Multimedia Tools and Applications (2022) 81:12371–12385
https://doi.org/10.1007/s11042-021-11434-6

1172: 5G MULTIMEDIA COMMUNICATIONS FOR VEHICULAR, INDUSTRY AND ENTERTAINMENT APPLICATIONS

Check for updates

Content-centric framework for Internet of Things

Xilan Chen[1] · Xiaonan Wang[1]

Received: 8 August 2020 / Revised: 4 August 2021 / Accepted: 17 August 2021 /
Published online: 17 September 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
The Internet of Things (IoT) concentrates on content dissemination and retrieval, so it is significant to achieve efficient content delivery. However, the Internet focuses on end-to-end communications, which might degrade the content retrieval performance in mobile environments. By contrast, the content-centric mechanism might be an ideal method for achieving efficient content delivery although it suffers from flooding and reverse-path disruptions. Therefore, we are motivated to exploit the content-centric mechanism to achieve IoT-based content delivery, and employ the address-centric anycast to overcome the limitations of the content-centric mechanism. Inspired by the idea, we propose a content-centric framework for IoT. The experimental results show that the proposed framework reduces the content communication cost and improves the content acquisition success rate.

Keywords Content-centric · Internet · IoT · Anycast

1 Introduction

With the great success of the Internet and the popularity of personal mobile devices with powerful processing and abundant storage capabilities, mobile devices become actually the main force of generating, consuming and providing multimedia contents via the Internet [1, 13]. As mobile devices usually work as Internet of Things (IoT) [9, 20, 21], so it is significant to achieve efficient IoT-based content delivery [7, 19]. This paper focuses on the content communication issue in the multi-hop IoT [22]. The Internet concentrates on end-to-end communications. If end-to-end communications are used for multimedia content retrieval in mobile environments, they might increase content communication costs and degrade success rates. First, each end-to-end communication process is performed independently [22], which results in redundant content request and response messages, and greatly increases content communication costs. Second, a consumer must acquire contents from a target provider even though the provider may

✉ Xiaonan Wang
ninawang9@163.com

1 Changshu Institute of Technology, Suzhou, China

be overloaded. If the target provider is unreachable, the content communication failure occurs. Third, if a consumer moves between subnets, it has to perform the care-of address (CoA) configuration and binding operations. These time-consuming operations might incur the packet loss and further increase the content communication cost.

The content-centric mechanism is a novel content communication model [2, 12], where a consumer starts a content communication process by sending an *Interest* with a name. The *Interest* is forwarded towards potential providers. Any provider receiving *Interest* sends a Data with target contents back to consumers via pending interest table (PIT). If n consumers send *n Interest* to acquire contents, these *Interest* can be aggregated via PIT. This greatly reduces content communication costs. Moreover, a consumer may retrieve contents from any provider [17], and either the CoA configuration or the binding operation is not required. Hence, the content-centric mechanism might be an ideal method to perform efficient content communications.

Based on the observation, we are motivated to exploit the content-centric mechanism to achieve IoT-based content communication. However, the content-centric mechanism also suffers from the limitations if it works in mobile IoT environments. For example, reverse paths may frequently disrupt due to node mobility, which leads to frequent content communication failures. The replicas of contents are cached on different providers, and the content-centric mechanism employs flooding to acquire and update contents, which incurs huge costs.

Anycast is a unicast-based communication model that can achieve content delivery without relying on reverse paths and reduce content communication costs compared to flooding [14, 16]. Moreover, any anycast member may provide contents [14]. To overcome the limitations of the content-centric mechanism, we are motivated to employ anycast to achieve the content-centric mechanism in IoT, and propose a content-centric framework in IoT (CCI) to lower content communication costs and enhance success rates via the following novelties:

(1) CCI exploits anycast to achieve the content-centric mechanism so that consumers can employ unicast to retrieve contents from the nearest anycast member without relying on reverse paths.
(2) CCI proposes a pending request mechanism to achieve aggregation, and based on this mechanism consumers can retrieve contents via one content communication process. Also, CCI extends multicast to the content-centric communications so that the contents can be updated in the multicast way instead of in the flooding way.
(3) CCI proposes an address separation mechanism where a mobile node is identified by a node ID instead of an address, so either the CoA configuration or the binding operation is not needed.

This work has the following differences from [20] and [21]:

(1) The architectures are different. The work [21] deals with a fixed network and in each subnet multiple content servers directly link with switches or access routers. The work [20] and this proposal handle a mobile network, and mobile nodes are multi-hop away from an access router. However, in [20] a content server is deployed in the center of a subnet and is multi-hop away from an access router, while in this work a content server is integrated with an access router and is located at the edge of a subnet.

(2) The address structures are different. The work [21] does not discuss the address structure issue. The work [20] defines the content address structure and employs the content address to perform data communications. By contrast, this work proposes the anycast and multicast address structures to deliver data.
(3) The data communication algorithms are different. In [21], routers and switches maintain forwarding tables and perform forwarding functions. In [20] and this proposal, mobile nodes perform forwarding functions. However, in [20] a mobile node independently retrieves data from a server via one data delivery procedure whereas in this proposal multiple mobile nodes employ the anycast technology to share data from the nearest anycast member via one data delivery process.
(4) The content update algorithms are different. The work [20] does not address the content update issues. The work [21] relies on forwarding tables to update contents in a limited flooding way while this work extends multicast to content-centric communications and employ multicast instead of flooding to update contents.

