Rui *et al. EURASIP Journal on Wireless Communications and Networking* (2017) 2017:207
DOI 10.1186/s13638-017-0991-3

EURASIP Journal on Wireless Communications and Networking

RESEARCH **Open Access**

ICN routing selecting scheme based on link quality for the urban vehicles' communication

Lanlan Rui, Hui Guo*, Ruichang Shi, Haoqiu Huang and Xuesong Qiu

Abstract

It is difficult for TCP/IP to manage the mobile nodes in the dynamic environment of VANET (Vehicular Ad hoc Network), so combining urban VANET with a new network architecture—ICN (information-centric network)—this paper proposes a FIB (forwarding information base) selecting scheme based on link quality, and we name it as RSBLQ (Routing Selecting Based on Link Quality). The metrics of link quality in RSBLQ are mainly *LET* (*link expiration time*) and *link available probability*. To reduce network delay, cache redundancy, and network traffic, our RSBLQ has the following designs: (1) we modify the structure of PIT (pending interest table) by adding two domains: *receive time* and *tolerance time*; (2) we introduce the algorithm of LET to help with the content routing selection in FIB; (3) RSBLQ also gets the link available probability to be another metric for our algorithm. Simulation results show that in terms of average response time, the performance improvement of RSBLQ is about 10, 15–20, and 60% compared to CCVN, V-NDN, and TCP/IP, respectively; it approximately has 8 and 13.8% improvement than CCVN and V-NDN about cache hit ratio; and the network traffic also decreases a lot.

Keywords: ICN, Urban VANET, LET, Link available probability, FIB selecting

1 Introduction

Due to the significant research and technology advancements in wireless communication, the traditional intelligent transport system (ITS) [1] has evolved towards vehicular communication. The concept of vehicle-to-vehicle (V2V) and vehicle-to-roadside (V2R) unit communication came into existence in research and developments as another communication network known as Vehicular Ad hoc Networks (VANETs) [2, 3].

1.1 Limitations in conventional VANET

The communication architecture [4] of conventional VANETs can be divided into three categories including Wireless Access in Vehicular Environments (WAVE)-based Wi-Fi, ad hoc, and hybrid. In WAVE-based Wi-Fi-driven architecture, RSUs (road-side units) alongside roads are used as wireless access points which provide communication coverage to the vehicles inside its coverage area. In ad hoc architecture, group of on-road vehicles form ad hoc networks using WAVE. These networks perform operations independently without any infrastructure support. In hybrid architecture, cellular and ad hoc architectures both using WAVE perform their operations in collaborations. However, there are some drawbacks for VANET based on TCP/IP.

First, assigning IP addresses to moving nodes is difficult, because IP addresses are a very limited and precious resource, its management is tightly controlled and requires infrastructure support, such as a central DHCP (Dynamic Host Configuration Protocol) server. However, the above idea directly conflicts with the ad hoc networks' desire to operate in an infrastructure-free manner.

Second, as technological advances have brought greater-than-ever numbers of computing devices, most of which are now mobile, thus managing IP address assignments for all of these devices is becoming increasingly infeasible and less meaningful.

Third, all communication is meant to serve the purpose of delivering data to application processes, and it is the data itself that is meaningful to applications. However, because data are unknown in today's IP-centric architecture,

* Correspondence: guohui@bupt.edu.cn
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

a node meets an application's request only by selecting a specific node to send the data to, which is inefficient and is easy to result in suboptimal data delivery.

Finally, in an ad hoc network, there is an inherent trade-off between the accuracy of the routing state maintained at each node and the overhead to keep this state consistent. Since packets are delivered over a single path, the binding between a mobile node and its current IP address is critical. With high node mobility, a node suffers from high overhead to keep the binding updated, a loss of connectivity due to outdated binding information, or both. Therefore, some researchers begin to consider applying other technologies to VANET to shield the limitation mentioned above.

1.2 New VANET schemes

1) VANET with SDN

Vehicular networks typically comprise various communication technologies, including dedicated short-range communication (DSRC), Wi-Fi, fourth generation (4G), 5G, and TV white space. Although these technologies can ensure reliable and ubiquitous mobile coverage, several salient features of VANETs introduce new challenges, such as unbalanced traffic flow in a multi-path topology and inefficient network utilization. Thus, flexible and programmable architectures are key requirements for VANET.

The convergence of software-defined networking (SDN) with VANET technology can play an important role in addressing most challenges. An illustration of a software-defined vehicular network (SDVN) is provided in [5], where SDN is introduced into VANET to improve network efficiency. In addition, equipment and radio devices are simply reconfigured in SDN by adding a network programmability feature to vehicular networks through external applications. Consequently, SDN provides flexibility in developing vehicular network infrastructure.

But there are also some challenges that need to be considered in the future.

Firstly, the high mobility of vehicles causes a change in SDVN topology and instability in wireless channels. High mobility also hinders the real-time collection of the information of vehicles and the network using the controller. Thus, the controller experiences delays in distributing commands. Efficient control for high mobility management is a significant concern that requires serious attention to promote the adoption of SDVN.

Secondly, in VANET, various types of networks are involved to ensure connectivity among vehicles. However, the lack of efficient internetworking mechanisms leads to connectivity issues among heterogeneous networks in a vehicular network. In SDVN, interconnection among heterogeneous networks has become a challenge because of the lack of standardized eastbound/westbound application programming interfaces (APIs) and northbound APIs for vehicular applications.

Thirdly, in SDVN, the propagation of misinformation from unauthorized entities can lead to serious accidents. Therefore, security is one of the key concerns that require serious attention. The controller should be protected because it is the centralized decision point in SDVN.

2) VANET with cloud computing

Despite the surge in VANET research, future high-end vehicles are expected to under-utilize on-board computation, communication, storage resources, etc. In [6], the authors envisioned the next paradigm shift from conventional VANET to Vehicular Cloud Computing (VCC) by merging VANET with cloud computing.

The basic idea of VANET (Vehicular Ad hoc Network) is to take the widely adopted and inexpensive wireless local area network (WLAN) technology, with a few tweaks, and install it on vehicles. Nevertheless, despite the surge in VANET research, security and privacy issues have been the root cause of impeded momentum in VANET deployment. One of the many goals of VANET is to support traffic safety and make the driving experience more safe and comfortable. In VANET, vehicles and RSUs, i.e., network nodes, will be equipped with on-board computation and communication modules to make sure fruitful communication possible among them. The main driving force behind VCC is that in the near future, the huge vehicular fleets on our roadways, streets, and parking lots will be recognized as abundant and under-utilized computational and communication resources.

VCC divides VANET clouds into three major architectures namely vehicular clouds (VC), vehicles using clouds (VuC), and hybrid clouds (HC). VC is further divided into two scenarios from the movement standpoint. Static clouds refer to the stationary vehicles providing cloud services. In case of static VANET clouds, the infrastructure (communication, storage, and process) can be rented out to make revenue as well. IaaS and data storages services are feasible for such arrangements. On the other hand, dynamic clouds are formed on demand in an ad hoc manner. VuC connects the VANET to traditional clouds where VANET users can use cloud services on the move such as infotainment, traffic information, and CAA. In HC, vehicular clouds will interact with traditional cloud for services exchange. The vehicles and RSUs will serve as gateways on the VANET part thereby communicating with the gateways of traditional clouds.

However, there are many risks involved with renting virtual resources in cloud environment or storing data in cloud thereby releasing control over data. One concern that many users are aware of is loss of privacy and data storage security. Logically, VANET clouds inherit their parental long-chased security and privacy issues from both VANET and cloud computing. The security and privacy challenges faced by stand-alone VANET and cloud computing will remain unchanged even if the two technologies are merged to form VANET clouds. The main challenges for VANET clouds are expected to be gossip interval, mobile authentication, conditional anonymity and virtualization, insiders and outsiders, renting out resources, autonomy, control, and cooperation middlewares.

