Computer Networks 131 (2018) 38–51

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Energy-efficient ICN routing mechanism with QoS support

Xingwei Wang[a], Jianhui Lv[b,*], Min Huang[c], Keqin Li[d], Jie Li[e], Kexin Ren[b]

[a] College of Software, Northeastern University, Shenyang 110169, China
[b] College of Computer Science and Engineering, Northeastern University, Shenyang 110169, China
[c] College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
[d] Department of Computer Science, State University of New York, New York 12561, USA
[e] Department of Computer Science, University of Tsukuba, Tsukuba Science City, Ibaraki 305-8573, Japan

ARTICLE INFO

Article history:
Received 19 June 2017
Revised 5 November 2017
Accepted 9 December 2017
Available online 13 December 2017

Keywords:
Information-Centric Networking
Quality of Service
Energy efficiency
Cauchy distribution
Routing

ABSTRACT

Information-Centric Networking (ICN) brings a promising networking paradigm which changes host-centric communication mode, and its routing decision depends on the unknown and named content item rather than the known IP address. In this paper, we propose a novel Energy-efficient Quality of Service (QoS) Routing mechanism for ICN (EQRI). Firstly, we evaluate the suitability of link state to user's QoS requirements by Cauchy distribution model and formulate the energy efficiency of link by monitoring the corresponding traffic. Secondly, we design a priority determination strategy based on QoS and energy efficiency, a color management strategy to assign color for outgoing interface, and a backtracking strategy to cope with the failed Interest packet. Thirdly, we propose a link selection algorithm based on color management, priority determination and backtracking strategy. Finally, we devise an ICN routing mechanism which consists of Interest packet routing and Data packet routing. The experimental results show that EQRI not only retrieves the content effectively but also outperforms existent mechanisms.

1. Introduction

The Internet has played an important role in promoting the transition of economy, technology and society from industrial era to information era. However, it is widely agreed that the current Internet is very complicated and thus difficult to manage and change [1]. Nowadays, the Internet users have been mainly interested in accessing information irrespective of its physical location, which makes the current Internet design with a host-centric communication mode to become inadequate for the newly emerging usage patterns [2]. In addition, the current Internet has become considerably difficult to support the unexpected increasing demands on content distribution, where the corresponding applications (*e.g.*, YouTube) are found to increasingly generate more traffic compared with the traditional applications (*e.g.*, web browsing) [3], having greatly motivated the development of some new networking paradigms based on the named data objects (*e.g.*, web pages, videos, documents or other pieces of information [4]). As a result, the studies on future Internet in the United States, European Union, Japan and China [5] have been proposed to promote a clean-slate networking paradigm to replace IP-centric one [6]. Information-Centric Networking (ICN) is a promising candidate, in which a user retrieves the content by its name [7]. ICN aims to provide a network infrastructure service which is more suitable for distributing and retrieving the content efficiently, and it is capable of overcoming some limitations of the current Internet, such as content distribution [4] and mobility [8]. As a matter of fact, ICN shows that what is being exchanged is more important than which network entities are exchanging information, that is, it focuses on the content rather than IP address [9]. In particular, the content is an abstract notion, and it can be any type of object, such as real-time multimedia stream, service or even network entity [6]. At present, a lot of ICN-related projects have been widely studied, such as Content-Based Networking (CBN) [10], Combined Broadcast and Content-Based Networking (CBCBN) [11], Data Oriented Network Architecture (DONA) [12], Network of Information (NetInf) [13], Publish/Subscribe Internet Technology (PURSUIT) [14], Service-Centric Networking (SCN) [15], Named Data Networking (NDN) [16] and Content-Centric Networking (CCN) [17]. As we know, NDN/CCN architecture is the most popular paradigm, thus our study relies on NDN/CCN rather than others.

Regarding ICN-related researches, naming, caching and routing schemes have attracted more and more attention, because ICN inherently relies on the location-independent naming, in-network caching and name-based routing for effective content retrieval. Among them, the naming aims to identify the named information object, and the content name is persistent, available and au-

* Corresponding author.
E-mail address: lvjianhui2012@163.com (J. Lv).

https://doi.org/10.1016/j.comnet.2017.12.002

thentic wherever it is [3]. The caching aims to deploy in-network storage attached to each router, where the cached copy can satisfy content request, facilitating routing efficiency especially when the same content is requested by multiple users. The routing is to select the appropriate outgoing interface to forward interest request by adopting the distributed forwarding model while guaranteeing route loop-free [14]. Particularly, the content retrieval in ICN is the most important, and it depends on the corresponding routing. As a result, ICN routing has drawn much attention, and many interesting and effective solutions have been proposed.

With various kinds of networked applications emerged and user's demands diversified, Quality of Service (QoS) support becomes critical to the network, so that their users can get high Quality of Experience (QoE) [18]. For example, video conference usually asks for high bandwidth and small delay, and it can tolerate occasional packet loss; video on demand usually asks for high bandwidth and low delay jitter, and it can tolerate relatively large delay (because large delay only means that the user begins to view video playout after long time waiting, as long as the delay keeps almost constant) and occasional packet loss; bulk data transfer usually asks for high bandwidth and no error, and it can tolerate relatively large delay and delay jitter; email asks for no error, while it can tolerate arbitrary bandwidth, delay and delay jitter. Compared with the traditional best-effort model, providing QoS support is more essential in all networks. Especially, the application types in ICN have been far more than that in the current Internet. Given this, ICN should also consider QoS support when designing its routing scheme.

With the explosive growth of network scale and the amount of networked devices, the network energy consumption increases sharply. The power consumption of Information and Communication Technologies (ICT) sector and the corresponding greenhouse gas emissions are expected to reach 95 GW [19] and 1.43 $GtCO_2e$ in 2020 [20] respectively if no energy-efficient networking technologies used. Thus, the energy saving in ICT sector, especially in networking, has recently become a major issue in the current and even future Internet [21]. Furthermore, in 2010, the Internet Engineering Task Force (IETF) set up a working group regarding energy management [22], revealing that the energy-efficient networking is becoming increasingly important. At present, the proposed energy-efficient methods are usually divided into two classes: one is on device-level, which provides the energy-efficient network devices and terminals; the other is on system-level, which organizes the usage of devices and terminals energy-efficiently in a network-wide manner. The energy-efficient routing algorithm belongs to system-level energy saving scheme, which aims to find the most energy-efficient route for users and thus reduce energy consumption when transferring traffic along the route. In addition, as far as we know, the energy saving efficiency from the perspective of algorithm level is expected to reach 75% [23]. There is no doubt that ICN also needs to save energy and the energy-efficient routing is essential.

The major contribution of this paper is to propose a novel Energy-efficient QoS Routing mechanism for ICN (EQRI). (i) The improved cauchy distribution model is used to evaluate the suitability of link state to user's QoS requirements and formulate the energy efficiency of link by monitoring the corresponding traffic, and further the priority of outgoing interface is determined based on QoS and energy efficiency. (ii) A color management strategy is proposed in order to select the proper outgoing interface to forward Interest packet efficiently. (iii) To reduce network overhead while guarantee Interest packet being forwarded effectively, a backtracking strategy is proposed to cope with the failed Interest packet. (iv) An ICN routing mechanism is proposed, which consists of Interest packet routing and Data packet routing.

The rest of this paper is organized as follows. The related work is reviewed in Section 2. In Section 3, the system framework of EQRI is introduced. Section 4 evaluates QoS and energy efficiency. The link selection scheme is proposed in Section 5. In Section 6, the routing decision is done. Section 7 shows the experimental results. Finally, Section 8 concludes this paper.

2. Related work

A number of researches paid attention to ICN routing, for example, simple interest forwarding and introducing bloom filter, Ant Colony Optimization (ACO), in-network caching and Software-Defined Networking (SDN). Regarding simple interest forwarding, in [24], some alternatives of blind routing algorithms were proposed to solve the problem of which Forwarding Information Base (FIB) miss occurs. In [25], a routing algorithm based on dynamic interest forwarding was proposed to discover path to the content copy not addressed in routing table. It forwarded interest request for such content towards the best outgoing interface at each hop. Regarding bloom filter, in [26], a scalable routing algorithm based on bloom filter was proposed to exchange the cached contents in order to address scalability issue. Besides, bloom filter in [27] was used to minimize signaling overhead and flooded interests. Regarding ACO, in [28], a service routing algorithm was proposed, which supported the integration of CCN and SCN. In addition, other ACO based ICN routing schemes were also investigated in [29–31]. Regarding in-network caching, in [32], a smart routing scheme based on cache capacity aware was proposed. It first used a selective caching method to balance caching load among routers and reduce caching redundancy dramatically. Then, it provided a scalable method to maintain routing information for the cached contents. In [33], five different hash based routing schemes were proposed to efficiently exploit in-network caching without requiring routers to maintain the state information of content. Regarding SDN, in [34], a routing scheme with the assistance of SDN was proposed, which separated control plane from data plane. However, the above ICN routing proposals neither provided QoS support nor considered energy saving.

There were also some ICN routing schemes with QoS support. In [35], three QoS aware path selection algorithms were proposed based on ACO. The first one was to retrieve the content from the multiple sources simultaneously while maximizing aggregate bandwidth, the second one was to select the highest bandwidth path to a single source for high bandwidth application, and the last one was to minimize per-packet delay for real-time traffic. In [36], the problem of meeting network delay requirements for the differentiated services was investigated. It solved delay guarantee task as a nonlinear integer programming problem. In [37], QoS was considered to make forwarding decision to adapt to network conditions and user preferences. In [38], a context aware forwarding plane with QoS support was proposed. It examined a typical scenario which encompasses different user applications with varying demands. However, [35–38] did not consider energy saving. In addition, different from them, this paper considered QoS from bandwidth, delay and error rate, and further used the fuzzy mathematical theory to do QoS evaluation.

Although a lot of ICN researches with energy saving were studied, they did not belong to ICN routing category, or they were beyond the pure ICN paradigm, for example, under or incorporating wireless network [21]. In spite of this, a few energy-efficient ICN routing researches were developed. In [39], an energy aware routing algorithm was proposed. It formulated the model of energy saving in traffic routing to achieve link rate adaption and predicted traffic to preserve network stability, in which the contents were placed at the edge routers to decrease traffic going through the backbone. In [40], an invention patent on energy-efficient con-

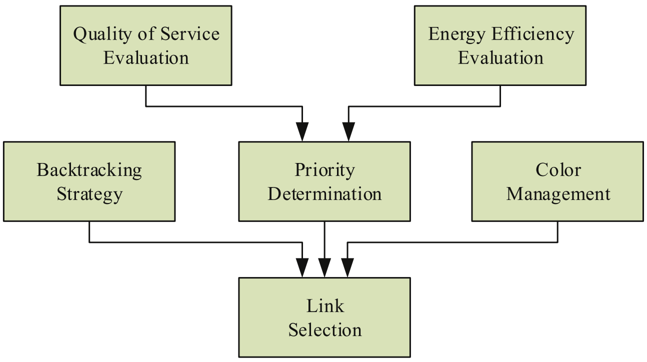


Fig. 1. The system framework of EQRI.

tent retrieval strategy was proposed. It used mobile device and content retrieval agent to achieve interest forwarding. In [41], an energy-efficient scheme by turning off redundant Content Routers (CRs) and links was proposed. It formulated energy consumption management problem as a mixed integer linear programming and proposed a centralized solution via spanning tree heuristic. However, [39–41] only considered energy saving irrespective of providing QoS support.