This paper is organized as follows. In Sect. 2 the related work on the content-centric mechanism is discussed, in Sects. 3 and 4 CCI is presented, in Sects. 5 and 6 CCI is evaluated, and Sect. 7 concludes the paper with a summary.

2 Related work

The Internet faces some challenges when it is used for dissemination and retrieval of multimedia contents. For example, each consumer independently acquires target contents from a specific provider identified by a destination address [16]. By contrast, the content-centric mechanism overcomes these challenges because any provider can provide the content. Therefore, the content-centric mechanism might be an ideal method to perform efficient contents communications for IoT. The content-centric standard [12] proposes a content-to-consumer framework to achieve efficient content delivery. This framework employs *Interest* and *Data* so that consumers can retrieve contents from the nearest providers. Moreover, the content-centric standard achieves the content update by employing flooding to guarantee that consumers retrieve real-time contents. In [4], a forwarding algorithm is presented and this algorithm chooses forwarders using a variety of metrics such as distance to mitigate redundant control information caused by flooding Interest. In [3], a search-and-routing method is proposed to obtain data from the nearest device with target data. This method employs the probe mechanism to discover devices with target data, but this probe procedure might incur additional overheads. In [15], the authors combine edge computing with the content-centric mechanism and decouple their data and control planes to enhance the performance of content communications. The authors implements two typical applications to justify the advantages of the solution. In [23], the authors propose a content-aware data delivery method. This solution employs fog computing to reduce content communication latency. Moreover, the content-aware filtering technology is used to achieve accurate filtering to further improve the content communication performance. In [25], the authors employ the content naming and content-based routing to organize networks for disaster recovery. This solution exploits the content-centric mechanism to connect rapidly users in post-disaster scenarios. This work demonstrates that the content-centric mechanism can efficiently improve the routing performance and satisfy the requirements of disaster recovery. In [18, 20, 21], the authors demonstrate that the content-centric mechanism can assist in performing efficient content communications, but node mobility and flooding are significant challenges.

The above solutions exploit the content-centric mechanism to improve the content communication performance. However, the improvements are limited because reverse-path disruptions caused by node mobility lead to frequent content communication failures and flooding used to acquire contents incurs huge overheads. Moreover, only the content-centric standard [12] addresses the content update issue whereas other existing solutions [3, 4, 15, 18, 23, 25] do not discuss how to solve the content update issue. Although the content-centric standard [12] ensures validity of contents, it performs the content update by employing flooding. Consequently, the costs and delays of the content update are relatively considerable.

Anycast is a unicast-based communication model that can deliver contents without relying on reverse paths and help suppress content delivery costs caused by content-centric flooding [16]. In [9], the authors employ unicast to retrieve contents to suppress the content communication costs and delays. However, this solution performs the content communications between a consumer and a target provider, so the performance improvements are limited. Moreover, this solution does not address the content update issue. In [22], the authors propose an anycast-based content-centric IoT (ACCM) to achieve content communications. In ACCM, the providers form an anycast group. The strength of ACCM is that the content update is achieved to ensure that consumers can acquire real-time contents. However, as each anycast member independently performs the content update, the costs and delays of the content update are relatively considerable. In [14], an anycast-based content communication method is proposed to reduce content communication delays and overheads. The analytical results justify the advantages of anycast in content communications because target contents are provided by the nearest anycast member. In [8], anycast is employed to reduce latency of content communications. In the solution a variety of metrics are used to establish an anycast tree and contents are delivered by using the anycast tree. The analytical results demonstrate superiority of anycast in content delivery because contents are delivered by the optimal anycast member.

Based on the advantages of anycast, we are motivated to employ anycast to overcome the disadvantages of the content-centric mechanism such as reverse-path disruptions and flooding so that consumers can employ unicast to retrieve contents from the nearest provider without replying on reverse paths.

3 Architecture

The CCI architecture consists of K access routers (ARs) and mobile nodes. Each AR_k ($1 \leq k \leq K$) is defined by its unique location coordinates (x_k, y_k). A mobile node performs the forwarding function, and achieves the content communications via the nearest AR. An AR and the mobile nodes performing communications via the AR form a subnet that is actually an Internet-based IoT, so the concept of the subnet in CCI is the same as the one in the traditional IP network. In a subnet, mobile nodes may be multi-hop away from the local AR. In a subnet, there is a content server whose function is to cache local contents, and the content server and AR are integrated together and share one address. The purpose of the content server is twofold. First, the content server has abundant storage resources and can cache and provide contents even if all mobile anycast members leave the subnet. This helps improve the success rates of content communications.

Second, the content server can assist in shortening the distance between a consumer and the remote contents in the inter-subnet communication scenario, which helps reduce the content communication delays and costs. The CCI architecture is shown in Fig. 1, where AR_k is identified by location coordinates (x_k, y_k), and AR_k and the mobile nodes achieving communications via AR_k construct subnet B_k.