3) VANET with ICN

Different from TCP/IP, information-centric networking (ICN) [7] focuses on whether the Interest packets can be satisfied. It neglects the position of a source node and its destination node. Thus, ICN transforms the network mode from host-centric to content-centric, and it is mainly designed around the following idea: (1) packets in ICN are divided into two types: Interest and Data, which are used to ask for data and reply to the request, respectively, and communication in ICN is driven by data consumers. (2) An on-path [8] caching mechanism reduces network traffic and improves the efficiency of information distribution. (3) Network security can be realized by ensuring that the content is secure. (4) Routing forwarding in ICN is complink expiration timed by three important structures: CS (content store), PIT (pending interest table), and FIB (forwarding information base). The CS caches contents that have been forwarded by the router so that future requests for these contents can be immediately served. The PIT records track the mapping between incoming interfaces and pending interest packets that have arrived but have not been served yet, then it can generate the hop-by-hop reverse paths towards consumers for Data. The FIB maps prefixes of content names to one or multiple output interfaces that indicate the next hop routers. When a node receives an Interest, it firstly searches for a content name match in its CS. If the content is found, a Data packet will be sent as a reply; otherwise, it will check whether there is a corresponding entry in PIT. If so, the node will record the interest packet and then drop it or it will forward the packet via FIB. However, if FIB is also missed, the Interest will be dropped. Together, the ICN features remove the address limitation in TCP/IP and make it easy to manage mobile nodes. Content-centric networking (CCN) [9] and named data networking (NDN) [10] are the most prominent interest-based ICN architecture.

Figure 1 has shown the forwarding and caching process of ICN in detail: the sender cannot directly send a content to the request nodes; the data is controlled by the receivers; the senders (or content provider) do not transport the content to the network actually, but they would send an advertisement to the whole network to tell what they have cached; and at this time, they also have no idea about who would be interested in their cached content. Then, a receiver or consumer declares what they are interested in through an Interest packet, but it does not know who can satisfy its request. The network starts a transmission path from the sender to the receiver only when the receiver's advertisement matches the Interest sent by a sender. Therefore, the interest matching process is not the process of finding the terminal and then establishing communication with it, ICN focuses on the content rather than the host that stores the content, and only the requested content is passed to the receiver.

1.3 Idea in this paper

In our study, the reasons for choosing ICN idea to improve VANET are based on the following consideration: firstly, there are many significant schemes merging VANET with ICN, which are described in Section 2.1, but the research of SDVN and VCC is at an initial stage, from the previous research, we think the ICN designing

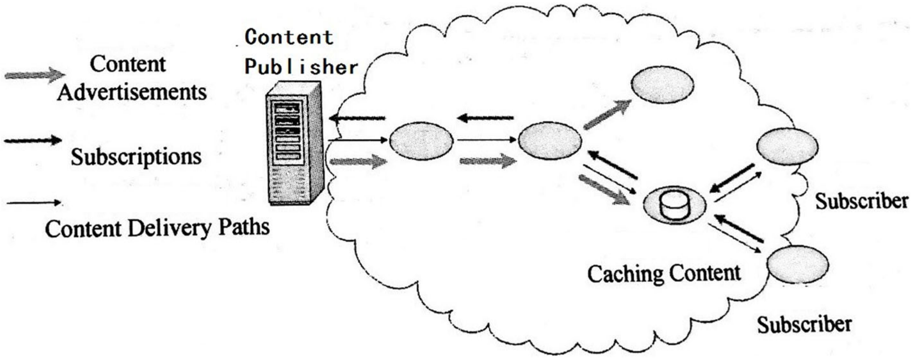

Fig. 1 The content forwarding and caching in ICN

idea can be of great help to improve the VANET performance and efficiency. Then, the ICN architecture can handle the limitation mentioned in Section 1.1 well. Moreover, we focus on the VANET routing in our study, and the routing process in ICN is very suitable for a dynamic VANET environment, and we believe it can handle the intermittent connection, nodes' mobility well.

Therefore, in this paper, we have applied ICN architecture to urban VANET and focused on the network routing process. To create stable routes and reduce the useless traffic, we design RSBLQ (Routing Selecting Based on Link Quality) as follows: (1) firstly, we improve the forwarding method of Interest packets, when they are forwarded by the guiding of FIB, considering there are always multiple outgoing interface choices in FIB, and in order to decrease useless traffic, we do not want to forward the Interest through all of the outgoing interfaces in FIB but have a selecting process. (2) Then, we modify the structure of PIT by adding two domains: *receive time* and *tolerance time*, which are used to record the Interests' arrive time and tolerance time respectively. (3) Next, we introduce the *LET* (link expiration time) and link available probability [11] as the metrics of link quality to help with the content routing selection in FIB; RSBLQ would calculate the LET and *link availability* of each link associated with each FIB outgoing interface. (4) Finally, we choose the outgoing interfaces with a bigger LET and link available probability to forward the Interest.

The contribution of our work lies in the following aspects: (1) by using ICN idea, RSBLQ makes the communication in VANET more flexible, the nodes can adapt to the dynamic environment well with ICN features, which is because there is no need for the request to arrive at a fixed destination node, an interest can be satisfied by any content nodes at any time, and it improves the network efficiency. (2) RSBLQ does not neglect the FIB function and avoids to apply a widely used flooding way in VANET but makes full use of FIB to guide the Interest forwarding process, which reduces the useless network traffic and lowers the probability of packet collision. (3) What is more, in order to make the routing paths in RSBLQ more stable and reduce broken link, we also introduce LET and link available probability as metrics for a FIB selecting process, getting more reliable FIB outgoing faces for Interest and ensuring the successful return of Data packet as much as possible.

The rest of the paper is organized as follows: Section 2 describes the related works about our study. Then, in Section 3, we give a detailed system definition to explain our system model and problem formations. Section 4 shows our main design. In Section 5, we use a simulation case to show the calculating process of RSBLQ. Section 6 analyzes the simulation results and demonstrates our scheme's efficiency and superiority. In Section 7, we conclude our paper and draw the future works.

2 Related works

2.1 VANET schemes based on ICN

In [12], the authors conclude the main shortcomings for mobile network based on IP protocol and point out solving the above problems via named data in mobile network and giving up the IP architecture.

In [13], the DMND (Collecting Data from Mobile Using Named Data) model is proposed aiming at building a new architecture that can effectively handle vehicular mobility, intermittent connectivity, and data security. Different from the idea of our RSBLQ, the authors neglect the forwarding guidance of FIB in ICN and believe the most important advantage of NDN is its capability to fully utilize the broadcast nature of wireless channels. Therefore, they use a base station to broadcast routing announcements, Interest packets, and even PIT entries. Then, through the following naming scheme, DMND eliminates the limitations of TCP/IP and achieves high data collection efficiency. In addition, the network can use the application names directly for data communication, eliminating the need for any mapping system between application names and flat identifiers. Furthermore, with each piece of data standing on its own, one can secure the data directly instead of securing its containers. Thus, requested content does not have to be delivered directly from its originator to the data requester, as long as the latter has effective means to verify the integrity and origin of incoming data. At the same time, caching Interest packets and being able to broadcast from multiple base stations solve the problem of high-speed mobility. Therefore, the design of DMND improves network performance from different angles; it certainly achieves some advantage and gives us good inspiration for the content naming mechanism. However, we do not want to apply a broadcast way for packets in VANET.

Authors in [14] investigate the content-centric networking paradigm in vehicular environments, and they design a CCVN framework, which relies on the main CCN pillars. Without using a completed broadcast method, CCVN enforces a simple counter-based approach coupled with deferred transmission timers and interest retransmission routines. Firstly, it divides Interest packets into B-Int and A-Int upon different functions. When there is an Interest, the requester uses a B-Int packet to locate the producers and then chooses a preferable content node to ask for series of data with an A-Int. Moreover, Interest forwarding in CCVN is no longer a completed broadcasting mechanism, and nodes forward a packet based on a counter; an Interest packet would be broadcast only the first time it is received. To ensure transmission reliability, CCVN also designs a timer for each Interest to deal with retransmission and prevent data loss. The design idea of CCVN is similar to our RSBLQ to a certain degree: we both think packet flooding is not a suitable way for VANET, and we choose to take advantage of FIB structure to choose several stable routing paths for Interest, but CCVN uses a hop counter as a broadcast limitation. But it is not enough, a hop counter can only decrease the broadcast range, which can also result in useless traffic.