With respect to energy-efficient ICN routing scheme with QoS support, in [42], a green domain and ACO based forwarding scheme was proposed. It first used popularity-inspired caching to decrease power consumption, and then designed ACO based interest forwarding to support QoS. Although [42] considered the combination of QoS and energy efficiency, its QoS evaluation method and formulation of energy saving needed to be improved further, which motivated the proposed EQRI in this paper.

3. System framework

3.1. Framework

In this paper, we propose a novel ICN routing mechanism to accomplish content retrieval, called EQRI, where both QoS and energy efficiency are considered. As depicted in Fig. 1, EQRI consists of six main modules, i.e., QoS Evaluation (QE), Energy efficiency Evaluation (EE), Backtracking Strategy (BS), Priority Determination (PD), Color Management (CM) and Link Selection (LS). Among them, QE is used to evaluate the suitability of link state to user's QoS requirements; EE is used to evaluate the energy efficiency of link by monitoring the corresponding traffic; PD is used to determine the priority of outgoing interface according to the outputs of QE and EE; CM is used to mark the color of outgoing interface; BS is used to cope with the failed Interest packet; LS is used to select an outgoing interface to forward Interest packet according to the outputs of BS, PD and CM.

EQRI has two kinds of messages, that is, Interest packet used to request the content and Data packet used to return the corresponding content to interest requester. As shown in Fig. 2, the structure of Interest packet consists of content name, Nonce, Time To Live (TTL), application type and Current CR (CCR). Among them, Nonce is used to identify Interest packet, that is, each Interest packet carries a random Nonce value generated by interest requester. A CR remembers both name and Nonce of each received Interest packet, and it has ability to check whether a newly arrived Interest packet carrying the same name as a previously received Interest packet from a different interest requester, or a previously forwarded Interest packet looped back. As a result, the closed loop problem of Interest packet is addressed. TTL means the survival time of Interest packet. Application type reflects the type of user's interest request. CCR is the latest location where Interest packet is. The structure of Data packet consists of content name and the corresponding content. In particular, two Interest packets are same means that they have same content name and Nonce. Meanwhile, multiple Interest packets with same content name and different Nonce values occupy multiple slots, but they belong to the same item no. field, which saves space overhead from the perspective of two fields in PIT, i.e., item no. and content name, and thus saves time overhead from the perspective of lookup.

3.2. Network model

We model ICN network as an undirected connected graph, denoted by $G=(V,E)$, where V and E are node set and link set respectively. Each node is regarded as one CR which consists of Content Store (CS), Pending Interest Table (PIT) and FIB. Among them, CS serves as in-network cache attached to CR to provide the content copy, PIT tracks incoming interface where pending Interest packet arrives, and FIB maps content name to outgoing interface in order to forward Interest packet. The structures of CS, PIT and FIB are shown in Fig. 3. In PIT, Nonce is used to distinguish a specific Interest packet from those with the same interest requests. In FIB, counter is used to determine the color of outgoing interface, and priority and color are used to select an outgoing interface to forward Interest packet.

In this paper, the link state is described by bandwidth, delay and error rate, and the interval is used to depict link state because of its difficulty for precise measurement. Specifically, $[B_L, B_H]$ denotes the available bandwidth of link, and $[D_L, D_H]$ and $[E_L, E_H]$ denote the tolerant delay and error rate of link respectively. Each link has two states, that is, the working state in which the link can be used to forward Interest packet and the closed state is just the opposite.

The user request is expressed as a tuple $<$ *Content name, Application type* $>$. Each application type corresponds to a specific set of QoS requirements, and it is mapped to a tuple $<$ *bw, del, era* $>$ which denote requirements on bandwidth, delay and error rate respectively, here $bw \in [B_L, B_H]$, $del \in [D_L, D_H]$ and $era \in [E_L, E_H]$.

4. QoS and energy efficiency evaluation

4.1. QoS evaluation

4.1.1. Cauchy distribution

The user's QoS requirements are hard to be expressed exactly, thus we use fuzzy distribution model to evaluate the suitability of link state to user's QoS requirements, such as rectangular distribution, trapezoidal distribution, parabolic distribution, cauchy distribution model, normal distribution model and ridge distribution model [43]. Among them, the change trend of function curve in cauchy distribution model is floating, that is, the change rate is always neither increasing nor decreasing. However, those in other models are stable, that is, the change rate is always either increasing or decreasing. To this end, Cauchy Distribution Model (CDM) is used, and the corresponding evaluation functions on QoS parameters are shown in Figs. 4(a) and (b). Their values vary from 0 to 1, being consistent with the condition of the suitability of these parameters to user's QoS requirements, where Figs. 4(a) and (b) follow partial-small and partial-large distribution respectively. In particular, the points a and b in Fig. 4 are two special segmentations, and they divide evaluation function into three parts, that is, $[0, a)$, $[a, b)$ and $[b, +\infty)$.

Regarding arbitrary link between CR_i and CR_j, denoted by $e_{i,j}$, user's strong demand on bandwidth means small suitability of the available bandwidth of $e_{i,j}$ to the user's bandwidth requirement. To be specific, when the bandwidth requirement bw exceeds the maximum available bandwidth provided by $e_{i,j}$, the corresponding

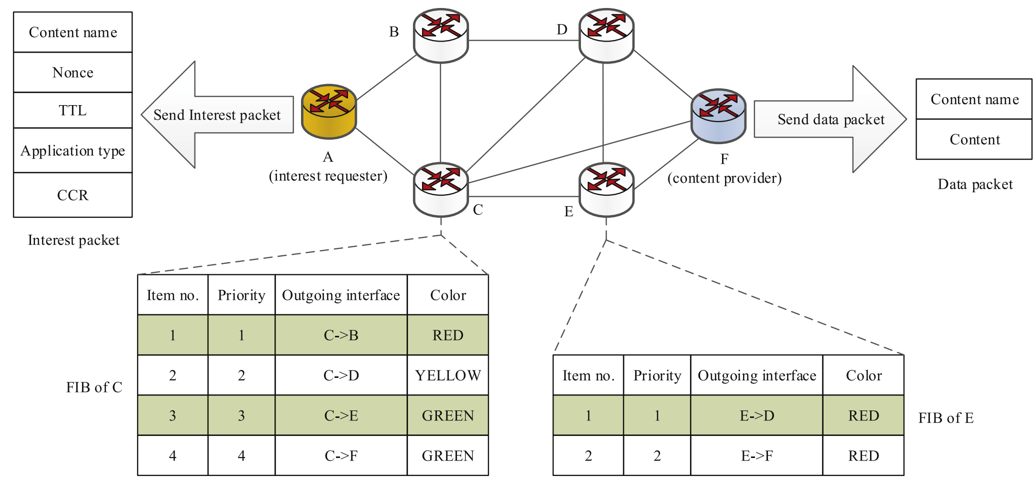


Fig. 2. The structures of Interest packet and Data packet.

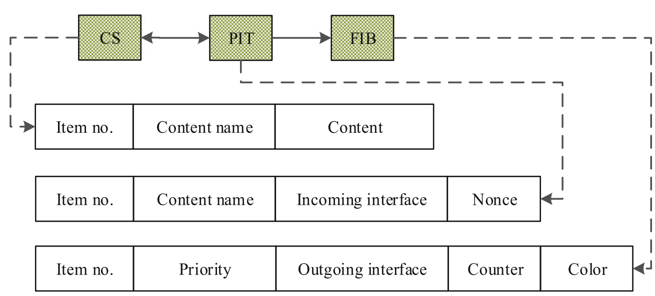


Fig. 3. The structures of CS, PIT and FIB.

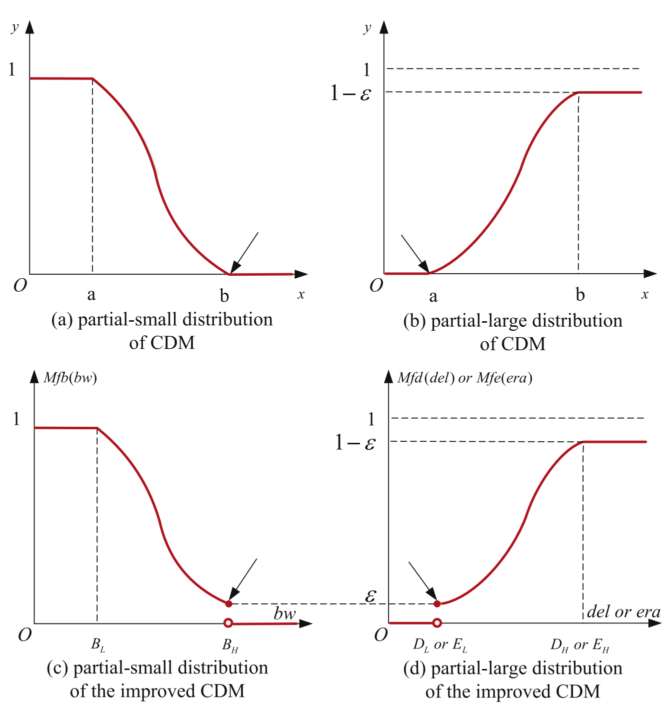


Fig. 4. CDM and improved CDM.

suitability is 0; in other words, $e_{i,j}$ has no capacity to satisfy user's bandwidth requirement. In addition, when *bw* is smaller than the minimum available bandwidth provided by $e_{i,j}$, the corresponding suitability is always 1, that is, $e_{i,j}$ has enough capacity to satisfy user's bandwidth requirement. The related bandwidth evaluation function is shown in Fig. 4(a).

Further, user's weak demand on delay (error rate) means small suitability of the tolerant delay (error rate) of $e_{i,j}$ to the user's delay (error rate) requirement. Similarly, when the delay (error rate) requirement *del* (*era*) exceeds the maximum tolerant delay (error rate) provided by $e_{i,j}$, the corresponding suitability is 1, because it does not reach the tolerance of $e_{i,j}$. In addition, when *del* (*era*) is smaller than the minimum tolerant delay (error rate) provided by $e_{i,j}$, the corresponding suitability is 0, because it is beyond the tolerance of $e_{i,j}$.

