cached contents, it is desirable to route request messages to the nearest caching node with content IDs through the networks. Breadcrumbs [6] is proposed to efficiently discover cached contents in the cache networks. With Breadcrumbs, each router guides request messages to the cached contents with in-network guide information called Breadcrumbs (BC). In order to solve various drawbacks of Breadcrumbs, several extensions are proposed, e.g., Active BC (ABC) [7], Hop-aware BC (HBC) [8], BC Scoping [9] and Mapping Server with Cache-location Resolution (MSCR) [10]. Because these extensions are not exclusive each other, they can be used together. When using some of them together, a certain synergy or conflict will be supposed. In this paper, mutual effects among several extentions are evaluted with the prototype system implementing Breadcrumbs and its extensions. Evalution results revealed that combination of ABC and MSCR has good synergic effect regarding the cache hit ratio. It is also observed that some extensions can compensate loss of performance each other. The remainder of this paper is organized as follows. Section II briefly explains the basic scheme of Breadcrumbs. Section III explains several extensions of Breadcrumbs. Section IV shows environments and assumptions for the evaluation. In Section V, evaluation results are discussed regarding performance metrics and network traffic. Finally, Section VI concludes this paper.

II. CACHE DISCOVERY WITH BREADCRUMBS

Breadcrumbs is used in the cache networks comprising cache-capable routers. Routers can cache the forwarded contents and record the trail information called Breadcrumbs along the download path. And they can guide messages to request contents (query) along the BC trail to the cached contents. Cached contents are replaced based on the Least Recently Used (LRU) policy when cache storage in a router is exhausted. A BC entry includes following 5 information. It is assumed that each content has global unique ID.

- Content ID
- ID of node from which the content arrived (Upstream Node)
- ID of node to which the content was forwarded (Downstream Node)
- Most recent time the content passed through the node (Download Time)
- Most recent time the content was requested at the node (Request Time)

Fig.1 shows an example of the cache network. The network consists of a contents server S1, three routers R1˜R3 and two user nodes U1˜U2. The basic bahavior of

 ISBN: 978-1-61208-300-1

Breadcrums is explained with Fig.1. First, user U1 requests content X. The query message sent by U1 reaches S1 via R2 and R1 (dotted arrow in Fig.1 (a)). The contents server S1 replies the data of content X to U1 via R1 and R2 (solid arrow in Fig.1 (a)). While forwarding the content X, router R1 and R2 cache the data and record the direction of the download in BC. Upstream and Downstream nodes of BC entries in R1 and R2 are (S1, R2), (R1, U1) respectively. A router also records Download time and Request time in BC entry.

Suppose that user U2 requests same content after R1 deleted it from cache storage. When the query sent by U2 reaches R1, R1 redirects the query to R2 because R1 has no cache but the valid BC entry for the content X (dotted arrow in Fig.1 (b-1) and (b-2)). BC entry becomes valid only if the Download time is within T_f or the Request time is within T_q. If the recorded time exceed such thresholds, BC entry expires. When the query reaches R2, the cache of content X in R2 is hit. Then R2 starts sending the data to U2. There are two options for download path to the requesting node. One is DFS (Download Follows Query), which forwards the contents along the reverse path of the query(solid arrow in Fig.1 (b-1)). Another one is DFSP (Download Follows Shortest Path), which forwards the contents through the shortest path from the contents holder to the requesting node(solid arrow in Fig.1 (b-2)). When the content is downloaded from R2 to U2, R1 and R2 overwrite their BC entries and R3 creates new BC entry. Upstream and Downstream node of BC entries in R2, R1 and R3 become (R1,R1), (R2,R3), (R1,U2) in DFQ case, and (R1,R3), (S1,R2), (R2,U2) in DFSP case respectively. If no caches exist along the BC trail, the query is forwarded to S1 (dotted arrow in Fig.1 (c)). When R1 receives the query from R2, R1 can recognize that no cache exists in the direction of R2 anymore. Therefore R1 deletes the BC entry for content X. While the content is downloaded from S1 to U2 (solid arrow in Fig.1 (c)), R1 creates BC entry (S1,R3) and R3 creates BC entry (R1,U2). R1 and R3 also create cache of the content X.