In [15], the authors discuss the trade-offs between proactive and reactive content-oriented VANET routing for different content types. To enable scalable proactive ICN routing in VANET, HBFR—a scalable routing method using Bloom filter for content advertisement—is presented, and its procedure utilizes hierarchical geographical partitioning. Simulation results show that CCN and HBFR achieve comparable response time for popular sharable data; however, HBFR requires more overhead than the reactive CCN approach, and for popular non-sharable data, HBFR achieves shorter response time and less traffic than CCN. The above results confirm the reason to use a hybrid content-oriented routing policy in VANETs.

With the inspiration of NDN architecture, Grassi and others put forward the concept of V-NDN [16]. In V-NDN, a vehicle node can play four roles: consumer, producer, forwarder, and "data mule." The main idea in V-NDN is as follows. First, it takes full advantage of the nature of wireless broadcast, instead of only accepting data with matching entries in PIT, and a vehicle may want to cache all received data regardless of whether it has a matching PIT entry or whether it needs the data for itself. Second, Data packets can be carried in moving vehicles even without network connectivity. When vehicles move around, they serve as data mules carrying the content to a wider area, enlarging the physical bounds that encompass the data and increasing the rendezvous opportunity between consumers looking for a specific piece of data and mules carrying a copy of it. Furthermore, V-NDN applies a simple greedy forwarding strategy to spread NDN interest packets in all directions. Therefore, the design in V-NDN is exactly the opposite of our idea. It tries its best to take advantage of the broadcast channel to decrease the negative impacts caused by high volatility and intermittent connectivity, but our RSBLQ does not want to flood the packets.

Similar with CCVN, the *Strategy Based on Hops* is mentioned in [17]. It makes an improvement of PIT structure by adding three fields: *Hop*, *Sendstamp*, and *Coun*t, which are used to record the fixed hop number, forwarding time, and forwarding hop number separately. In the Datas' returning process, a packet would memorize hop number all the time, then it makes a comparison with the local value stored in PIT, by using a threshold T to decide whether the Data should be forwarded or not. The strategy reduces the flooding range and network traffic through threshold T, but there is also useless traffic actually.

To examine whether the potential ICN benefits are suitable to a VANET environment, in [18], the authors consider two ICN-specific design options: data source selection policies and caching policies. For the data source selection, it considers Nearest Replica Routing and Original Source-only Routing as the two fundamental policies and adopts the broadcast-based shortest path geo-forwarding as the forwarding mechanism. Through a shortest path algorithm, each potential next hop calculates the shortest intersection routes by using the area map. Upon receiving an Interest packet, the node sets a waiting timer based on its aforementioned calculated results; when the time is up, if the potential node does not receive the same Interest broadcast by others, it would broadcast the interest. Otherwise, it would cancel the process. The design in this paper also attempts to avoid broadcasting packets, and it adopts a "listen" mechanism to make the potential nodes determine whether it should broadcast the packets further. It is smart and efficient, and it also suggests that future-ICN-based VANETs can exploit the shortest path to known sources to avoid the time consumption caused by the nearest cache discovery.

And we have provided a table to highlight the main previous schemes (Table 1).

2.2 LET and link availability in mobile network

In [19, 20], in order to reduce link breakages during data services, the author proposes a ranging-based link availability routing scheme. It modifies the definition of link availability according the different node location and maximum transmission range; then, the link cost is defined to make a trade-off between link availability and network traffic. What is more, to ensure successful routings, the scheme also gets a backup link for each direct link, and when the direct nodes are going to out of transmission range, the corresponding backup links start to work.

Aiming at solving the unpredictable topology changes and frequent link failure in mobile adhoc network (MANET), [21] puts forward a link availability algorithm based on rough distance between nodes in a short period, then it presents a new routing protocol—a novel link availability-based routing protocol (LBRP)—for MANET. The design is based on a random mobility model, and it calculates the PDF (probability density function) of speed, direction, and relative distance, then by coming with the conditional PDF of distance between two nodes, it concludes the specific expression of link availability. Moreover, LBRP also has a detailed design for the packets definition and routing process, it gets the optimal paths through broadcasting RREQ and sending back RREP

Table 1 The comparison of previous schemes

Strategy name	Packet transmission mode	Features	Comparison with RSBLQ
DMND	Broadcast	(1) Base stations broadcast all information. (2) Application names act as forwarding destination, which eliminates the need for any mapping system. (3) Intermediate caching nodes can reply to the Interest directly.	(1) The broadcast in DMND causes too much useless network traffic than RSBLQ. (2) The security cannot be ensured when DMND uses application names in the data communication process. (3) The two methods both allow intermediate caching nodes to reply the Interest, which efficiently decreases the network delay.
CCVN	Counter-based broadcast	(1) Appling a counter-based broadcast to limit the network traffic. (2) Separating the Interest into A-Int and B-Int, dividing the request process into locating content nodes and real data transmission.	(1) CCVN uses a hop counter to decrease the traffic, which has a same goal with RSBLQ, but the two schemes accomplish at a different degree.
V-NDN	Broadcast	(1) Broadcasting all information by greedy strategy. (2) Nodes cache all received content. (3) Every node can act as a data mule to carry the packets.	(1) The broadcast in V-NDN causes too much useless network traffic than RSBLQ. (2) All-cached scheme can improve the cache hit ratio, but it enlarges the nodes' workload in V-NDN.
HBFR	Broadcast	(1) Using Bloom filter in CS. (2) Nodes cache the content advertisement by a proactive method.	(1) HBFR applies a proactive routing and focuses on the content caching, but our RSBLQ aims at improving the mobile ICN routing process.

messages. It is proved that LBRP has obtained a good performance, but its communication is limited by TCP/IP model at a certain degree.

Based on the urban scenario, [22] divides four different motion cases according to the velocity and direction of nodes, then it deduces the calculating formulas for link availability probability, and finally, it has a comparison with other routing schemes and outperforms than others. However, the design only considers the simple cases regardless of the complex motion for nodes, such as the motor direction may be not parallel but at an angle, so its research is not sufficient at a certain degree.

By using the Dynamic Source Routing (DSR) [23], the Flow-Oriented Routing Protocol (FORP) [24] was proposed to get stable routes that have even twice the lifetime of the minimum-hop routes [25, 26], but it makes too much hops. So in [27], the author propose the MILET scheme, which aims at simultaneously optimizing the lifetime for each multicast tree as well as the hop counts per source-receiver paths in these trees, it cleverly assigns the weight of a link to be 1 plus the inverse of its LET value, and the number 1 represents one hop, thus making the hop number as the weight too, then the receiver nodes of the multicast group choose the source-receiver path with the minimum sum of the link weights. Simulation results show it outperforms the other schemes.

Different with the other schemes, two novel routing metrics are proposed in [28]. The first one is link occupancy ratio (LOR), which is used to measure the occupancy level of a link, and based on this estimation, it selects links which are “freer” and the most available to transmit data traffic. Then, motivated by the inability of LOR to measure the accurate link capacity to support a specific amount of data, especially when links have a same LOR but different bandwidth values, the authors define Residual Link Capacity (RLC), it represents accurately how much of additional traffic can the link support, and by being relative to the amount of transmitted data, it allows a more efficient routing. The idea of this paper gives us a new thinking about how to define link quality properly in different kinds of network environments.