In IoT, a mobile node uses a unicast address, an anycast address and a multicast address to perform content communications and update. A unicast address is used for routing, an anycast address uniquely identifies a type of contents and is used for seeking target contents, and a multicast address is used for updating target contents. As content delivery based on geo-location information can assist in improving the communication performance [5], we propose a location-based unicast address structure and aim to retrieve contents from the nearest anycast member. A unicast address contains the network prefix (NP) and the interface ID that consists of the node ID and the location coordinates, as shown in Table 1. The node ID of an AR is zero. An anycast address consists of the NP and the interface ID whose value is zero. This means that NP actually identifies a type of contents, as shown in Table 2. The multicast address structure includes the multicast prefix (MP) and the group ID that includes the NP and the reserved field, as shown in Table 3.

In Table 1, the node ID space is [1, $2^{2i-2j}-1$] that is partitioned into K parts, and each part is for each subnet. The node ID space A_k in the kth subnet B_k is shown in (1) and is maintained by AR_k. In B_k, the node ID of AR_k is zero. After a mobile node in B_k starts, it uses the existing addressing method [24] to acquire a unique node ID from AR_k, and is defined by the node ID during its lifetime.

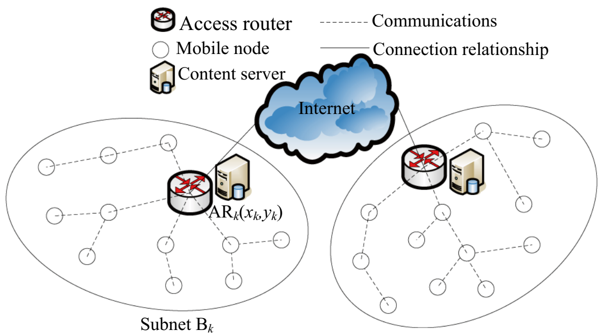


Fig. 1 CCI architecture

Table 1 Unicast address

Bits: 128-2i	2i-2j	j	j
Network prefix	Node ID Interface ID	Abscissa	Ordinate

Table 2 Anycast address

Bits: 128-2i	2i
Network prefix	Reserved Interface ID

Table 3 Multicast address

Bits: 16	128-2i	2i-16
Multicast prefix	Network prefix Group ID	Reserved

$$A_k = \begin{cases} [\frac{(k-1)\cdot 2^{2i-2j}}{K} + 1, \frac{k\cdot 2^{2i-2j}}{K}]; 1 \le k < K \\ [\frac{(k-1)\cdot 2^{2i-2j}}{K} + 1, \frac{k\cdot 2^{2i-2j}}{K} - 1]; k = K \end{cases} \quad (1)$$

4 Content-centric IoT famework

Each AR or mobile node maintains a content index table to store the information on providers, and an entry contains the anycast address, the node ID, the coordinates and the lifetime. The AR and mobile nodes in a subnet share one index table that is named as the coordinates of the AR. For example, the index table name in B_k is (x_k, y_k). After a mobile node starts, it acquires the coordinates of each AR by using the pre-load map. Anycast address A_1 defines content C_1 and anycast group G_1. The server whose NP is equal to the one of A_1 and the mobile nodes which can provide C_1 form G_1 and multicast group P_1 identified by multicast address U_1 whose NP is equal to the one of A_1.

4.1 Content announcement

The node ID of mobile node M_1 is I_{M1} and the coordinates are (x_{M1}, y_{M1}). M_1 is closest to AR_k and can produce content C_1 identified by anycast address A_1. The members in anycast group G_1 form multicast group P_1. AR_k is located in subnet B_k with NP_k and its coordinates are (x_k, y_k). After M_1 produces C_1, it does the following content announcement operations:

(1) M_1 creates the index table T_h with name (x_k, y_k) and creates an entry where the anycast address is A_1, the node ID is I_{M1}, and the coordinates are (x_{M1}, y_{M1}). Then, M_1 piggybacks T_h in *hello* [10] and becomes a member of G_1 and P_1.
(2) The mobile node receives *hello* with T_h. If the mobile node is in subnet B_k, it performs the operations based on the following cases:

Case 1: There is the entry E_h in T_h where neither the anycast address nor the node ID is the one of any entry in T_m

The mobile node adds E_h in T_m.

Case 2: There is the entry E_h in T_h where the anycast address and node ID are equal to ones of the entry E_m in T_m and the lifetime is larger than the one of E_m

The mobile node updates the coordinates and lifetime in E_m with the ones in E_h.

Case 3: Neither *Case* 1 nor *Case* 2 is satisfied.

It goes to 4).

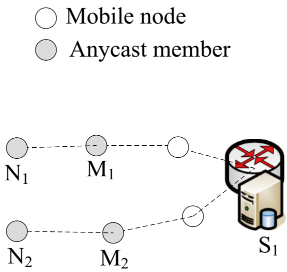
(3) The mobile node piggybacks T_m in *hello* and goes to 2).
(4) The process is complete.

If M_1 changes its location, it updates the corresponding index entry, and piggybacks the index table in *hello*. In this way, the nodes in B_k can share the real-time index table. As shown in Fig. 2, the ID of mobile node M_1/M_2 is I_{M1}/I_{M2} and the coordinates are (x_{M1}, y_{M1})/(x_{M2}, y_{M2}). The ID of content server S_1 is I_{S1} and the coordinates are (x_{S1}, y_{S1}). Content C_1/C_2 is defined by anycast address A_1/A_2. After M_1/M_2 produces C_1/C_2, it creates an index entry. Then, S_1 generates C_1 and creates an index entry. At time T_1, M_1/M_2/S_1 becomes a member of G_1 and P_1, and shares the index table with name (x_k, y_k) at T_1.