In fact, the available bandwidth provided by $e_{i,j}$ has significant influence on congestion, packet collision and traffic rate. In addition, the tolerant delay and error rate are also dynamic change, which are influenced by sending rate and sending number of packets into network, that is, bandwidth, delay and error rate are effectively multifaceted aspects. However, such dynamic cases are difficult to simulate in the practice network environment. Therefore, in this paper, these evaluation functions with respect to suitability depend on the assumption that the available bandwidth, tolerant delay and error rate are changeless within a certain period of time.

However, the evaluation value of link's available bandwidth to user's bandwidth requirement approaches 0 rather than being equal to 0 when his bandwidth requirement reaches the maximum available bandwidth of link, that is, the condition of $x = b$ in Fig. 4(a). Therefore, we improve partial-small CDM from Fig. 4(a) to (c). Similarly, the evaluation value of link's tolerant delay (error rate) to user's delay (error rate) requirement also approaches 0 rather than being equal to 0 when his delay (error rate) requirement reaches the minimum delay (error rate) of link, that is, the condition of $x = a$ in Fig. 4(b). Therefore, we improve partial-large CDM from Fig. 4(b) to (d).

4.1.2. Improved CDM based QoS evaluation

According to the improved partial-small CDM, we define the evaluation function of link's available bandwidth to user's bandwidth requirement, denoted by $Mfb(bw)$, as follows:

$$Mfb(bw) = \begin{cases} 0, & bw > B_H \\ \varepsilon, & bw = B_H \\ \left[1 + \alpha\left(bw - B_L\right)^{\beta}\right]^{-1}, & B_L < bw < B_H \\ 1, & bw \le B_L \end{cases} \quad (1)$$

where α, $\beta > 0$ and $\varepsilon \to 0$.

Similarly, we define two other evaluation functions: one is on link's tolerant delay to user's delay requirement, and one is on link's tolerant error rate to user's error rate requirement, denoted by $Mfd(del)$ and $Mfe(era)$ respectively, as follows:

$$Mfd(del) = \begin{cases} 0, & del < D_L \\ \varepsilon, & del = D_L \\ \left[1 + \alpha\left(del - D_L\right)^{-\beta}\right]^{-1}, & D_L < del < D_H \\ 1 - \varepsilon, & del \geq D_H \end{cases} \quad (2)$$

$$Mfe(era) = \begin{cases} 0, & era < E_L \\ \varepsilon, & era = E_L \\ \left[1 + \alpha\left(era - E_L\right)^{-\beta}\right]^{-1}, & E_L < era < E_H \\ 1 - \varepsilon, & era \geq E_H \end{cases} \quad (3)$$

The suitability of link state to user's QoS requirements depends on $Mfb(bw)$, $Mfd(del)$ and $Mfe(era)$, where $Mfb(bw) \in [0, 1]$ by Eq. (1), $Mfd(del) \in [0, 1)$ by Eq. (2) and $Mfe(era) \in [0, 1)$ by Eq. (3). The weighted average method is used to evaluate the suitability of link state to user's QoS requirements because $Mfb(bw)$, $Mfd(del)$ and $Mfe(era)$ have the same order of magnitude. Let *Efl_QoS* denote the comprehensive evaluation function on QoS, and it is defined as follows:

$$Efl\_QoS = \alpha_B Mfb(bw) + \alpha_D Mfd(del) + \alpha_E Mfe(era) \quad (4)$$

where α_B, α_D and α_E are the weights with respect to bandwidth, delay and error rate respectively, reflecting their relative importance to user's QoS requirements. In particular, $0 < \alpha_B, \alpha_D, \alpha_E < 1$ and $\alpha_B + \alpha_D + \alpha_E = 1$.

We present a theorem about the performance of *Efl_QoS* as follows.

Theorem 1. *To get the best performance of EQRI, Efl_QoS is equal to* $\frac{1}{2} + \frac{\alpha_B - \alpha_D - \alpha_E}{2\beta}$. *The related proof is found in Appendix A.*

4.2. Energy efficiency evaluation

In this paper, energy efficiency is used to measure energy saving, and it is defined as the number of bits which can be successfully transmitted with the unit energy consumption. As a matter of fact, a link, no matter wired or wireless, has no any energy consumption; instead, configuration and traffic information from network devices consume energy [44]. In other words, the energy consumption of link is used to express the interface's energy consumption because each outgoing interface corresponds to one link. However, in order to display the consistency with the suitability of link state, we use the energy consumption of link rather than that of outgoing interface. Inspired by Mahadevan et al. [44], the energy consumption depends on the corresponding traffic and configuration. Although different types of network devices have different configurations, which causes different energy consumption models, this paper only considers traffic irrespective of configuration for convenient modeling and implementation. In particular, with respect to traffic, Mahadevan et al. [44] builds a linear model for power consumption which is linearly proportional to energy consumption, thus we consider the relationship between energy consumption and traffic as linearity. Let *Ecl_energy* denote the energy consumption function of link, and it is defined as follows:

$$Ecl\_energy = \begin{cases} \sigma + \lambda \cdot tra, & link\ is\ working \\ \sigma, & otherwise \end{cases} \quad (5)$$

where σ means the fixed energy consumption on condition that link is working, λ is the consumed energy by processing per bit traffic, *tra* is the transmitted traffic.

Then, let *Eel_energy* represent the energy efficiency function of link, and it is defined as follows:

$$Eel\_energy = \begin{cases} \dfrac{\lambda \cdot tra}{\sigma + \lambda \cdot tra}, & link\ is\ working \\ \dfrac{\lambda \cdot tra}{\sigma}, & otherwise \end{cases} \quad (6)$$

By now, two evaluation values, i.e., the suitability of link state to user's QoS requirements and the energy efficiency of link, are obtained, and they are used to determine the priority of outgoing interface.

5. Link selection

When CR_i receives an Interest packet *inp*, it searches CS_i, PIT_i and FIB_i, here $1 \leq i \leq n$ and n is the number of CRs. If FIB_i is searched, it means that an appropriate outgoing interface will be selected to forward *inp*. The link selection depends on color and priority of outgoing interface. Specifically, a backtracking strategy is devised to cope with the failed *inp*.

5.1. Color management

As mentioned above, FIB has color field which is used to mark outgoing interface state. To check the availability of outgoing interface conveniently, besides using the extended OSPF protocol to monitor and distribute link state [45], we mark the outgoing interface as three colors, i.e., *RED, YELLOW* or *GREEN,6*. In particular, when an FIB entry is not available, the corresponding color of outgoing interface is modified as RED, instead of deleting this entry from FIB. This way which adjusts the color of outgoing interface can save much more time due to the frequent addition and deletion of FIB entries. *RED* means that outgoing interface cannot be used to forward Interest packet; *YELLOW* and *GREEN* means that outgoing interface can be used to forward Interest packet, while outgoing interface in *GREEN* has higher priority than that in *YELLOW*.

As shown in Fig. 3, each outgoing interface is equipped with a counter and its value is denoted by *count*. Initially, all outgoing interfaces are marked in *YELLOW*, and their *count*s are set as 0. However, during the following forwarding, some interest requests may be failed frequently, in order to ensure the fair forwarding, GREEN or YELLOW interface should be changed to RED. To this end, in terms of one outgoing interface, if it is not used to forward Interest packet, its *count* remains unchanged; if the content returns within TTL, its *count* is increased by 1, otherwise its *count* is decreased by 1 each time. According to the above statements, the transition relation among *RED, YELLOW* and *GREEN*, i.e., the color determination of outgoing interface, is defined as follows:

$$color := \begin{cases} RED, & count < -\Omega \\ YELLOW, & -\Omega \leq count \leq 0 \\ GREEN, & count > 0 \end{cases} \quad (7)$$

where Ω is a threshold to indicate the permitted maximum failure times of interest forwarding.

5.2. Priority determination

LS gets *Efl_QoS* and *Eel_energy* from QE and GE respectively, which are used to determine the priority of outgoing interface. Let *Dfl_pri* denote the priority determination function, and it is defined as follows:

$$Dfl\_pri = Efl\_QoS \cdot Eel\_energy \quad (8)$$

Apparently, outgoing interface with large *Dfl_pri* has high priority. In this paper, we use the quick sort method to sort these priority values in descending order, where outgoing interface with high priority is assigned as small number.

We present a definition about the optimal outgoing interface as follows.

Definition 1 (the optimal outgoing interface)**.** If an outgoing interface meets that (i) its color is *YELLOW* or *GREEN* and (ii) its *Dfl_pri* is the highest among all outgoing interfaces, then it is regarded as the optimal one.

For example, in Fig. 2, when *inp* arrives at *C*, the third entry in FIB is selected, where *inp* is forwarded via "$C \rightarrow E$". According to CM, only "$C \rightarrow E$" and "$C \rightarrow F$" can be used to forward *inp*; however, according to PD, "$C \rightarrow E$" is finally selected to forward *inp* due to its higher priority.

5.3. Backtracking strategy

In terms of one CR, when all outgoing interfaces are marked in *RED*, it means that the CR is not able to forward Interest packet. Under such case, the original ICN generates and send a new interest request [6]; instead, this paper devises a backtracking strategy to return the failed Interest packet to the previous CR in order to reduce network overhead. We present a definition about the previous CR as follows.

Definition 2 (the previous CR)**.** $\forall$ CR_i and CR_j, here $i \neq j$, if (i) there exists an edge between CR_i and CR_j and (ii) CR_i receives *inp* from CR_j, then CR_j is the previous CR of CR_i.

Consider that *inp* is forwarded along CR_k, CR_j and CR_i in turn, here $i \neq j \neq k$. When CR_i is not capable of forwarding *inp*, *inp* is backtracked to CR_j and another outgoing interface of CR_j is selected to forward *inp*. According to the statement, we know that a single CR can send multiple and same Interest packets, that is, it is possible to send multiple *inp* from CR_j. If the remaining outgoing interface(s) of CR_j cannot be used to forward *inp*, *inp* is backtracked to CR_k; otherwise, the *YELLOW* or *GREEN* one with the highest priority is used to forward *inp*, and it is regarded as the suboptimal outgoing interface. For example, in Fig. 2, *C* selects "$C \rightarrow E$" to forward *inp*; however, the colors of all outgoing interfaces in *E* are *RED*, thus *inp* is backtracked to *C* and it is forwarded via "$C \rightarrow F$".

We design a Link Selection algorithm based on CM, PD and BS (LSCPB) to forward Interest packet, and it is described in Algorithm 1, where lines 11–17 are to determine how to process Interest packet.

Algorithm 1 LSCPB algorithm.