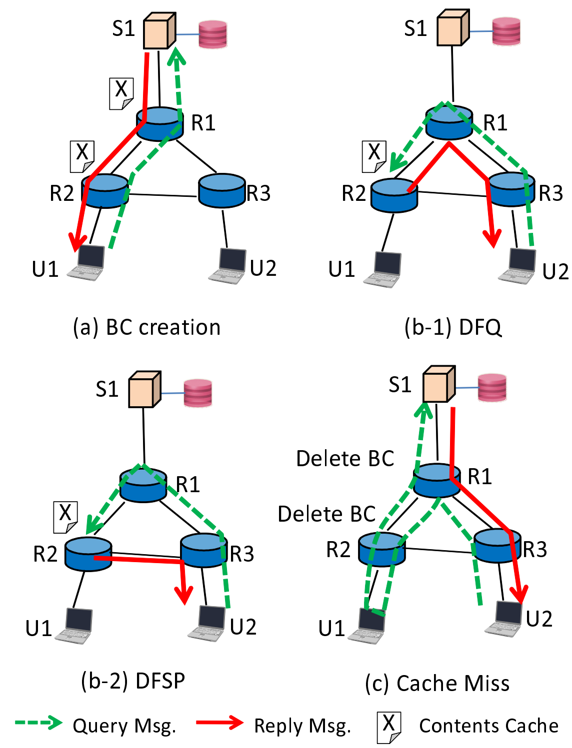

Figure 1. Basic behavior of Breadcrumbs

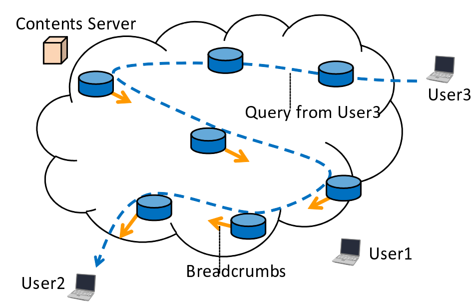

Figure 2. Lengthened BC Trail

III. EXTENSIONS OF BREADCRUMBS

A. Active Breadcrumbs

Active Breadcrumbs (ABC) is the extension of Breadcrumbs proposed in [7]. With Breadcrumbs, routers always overwrite BC entries to the newest direction of download. Therefore if a content is downloaded from a node having the cache, BC entries in its adjacent routers are overwritten to the direction of download users. For example in Fig.2, after User1 downloads a content from Server and User2 downloads it from User1, BC trail is formed like short arrows in Fig.2. Therefore the query sent by User3 travels a long trail along the dotted line. Even if the query passes close to the server and User1 which have desired content, it is redirected toward the further node User2. The objective of ABC is to avoid such ineffective redirection of queries. In addition, ABC can draw queries travling near the nodes which have the contents cache even if no adjacent routers have BC entries. As a result, ABC can improve cache hit ratio and reduce the load of contents server. In Fig.2, suppose that User1 distributes ABC to the adjacent router, the query from User3 is guided to User1. So User3 can download contents from User1. If User1 removed the contents from its cache storage, it also invalidates ABC for the removed contents. The range of distribution of ABC can be adaptively determined in terms of contents popularity, load of the node and so on.

 ISBN: 978-1-61208-300-1

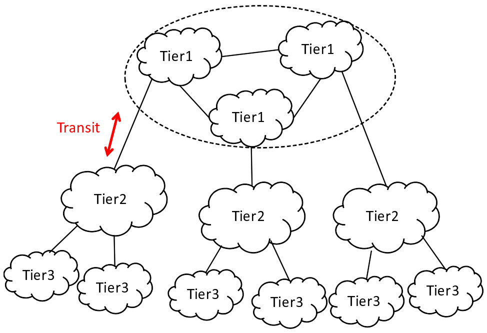

Figure 3. Hierarchy of ASes

B. Hop-aware Breadcrumbs

Hop-aware Breadcrumbs (HBC) is proposed in [8]. As same objective as ABC, HBC avoids redirection of queries near the server to further node. HBC aims to decrease delay of contents discovery. HBC extends BC entry to have the download node in addition to the downstream node of BC trail. A router compares hop count to the contents server and that to download node. If contents server is closer than the download node, the router ignores BC information and forwards the query to the server direction.

C. BC Scoping

BC Scoping is proposed in [9]. BC Scoping assumes a hierarchical structure of the network as shown in Fig.3. The Internet consists of a large number of Autonomous Systems (AS). ASes have vertical relations called Tier. Lower Tier ASes have transit links to connect upper Tier ASes. When lower ASes send or receive traffic through the transit links, they need to pay to upper ASes according to the amount of traffic. Transit cost per Mbyte in US was predicted $2.34 in 2012 [11]. Therefore it will be possible that lower ASes reduce their transit cost with utilizing contents cache in their own networks. In addition, when users can download desired contents within the belonging AS, users have benefits such as faster download and so on. Since original BC doesn't take hierarchy of the network into consideration, queries will be guided toward the different ASes via transit links. BC Scoping permits only routers which belong to the same ASes as the download user to create BC entries. This scheme restricts inter-AS redirection of queries and increases cache hit ratio within the same AS.