In [29], the authors present a Vehicular Ad hoc Network Routing Protocol (VARP-LET) that uses LET information to create routes and increase the reliability and stability of the routes. It divides the transmission area of each node into three categories and chooses the reliable nodes with LET in the area between the maximum transmission distance and the minimum distance. Moreover, VARP-LET introduces a route break indicator (RBI) message to reduce the packet loss. It shows the design of VARP-LET has obtained a good performance for VANET, which also gives a good point about the ICN routing in VANET.

Although it has been proved that the above works about link quality in mobile ad hoc networks are efficient, they are all limited by the fixed communication model in TCP/IP, which means the source and destination nodes in TCP/IP are known, and the routing paths have already designed, it is suitable for the static network, but in a dynamic environment, the fixed routing process seems to be more inappropriate. Moreover, TCP/IP architecture makes a difficult management of mobile nodes and results in lots of useless traffic. Therefore, we decide to apply ICN to VANET to shield the above shortcomings.

3 System definition

3.1 Application scenarios

As shown in Fig. 2, we take the urban crossroads environment as an example, which is more complicated than other scenarios, and the routing schemes in Fig. 2 are also available in other urban VANETs.

In the real environment, vehicular nodes can act not only as consumers but also as providers, they all have a transmission range—R, and all the neighbor nodes within the transmission range can arrive at the node by a one-hop routing. A node would collect its neighbors' dynamic information through beacon messages, and the information would be used to calculate LET and link availability of each connection. Based on the designing of ICN, there would be a CS, PIT, and FIB in each node.

3.2 Problem description

Based on the above scenarios, we have described a detailed Interest forwarding process in Fig. 3. Node A acts as a consumer who wants Content 1, and then it builds an Interest to ask for the Content 1 with the following steps. It should be noted that we only show a simple case when the intermediate nodes' FIB all hit, if FIB misses, the forwarding process would be over.

A: 1. I want to get the Content 1, but my CD and PIT all miss, I need to forward my Interest by FIB…
A: 2. FIB tells me to forward the Interest to B or C node, who is my best choice?
A: 3. Based on RSBLQ scheme, I choose node B as my next hop…
B: 4. My CS and PIT miss, and based on my FIB and RSBLQ, I think I should the Interest to node E…
E: 5. My CS and PIT miss, based on my FIB and RSBLQ, I think I should the Interest to node F…
F: 6. My CS and PIT miss, based on my FIB and RSBLQ, I think I should the Interest to node G…
G: 7. My CS and PIT miss, based on my FIB and RSBLQ, I think I should the Interest to node I…
I: 8. My CS hits, I can return a Data packet…

The above process has described a simple Interest forwarding process by using RSBLQ scheme, which aims at building unicast routing paths for Interest and decreasing the useless network traffic caused by widely used broadcast way.

3.3 System mapping model

As shown in Fig. 4, we have built a system mapping model between the application scenarios and ICN network layer. Each vehicular node maps to a dynamic ICN routing node, and it would have several link connections with different neighbors through different local interfaces. We have also shown the FIB entry of each intermediate node to help with our explanation. Corresponding to the working process in Section 3.2, the Interest from node A arrives at a content node I through a routing path A-B-E-F-G-I, and the Data built by I would be returned along I-G-F-E-B-A.

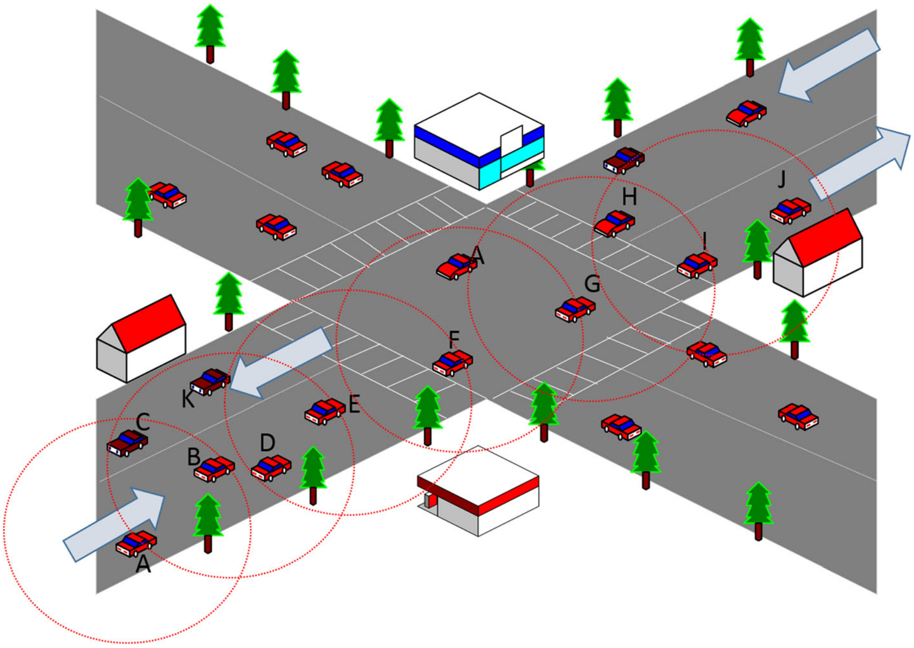

Fig. 2 An application scenario

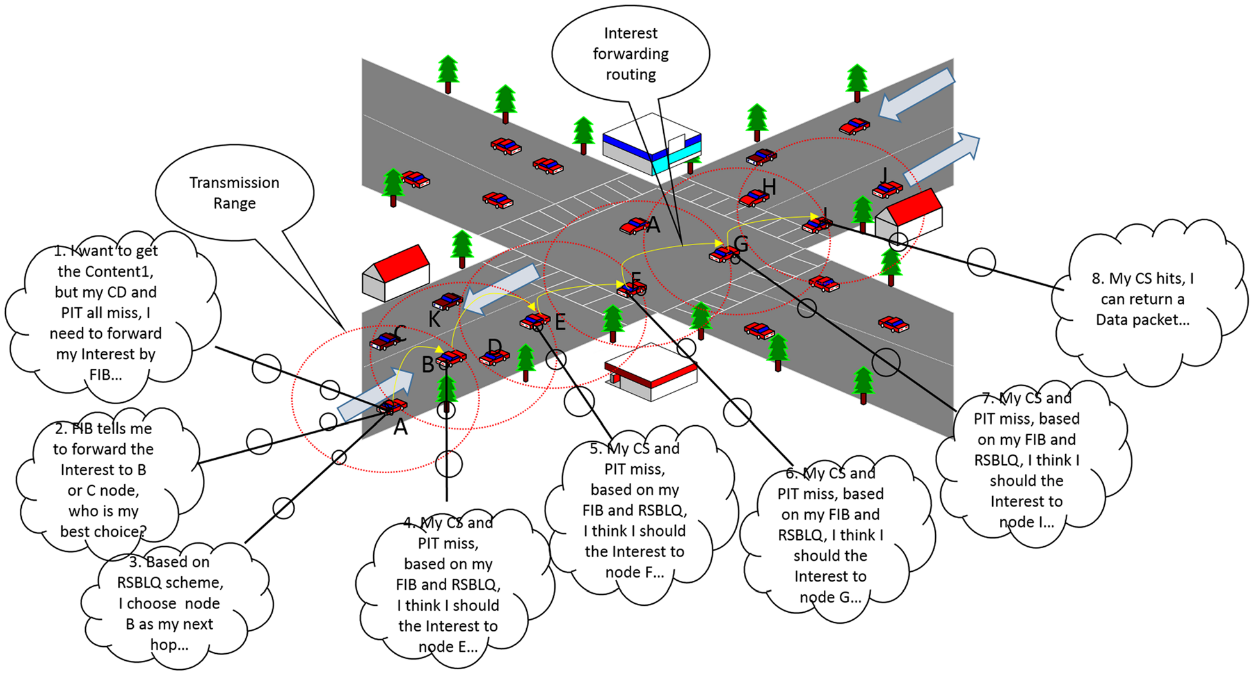
Interest forwarding routing
Transmission Range
1. I want to get the Content1, but my CD and PIT all miss, I need to forward my Interest by FIB...
2. FIB tells me to forward the Interest to B or C node, who is my best choice?
3. Based on RSBLQ scheme, I choose node B as my next hop...
4. My CS and PIT miss, based on my FIB and RSBLQ, I think I should the Interest to node E...
5. My CS and PIT miss, based on my FIB and RSBLQ, I think I should the Interest to node F...
6. My CS and PIT miss, based on my FIB and RSBLQ, I think I should the Interest to node G...
7. My CS and PIT miss, based on my FIB and RSBLQ, I think I should the Interest to node I...
8. My CS hits, I can return a Data packet...
A
B
C
K
D
E
F
A
G
H
I
J