Input: An Interest packet (*inp*)
Output: An outgoing interface
Begin
01: CR_i receives *inp* from CR_j;
02: **for** $i = 1$ to m, **do** // m is the number of outgoing interfaces;
03: Evaluate QoS according to Eq. (4);
04: Evaluate energy efficiency according to Eq. (6);
05: Determine priority according to Eq. (8);
06: **end for**
07: Sort m priority values in descending order;
08: **for** $i = 1$ to m, **do**
09: Assign a number to outgoing interface;
10: **end for**
11: **if** CR_i has been traversed, **then**
12: **Return** the suboptimal outgoing interface;
13: **else**
14: **if** CR_i is capable of forwarding *inp*, **then**
15: **Return** the optimal outgoing interface;
16: **else**
17: Backtrack to CR_j;
18: **end if**
19: **end if**
End

6. Routing decision

6.1. Interest routing

When CR_i receives *inp*, it searches its CS to see whether the content exists. If yes, the forwarding of *inp* is finished; as a result, the content is sent back to the incoming interface of *inp*. If the content cannot be found in CS, CR_i checks its PIT to see whether there is another Interest packet *jnp* with the same interest request as *inp*.

If the matched item is found in PIT, CR_i checks whether the link to which the incoming interface of *jnp* corresponds can meet user's QoS requirements. If yes, CR_i adds the incoming interface of *jnp* to PIT and *jnp* waits the content. If the content is returned within TTL, the content is sent back to the incoming interface of *jnp*; otherwise, *inp* is backtracked to the previous CR.

If the matched item cannot be found in PIT or the link to which the incoming interface of *jnp* corresponds cannot meet user's QoS requirements, CR_i searches its FIB to forward *inp* based on LSCPB algorithm. In this case, if the optimal outgoing interface is found,

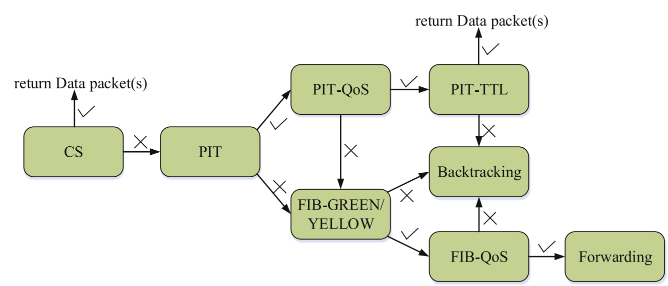


Fig. 5. Interest routing process.

CR_i checks whether the link to which the optimal outgoing interface corresponds can meet user's QoS requirements. If yes, *inp* is forwarded via the optimal outgoing interface. If the optimal outgoing interface cannot be found or the link to which the optimal outgoing interface cannot meet user's QoS requirements, *inp* is backtracked to the previous CR and the suboptimal outgoing interface is selected to forward *inp*. Specifically, if *inp* is forwarded from CR_i to CR_j rather than other CRs, then in order to save energy further, the CRs which have no probability to receive *inp* are switched off while CR_j is switched on.

The Interest routing algorithm is described in Algorithm 2 and depicted in Fig. 5, where lines 8 and 15 are to check whether the obtained outgoing interface can meet user's QoS requirements.

6.2. Data routing

In this paper, Data packet(s) is(are) sent to interest requester along the same path which is reverse with that of Interest packet. When CR_i finds the content in its CS or waits the content within TTL, Data packet(s) is(are) generated to carry the content back to interest requester and then CR_i searches its PIT to find the incoming interface from which the corresponding Interest packet has been received. If the matched item is found in PIT, CR_i does three operations: (i) sends Data packet(s) to the incoming interface; (ii) puts the content into CS of the previous CR; (iii) deletes the corresponding PIT item about the incoming interface. If the

Algorithm 2 Interest packet routing algorithm.

Input: An Interest packet (*inp*)
Output: Data packet(s) or failure
Begin
CR_i receives *inp*;
if CR_i has forwarded *inp*, **then**
  Return failure;
else
  if the content is found in CS, **then**
    Return Data packet(s);
  else if the matched item is found in PIT
    if user's QoS requirements is met, **then**
      if the content cannot be returned within TTL, **then**
        Backtrack *inp* to its previous CR;
      else
        Return Data packet(s);
      end if
    end if
  else // CR_i searches its FIB;
    Select the optimal outgoing interface by LSCPB;
    if user's QoS requirements is met, **then**
      Forward *inp* via the optimal outgoing interface;
      Switch off the CRs which have no probability to receive *inp*;
    else
      Backtrack *inp* to its previous CR;
    end if
  end if
end if
End

matched item cannot be found in PIT, Data packet(s) is(are) discarded. Meanwhile, if CS is not full, the content is cached; otherwise, the First In First Out (FIFO) replacement strategy is performed.

The Data routing algorithm is described in Algorithm 3.

Algorithm 3 Data packet routing algorithm.

Input: Data packet(s)
Output: Null
Begin
if the matched item is found in PIT, **then**
  Send Data packet(s) to the incoming interface;
  if CS is not full, **then**
    Put the content into CS;
  else
    Perform FIFO replacement strategy;
  end if
  Delete PIT item;
Return Null;
end if
End

7. Performance evaluation

7.1. Simulation setup

7.1.1. Experiment method

We compare EQRI with four well-known algorithms, i.e., NDN Forwarding mechanism (NDNF) [6], QoS Aware Path Selection mechanism [35], Energy Aware Traffic Routing mechanism (EATR)

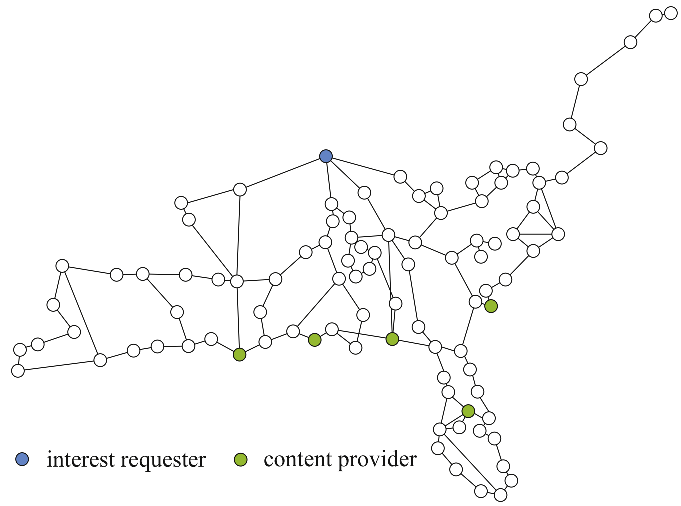


Fig. 6. 2010-USA-Deltacom topology with 97 nodes and 124 edges.

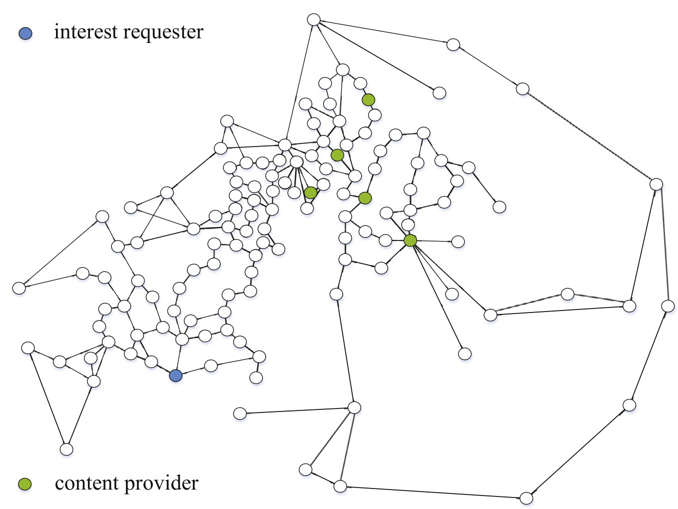


Fig. 7. 2010-Europe-GTS topology with 130 nodes and 168 edges.

[39] and ACO based forwarding with Domain Partition (ADP) [42] based on 100 times simulations. In order to make the comparison as fair as possible, we integrate QoS and energy efficiency into NDNF, energy efficiency into QAPS, and QoS into EATR respectively. Then, Average Routing Success Rate ($\mathbb{ARSR}$), Average Hop Count ($\mathbb{AHC}$), Average Delay ($\mathbb{AD}$), Average Throughput ($\mathbb{AT}$), Average Energy Efficiency ($\mathbb{AEE}$) and stability are considered as six evaluation metrics over Deltacom topology (with 97 nodes and 124 edges) in Fig. 6 and GTS topology (with 130 nodes and 168 edges) in Fig. 7. For two topologies, they have 1 interest requester and 5 content providers, where bandwidth, delay and error rate are set as 1Gbps, 30ms and 0.01 respectively. In addition, we generate 500, 600, 700, 800, 900 and 1000 interest requests and send them into network following uniform distribution. Moreover, we implement these three algorithms by C++ programming language; meanwhile, the test environment is set up on a personal computer with the Intel (R) i5-4590, 3.30 GHZ CPU, 4G RAM over Windows 7 system.

7.1.2. Parameter setting

At first, we capture data from Sohu website every day for one hour during one week. Then, we extract the names of interest requests from these HTTP requests and use them as experimental data, where the volume of data is 100GB. In particular, the traffic is simulated via network bandwidth, and regarding its measure-

Table 1
Simulation parameter settings.

Parameter	α	β	α_B	α_D	α_E	λ	σ
Setting	1	1.0001	0.8	0.1	0.1	0.3	1KJ
Parameter	counter (Ω)	CS size	PIT size	FIB size	buffer size	packet size	TTL
Setting	6	1.5G	200MB	500MB	0.5GB	128B	64

Table 2
Average QoS evaluation values for EQRI.

The number of interest requests	500	600	700	800	900	1000
QoS evaluation value in Deltacom	0.817081	0.799999	0.816478	0.799963	0.781178	0.808701
Difference in Deltacom	0.017111	0.000029	0.016508	0.000007	0.018792	0.008731
QoS evaluation value in GTS	0.804752	0.813006	0.798642	0.815537	0.808631	0.810574
Difference in GTS	0.004782	0.013036	0.001328	0.015567	0.008661	0.010604

ment within a period of time, we use sFlow [46] to monitor the real-time change of bandwidth and convert to the corresponding traffic. For network parameters and computation parameters, their settings are shown in Table 1. Meanwhile, each CR stores 500 content items by adopting FIFO replacement strategy. Especially, this paper does not pay attention to caching, thus the buffer size is set to be sufficient, that is, 0.5GB. In addition, the best performance of EQRI is irrelevant to the setting of α according to Theorem 1, thus α is set as 1 for simple computation. The initial energy availability of CR is considered and set as 350J, and it is only used to accomplish CR startup.