D. MSCR

Mapping Server with Cache-location Resolution (MSCR) is proposed in [10]. With Breadcrumbs, it is necessary for requesting users to know the location of contents server prior to sending a query. MSCR assumes Mapping Server (MS) which can provide server locations correspond to the content IDs. Users who want some contents inquire contents server information to MS. With MSCR, MS also keeps Prospective Cache Location (PCL) in addition to the server locations. PCL includes location of users who recently inquired the server location, in other words, locations where desired contents are potentially cached. When other user inquires server location of the contents to MS, MS replies PCL which is the closest to the user, too. The user puts the PCL information to the query message. When a router receives the query with PCL, the router forwards the query not toward the contents server but toward the location where the PCL indicates. If no caches exist on the way to the location, PCL information is removed from the query message and the query is forwarded toward the contents server. As a result, since the users can get the contents from closer location, network traffic can be reduced. For the prototype used in the evaluation, MS selects best PCL as the following priority order.

1) PCL having the same domain which the requesting user belongs to.
2) PCL having the location address which is longest-matched with requesting user's address.
3) PCL registered most recently.

E. Priority among extensions

Our prototype implemented above mentioned four extensions based on Breadcrumbs. Because the four extensions can be used simultaneously, routers may have several options for query forwarding. Following is the order of priority for routers to forward queries.

1) ***ABC***: If there exists ABC, the query is forwarded by ABC.
2) ***HBC***: If valid HBC exists AND contents server is closer than download node in HBC, query is forwarded toward the server. If server is not closer, the query is forwarded to the downstream node of HBC.
3) ***MSCR***: If PCL exists in the query, the query is forwarded toward the location indicated by PCL.
4) Otherwise the query is forwarded toward the contents server.

IV. Evaluation Environment

Breadcrumb-based contents distribution system was developed on Linux with C language. The system implements basic Breadcrumbs and four extensions, i.e., ABC, HBC, BC Scoping and MSCR. The structure of the system is shown in Fig.4. Every router, user node and contents server runs the contents distribution module. The system also has a mapping server which provides location (i.e., IP address) of cotents servers to users. All modules are implemented as user level processes. All messages among processes are defined in XML.

In this evaluation, the performance metrics and network traffic were measured when plural extensions work at the same time. 14 patterns in which more than one extensions are used were compared with basic BC case. Measured

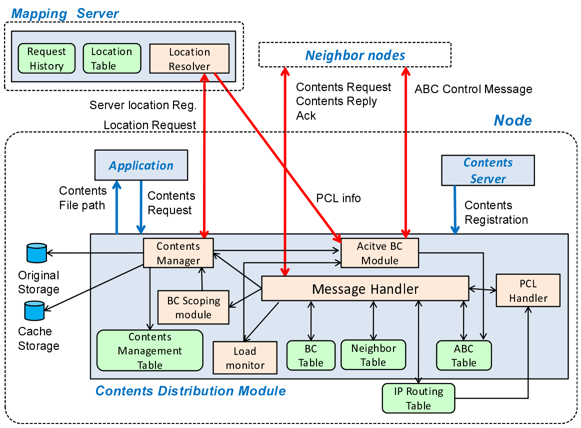

Figure 4. Structure of breadcrumbs system

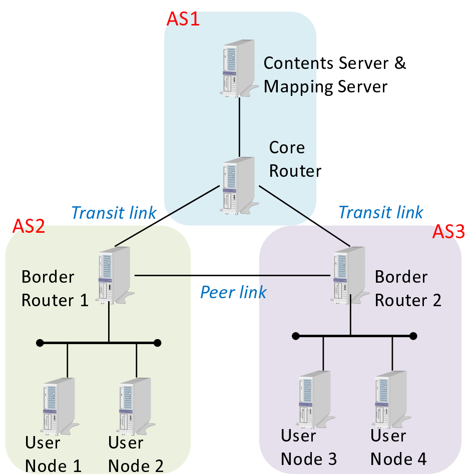

Figure 5. Evaluation Topology

metrics are cache hit ratio, query hop count, download hop count, number of cache miss and average cache lifetime.

A topology used for the evaluation is shown in Fig.5. The network consists of one contents server (CS), which has the role of both mapping server and contents server, one core router (CR), two border routers (BR1˜2) and four user nodes (UN1˜4). There are three Autonomous Systems (AS) in the network. It is assumed that AS1 is upper AS and AS2 and AS3 are lower ASes. Two topologies with and without a peer link between AS2 and AS3 were evaluated. All network interfaces equiped in the nodes are FastEther (100Mbps).