4.2 Intra-subnet content communication

The node ID of mobile node N_1 is I_{N1} and the coordinates are (x_{N1}, y_{N1}). Mobile node N_1 is in B_k and desires content C_1 identified by anycast address A_1. If there is at least an index entry where the anycast address is A_1, N_1 acquires C_1 via the following process:

(1) N_1 selects the nearest anycast member M_1 with anycast address A_1, and uses A_1's NP and M_1's node ID and coordinates to construct M_1's unicast address. Then, N_1 sends a *C-Req* message. In *C-Req*, the destination address is M_1's unicast address, and the source address is N_1's unicast address where the NP is zero, the node ID is I_{N1} and the coordinates are (x_{N1}, y_{N1}).
(2) If any member of group G_1 receives *C-Req*, it sends a *C-Rep* message with C_1 and goes to 4).
(3) If M_1 receives *C-Req*, it returns *C-Rep* with C_1. If the previous hop of M_1 detects that M_1 is unreachable, it selects another member of G_1 and updates the node ID and coordinates of the destination address in *C-Req* with the ones of the member, and forwards *C-Req*. In the latter case, after the member receives *C-Req*, it returns *C-Rep*.



Index table (x_k,y_k) at T_1

Anycast	ID	Coordinates
A_1	I_{M1}	(x_{M1},y_{M1})
A_2	I_{M2}	(x_{M2},y_{M2})
A_1	I_{S1}	(x_{S1},y_{S1})

Index table (x_k,y_k) at T_2

Anycast	ID	Coordinates
A_1	I_{M1}	(x_{M1},y_{M1})
A_2	I_{M2}	(x_{M2},y_{M2})
A_1	I_{S1}	(x_{S1},y_{S1})
A_1	I_{N1}	(x_{N1},y_{N1})
A_2	I_{N2}	(x_{N2},y_{N2})

Fig. 2 Content announcement

(4) After N_1 or an intermediate node that desires C_1 receives *C-Rep*, it caches C_1, becomes a member of G_1 and P_1, creates an entry in the index table with name (x_k, y_k) and piggybacks the table in hello, as shown in Fig. 2.

Since CCI can update the index table in real time, the anycast members in the index table are usually active and reachable. As shown in Fig. 2, the ID of mobile node N_1/N_2 is I_{N1}/I_{N2} and the coordinates are $(x_{N1}, y_{N1})/(x_{N2}, y_{N2})$. Content C_1/C_2 is defined by anycast address A_1/A_2. At time T_2, based on the index table, N_1/N_2 acquires C_1/C_2 from the nearest provider M_1/M_2, becomes a member of G_1 and P_1, and updates the index table with name (x_k, y_k) at T_2. In CCI, a mobile node acquires data from the nearest anycast member to reduce the content communication latency and cost. As shown in Fig. 2, mobile node N_1 requesting content C_1 is closest to anycast member M_1 rather than content server S_1, so it uses M_1's unicast address to retrieve content C_1 from M_1.

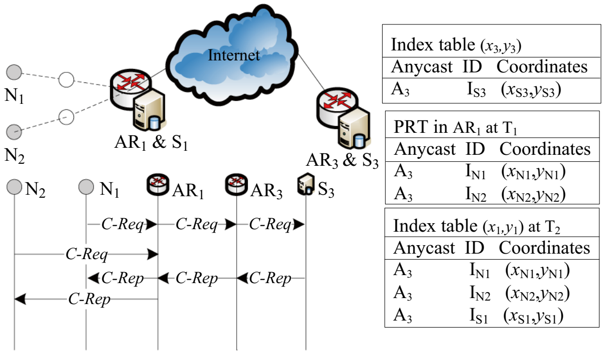
4.3 Inter-subnet content communications

An AR maintains a pending request table (PRT) to store the information on consumers, and each entry includes the anycast address, the node ID and the coordinates. Mobile node N_1 is in subnet B_k where the AR is AR_k, and desires content C_3 identified by anycast address A_3 whose NP is NP_3. The coordinates of AR_k with NP_k are (x_k, y_k). If there is no index entry for A_3, N_1 acquires C_3 via the following process:

(1) N_1 constructs a unicast address where the NP is NP_3, the node ID is zero and the coordinates are (x_k, y_k). Then, N_1 sends *C-Req* where the destination address is the unicast address, and the source address is N_1's unicast address.
(2) If an intermediate node or AR_k is a member of anycast group G_3 identified by A_3, it returns *C-Rep* with C_3, and goes to 6).
(3) If AR_k does not have the PRT entry for A_3, it updates the NP of the source address in *C-Rep* with NP_k, forwards *C-Req*, and creates one PRT entry where the anycast address is A_3, the node ID is I_{N1} and the coordinates are (x_{N1}, y_{N1}).
(4) After *C-Req* reaches AR_3 whose NP is NP_3 via the Internet, AR_3 asks the local content server S_3 to return *C-Rep* with C_3.
(5) *C-Rep* reaches AR_k via the Internet. Based on the PRT, AR_k forwards *C-Rep* to each node identified by the entry where the anycast address is A_3, and removes the PRT entries.
(6) After N_1 or an intermediate node that desires C_3 receives *C-Rep*, it stores C_3, becomes a member of anycast group G_3 and multicast group P_3, and creates an entry in the index table with name (x_k, y_k).