7.2. QoS test for EQRI

In this section, we test average QoS evaluation values for EQRI over Deltacom and GTS, and the corresponding results are shown in Table 2. These average evaluation values are around 0.79997 and the corresponding differences are within 0.02. As a matter of fact, we have proved that EQRI can reach the best performance in case of $Efl\_QoS = \frac{1}{2} + \frac{\alpha_B - \alpha_D - \alpha_E}{2\beta}$ in Theorem 1. Putting the simulation parameters ($\beta = 1.0001,\ \alpha_B = 0.8,\ \alpha_D = 0.1$ and $\alpha_E = 0.1$) into it, and $Efl\_QoS = 0.79997$, which indicates that the suitability of link state to user's QoS requirements can reach 0.79997. To some extent, these average QoS evaluation values are acceptable. In addition, the experimental results are basically consistent with the proposed Theorem 1, which reveals the integration of theory and practice.

7.3. Average routing success rate

The routing success rate is defined as the ratio of the number of interest requests which have successfully retrieved the content to the total number of interest requests. The $\mathbb{ARSR}$s for EQRI, NDNF, QAPS, EATR and ADP are reported in Fig. 8.

Over the same topology, we observe that ADP has the highest $\mathbb{ARSR}$, followed by EQRI, NDNF, QAPS and EATR, especially EQRI and ADP have the virtually identical $\mathbb{ARSR}$. For ADP, it sends a group of interest requests from interest requester by using the distributed parallel ability of ACO, thus has the highest $\mathbb{ARSR}$. For EQRI and NDNF, when one CR is not capable of forwarding interest request, EQRI uses backtracking strategy to return interest request to the previous CR while NDNF generates a new interest request from interest requester. In such condition, interest request has the chance to retrieve the content during the backtracking process in EQRI, thus EQRI has higher $\mathbb{ARSR}$ than NDNF. For NDNF and QAPS, when the content cannot be found in CS, NDNF checks PIT and waits the content within TTL while QAPS finds an outgoing interface with the highest priority to forward interest request. Thus, NDNF has higher $\mathbb{ARSR}$ than QAPS. For ADP, EQRI, NDNF and QAPS, the contents are placed evenly in CRs, that is, each CR in network caches a certain number of contents. However, those are placed at the CRs located at the network edge for EATR, which decreases the probability of content retrieval, thus EATR has the lowest $\mathbb{ARSR}$.

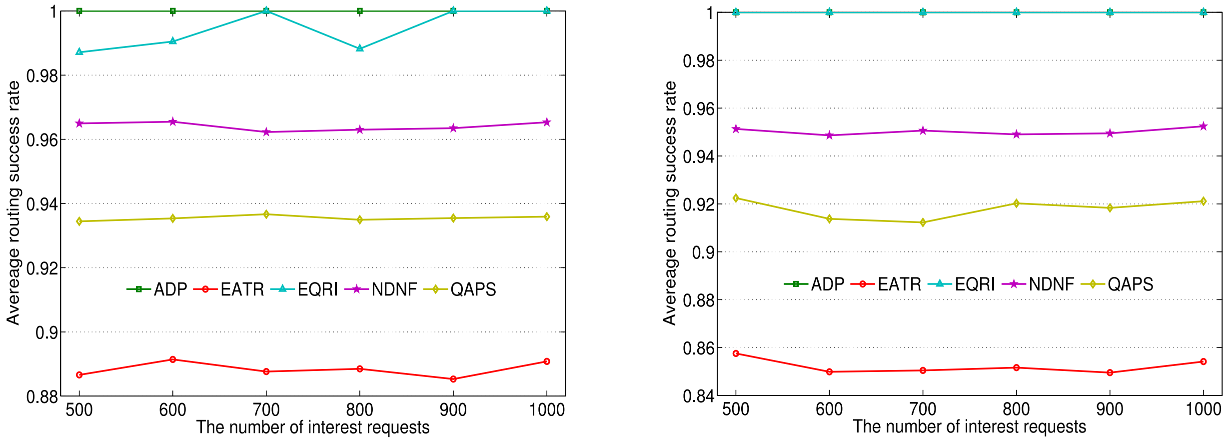
Over the different topologies, only $\mathbb{ARSR}$ of ADP has no obvious change, and the corresponding $\mathbb{ARSR}$ reaches 100%. For EQRI, its $\mathbb{ARSR}$ over GTS is higher than that over Deltacom in cases of sending 700, 900 and 1000 interest requests, which indicates that EQRI have better adaptability over GTS than Deltacom. In addition, $\mathbb{ARSR}$s of NDNF, QAPS and EATR are lower over Deltacom than that over GTS, this is because re-sending interest request, checking PIT and placing content in larger network have more significant opposite influence on NDNF, QAPS and EATR respectively.

7.4. Average hop count

The $\mathbb{AHC}$s for EQRI, NDNF, QAPS, EATR and ADP are reported in Fig. 9.

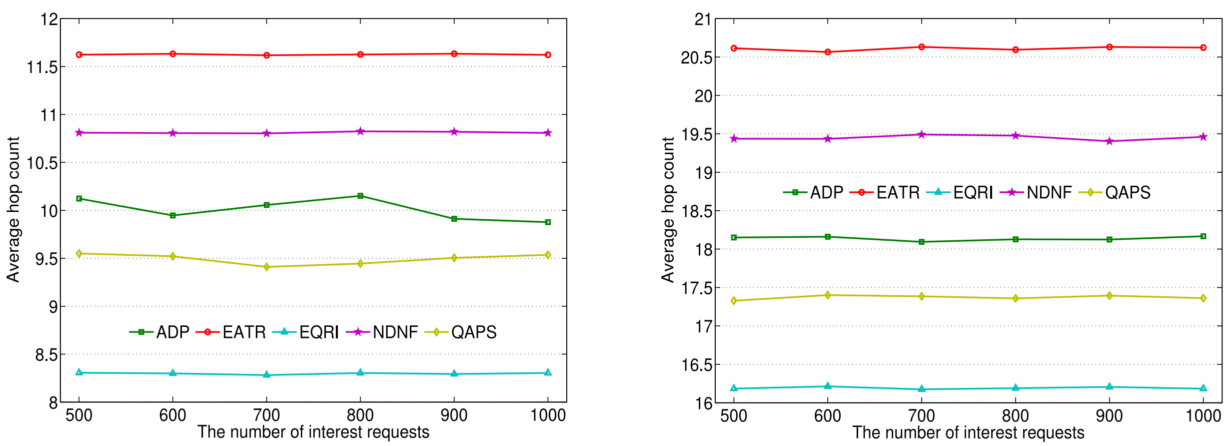
Over the same topology, we observe that EQRI has the smallest $\mathbb{AHC}$, followed by QAPS, ADP, NDNF and EATR. Although EQRI uses backtracking way which increases $\mathbb{AHC}$ to a certain extent, the failed number of interest requests has the considerably low proportion according to Fig. 9. That is, the influence on $\mathbb{AHC}$ by using backtracking way is not very significant. For EQRI and QAPS, they are to find a single path and select the optimal outgoing interface to forward interest request; in contrast, NDNF, EATR and ADP are to find the multiple alternative paths, and they have to traverse much more hops and thus have larger $\mathbb{AHC}$ than EQRI and QAPS. For EQRI and QAPS, the latter does not perform the operation of checking PIT. Especially when the same interest request arrives at a CR, QAPS forwards it straightly while EQRI checks PIT and waits the possible content within TTL. Thus, EQRI traverses much less hops and thus has smaller $\mathbb{AHC}$ than QAPS. For NDNF, EATR and ADP, the first two algorithms always forward interest request based on the multiple same outgoing interfaces, while the last one forwards a group of interest requests from interest requester and the alternative number of outgoing interfaces becomes small as the forwarding goes on. In addition, ADP inherently supports QoS, thus it always selects the relatively optimal outgoing interface to forward interest request. Given the above two aspects, ADP has smaller $\mathbb{AHC}$ than NDNF and EATR. For NDNF and EATR, instead of placing the contents at the CRs of network edge, NDNF caches and distributes them at CRs evenly. Under such condition, EATR has to forward interest request to the edge CRs, thus NDNF has smaller $\mathbb{AHC}$ than EATR.

Over the different topologies, we observe that the five algorithms have larger $\mathbb{AHC}$ over GTS than that over Deltacom, because GTS has larger physical distance between interest requester and content provider than Deltacom, which can be seen from Figs. 6 and 7.



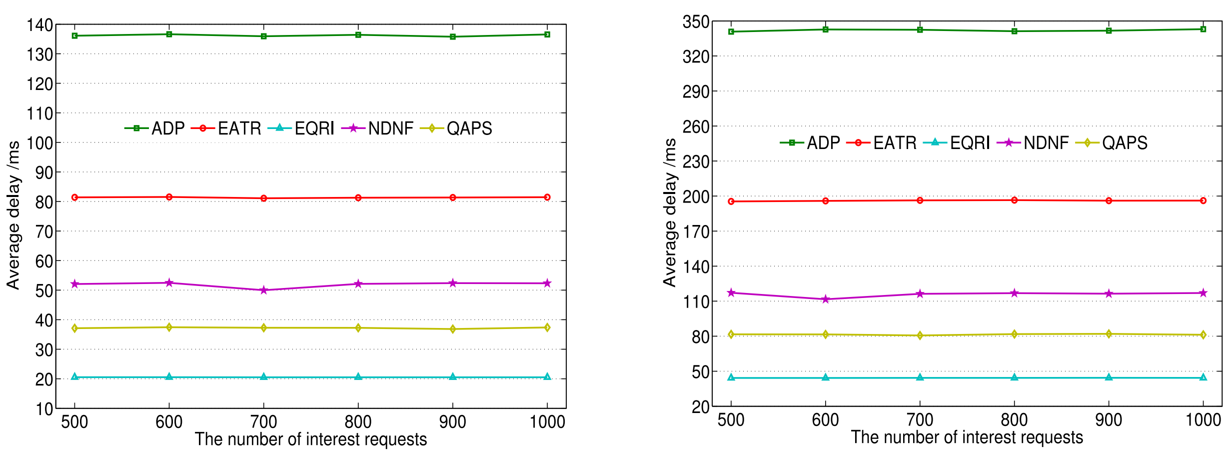
(a) # Deltacom (b) # GTS

Fig. 8. Average routing success rate among EQRI, NDNF, QAPS, EATR and ADP.



(a) # Deltacom (b) # GTS

Fig. 9. Average hop count among EQRI, NDNF, QAPS, EATR and ADP.