Only routers can forward queries with BC. User nodes do not create any BC entries. However user nodes can distribute ABC when contents in its cache are hit. The range of ABC distribution depends on the hit count of cached contents. Because hit count is higher, the content is supposed to be more popular. The pair of (Cache Hit Count, ABC TTL) are set (1,1), (5,2), (10,3) in this evaluation. Contents data are downloaded along the shortest path. Every router and

Table I
PARAMETERS

Param	Value
Router Cache Size	10
User Cache Size	10
Download policy	DFSP
Cache Replace Policy	LRU
T_f	1800sec
T_q	600sec
Total Eval. time	1000sec

Table II
RESULTS WITH BASIC BC

	Cache Hit Ratio (%)	Query Hop Count	Download Hop count	# Cache Miss	Cache Lifetime (sec)
with peer link	36.2	3.46	2.20	1666	9.80
w/o peer link	37.5	3.49	2.24	1502	9.86

user node have cache storage, which size is as 10 contents. Total number of contents is 1000 and size of a content is 100Kbytes. The popularity of k^{th} content $p(k)$ is calculated in eq.(1) as followed in Zipf distribution. In this experiment, $\alpha = 1.0$ was used.

$$p(k) = \frac{1/k^{\alpha}}{\sum_{n=1}^{1000} 1/n^{\alpha}} \tag{1}$$

All user nodes send one query per second. Requested contents are chosen with probability in proportional to $p(k)$. Becasue one trial lasts 1000 seconds, the number of total reqeusts from all user nodes is 4000. Three trials were performed and results are averaged. Parameters configured for the evaluation are shown in Table I.

V. EVALUATION RESULTS

A. Performance of Contents Distribution

The results in the case using only basic BC are shown in Table II. The results between topologies with and without peer link have no big differences. Cache hit ratio is slightly better than 36%. Query hop count is about 3.5, a little longer than the hop count from user node to content server (3 hops). However, download hop count is around 2.2, shorter than 3. It means that users can get the contents from closer cache. Cache miss events happen more than 1500 times out of 4000 requests. Each content cache can survive less than 10 seconds on average.

Compared with this basic BC results, Table III and IV show increase or decrease percentage of the results for combinations of extensions. The results in the topology with peer link are shown in Table III, and those in the topology without peer link are shown in Table IV. Shaded cells mean degradation from basic BC results. Bold figures mean the best results among the combinations. Same results regarding each metric are also shown in Fig.6 ˜10. Circled bars in the figures mean the best results.

 ISBN: 978-1-61208-300-1

Table III
INCREASE/DECREASE % IN TOPOLOGY WITH PEER LINK

	Cache Hit Ratio	Query Hop Count	Download Hop count	# Cache Miss	Cache Lifetime
ABC	4.93	−2.24	−2.05	−4.90	3.37
BC Scope	−14.07	−22.92	−0.34	−42.37	2.14
MSCR	2.79	−3.11	−1.25	0.68	1.33
HBC	−14.46	−22.56	−0.11	−41.69	1.43
ABC +BC Scp	−9.33	−23.43	−1.14	−43.91	2.24
ABC +MSCR	**6.34**	−5.28	**−2.61**	−5.42	**4.49**
ABC +HBC	−4.37	**−24.15**	−2.05	**−45.35**	3.88
BC Scp +MSCR	−14.08	−14.90	0.23	−24.66	1.12
BC Scp +HBC	−13.84	−22.56	−0.45	−40.41	1.33
MSCR +HBC	−11.97	−14.68	−0.45	−23.96	1.53
ABC +BC Scp +MSCR	−5.23	−16.70	−1.70	−29.15	2.65
ABC +BC Scp +HBC	−9.07	−23.43	−0.91	−44.51	2.14
ABC +MSCR +HBC	−3.45	−15.33	−1.36	−27.47	2.86
BC Scp +MSCR +HBC	−11.04	−15.69	−1.14	−24.97	1.94
All	−4.72	−16.92	−1.93	−28.75	3.37

Table IV
INCREASE/DECREASE % IN TOPOLOGY W/O PEER LINK

	Cache Hit ratio	Query Hop count	Download Hop count	# Cache Miss	Cache Lifetime
ABC	2.29	−0.29	0.00	−2.04	1.72
BC Scope	−13.96	−24.30	−3.57	−35.41	1.42
MSCR	−1.55	1.79	0.67	1.04	−1.01
HBC	−17.56	−23.37	−2.23	−35.21	1.12
ABC +BC Scp	−11.17	−24.44	−2.90	−39.20	2.43
ABC +MSCR	**2.84**	0.07	−1.00	−2.15	2.33
ABC +HBC	−12.27	**−24.59**	−2.57	−41.31	1.83
BC Scp +MSCR	−15.27	−12.26	−1.90	−42.49	0.81
BC Scp +HBC	−18.76	−22.80	−1.45	−33.70	0.51
MSCR +HBC	−18.04	−16.20	−1.79	−35.52	0.81
ABC +BC Scp +MSCR	−9.85	−13.12	−2.34	**−45.64**	1.83
ABC +BC Scp +HBC	−10.52	−24.01	−2.90	−37.36	2.33
ABC +MSCR +HBC	−10.92	−17.49	−2.79	−40.02	1.93
BC Scp +MSCR +HBC	−13.28	−18.06	**−3.79**	−36.85	2.94
All	−8.04	−17.99	**−3.79**	−39.56	**3.45**