Fig. 3 An Interest forwarding process

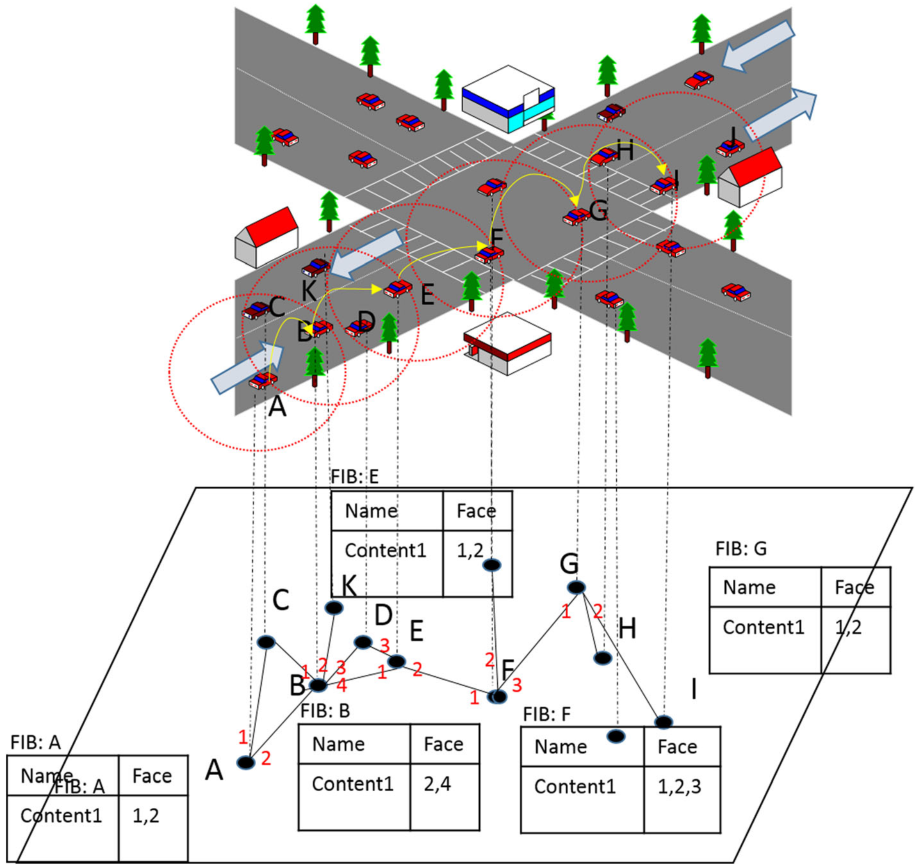

Fig. 4 System mapping model

Based on ICN and REBLQ designing, when Interest arrives at a node, for example node F, it would find out the corresponding FIB entry guide the forwarding through faces 1, 2, and 3, and to decrease the network traffic and choose a more stable connection, it would calculate the LET and link availability of the three connections, selecting the face 3 eventually.

4 Main designing

4.1 Main idea

Our main goals lie in the following two aspects: on the one hand, we hope to build stable and efficient routings to adapt well to the mobile environment; so based on the dynamic metrics, we calculate the value of LET and link availability probability for each FIB outgoing interface, and on the other hand, to reduce the useless traffic caused by flooding, we have a selecting process to choose the optimal outgoing FIB interface with bigger LET and link availability, forwarding the Interest through the final choice.

In ICN, when a node receives an Interest, if CS and PIT are all missed, it would look up FIB, and when FIB hits, RSBLQ does not forward the Interest to all the outgoing interfaces stored in the table entry, it would calculate LET and link available probability for each interface, then selects the best one to be outgoing interface for Interest, and inserts the information of the packet into PIT; but if FIB missed, the node would drop the Interest directly. In addition, to update PIT, routing nodes would scan the table periodically and delete table entries which are beyond tolerance time. The completed forwarding flow chart of nodes is shown in Fig. 5.

4.2 The designing of PIT

To achieve our designing goals and on the basis of the original ICN forwarding mechanism, we improve the structure of PIT, which is shown in Table 2, for each table entry, we add two domains of receive time and tolerance time, which represent the time when the first Interest arrives and the biggest time that each table entry can tolerate respectively. Moreover, at each node, we add a neighbor table (NT) to record the dynamic parameters of its neighbor nodes (Table 3).

The algorithm of RSBLQ

```
Input: Interest1 packet, the CS, PIT and FIB in nodeA
while (Interest1 arrives at nodeA)
    get the requested content name in Interest1;
    if (there is a CS entry of the requested content name)
        build a Data packet to return;
    else if (there is a PIT entry of the requested content name)
        if (the PIT entry does not include the incoming interface of Interest1)
            add the incoming interface to the PIT entry;
        end if
    else if (there is a FIB entry of the requested content name)
        calculate the LET and link availability of each outgoing interface in FIB;
        choose the best one to forward the Interest1;
    else
        drop the Interest1;
    end if
end while
```

To ensure the information freshness of PIT, we need to set a simple PIT updating scheme. When the nodes add a content to a table entry, it would record the receive time and tolerance time, the tolerance time is designed as 2 × TTL, because we want to allow at least once retransmission of the Interest, and according to the biggest distance and propagation rate, we can get the value of TTL which is 3.200 s in average, then our tolerate time is 6.400 s. The existing DSRC (dedicated short-range communication) and WAVE work with a bandwidth of 5.9 Ghz, and their effective transmission distance is 100–500 m. In this range, one-hop communication can be connected with 20 to hundreds of neighbors and vehicles can exchange at least 10 pieces of packet per second, so it means our design can allow at most 64 times of packet delivery during the tolerance time, and if a table entry is beyond the tolerance time, it would be updated or deleted soon.

4.3 The LET algorithm

In our RSBLQ scheme, firstly, on the basis of the vehicular mobility metrics, we need to calculate the LET of each link associated with each FIB outgoing interface, then the nodes can choose a better FIB interface with a longer lifetime.

Upon an urban environment, we choose one vehicle node as the reference point and calculate the relative distance between the reference point and other nodes based on the following preconditions: (1) vehicles are equipped with satellite based on global positioning system (GPS) that can get their speed, direction, location, and so on; (2) the beacon or hello messages are exchanged to identify the neighbors' presence, position, and direction; (3) as shown in Fig. 6, assuming that at the initial time, the distance between two cars p and q is d, their position are represented as $(x_{\mathrm{p}}, y_{\mathrm{p}})$ and $(x_{\mathrm{q}}, y_{\mathrm{q}})$, after a t time, their relative distance is d_1, and their position become $(x_{\mathrm{pt}}, y_{\mathrm{pt}})$ and $(x_{\mathrm{qt}}, y_{\mathrm{qt}})$ separately; (4) vehicular nodes are moving with a constant speed, the speed of p and q are v_{p} and v_{q} , respectively, then we can get their component speed and the angle with axis X; (5) the nodes' biggest transmission range is $R_{\max}$, we assume that p and q are in communication range, which is to say $d \le R_{\max}$, and the communication between p and q would be disconnected when d_1 is greater than $R_{\max}$. Then, we can follow the following formulas to get the LET.