As shown in Fig. 3, the ID of mobile node N_1/N_2 is I_{N1}/I_{N2} and the coordinates are $(x_{N1}, y_{N1})/(x_{N2}, y_{N2})$. The ID of content server S_3 is I_{S3} and the coordinates are (x_{S3}, y_{S3}). N_1 sends *C-Req* to retrieve C_3. AR_1 receiving *C-Req* creates one PRT entry for A_3 and routes *C-Rep* towards AR_3. Then, N_2 sends *C-Req* to get C_3. After AR_1 receives *C-Req* at time T_1, it creates one PRT entry for A_3 and waits for C_3. In this way, the aggregation is achieved. AR_3 receiving *C-Req* forwards *C-Req* to S_3 that returns *C-Rep* with C_3. AR_1 forwards *C-Rep* to N_1 and N_2 based on PRT, caches C_3 and creates an index entry. After N_1/N_2 retrieves C_3, it caches C_3 and creates an index entry. At time T_2, $N_1/N_2/S_1$ becomes a member of G_3 and P_3, and shares the index table with name (x_k, y_k) at T_2. As shown in Fig. 3, content server

Fig. 3 Inter-subnet content communications



S_3 in subnet B_3 is the anycast member caching content C_3. If in subnet B_3 there are mobile anycast members caching content C_3, the distance from mobile node N_1 to content server S_3 is smaller than the distance from N_1 to any mobile anycast member in subnet B_3. In CCI, a mobile node acquires content from the nearest anycast member to reduce the content retrieval latency and cost, so N_1 retrieves content C_3 from server S_3 rather than other mobile anycast members in B_3.

4.4 Content update

The members of anycast group G_1 identified by anycast address A_1 whose NP is NP_1 form multicast group P_1 identified by multicast address U_1. A_1 identifies content C_1 and mobile node M_1 is a member of G_1 and P_1. If M_1 updates C_1, it does the following content update process:

(1) M_1 constructs a multicast address where the MP is equal to the multicast prefix, the NP is equal to NP_1 and the reserved field is zero. Then, M_1 sends a *C-Update* message. In *C-Update*, the destination address is the constructed multicast address, the source address is M_1's unicast address where the NP is zero, the node ID is I_{M1} and the coordinates are (x_{M1}, y_{M1}), and the payload is the updated content.
(2) After a multicast member of P_1 receives *C-Update*, it updates C_1 with the content in *C-Update*.
(3) The process ends.

5 Performance analysis

CCI aims to lower content communication costs and delays and enhance success rates, so these parameters are analyzed. According to the intra-subnet content communication algorithm, the intra-subnet content communication cost C_{Intra} consists of the content request cost $C_{Intra\text{-}Req}$ and the content response cost $C_{Intra\text{-}Rep}$, as shown in (2). As a consumer retrieves contents from the nearest anycast member, $C_{Intra\text{-}Req}$ and $C_{Intra\text{-}Rep}$ are shown in (3), where c is the cost of delivering a message between neighbors, l_{Intra} is the distance from a consumer to the nearest anycast member and l_D is the diameter of a subnet. The probability p_k of k mobile nodes becoming anycast members follows the Poisson process [6], so l_{Intra} is shown in (4), where m is the number of mobile nodes in a subnet and $l_{C\text{-}j}$ is the distance from a consumer to the *j*th anycast member. The content communication latency T_{Intra} includes the content request latency $T_{Intra\text{-}Req}$ and the content response latency $T_{Intra\text{-}Rep}$, as shown in (5–6), where t is the latency of transmitting a message between neighbors. In CCI, the content communication fails only if all the target members in the index table are unreachable. Therefore, the intra-subnet success rate S_{Intra} is shown in (8) where n is the number of the anycast members in the index table and p is the probability of an anycast member being unreachable.

$$C_{Intra} = C_{Intra-Req} + C_{Intra-Rep} \tag{2}$$

$$C_{Intra-Rep} = C_{Intra-Req} = l_{Intra} \cdot c;1 \le l_{Intra} \le l_D \tag{3}$$

$$l_{Intra} = \sum_{k=1}^{m} p_k \cdot k \cdot \underset{j=1}{\overset{k}{\mathrm{MIN}}}\, l_{\mathrm{C}\text{ - }j} \tag{4}$$

$$T_{Intra} = T_{Intra-Req} + T_{Intra-Rep} \tag{5}$$

$$T_{Intra-Rep} = T_{Intra-Req} = l_{Intra} \cdot t \tag{6}$$

$$S_{Intra} = 1-p^n \tag{7}$$

$$n = \sum_{k=1}^{m} p_k \cdot k \tag{8}$$

According to the inter-subnet content communication algorithm, the content communication cost C_{Inter} is comprised of the content request cost $C_{Inter\text{-}Req}$ and the content response cost $C_{Inter\text{-}Rep}$, as shown in (9). The probability p_j of j mobile nodes requesting a type of contents follows the Poisson process. If *n'* mobile nodes request contents in parallel, the inter-subnet content communication process is performed only once. Therefore, $C_{Inter\text{-}Req}$ and $C_{Inter\text{-}Rep}$ are shown in (10, 11, 12, 13 and 14), where l_{Inter} is the distance from a consumer to the destination server, l is the distance from the source AR to the destination AR, $l_{Intra\text{-}AR}$ is the average distance from *n'* consumers to the local AR, and $l_{i\text{-}AR}$ is the distance from the *i*th consumer to the local AR. The inter-subnet content communication latency T_{Inter} includes the content request latency $T_{Inter\text{-}Req}$ and the content response latency $T_{Inter\text{-}Rep}$, as shown in (15–16). The inter-subnet success rate S_{Inter} is shown in (17).