1) *cache hit ratio:* The result of cache hit ratio in both topologies are shown in Fig.6. As shown in Fig. 6, all combinations of extensions do not significantly improve cache hit ratio. This is because the topolgy is almost tree structure, the core router can surely redirect queries only with basic BC. Since MSCR can guide queries to the region in which ABC is locally distributed, the combination of ABC and MSCR can improve cache hit ratio best. It can be said that MSCR can enhance the benefit of ABC. Since the network is samll in this evaluation, synergy of ABC and MSCR is limited. The benefit of synergy between ABC and MSCR will be more in larger networks. On the other hand, BC Scoping and HBC worsen cache hit ratio compared with basic BC. This is because both extensions will restrict query redirection at the core router, so more queries tend to go to the contents server. When ABC is used with HBC in the topology with peer link, degradation of cache hit ratio can be minimized with keeping shorter query and download hop count. This is because ABC can redirect queries between BRs directly via peer link only if cache in another BR is available.

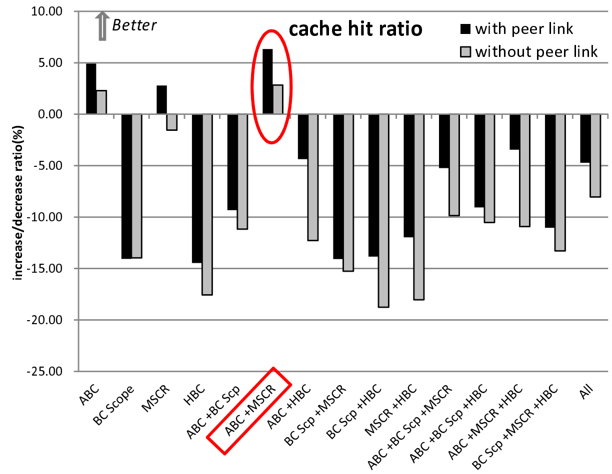

Figure 6. Increase/decrease of Cache Hit Ratio

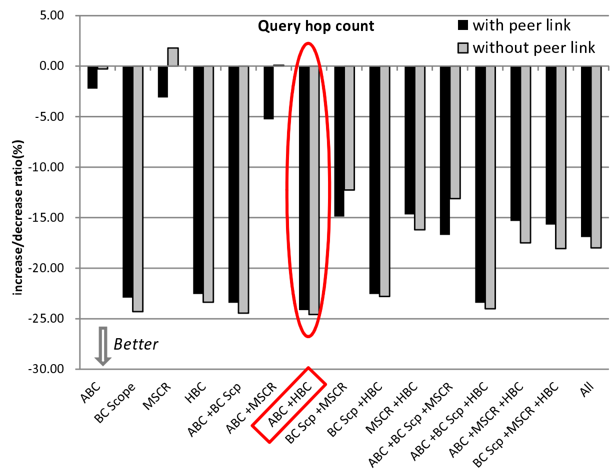

Figure 7. Increase/decrease of Query Hop Count

2) *query hop count:* Query hop count is related to the time to find contents. Shorter the query hop is, faster the user can find contents. According to Fig.7, thanks to the restriction of redirection at CR, HBC and BC Scoping can shorten query hop count by more than 20%. With using only ABC or MSCR individually, reduction of query hop count is limited. This is because the server is as close as cache locations in BRs and UNs in the evaluation topology. However, combination of ABC and HBC/BC Scoping can enhance query hop reduction. This is because queries for only popular contents are redirected between BRs by ABC, but queries for less popular contents are not redirected from CR to BRs due to BC Scoping or HBC. Therefore combination of ABC and HBC/BC Scoping is good choice to reduce query hop count with less degradation of cache hit ratio.