According to the Pythagorean theorem, we can have the expression of the relative distance d_1:

$$d_1 = \sqrt{\left(\left(x_{pt}-x_{qt}\right)^2 + \left(y_{pt}-y_{qt}\right)^2\right)} \tag{1}$$

We use v_{px}, v_{py}, v_{qx}, and v_{qy} to represent the component speed in the direction of coordinates, then we have their position after a t time:

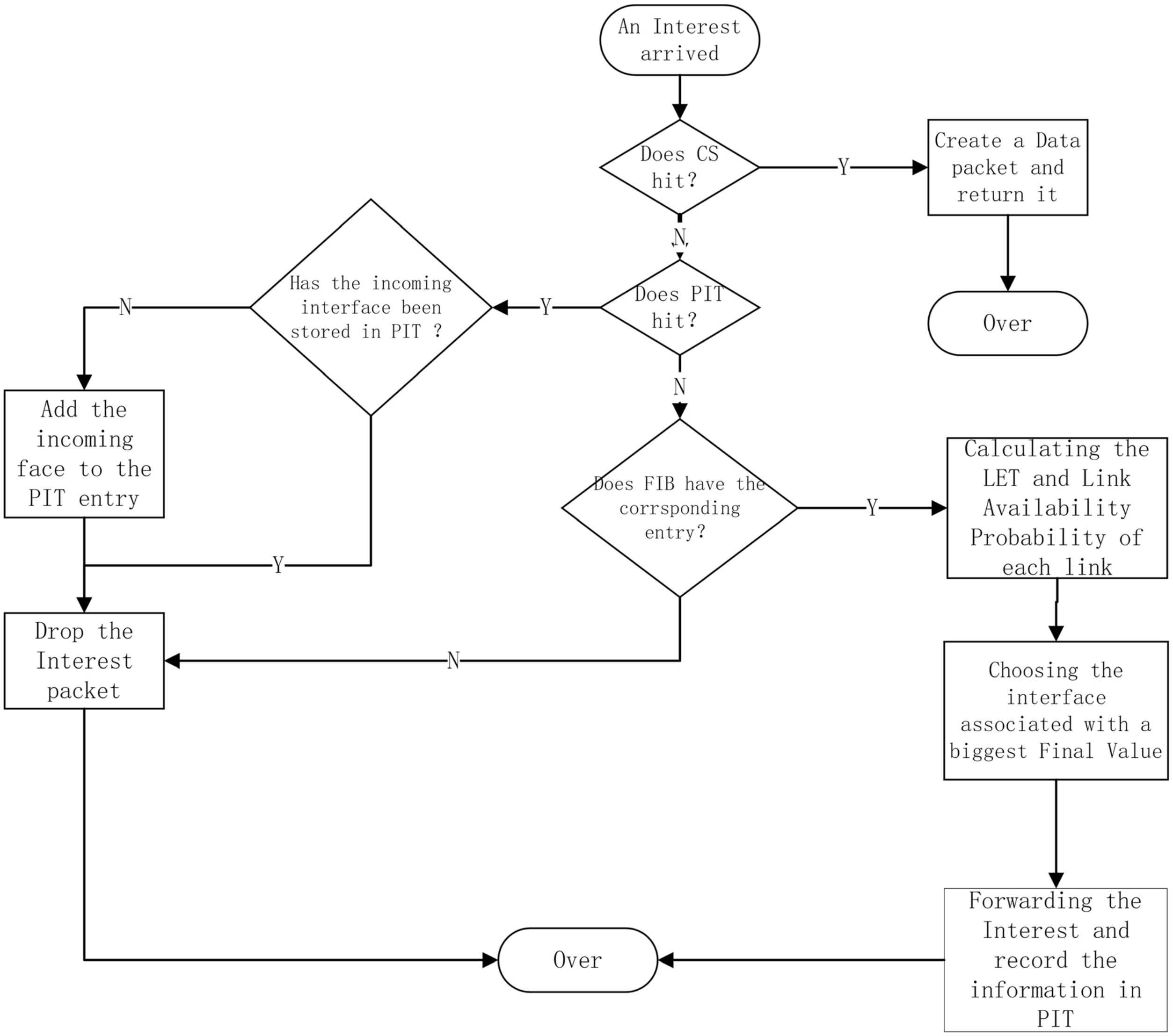
An Interest arrived
Does CS hit?
Y
Create a Data packet and return it
Over
N
Does PIT hit?
Y
Has the incoming interface been stored in PIT ?
N
Add the incoming face to the PIT entry
Y
N
Does FIB have the corrsponding entry?
Y
Calculating the LET and Link Availability Probability of each link
Choosing the interface associated with a biggest Final Value
N
Drop the Interest packet
Forwarding the Interest and record the information in PIT
Over

Fig. 5 The forwarding flow chart

$$x_{\mathrm{pt}} = x_{\mathrm{p}} + v_{\mathrm{px}} \times t = x_{\mathrm{p}} + v_{\mathrm{p}} \times \cos\theta_{\mathrm{p}} \times t \tag{2}$$

$$x_{\mathrm{qt}} = x_{\mathrm{q}} + v_{\mathrm{qx}} \times t = x_{\mathrm{q}} + v_{\mathrm{q}} \times \cos\theta_{\mathrm{q}} \times t \tag{3}$$

$$y_{\mathrm{pt}} = y_{\mathrm{p}} + v_{\mathrm{py}} \times t = y_{\mathrm{p}} + v_{\mathrm{p}} \times \sin\theta_{\mathrm{p}} \times t \tag{4}$$

$$y_{\mathrm{qt}} = y_{\mathrm{q}} + v_{\mathrm{qy}} \times t = y_{\mathrm{q}} + v_{\mathrm{q}} \times \sin\theta_{\mathrm{q}} \times t \tag{5}$$

Combining the above formulas together, the specific expression of $d_1$is as follows:

$$\begin{aligned} d_1 &= \sqrt{\left(\left(x_{\mathrm{pt}}-x_{\mathrm{qt}}\right)^2 + \left(y_{\mathrm{pt}}-y_{\mathrm{qt}}\right)^2\right)} \\ &= \sqrt{\begin{bmatrix} \left(\left(x_{\mathrm{p}} + v_{\mathrm{p}} \times \cos\theta_{\mathrm{p}} \times t\right)-\left(x_{\mathrm{q}} + v_{\mathrm{q}} \times \cos\theta_{\mathrm{q}} \times t\ \right)\right)^2 + \\ \left(\left(y_p + v_p \times \sin\theta_{\mathrm{p}} \times t\right)-\left(y_{\mathrm{q}} + v_{\mathrm{q}} \times \sin\theta_{\mathrm{q}} \times t\ \right)\right)^2 \end{bmatrix}} \end{aligned} \tag{6}$$

The finial expression can be obtained as a quadratic equation about t:

$$\begin{aligned} &\left[{v_{\mathrm{p}}}^2 + {v_{\mathrm{q}}}^2-2v_{\mathrm{p}} \times v_{\mathrm{q}} \times \cos\left(\theta_{\mathrm{p}} + \theta_{\mathrm{q}}\right)\right] \times t^2 + 2t \\ &\quad \times\left[\left(x_{\mathrm{p}}-x_{\mathrm{q}}\right)\left(v_{\mathrm{p}} \times \cos\theta_{\mathrm{p}}-v_{\mathrm{q}} \times \cos\theta_{\mathrm{q}}\right)\right. \\ &\quad \left.+\left(y_{\mathrm{p}}-y_{\mathrm{q}}\right)\left(v_{\mathrm{p}} \times \sin\theta_{\mathrm{p}}-v_{\mathrm{q}} \times \sin\theta_{\mathrm{q}}\right)\right] + d^2-{d_1}^2 = 0 \end{aligned} \tag{7}$$

Table 2 The improved PIT

Prefix	Face	Receive time (s)	Tolerance time (s)
/car1/video/ip3.mp4	6	55.402	6.400
/car5/video/ip2.mp4	5	30.408	6.400

Table 3 Neighbor table

Neighbor nodes	V(m/s)	(*X*, *Y*)	cosθ	Link expiration time	Link available probability
Neighbor's ID	Motion velocity	Position in digital map	Motion direction		