$$C_{Inter} = C_{Inter-Req} + C_{Inter-Rep} \tag{9}$$

$$C_{Inter-Req} = (l_{Inter} \cdot c + \left(n' - 1\right) \cdot l_{Intra-AR} \cdot c)/n' \tag{10}$$

$$n' = \sum_{j=1}^{m} p_j \cdot j \tag{11}$$

$$l_{Inter} = l_{Intra-AR} + l; \quad 1 \le_{Intra-AR} \le l_D \tag{12}$$

$$l_{Intra-AR} = \sum_{i=1}^{n'} l_{i-AR}/n' \tag{13}$$

$$C_{Inter-Rep} = (l_{Inter} \cdot c + \left(n' - 1\right) \cdot l_{Intra-AR} \cdot c)/n' \tag{14}$$

$$T_{Inter} = T_{Inter-Req} + T_{Inter-Rep} \tag{15}$$

$$T_{Inter-Req} = T_{Inter-Rep} = l_{Inter} \cdot t \tag{16}$$

$$S_{Inter} = S_{Intra} \tag{17}$$

Based on the content update algorithm, if the multicast members are within one subnet, the intra-subnet content update cost $C_{Intra\text{-}Update}$ and latency $T_{Intra\text{-}Update}$ are shown in (18, 19 and 20), where d_i is the distance from the i^{th} multicast member to the nearest multicast member, l_M is the distance from the multicast member announcing the updated content to the farthest multicast member, and $l_{A\text{-}j}$ is the distance from the announcing member to the *jth* multicast member. If the multicast members are located in q subnets, the inter-subnet content update cost $C_{Inter\text{-}Update}$ and latency $T_{Inter\text{-}Update}$ are shown in (21, 22 and 23). In (21, 22 and 23), $l_{M\text{-}AR}$ is the distance from an AR to the fartest multicast member, l_{max} *is* the distance from the subnet where the announcing multicast member is located to the farthest subnet including providers, and $l_{j\text{-}AR}$ is the distance from an AR to the *jth* multicast member.

$$C_{Intra-Update} = \sum_{i=1}^{n} d_i \cdot c \tag{18}$$

$$T_{Intra-Update} = l_M \cdot t; 1 \le l_M \le l_D \tag{19}$$

$$l_M = \sum_{k=2}^{m} p_k \cdot k \cdot \underset{j=1}{\overset{k-1}{\mathrm{MAX}}}\, l_{\mathrm{A\ \text{-}}\ j} \tag{20}$$

$$C_{Inter-Update} = q \cdot C_{Intra-Update} + l_{max} \cdot c \tag{21}$$

$$T_{Inter-Update} = \left(2l_{M-AR} + l_{max}\right) \cdot t \tag{22}$$

$$l_{M-AR} = \sum_{k=1}^{m} p_k \cdot k \cdot \underset{j=1}{\overset{k}{\mathrm{MAX}}}\, l_{j\ \text{-}\ \mathrm{AR}} \tag{23}$$

Table 4 Parameters

Parameters	Values
Node population	200
Transmission radius	100 m
Speed	2–10 m/s
Pause time	10 s
MAC	IEEE 802.11
Mobility model	Random waypoint
Runs	20
Confidence level	0.95

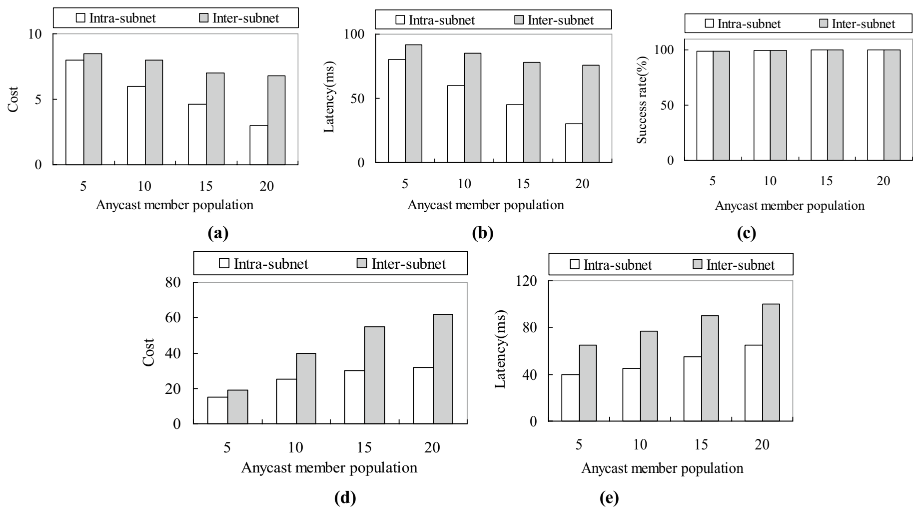


Fig. 4 Effects of anycast member population

6 Performance evaluation

The random waypoint mobility model is adopted to evaluate CCI because it is the most common mobility model in networks [7]. In the model, a node randomly selects a destination, and moves towards the destination. Once the node arrives at the target, it chooses another destination after the pause time. The MAC protocol uses IEEE 802.11 [11], the transmission radius is 100 m, and the node population is 200, as shown in Table 4.