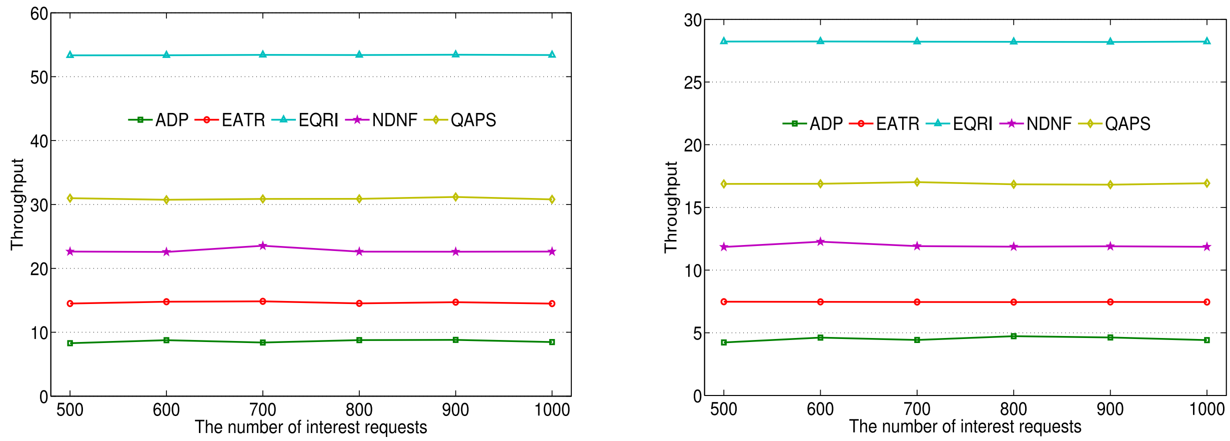
(a) # Deltacom (b) # GTS

Fig. 10. Average delay among EQRI, NDNF, QAPS, EATR and ADP.

7.5. Average delay

The delay is defined as the time-slot between the point-in-time when an Interest packet is sent from an interest requester and that when the interest requester receives the corresponding Data packet(s) which carries the demanded content or is informed as failure. The $\mathbb{AD}$s for EQRI, NDNF, QAPS, EATR and ADP are reported in Fig. 10.

Over the same topology, we observe that EQRI has the smallest $\mathbb{AD}$, followed by QAPS, NDNF, EATR and ADP. Among them, ADP consumes a lot of time to partition the network into several domains in order to reduce energy consumption; on the other hand, ADP uses ACO to find a feasible path in order to support



(a) # Deltacom (b) # GTS

Fig. 11. Average throughput among EQRI, NDNF, QAPS, EATR and ADP.

QoS, which needs much more computation time. Thus, ADP has the largest $\mathbb{AD}$. As a matter of fact, during the process of data routing, sending multiple data objects from multiple different content providers to interest requester influences transmission efficiency of data, and thus increases the corresponding transmission delay. In terms of the remaining four algorithms, NDNF and EATR multicast interest request via multiple outgoing interfaces and find many content providers, thus they have larger $\mathbb{AD}$ than EQRI and QAPS. For NDNF and EATR, the latter need consume more time to send interest request or find content provider to the edge CRs, thus EATR has larger $\mathbb{AD}$ than NDNF. For EQRI and QAPS, although the former uses both backtracking way and PIT checking, the failed number of interest requests has the considerably low proportion and checking PIT consumes the relatively small time. In addition, the former considers three QoS factors including delay, bandwidth and error rate while the latter only considers bandwidth, thus it has higher delay requirement than the latter. Furthermore, the latter always uses ACO to select outgoing interface with the highest priority to forward interest request, which consumes much more time than the latter. Given the above three aspects, EQRI has smaller $\mathbb{AD}$ than QAPS.

Over the different topologies, we observe that the five algorithms have larger $\mathbb{AD}$ over GTS than that over Deltacom, because they need to traverse more CRs over GTS than Deltacom.

7.6. Average throughput

The throughput is defined as the processed number of interest requests within unit time (μs). The $\mathbb{AT}$s for EQRI, NDNF, QAPS, EATR and ADP are reported in Fig. 11.

We observe that EQRI has the largest $\mathbb{AT}$, followed by QAPS, NDNF, EATR and ADP. The related reason is concluded as follows. An algorithm with smaller $\mathbb{AD}$ means that it has more time to process more number of interest requests, and thus has larger $\mathbb{AT}$. More inherent reasons can be found from the above sections.

7.7. Average energy efficiency

The network energy efficiency is defined as the ratio of the total traffic to the total consumed energy in network, where the total consumed energy is equal to the total energy consumed by all involved CRs and links during traffic transferring. The $\mathbb{AEE}$s for EQRI, NDNF, QAPS, EATR and ADP are reported in Fig. 12.

EQRI has the highest $\mathbb{AEE}$, followed by EATR, ADP, QAPS and NDNF. Since EQRI, ADP and EATR inherently support energy efficiency, they have higher $\mathbb{AEE}$ than the other two algorithms. For the first three algorithms, EQRI forwards interest request via the relatively outgoing interface, thus consumes the smallest energy. In addition, energy efficiency in EQRI can be guaranteed in terms of each link since it is considered when selecting outgoing interface. However, both EATR and ADP only consider energy efficiency from the global perspective and cannot guarantee the link's energy efficiency real-time. Given the above two aspects, EQRI has the highest $\mathbb{AEE}$. Although EATR and ADP forward interest request with the multiple alternative paths, consuming much more energy compared to EQRI, the latter forwards interest request via the edge CRs and thus avoids the traffic going through the backbone, consuming much smaller energy. In other words, EATR has higher $\mathbb{AEE}$ than ADP. In the similar way, QAPS forwards interest request via an outgoing interface straightly, thus has higher $\mathbb{AEE}$ than NDNF.

7.8. Stability

The stability is quantified as the fluctuation coefficient among EQRI, NDNF, QAPS, EATR and ADP with respect to $\mathbb{ARSR}$, $\mathbb{AHC}$, $\mathbb{AD}$, $\mathbb{AT}$ and $\mathbb{AEE}$, as follows.

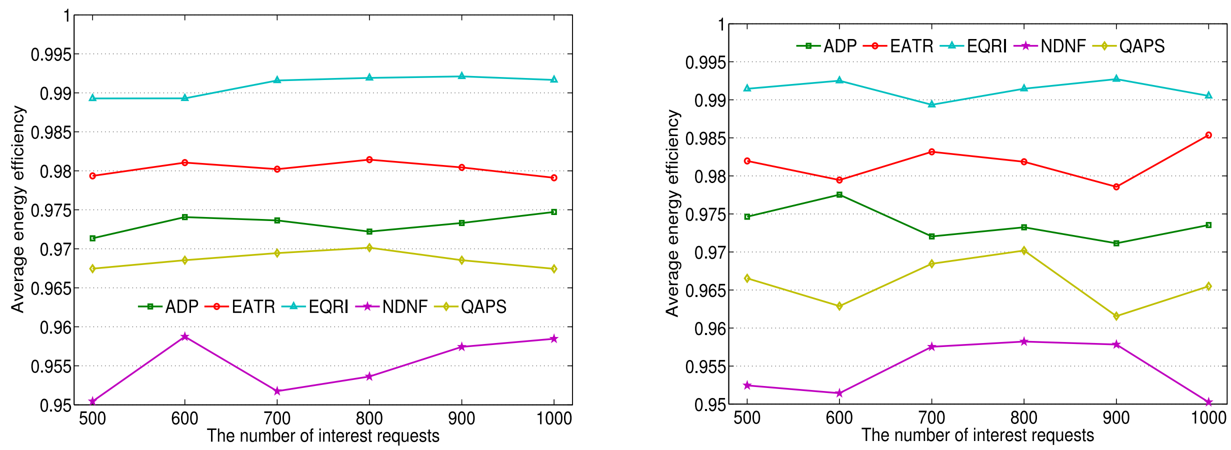
$$\rho_i = \frac{sv_i}{\sum_{i=1}^{5} sv_i} \tag{9}$$

$$sv_i = \sum_{j=1}^{6} \left\{ x_{max} - x_j \right\} \tag{10}$$

$$x_{max} = max_{j=1}^{6} x_j \tag{11}$$

Among them, sv_1, sv_2, sv_3, sv_4 and sv_5 denote the stability values for EQRI, NDNF, QAPS, EATR and ADP respectively; x_1, x_2, x_3, x_4, x_5 and x_6 denote the values of $\mathbb{ARSR}$, $\mathbb{AHC}$, $\mathbb{AD}$, $\mathbb{AT}$ and $\mathbb{AEE}$ respectively; ρ_1, ρ_2, ρ_3, ρ_4 and ρ_5 denote the fluctuation coefficients for EQRI, NDNF, QAPS, EATR and ADP respectively. Meanwhile, an algorithm with small fluctuation coefficient means it has good performance. The stabilities for EQRI, NDNF, QAPS, EATR and ADP are reported in Table 3.

According to Table 3, we observe that EQRI has the smallest fluctuation coefficient, that is, it is more stable than NDNF, QAPS, EATR and ADP. On one hand, EQRI is to find a single path based on color management, priority determination and backtracking strategy so that interest request can be forwarded via the optimal outgoing interface. On the other hand, EQRI considers QoS and energy efficiency inherently, thus user's QoS requirements and energy efficiency can be guaranteed effectively.



(a) # Deltacom (b) # GTS

Fig. 12. Average energy efficiency among EQRI, NDNF, QAPS, EATR and ADP.

Table 3
The stability test results among EQRI, NDNF, QAPS, EATR and ADP.

Algorithm	EQRI	NDNF	QAPS	EATR	ADP
Fluctuation coefficient over Deltacom	0.430217	1.062386	0.516551	0.699936	0.814231
Fluctuation coefficient over GTS	0.475824	0.989125	0.503398	0.641217	0.785863

Table 4
Wilcoxon testing results over Deltacom under 600 interest requests with respect to routing success rate.

Comparison	R^+	R^-	$p - value$
EQRI versus ADP	0	100	0
EQRI versus EATR	100	0	0
EQRI versus NDNF	100	0	0
EQRI versus QAPS	100	0	0

Table 5
Wilcoxon testing results over Deltacom under 600 interest requests with respect to hop count.

Comparison	R^+	R^-	$p - value$
EQRI versus ADP	0	100	0
EQRI versus EATR	0	100	0
EQRI versus NDNF	0	100	0
EQRI versus QAPS	0	100	0

Table 6
Wilcoxon testing results over Deltacom under 600 interest requests with respect to delay.

Comparison	R^+	R^-	$p - value$
EQRI versus ADP	0	100	0
EQRI versus EATR	0	100	0
EQRI versus NDNF	0	100	0
EQRI versus QAPS	0	100	0

Table 7
Wilcoxon testing results over Deltacom under 600 interest requests with respect to throughput.