Based on our theoretical analysis, when there is $d_1 > R_{\max}$, the link between two cars is broken, so we can get their LET when there is $d_1 = R_{\max}$:

$$\begin{aligned}&\left[{v_\mathrm{p}}^2+{v_\mathrm{q}}^2–2v_\mathrm{p}\times v_\mathrm{q}\times\cos\left(\theta_\mathrm{p}+\theta_\mathrm{q}\right)\right]\times t^2+2t\\&\quad\times[\left(x_\mathrm{p}–x_\mathrm{q}\right)\left(v_\mathrm{p}\times\cos\theta_\mathrm{p}–v_\mathrm{q}\times\cos\theta_\mathrm{q}\right)\\&\quad+\left(y_\mathrm{p}–y_\mathrm{q}\right)\left(v_\mathrm{p}\times\sin\theta_\mathrm{p}–v_\mathrm{q}\times\sin\theta_\mathrm{q}\right)]+d^2–{R_{\max}}^2=0\end{aligned}\tag{8}$$

As the value of v_p, v_q, x_p, x_q, θ_p, θ_q, y_p, y_q, d, and $\mathrm{R}_{\max}$ are all known, then we can have the solution of Eq. (8), and it is the actual LET value of this link.

With a bigger LET, the routing path can exist a longer time, and the probability of the Data's successful return would be also bigger, thus the Interest packets need to be forwarded through the outgoing interface associated with a bigger LET.

4.4 Link available probability

As the mobile information of vehicle node is independent with each other, and the state of each node is unpredictable, there is a great possibility that the content routing is not available after a period time, so it is not sufficient to have the FIB selecting process only based on LET, which is obtained by the actual mobility metrics. To ensure the stability of our routing paths, we also introduce the link availability probability [30] based on some theoretical analysis, and it can predict whether a link is also available after a certain time.

Firstly, we define the mobility, epoch lengths are exponential and the parameter is λ, and an epoch is a random length interval during which a node moves in a constant direction at a constant speed, then there is

$$E(x)\approx p\{\text{Epoch length}\le x\}=1–e^{-\lambda x}\tag{9}$$

Secondly, node mobility is uncorrelated. To simplify the discussion, we further assume that each node has the same mean epoch length. However, the following derivation can be extended for the case of different mean epoch lengths.

Given a prediction t on the continuously available time for an active link between two nodes at time t_0, the availability of this link, $L(t)$, is defined as:

$$L(t)\approx p\{\text{To last to }\ t_0+t\ \ |\text{available at }\ t_0+t\}\tag{10}$$

Which indicates the probability that the link will be continuously available from time t_0 to $t_0 + t$.

The calculation of $L(t)$ can be divided into two parts: the link availability when the velocities of two nodes

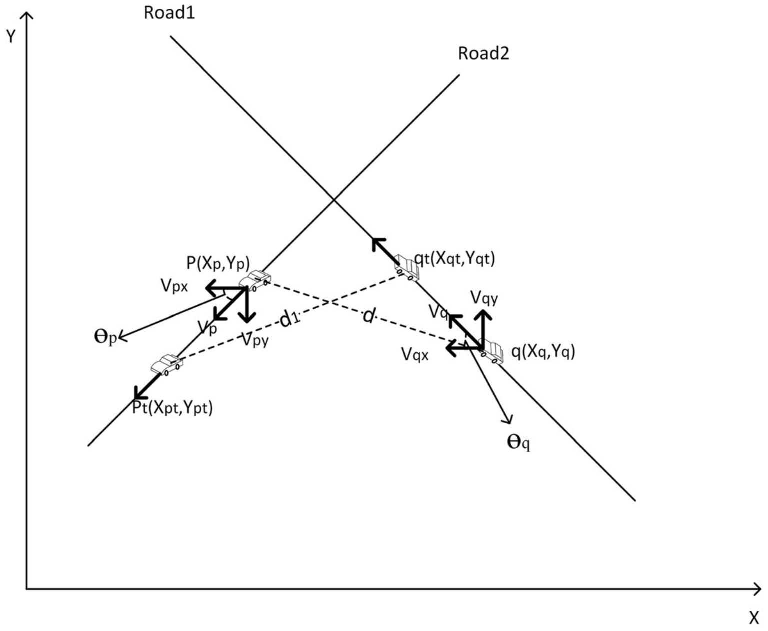

Fig. 6 The schematic diagram of LET

keep unchanged between t_0 to $t_0 + t$, $L_1(t)$ and the one for the other case, L$_2$(t), that is:

$$L(t) = L_1(t) + L_2(t) \tag{11}$$

It is easy to calculate $L_1(t)$, which is equal to the probability that the epochs from t_0 onwards for the two nodes are longer than t. Since nodes' movements are independent of each other and exponential distribution is "memoryless," then there is:

$$L_1(t) \approx p\{\text{Epoch length} \geq t\} \times p\{\text{Epoch length} \geq t\} = e^{-2\lambda t} \tag{12}$$

However, because of the difficulties in learning changes in link status caused by changes on a node's movement, it is difficult to give an accurate calculation for $L_2(t)$.

Denote $\varnothing < t$ as a random variable for the time interval between t_0 and $t_0 + t$ during which either of the two nodes or both change their movements. $P\{\phi \leq \varnothing \leq t\}$ indicates the probability for both nodes to keep their movements unchanged between t_0 and $t_0 + \phi$ while either of them or both change after $t_0 + \phi$. The calculation of $P\{\phi \leq \varnothing \leq t\}$ can be divided into two cases: when only one node changes its movements and when both nodes change their movements between $t_0 + \phi$ and $t_0 + t$, then $\mathrm{P}\{\phi \leq \varnothing \leq t\}$ is given by:

$$\begin{aligned} P\{\phi \leq \varnothing \leq t\} &= 2[E(t) - E(\phi)][1 - E(t)] + [E(t) - E(\phi)]^2 \\ &= e^{-2\lambda\phi} - e^{-2\lambda t} \end{aligned} \tag{13}$$

$L_2(\phi)$ is further used to estimate link availability corresponding to ϕ as follows:

$$L_2(\phi) = \frac{\phi + (t-\phi)pe^{-2\lambda(t-\phi)}}{t} + \epsilon \tag{14}$$

where p is the probability for the two nodes to move closer to each other after changing their movements, and $\epsilon \geq 0$ is an adjustment to the link availability.

We use T_t to represent the total time that a link will be continuously available between t_0 and $t_0 + t$ if the movement changes happen as discussed below. After the first movement change happens at $t_0 + \phi$, the link between two nodes will still be available if the change makes these two nodes to move closer to each other with the probability p. The link is expected to be continuously available from $t_0 + \phi$ to $t_0 + t$, if the two nodes keep their movements unchanged during this period with probability $e^{-2\lambda(t-\phi)}$ (the derivation is similar to Eq. 12). So, T_t can be calculated by:

$$T_t = \phi + (t-\phi)pe^{-2\lambda(t-\phi)} + \ldots \tag{15}$$

The "…" means the accurate T_t can be achieved by repeating the above calculation with the considering second possible change in movements during the remaining period and so on. However, doing so will complicate the calculation. On the other hand, it is reasonable to assume that the average contribution of "…" part to the overall link availability is relatively smaller than the other parts. For the sake of simplicity, ϵ is introduced to estimate the link availability contributed by this part. As discussed later, the use of ϵ can also make our calculation adaptive to the environmental changes.