6.1 The effects of anycast member population

The effects of anycast member population on the content communication and content update are shown in Fig. 4. In the intra-subnet and inter-subnet content communications, with the growth in anycast member population, the area where anycast members spread augments and the probability of anycast members becoming unreachable greatly decreases, so the distance from a consumer to the nearest anycast member is reduced and the success rate is improved. As the intra-subnet communication is included in the inter-subnet communication, it has lower costs and delays, as shown in Fig. 4a–c. The growth in anycast member population increases the content update costs and delays because the number of multicast members is equal to the one of anycast members. The intra-subnet update costs and delays are smaller than the inter-subnet ones because in the inter-subnet update process the multicast members are distributed over multiple subnets, as shown in Fig. 4d, e.

6.2 The effects of speed

The effects of speed on the content communication and update are shown in Fig. 5. In the intra-subnet and inter-subnet communications, the growth in speed augments the area where anycast members spread, so the distance between a consumer and the nearest anycast member is reduced. This results in reduction in communication costs and latency. Since the probability of anycast members becoming unreachable is relatively steady, the success rate is stable. The inter-subnet communication is built on the intra-subnet communication, so it has greater costs and latency, as shown in Fig. 5a–c. The growth in speed expands the area where multicast members spread, so the content update cost and latency grow. The inter-subnet content update cost and latency are greater because the multicast members spread over multiple subnets, as shown in Fig. 5d, e.

6.3 Comparison

CCI is compared to the anycast standard [16], Rowbee [9], the content-centric standard [12] and ACCM [22], as shown in Fig. 6. In CCI and the anycast standard, the content communication costs and delays reduce with the growth in provider population, and in Rowbee the costs and delays are steady. The primary reason is that Rowbee performs the content communications between a consumer and a target provider. CCI employs

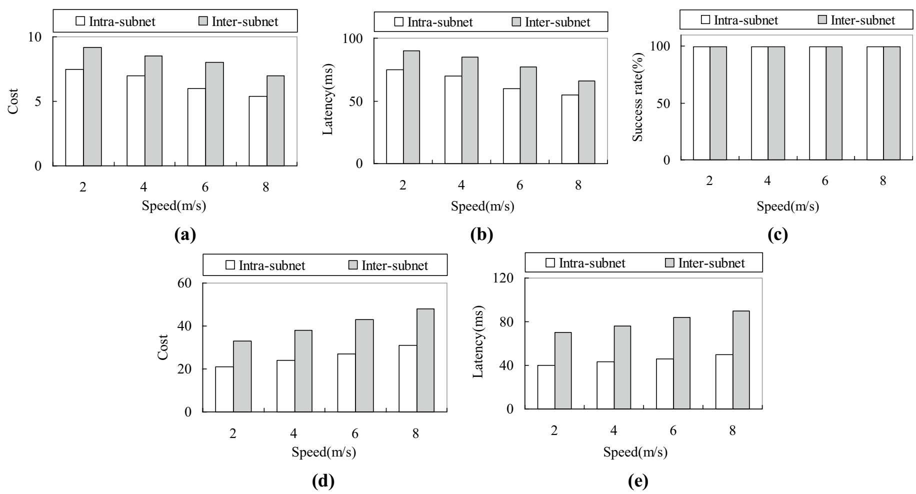


Fig. 5 Effects of speed

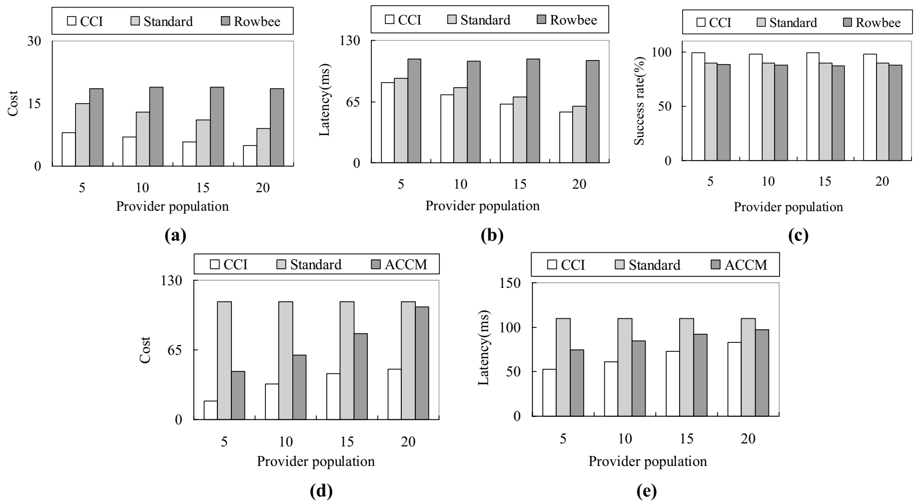


Fig. 6 Comparisons

PRT to perform content communications, so the cost and latency are minimal. The success rates in CCI, the anycast standard and Rowbee are stable, as shown in Fig. 6a–c. With the increase in provider population, the content update costs and delays in CCI and ACCM grow whereas the ones in the content-centric standard are steady. However, the costs and latency in the content-centric standard are greater than the ones in CCI and ACCM as the content-centric standard employs flooding to perform the content update. In ACCM, each anycast member independently performs the content update while in CCI the anycast members achieve the content update via one content update procedure. Consequently, the content update cost and latency in CCI are smaller than the ones in ACCM, as shown in Fig. 6d, e.