3) *download hop count:* Download hop count is related to both time to download contents and traffic load in the

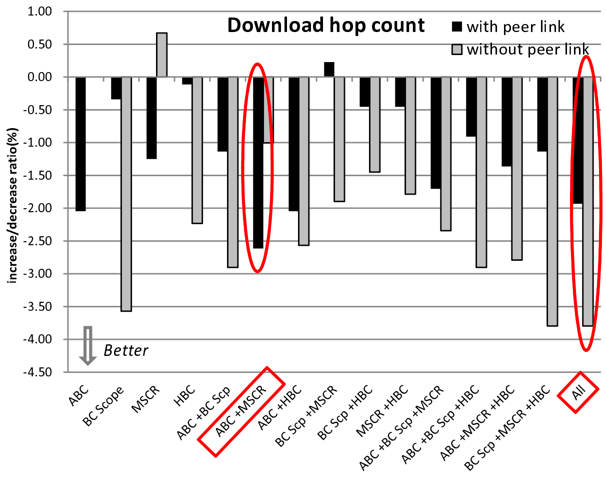

Figure 8. Increase/decrease of Download Hop Count

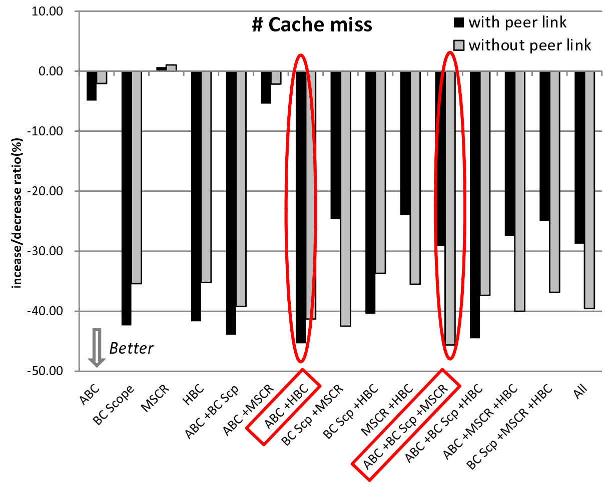

Figure 9. Increase/decrease of number of cache miss

network. Shorter the download hop is, faster the user can download contents, also lighter the traffic load in the network becomes. The results of increase or decrease of download hops are shown in Fig.8. In case of the topology with peer link, the combination of ABC and MSCR is best. ABC only and ABC+HBC also marked good scores. It means that ABC contributes downloading contents cache from BRs via peer link. Furthermore, MSCR and ABC encourage download between UNs in the same AS. Suppose that UN1 has a content cache and distribute ABC to BR1. Even if BR1 has no BC pointing to UN1, ABC in BR1 can guide queries from UN2 to the cache-holding UN1. MSCR can also guide queries directly to the neighbor UN in the same AS based on the PCL given by MS. Hop count between UN1 and UN2 in the same AS is only 1 hop. Therefore combination of ABC and MSCR can reduce more download hop count than other combinations in this topology. Unlike the case with peer link, using all extensions can reduce download hop most in the topology without peer link. When a UN downloads a content from other UN in the different AS, hop count will be 4. Since hop count from server to UN is 3, download between UN in different ASes lengthen the download hop count. HBC and BC Scoping can avoid such longer download path by restricting queries to go to other AS. On the one hand, ABC and MSCR help download between UNs in the same AS. This synergy causes the best result in terms of download hop reduction.

4) cache miss frequency: According to Table II, cache miss events happen more than 1500 times for total 4000 queries. It means that cache storage size in routers and user nodes are so small while BCs survive much longer than cache lifetime. According to Fig.9, HBC and BC Scoping can reduce cache miss event by around 40% because they restrict query redirection from CR to BRs with BC. Owing to the decrease of cache miss, useless forwarding of queries are suppressed. As the result, total messages processed in the network can be reduced by 7%. ABC reduces cache miss because ABC can guide queries to surely existing cache. When the cache in the node is removed, ABC distributed to the adjacent nodes are also purged. Therefore ABC does not worsen cache miss situation. Regarding MSCR, since PCL in the MS is not related to the removal of cached contents, MSCR is more prone to cache miss than ABC. In the topology with peer link, ABC+HBC is the best combination to minimize cache miss. This is because ABC can surely find cached contents via peer link, and HBC prevents unsure redirection of query at CR. On the other hand, in the topology without peer link, ABC+BC Scoping+MSCR is the most effective. With MSCR, MS possibly gives PCL in same AS as the requesting user. Therefore if neighbor UN in the same AS doesn't have the cached contents, a query merely goes to BR. Since BC Scoping prevents CR from creating BC, CR forwards queries to another AS with ABC only if BR or UNs in another AS surely have cached contents.