Comparison	R^+	R^-	$p - value$
EQRI versus ADP	100	0	0
EQRI versus EATR	100	0	0
EQRI versus NDNF	100	0	0
EQRI versus QAPS	100	0	0

Table 8
Wilcoxon testing results over Deltacom under 600 interest requests with respect to energy efficiency.

Comparison	R^+	R^-	$p - value$
EQRI versus ADP	100	0	0
EQRI versus EATR	100	0	0
EQRI versus NDNF	100	0	0
EQRI versus QAPS	100	0	0

7.9. Wilcoxon-based statistical testing

In order to make the experimental results in Figs. 8–12 more convincing, we do the statistical testing based on Wilcoxon [47,48], here the significance level (i.e., confidence interval) is set as 0.01. According to Figs. 8–12, 60 tables (5 performance evaluation metrics*6 groups of different interest requests*2 topologies) are generated to report Wilcoxon testing results. Due to the limited space, we only show five groups of Wilcoxon testing results over Deltacom under 600 interest requests with respect to five metrics, and they are shown in Tables 4–8 respectively. According to the above testing results, we observe that EQRI has the optimal $\mathbb{AHC}$, $\mathbb{AD}$, $\mathbb{AT}$ and $\mathbb{AEE}$, as well as the suboptimal $\mathbb{ARSR}$. In summary, from the perspective of comprehensive evaluation, EQRI has better performance than EATR, ADP, QAPS and NDNF.

8. Conclusions

ICN has been widely studied as one promising networking paradigm, which focuses on named content rather than host-centric IP address. In this paper, we propose a novel energy-efficient QoS routing mechanism for ICN to retrieve the content effectively. We evaluate QoS and energy efficiency, design color management, priority management and backtracking strategy, and further present link selection algorithm to forward interest request. In addition, we compare the proposed EQRI with four existent algorithms by considering routing success rate, hop count, delay, throughput, energy efficiency and stability, and the simulation results demonstrate that EQRI has better performance.

However, as a new ICN routing mechanism, EQRI has some limitations to be enhanced and improved. Although EQRI is evaluated over Deltacom and GTS topologies, it needs to be evaluated over other real and virtual topologies to see whether it has the stable and satisfiable performance, that is, EQRI should have strong robustness. In addition, EQRI does not consider the mobility of either content or interest requester, and we should enhance EQRI to support mobility.

Acknowledgments

We would like to thank the editors and all anonymous reviewers for helpful suggestions, which have considerably improved and enhanced the quality of this paper. This work is supported by the National Natural Science Foundation of China under Grant No. 61572123, and the National Science Foundation for Distinguished Young Scholars of China under Grant No. 71325002.

Appendix

Theorem 1. *To get the best performance of EQRI, Efl_QoS is equal to* $\frac{1}{2}+\frac{\alpha_B-\alpha_D-\alpha_E}{2\beta}$.

Proof. To get the best performance of EQRI, the optimal outgoing interface should be selected, thus *Efl_QoS* should be maximized according to Eq. (8), which means that $Mfb(bw)$, $Mfd(del)$ and $Mde(era)$ should be maximized and their change rates should be minimized. The maximization of $Mfb(bw)$, $Mfd(del)$ and $Mde(era)$ means that the available bandwidth, tolerant delay and error rate meet user's requirements as much as possible, while the minimization of their change rates means that the available bandwidth, tolerant delay and error rate provided by link keep stable as much as possible. □

1) The change rate of $Mfb(bw)$ reaches the minimum, which means that it exists a fixed Min_b in case of $B_L < bw < B_H$ to make

$$Min_b = \left[\frac{\partial Mfb(bw)}{\partial bw}\right]_{min} \tag{12}$$

For Eq. (1), when $B_L < bw < B_H$, we have

$$Mfb(bw) = Mfb(t+B_L) = \left(1+\alpha t^{\beta}\right)^{-1}, t = bw - B_L \tag{13}$$

The change rate of $Mfb(bw)$ is defined as follows:

$$\frac{\partial Mfb(bw)}{\partial bw} = \frac{\partial Mfb(t)}{\partial t} = \left[\left(1+\alpha t^{\beta}\right)^{-1}\right]' = -\alpha\beta t^{\beta-1}\left(1+\alpha t^{\beta}\right)^{-2} \tag{14}$$

Let $fb(t)$ replace the left of Eq. (14), and we have

$$fb(t) = -\alpha\beta t^{\beta-1}\left(1+\alpha t^{\beta}\right)^{-2}, 0 < t < B_H - B_L \tag{15}$$

The derivative function of $fb(t)$ is as follows:

$$\begin{aligned}\frac{\partial fb(t)}{\partial t} &= -\alpha\beta t^{\beta-1}\Big[(\beta-1)t^{\beta-2}\left(1+\alpha t^{\beta}\right)^{-2} - 2\alpha\beta t^{2\beta-2}\left(1+\alpha t^{\beta}\right)^{-3}\Big] \\ &= -\alpha\beta t^{\beta-2}\left(1+\alpha t^{\beta}\right)^{-3} \times \left[(\beta-1)(1+\alpha t^{\beta}) - 2\alpha\beta t^{\beta}\right]\end{aligned} \tag{16}$$

Since $fb(t) < 0$ by Eq. (15), Eq. (12) means that $fb(t)$ can reach the minimum. Let $gb(t)$ replace the left of Eq. (16). Consider $gb(t) \geq 0$, we have

$$\alpha\beta t^{\beta} + \alpha t^{\beta} \geq \beta - 1 \tag{17}$$

(i) When $0 < \beta \leq 1$, Eq. (17) always holds; thereby, $fb(t)$ is a monotone increasing function. Since $t \in (0, B_H - B_L)$, $fb(t)$ cannot reach the minimum.

(ii) When $\beta > 1$, $fb(t)$ can reach the minimum if and only if the equality of Eq. (17) holds, and we have

$$t = \sqrt[\beta]{\frac{\beta-1}{\alpha(\beta+1)}} \tag{18}$$

Putting Eq. (18) into Eqs. (15) and (13), and we have

$$Min_b = -\frac{\beta^2-1}{4\beta}\sqrt[\beta]{\frac{\alpha(\beta+1)}{\beta-1}} \tag{19}$$

$$Mfb(bw) = \frac{\beta+1}{2\beta} \tag{20}$$

2) The change rate of $Mfd(del)$ reaches the minimum, which means that it exists a fixed Min_d in case of $D_L < bw < D_H$ to make

$$Min_d = \left[\frac{\partial Mfd(del)}{\partial del}\right]_{min} \tag{21}$$

For Eq. (2), when $D_L < bw < D_H$, we have

$$Mfd(del) = \left(1+\alpha t^{-\beta}\right)^{-1}, t = del - D_L \tag{22}$$

The change rate of $Mfd(del)$ is defined as follows:

$$\frac{\partial Mfd(del)}{\partial del} = \frac{\partial Mfd(t)}{\partial t} = \left[\left(1+\alpha t^{-\beta}\right)^{-1}\right]' = \alpha\beta t^{-\beta-1}\left(1+\alpha t^{-\beta}\right)^{-2} \tag{23}$$

Let $fd(t)$ replace the left of Eq. (23), and we have

$$fd(t) = \alpha\beta t^{-\beta-1}\left(1+\alpha t^{-\beta}\right)^{-2}, 0 < t < D_H - D_L \tag{24}$$

The derivative function of $fd(t)$ is shown as follows:

$$\frac{\partial fd(t)}{\partial t} = t^{-\beta-2}\left(1+\alpha t^{-\beta}\right)^{-3}\left[\alpha\beta t^{-\beta} - \alpha t^{-\beta} - \beta - 1\right] \tag{25}$$

Since $fd(t) > 0$ by Eq. (24), Eq. (21) means that $fd(t)$ can reach the maximum. Let $gd(t)$ replace the left of Eq. (25). Consider $gd(t) \leq 0$, we have

$$\alpha t^{-\beta}(\beta-1) \leq \beta + 1 \tag{26}$$

(i) When $0 < \beta \leq 1$, Eq. (26) always holds; thereby, $fd(t)$ is a monotone decreasing function. Since $t \in (0, D_H - D_L)$, $fd(t)$ cannot reach the maximum.

(ii) When $\beta > 1$, $fd(t)$ can reach the maximum if and only if the equality of Eq. (26) holds, and we have

$$t = \sqrt[\beta]{\frac{\alpha(\beta-1)}{\beta+1}} \tag{27}$$

Putting Eq. (27) into Eqs. (24) and (22), and we have

$$Min_d = \frac{\beta^2-1}{4\beta}\sqrt[\beta]{\frac{\beta+1}{\alpha(\beta-1)}} \tag{28}$$

$$Mfd(del) = \frac{\beta-1}{2\beta} \tag{29}$$

3) In fact, both of evaluation functions, i.e., link's tolerant delay to user's delay requirements and link's tolerant error rate to user's error rate requirements, follow the improved partial-large CDM; similarly, it exists a fixed Min_e in case of $E_L < bw < E_H$, and we have

$$Min_e = \left[\frac{\partial Mfe(era)}{\partial era}\right]_{min} \tag{30}$$

$$Mfe(era) = \left(1+\alpha t^{-\beta}\right)^{-1}, t = era - E_L \tag{31}$$

$$Min_e = \frac{\beta^2 - 1}{4\beta} \sqrt[\beta]{\frac{\beta + 1}{\alpha(\beta - 1)}} \tag{32}$$

$$Mfe(era) = \frac{\beta - 1}{2\beta} \tag{33}$$

Based on Eqs. (4), (20), (29) and (33), we have

$$Efl\_QoS = \alpha_B \frac{\beta + 1}{2\beta} + \alpha_D \frac{\beta - 1}{2\beta} + \alpha_E \frac{\beta - 1}{2\beta} = \frac{1}{2} + \frac{\alpha_B - \alpha_D - \alpha_E}{2\beta} \tag{34}$$