7 Conclusion and future works

In this paper, we look into the advantages and limitations of the content-centric solutions. Based on the advantages, we exploit the content-centric mechanism to perform IoT-based content delivery. To overcome the limitations, we employ anycast to achieve the content-centric mechanism in IoT, and propose CCI to reduce content communication costs and improve success rates.

This work aims to perform efficient content communications in the Internet environments. In some cases, mobile nodes such as vehicles form an ad hoc network that might not support the Internet connectivity due to the lack of infrastructures such as AR. In our future work, we plan to exploit resources of mobile nodes to enhance efficiency of content communications in the network without the Internet support.

Funding This work is supported by the CERNET Innovation Project under Grant No. NGII20170106.

References

1. Ahmed E, Gharavi H (2018) Cooperative vehicular networking: a survey. IEEE Trans Intell Transp Syst 19(3):996–1014
2. Ahmed SH, Bouk SH, Yaqub MA, Kim D, Song H (2019) DIFS: distributed interest forwarder selection in vehicular named data networks. IEEE Trans Intell Transp Syst 19(9):3076–3080
3. Bastos IV, Moraes IM (2019) A diversity-based search-and-routing approach for named-data networking. Comput Netw 157:11–23
4. Bouk SH, Ahmed SH, Park KJ, Eun Y (2019) Interest broadcast suppression scheme for named data wireless sensor networks. IEEE Access 7:51799–51809
5. Cadger F, Curran K, Santos J, Moffett S (2013) A survey of geographical routing in wireless ad-hoc networks. Commun Surv Tutor IEEE 15(2):621–653
6. Carofiglio G, Gallo M, Muscariello L (2012) ICP: design and evaluation of an interest control protocol for content-centric networking. In: Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on. IEEE, pp 304–309
7. Dou Z, Wang X, Li Y (2019) Coordinate-based addressing for MANET. Telecommun Syst 71(1):121–139
8. Gao D, Lin H, Liu X (2016) Routing protocol for k-anycast communication in rechargeable wireless sensor networks. Comput Standards Interfaces 43:12–20
9. Gao D, Zhang S, Zhang F, He T, Zhang J (2019) Rowbee: a routing protocol based on cross-technology communication for energy-harvesting wireless sensor networks. IEEE Access 7:40663–40673
10. Han SY, Lee D (2013) An adaptive hello messaging scheme for neighbor discovery in on-demand MANET routing protocols. IEEE Commun Lett 17(5):1040–1043
11. IEEE 802.11 Working Group (2016) Part11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications. ANSI/IEEE Std. 802.11
12. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL (2012) Networking named content. Commun ACM 55(1):117–124
13. Khelifi H, Luo S, Nour B, Moungla H, Faheem Y, Hussain R, Ksentini A (2020) Named data networking in vehicular Ad hoc networks: state-of-the-art and challenges. IEEE Commun Surv Tutor 22(1):320–351
14. Kostin AE, Fanaeian Y, Al-Wattar H (2016) Anycast tree-based routing in mobile wireless sensor networks with multiple sinks. Wireless Netw 22(2):579–598
15. Li H, Ota K, Dong M (2018) The content-centric mechanism: orchestration of edge-centric computing and content-centric networking in the 5G radio access network. IEEE Wirel Commun 25(3):88–93
16. McPherson D, Oran D, Thaler D, Osterweil E (2014) Architectural considerations of IP anycast. RFC 7094.
17. Ortega V, Bouchmal F, Monserrat JF (2018) Trusted 5G vehicular networks: blockchains and content-centric networking. IEEE Veh Technol Mag 13(2):121–127
18. Rezaeifar Z, Wang J, Oh H, Lee SB, Hur J (2019) A reliable adaptive forwarding approach in named data networking. Futur Gener Comput Syst 96:538–551
19. Su Z, Xu Q (2015) Content distribution over content centric mobile social networks in 5G. IEEE Commun Mag 53(6):66–72
20. Wang X (2019) Content-centric networking for mobile networks. Wirel Pers Commun 109(1):89–110
21. Wang D, Wang X (2020) Content-centric Framework over the Internet Environments. Wirel Pers Commun 116:2135
22. Wang X, Zhu X (2018) Anycast-based content-centric MANET. IEEE Syst J 12(2):1679–1687
23. Wu J, Dong M, Ota K, Li J, Guan Z (2019) FCSS: Fog computing based content-aware filtering for security services in information centric social networks. IEEE Trans Emerg Top Comput 7(4):553–563
24. Xiaonan W, Haili H, Hongbin C, Rong Z (2015) A scheme for connecting vehicular networks to the Internet. Trans Emerg Telecommun Technol 26(5):836–850
25. Xu J, Ota K, Dong M (2020) Fast networking for disaster recovery. IEEE Trans Emerg Top Comput 8(3):845–854

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.