5) cache lifetime: Because average cache lifetime shown in Fig.10 is around 10 seconds, BC lifetime (T_f) should be similar value as the cache lifetime. If BC lifetime is too long, cache miss will happen more frequently. Conversely BC lifetime is shorter than cache lifetime, cache hit ratio must be degraded. However we cannot know the average cache lifetime in advance and it will be changeable due to many factors such as popularity of the contents, user distribution, network topology, router capacity and so on. Therefore how to decide optimal BC lifetime is still open issue. Since ABC is distributed when the cache is hit in the node, ABC will guide queries for popular contents and cache of the popular contents can be used more frequently. As the result, cache lifetime become longer when using ABC. In the topology with peer link, the combination of ABC and MSCR can prolong the cache lifetime most. While in the

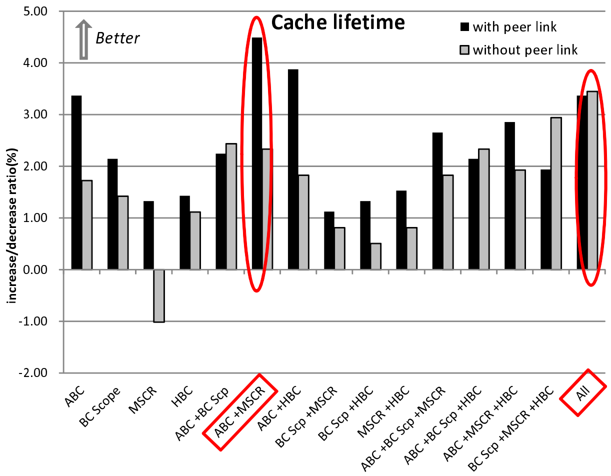

Figure 10. Increase/decrease of average cache lifetime

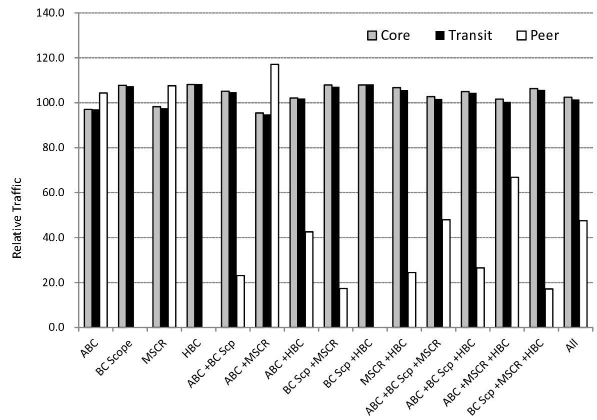

Figure 11. Relative traffic in topology with a peer link

Table V
NETWORK TRAFFIC WITH BASIC BC (MBYTES)

	Core	Transit	Peer	Intra	Total
with peer link	395.9	410.9	38.9	519.7	1365.4
w/o peer link	386.8	485.6	N/A	517.0	1389.5

topology without peer link, the lifetime is the longest when using all extensions. Although cache hit ratio is degraded in most combinations, cache lifetime becomes longer. Main reason is that the cache in the BRs can survive longer. In the topology without peer link, the cache lifetime in UNs tend to be shorter. While in the topology with peer link, that in the CR becomes shorter. However in both topologies, the cache lifetime in BRs becomes longer by 6˜10%. If cache lifetime becomes longer, it is expected that load to replace cache in routers can be reduced.

B. Network traffic

Total amount of network traffic (Mbytes) are shown in Table V when using basic BC individually. $Core$ means traffic on the link between CR and contents server. $Transit$ means traffic on the transit links between CR and BRs. $Peer$ means traffic on the peer link between BRs. $Intra$ means traffic within AS2 and AS3. $Total$ is the traffic throughout the whole network. Total traffic increases in case of no peer link. This is becasue download across ASes travels two transit links instead of the peer link.

For all combinations, relative amount of traffic to basic BC results as 100 are shown in Fig.11 and Fig.12.

1) The topology with peer link: From the viewpoint of operators in lower Tier ASes, it is desirable that users (their customers) can download contents within the AS or can download from other ASes connected with a peer link. According to Fig.11, it reveals that HBC can not utilize the peer link at all. This result depends on the topology. Since the contents server is closer than UNs from CR, HBC always selects to forward queries toward the server in CR. Since BC Scoping also prevents inter-AS download, it doesn't utilize peer link at all, too. In terms of the operation cost, ABC+MSCR is the best combination becasue it can reduce the transit traffic most. This combination can also reduce total traffic most becasue download hop count is minimized by exploiting the peer link.

2) The topology without peer link: As shown in Fig.12, when using HBC or BC Scoping, more transit traffic can be reduced. This is because inter-AS download is restricted. Instead, core traffic increases becasue more users download the contents from the server. BC Scoping+MSCR+HBC is the best combination for the operators in lower ASes because it can reduce tansit traffic most. This combination is also the best from global point of view becasue it can minimize total traffic. From viewpoint of core network operator, ABC+MSCR is the best to reduce the most core network traffic. According to this observation, there is contradiction between the benefits of core network operator and lower AS operators. Core network operators have no incentive to use HBC and BC Scoping, because they want to minimize traffic in core network and maximize transit cost. Lower AS operators want core network to use BC Scoping and HBC to minimize transit cost. If each operator can configure which extensions are used in thier routers independently, the benefit will be reduced. How to compromise the benefit of operators in different ASes and find best mix of extensions for all operators is the essential but difficult issue.