References

[1] M. Casado, T. Koponen, S. Shenker, A. Tootoonchian, Fabric: a retrospective on evolving SDN, in: ACM Workshop on Hot topics in Software Defined Networks, 2012, pp. 85–90.
[2] M.F. Bari, S.R. Chowdhury, R. Ahmed, R. Boutaba, A survey of naming and routing in information-centric networks, IEEE Commun. Mag. 50 (12) (2012) 44–53.
[3] G. Xylomenos, C.N. Ververidis, V.A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K.V. Katsaros, G.C. Polyzos, A survey of information-centric networking research, IEEE Commun. Surv. Tutorials 16 (2) (2014) 1024–1049.
[4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, 2012, IEEE Commun. Mag., 50, 7, 26–36.
[5] J.L. Pan, S. Paul, R. Jain, A survey of the research on future internet architectures, IEEE Commun. Mag. 49 (7) (2011) 26–36.
[6] C. Yi, A. Afanasyev, L. Wang, B. Zhang, L. Zhang, Adaptive forwarding in named data networking, ACM SIGCOMM Comput. Commun. Rev. 42 (3) (2012) 62–67.
[7] J. Ren, K. Lu, S. Wang, X. Wang, S. Xu, L. Li, S. Liu, VICN: a versatile deployment framework for information-centric networks, IEEE Netw. 28 (3) (2014) 26–34.
[8] G. Tyson, N. Sastry, R. Cuevas, I. Rimac, A. Mauthe, A survey of mobility in information-centric networks, Commun. ACM 56 (12) (2013) 90–98.
[9] K. Jim, Information-centric networking: the evolution from circuits to packets to content, Comput. Netw. 66 (2014) 112–120.
[10] A. Carzaniga, A.L. Wolf, Content-based networking: a new communication infrastructure, in: NSF Workshop on an Infrastructure for Mobile and Wireless Systems, 2001, pp. 56–68.
[11] A. Carzaniga, M. Rutherford, A. Wolf, A routing scheme for content-based networking, in: IEEE International Conference on Computer Communications, 2004, pp. 918–928.
[12] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K.H. Kim, C. Shenker, I. Stoica, A data-oriented (and beyond) network architecture, in: ACM SIGCOMM, 2007, pp. 181–192.
[13] C. Dannewitz, J. Golic, B. Ohlman, B. Ahlgren, Secure naming for a network of information, in: IEEE International Conference on Computer Communications, 2010, pp. 1–6.
[14] T. Braun, V. Hilt, M. Hofmann, I. Rimac, M. Steiner, M. Varvello, Service-centric networking, in: IEEE International Conference on Communications, 2011, pp. 1–6.
[15] A. Carzaniga, M. Papalini, A.L. Wolf, Content-based publish/subscribe networking and information-centric networking, in: ACM SIGCOMM Workshop on Information-Centric Networking, 2011, pp. 56–61.
[16] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, Named data networking, ACM SIGCOMM Comput. Commun. Rev. 44 (3) (2014) 66–73.
[17] J. Kurose, Content-centric networking: technical perspective, Commun. ACM 55 (1) (2011). 116–116
[18] Y. Chen, K. Wu, Q. Zhang, From qos to qoe: a tutorial on video quality assessment, IEEE Commun. Surv. Tutorials 17 (2) (2015) 1126–1165.
[19] A.P. Bianzino, C. Chaudet, D. Rossi, J.L. Rougier, A survey of green networking research, IEEE Commun. Surv. Tutorials 14 (1) (2012) 3–20.
[20] The Climate Group, 2017http://www.smart2020.org/ [accessed 5.11.17].
[21] C. Fang, F.R. Yu, T. Huang, J. Liu, Y. Liu, A survey of green information-centric networking: research issues and challenges, IEEE Commun. Surv. Tutorials 17 (3) (2015) 1455–1472.
[22] B. Claise, J. Parello, EMAN: energy-management activities at the IETF, IEEE Internet Comput. 17 (3) (2013) 80–82.
[23] F. Alagoz, G. Gur, Energy efficiency and satellite networking: a holistic overview, Proc. IEEE 99 (11) (2011) 1954–1979.
[24] M.U. Haque, K. Pawlikowski, A. Willig, L. Bischofs, Performance analysis of blind routing algorithms over content centric networking architecture, in: International Conference on Computer and Communication Engineering, 2012, pp. 922–927.
[25] R. Chiocchetti, D. Perino, G. Carofiglio, D. Rossi, G. Rossini, INFORM: a dynamic interest forwarding mechanism for information centric networking, in: ACM SIGCOMM Workshop on Information-Centric Networking, 2013, pp. 6–14.
[26] M. Lee, K. Cho, K. Park, T.T. Kwon, Y. Choi, SCAN: scalable content routing for content–aware networking, in: IEEE International Conference on Communications, 2011, pp. 1–5.
[27] M. Tortelli, G.L. Alfredo, G. Boggia, CCN forwarding engine based on bloom filters, in: ACM International Conference on Future Internet Technologies, 2012, pp. 13–14.
[28] S. Shanbhag, N. Schwan, I. Rimac, M. Varvello, SoCCer: services over content–centric routing, in: ACM SIGCOMM Workshop on Information–Centric Networking, 2011, pp. 62–67.
[29] J. Lv, X. Wang, M. Huang, Ant colony optimization-inspired ICN routing with content concentration and similarity relation, IEEE Commun. Lett. 21 (6) (2017) 1313–1316.
[30] J. Lv, X. Wang, M. Huang, ACO-inspired ICN routing mechanism with mobility support, Appl. Soft Comput. 58 (2017) 427–440.
[31] J. Lv, X. Wang, K. Ren, M. Huang, J. Li, ACO-inspired information-centric networking routing mechanism, Comput. Netw. 126 (2017) 200–217.
[32] Q. Li, Z. Zhao, M. Xu, Y. Jiang, Y. Yang, A smart routing scheme for named data networks, Comput. Commun. 103 (2017) 83–93.
[33] L. Saino, L. Psaras, G. Pavlou, Hash-routing schemes for information centric networking, in: ACM SIGCOMM Workshop on Information-Centric Networking, 2013, pp. 27–32.
[34] J. Lv, X. Wang, M. Huang, J. Shi, K. Li, J. Li, RISC: ICN routing mechanism incorporating SDN and community division, Comput. Netw. 123 (2017) 88–103.
[35] A.Z. Khan, S. Baqai, F.R. Dogar, Qos aware path selection in content centric networks, in: IEEE International Conference on Communications, 2012, pp. 2645–2649.
[36] W. Chao, L. Wang, H. Xie, Z. Zhang, Z. Jiang, Network delay guarantee for differentiated services in content-centric networking, Comput. Commun. 76 (2016) 54–66.
[37] A. Kerrouche, M.R. Senouci, A. Mellouk, Qos-FS: a new forwarding strategy with qos for routing in named data networking, in: IEEE International Conference on Communications, 2016, pp. 1–7.
[38] D. Posch, B. Rainer, H. Hellwagner, Towards a context-aware forwarding plane in named data networking supporting qos, ACM SIGCOMM Comput. Commun. Rev. 47 (1) (2017) 6–14.
[39] Y. Song, M. Liu, Energy-aware traffic routing with named data networking, China Commun. 9 (6) (2012) 71–81.
[40] V. Jacobson, Energy-efficient content retrieval in content-centric networks, 2014, Palo Alto, CA, US 8644211B21.
[41] C. Fang, F.R. Yu, T. Huang, J. Liu, Y. Liu, Distributed energy consumption management in green content-centric networks via dual decomposition, IEEE Syst. J. 11 (2) (2017) 625–636.
[42] C. Li, W. Liu, I. Wang, M. Li, K. Okamura, Energy-efficient quality of service aware forwarding scheme for content-centric networking, J. Netw. Comput. Appl. 58 (2015) 241–254.
[43] G.A. Anastassiou, Fuzzy Mathematics: Approximation Theory, Springer Press, 2010.
[44] P. Mahadevan, P. Sharma, S. Banerjee, P. Ranganathan, A power benchmarking framework for network devices, in: IFIP International Conference on Networking, 2009, pp. 795–808.
[45] A.K.M.M. Hoque, S.O. Amin, A. Alyyan, B. Zhang, L. Zhang, L. Wang, NLSR–named data link state routing protocol, in: ACM SIGCOMM Workshop on Information–Centric Networking, 2013, pp. 15–20.
[46] http://www.inmon.com/technology/index.php [accessed 5.11.17].
[47] R. Jain, The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling, Wiley Press, 1991.
[48] J. Derrac, S. Garcia, D. Molina, F. Herrera, A practical tutorial on the use ofnonparametric statistical tests as a methodology for comparing evolutionaryand swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.

Xingwei Wang received the B.S., M.S., and Ph.D. degrees in computer science from the Northeastern University, Shenyang, China in 1989, 1992, and 1998 respectively. He is currently a Professor at the College of Software, Northeastern University, Shenyang, China. His research interests include future Internet and cloud computing, etc. He has published more than 100 journal papers, books and book chapters, and refereed conference papers. He has received several best paper awards.

Jianhui Lv received the B.S. degree in mathematics and applied mathematics from the Jilin Institute of Chemical Technology, Jilin, China in 2012, and the M.S. degree in computer science from the Northeastern University, Shenyang, China in 2014. He is currently working toward the Ph.D. degree in the Northeastern University, Shenyang, China. His research interests include ICN routing and operational research, etc. He has published more than 20 journal and conference papers.

Min Huang received the B.S. degree in automatic instrument, the M.S. degree in systems engineering, and Ph.D. degree in control theory from the Northeastern University, Shenyang, China in 1990, 1993, and 1999 respectively. She is currently a Professor at the College of Information Science and Engineering, Northeastern University, Shenyang, China. Her research interests include modeling and optimization for logistics and supply chain system, etc. She has published more than 100 journal papers, books, and refereed conference papers.

Keqin Li is a SUNY Distinguished Professor of computer science. His research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU-GPU hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things and cyber-physical systems. He has published over 440 journal papers, book chapters, and refereed conference papers, and has received several best paper awards. He is currently or has served on the editorial boards of IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Cloud Computing, Journal of Parallel and Distributed Computing. He is an IEEE Fellow.

Jie Li is a Professor in computer science in Division of Information Engineering, Faculty of Engineering, Information and Systems, University of Tsukuba, Japan. His research interests are in Intelligent Mobile and Ubiquitous Computing, Networking, and Security, Internet Computing and Networking, Big Data, Cloud, OS, modeling and performance evaluation of information systems. He is a senior member of IEEE and ACM, and a member of IPSJ (The Information Processing Society of Japan). He is the founding chair of the IEEE Technical sub-Committee on Big Data (TSCBD), ComSoc. He received the B.E. degree in computer science from Zhejiang University, Hangzhou, China, the M.E. degree in electronic engineering and communication systems from China Academy of Posts and Telecommunications, Beijing, China. He received the Dr. Eng. degree from the University of Electro-Communications, Tokyo, Japan. He had served as a secretary for Study Group on System Evaluation of IPSJ and on Steering Committees of the SIG of System EVAluation (EVA) of IPSJ, the SIG of DataBase System (DBS) of IPSJ, and the SIG of MoBiLe computing and ubiquitous communications of IPSJ. He has served on several editorial boards for the IPSJ Journal and many international professional journals He has also served on the program committees for several international professional conferences.

Kexin Ren received the B.S. degree in network engineering from the Harbin University of Science and Technology, Harbin, China in 2015. She is currently working toward the M.S. degree in the Northeastern University, Shenyang, China. Her research interests include ICN routing, etc.