VI. CONCLUSION

In this paper, synergic effects among multiple extensions of Breadcrumbs, which is the distributed mechanism to discover cached contents in the networks were evaluated.

According to the evaluation results, ABC and MSCR can improve cache hit ratio most regardless the existence of peer links. In the topology with peer links, ABC and MSCR

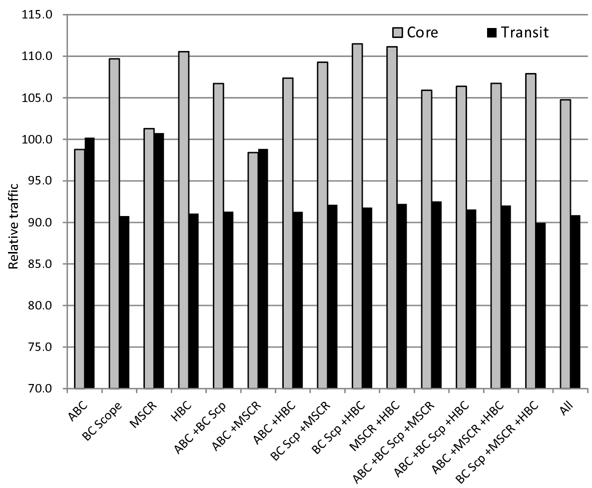

Figure 12. Relative traffic in topology without peer link

is also the best combination for both performance metrics and network traffic. If it is important to reduce control messages, ABC and HBC is the best combination. In the topology without peer links, using all extensions is the best choice to improve performance metrics. From the viewpoint of network traffic, BC Scoping, HBC and MSCR is good combination for lower AS operators.

Because these results fairly depend on the topologies used in the evaluation, evaluation with larger networks will be performed near future. It is also necessary to consider real network scenario for applications of Breadcrumbs system. Automatic BC lifetime adaptation scheme will be another challenge to optimize performance of cache discovery with BC.

ACKNOWLEDGMENT

This work was supported by National institute of Information and Communications Technology (NICT) Japan.

REFERENCES

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010-2015, Feb. 2011.

[2] G.Tyson, S.Kaune, S.Miles, Y.El-khatib, A.Mauthe, and A.Taweel, "A Trace-Driven Analysis of Caching in Content-Centric Networks", Proc. ICCCN'12, July 2012, pp.1-7.

[3] S.Paul, R.Yates, D.Raychaudhuri, and J.Kurose, "The cache-and-forward network architecture for efficient mobile content delivery service in the future internet," First ITU-T Kaleidoscope Academic Conference, May 2008, pp.367-374.

[4] T.Koponen, M.Chawla, B-G.Chun, A.Ermolinskiy, K.H.Kim, S.Schenker, and I.Stoica, "A Data-Oriented (and Beyond) Network Architecture," in Proc. ACM SIGCOMM 2007, Aug. 2007, pp.181-192.

[5] V.Jacobson, D.Smetters, J.Thorton, M.Plass, N.Briggs, and R.Braynard, "Networking Named Content," in Proc. ACM CoNEXT 2009, Dec. 2009, pp.1-12.

[6] E.J.Rosenweig and J.Kurose, "Breadcrumbs: efficient, best-effort content location in cache networks," in Proc. IEEE INFOCOM 2009, Apr. 2009, pp.2631-2635.

[7] M.Kakida, Y.Tanigawa, and H.Tode, "Active Breadcrumbs: Aggressive Distribution Method of In-network Guidance Information for Content-Oriented Networks," in Proc. LCN2012, Oct. 2012, pp. 184-187.

[8] K.Hashimoto, Y.Takaki, C.Ohta, and H.Tamaki, "In-network Hop-aware Query Induction Scheme for Implicit Coordinated Content Caching," in Proc. AFIN2011, Aug. 2011, pp. 69-73.

[9] M.Kakida, Y.Tanigawa, and H.Tode, "Distribution Method of In-network Guidance Information for Inter-AS Content-Oriented Network Topology," World Telecomm. Congress (WTC) 2012, Mar. 2012, Poster session.

[10] H.Kawabata, K.Hashimoto, T.Inamoto, Y.Takaki, C.Ohta, and H.Tamaki, "Content/Location Mapping with Cache-Location Resolution for In-network Guidance", Proc. AFIN2012, Aug. 2012, pp.1-6.

[11] W.B.Norton, "Internet transit prices - historical and projected," tech. rep., Dr Peering International, 2010.