The average $L_2(\phi)$ is used to estimate $L_2(t)$:

$$\overline{L}_2 = \int_0^t L_2(\phi) f(\phi) d\phi \tag{16}$$

where $f(\phi) \geq 0$ is given by

$$\begin{aligned} f(\phi) &= \lim_{\Delta\phi \to 0} \frac{P\{\phi \leq \varnothing \leq t\} - P\{\phi + \Delta\phi \leq \varnothing \leq t\}}{\Delta\phi} \\ &= \frac{dP\{\phi \leq \varnothing \leq t\}}{d\phi} = 2\lambda e^{-2\lambda\phi} \end{aligned} \tag{17}$$

Then, combining the above equations, we can have:

$$\begin{aligned} \overline{L}_2 \approx \int_0^t \left(\frac{\phi + (t-\phi)pe^{-2\lambda(t-\phi)}}{t} + \epsilon \right) 2\lambda e^{-2\lambda\phi} d\phi = \frac{1}{2\lambda t} \\ + \epsilon + e^{-2\lambda t} \left(p\lambda t - \frac{1}{2\lambda t} - \epsilon - 1 \right) \end{aligned} \tag{18}$$

So, we can have an estimation of $L(t)$:

$$L(t) \approx L_1(\mathrm{t}) + \overline{L}_2 = \frac{1}{2\lambda t} + \epsilon + e^{-2\lambda t} \left(p\lambda t - \frac{1}{2\lambda t} - \epsilon \right) \tag{19}$$

Now, we discuss how to obtain p and ϵ. It is easy to get p for a free environment with the following consideration. After a change in the movements of two nodes, they will move either close to each other or far away from each other, so there is $p = 0.5$.

$$L(t) \approx \left(1 - e^{-2\lambda t}\right) \left(\frac{1}{2\lambda t} + \epsilon \right) + \frac{\lambda t e^{-2\lambda t}}{2} \tag{20}$$

The value of ϵ mainly depends on environmental factors, from the point of view of application, and to devise a measurement method for ϵ, we propose the following measurement.

After a node has a prediction t on an active link with another node at time t_0, it then measures how long this link will really last from t_0, say T_r, if this link is still available just after $t_0 + t$, the node sets $T_r = t$ and does another prediction. Repeating the above operation, the node may have a series of $<T_{r,\ i}, N_i>$ for the same t, where the N_i is the occurrence times of the same T_r(namely $T_{r,\ i}$). Then, we can have a measured $L(t)$ corresponding to t, $L_m(t)$, which is calculated by

$$L_m(t) = \frac{\sum_{i=1}^{N_{T_r}} \left(\frac{T_{r,i}}{t} \times N_i\right)}{\sum_{i=1}^{N_{T_r}} (N_i)} \tag{21}$$

where N_{T_r} is the total number of different T_r, substituting $L(t)$ in Eq. 20 with $L_m(t)$, we can have a measured ϵ, ϵ_m:

$$\epsilon_m \approx \frac{L_m(t) - \frac{\lambda t e^{-2\lambda t}}{2}}{1-e^{-2\lambda t}} - \frac{1}{2\lambda t} \tag{22}$$

Repeating the above operation and calculation with different t values, say, $t_j(j = 1,2,\ldots)$, we can have a series of ϵ_m, $\epsilon_{m,\ j}$, then the final ϵ can be estimated by

$$\epsilon \approx \frac{\sum_{J=1}^{N_t} M_j \epsilon_{m,j}}{\sum_{J=1}^{N_t} M_j} \tag{23}$$

where M_j is the occurrence times of t_j and N_t is the total number of different t values predicted. At the beginning, a node can set $\epsilon = 0$, then we can have a conservative prediction for the link availability, $L_{\min}(t)$, that is,

$$L_{\min}(t) = \frac{1-e^{-2\lambda t}}{2\lambda t} + \frac{\lambda t e^{-2\lambda t}}{2} \tag{24}$$

4.5 Final decision-making

We can get the value of LET and link availability probability for any link based on the above design, then it is necessary to combine LET and link availability probability to make a final decision; therefore, we design the following formula:

$$\text{Final Value} = \alpha \times LET + \beta \times \mathrm{L(t)} \tag{25}$$

where α and β are the weight value for LET and link availability probability, respectively. We can change their influence on the *Final Value* by changing the value of α and β. Thus, our RSBLQ can select an optimal FIB interface with the biggest *Final Value* to forward the Interest packets, instead of choosing all of the FIB interfaces or just flooding.

5 Simulation experiment case

We have obtained a part of a simulation network and its real data in this section, and here, we would show the construction of the information in NT.

We choose one node as the requester, and it has two neighbors, the specific dynamic parameters of each node are shown in Table 4.

Table 4 Dynamic parameters

Node	Parameters			
	(X, Y)	$\overrightarrow{V}$	cosθ	$\lvert\overrightarrow{V}\rvert$(m/s)
District0/northroad0/node0 (requester)	(0, 0)	(3, 4)	0.6	5
District1/northroad1/node1 (neighbor 1)	(15, 20)	(6, 0)	1	6
District2/northroad2/node2 (neighbor 2)	(11, 10)	(3, 2.6)	0.75	4

Based on the information in Table 4, we can get the table items of NT, the cosθ refers to the cosine value of the angle between a velocity vector and the X-axis.

For the convenience of calculation, we take some approximate integral values to replace the real experiment data, but it would not influence the results poorly. Then, we can get the specific NT as follows (Table 5).

Comparing the experimental data in the tables above, we can find out that our RSBLQ has a consistent choice. Based on the results of link expiration time and link available probability, respectively, we can see neighbor 1 outperforms the others in the FIB selecting process. Hence, the results verify the accuracy and consistency of our scheme.

6 Simulation results and analysis

6.1 System settings

As shown in Fig. 7, Similar with TCP/IP architecture, ICN applies an hourglass model too, a type of thin waist network architecture that does not have any unnecessary restrictions on the lower technology innovation, so NDN keep the same "thin waist type" structure, but its central layer is not the network layer but the contents, and here, we focus on the content and security layer. The architecture must incorporate security features. The security of TCP/IP was built later, and it did not take into account the problem at the beginning of its design, so it has many limitations. ICN sets up a barrier at the waist for all named data to provide basic security, and ICN data signature can not only ensure the integrity of the data, but also ensure the authenticity of the data sources, so in ICN, although the producers and consumers do not communicate directly, it is able to ensure the end-to-end communication security. What is more, ICN has a function of network traffic self-regulation. Different from the open loop data transmission of IP, ICN designs a

Table 5 Neighbor table

Neighbor node label	V(m/s)	*(X, Y)*	*cosθ*	Link expiration time	Link available probability
District 1/northroad1/ node1	6	(15, 20)	1	51	0.73
District 2/northroad2/ node2	4	(11, 10)	0.75	49.8	0.52

flow balance feedback loop in the "thin waist" model, which matches each data and Interest in each link and provides a powerful mechanism to balance the network traffic. Therefore, ICN can adjust traffic automatically in unicast and multicast cases and does not rely on transport protocols. At the same time, NDN also separate the routing scheme and forwarding mechanism.

Following the ICN designing, the nodes structure in our simulation is shown in Fig. 8, and each vehicular node would have a CS to cache the content, a PIT to record the pending Interest information and guide the Data's return, and a FIB to act as a routing table function. Moreover, the nodes would have several interfaces connecting with different nodes (other vehicles, RSU, and so on).

Figure 9 has shown the structure of the Interest and Data packets, for simplicity, we only set three fields for the Interest, a content name field to record the requested content ID, a nonce field to identify an Interest uniquely, and a selector field to record some extra information. To ensure security, we also add a signature and a signed information field to the Data packets.

At first, it is necessary to make our simulation targets clear; according to our design, our simulation contains the following aspects:

1. Rewriting the structure of PIT
2. Creating a neighbor table for each node to collect the dynamic information
3. Rewriting the forwarding strategy and adding the calculating process of LET and link availability
4. Setting FIB forward the packets according the calculating results

More concretely, to realize the settings of the PIT and FIB, we have rewritten the corresponding codes in the documents of *ndn-pit-entry.cc* and *ndn-fib-entry.cc*, where a function *Entry:: AddOrUpdateRoutingMetric ()* is used to add and update the routing metric in FIB, and we have created two arrays face[] and FIBmetric[] to record the calculation result of each FIB interfaces, which would be used to the FIB selecting process of LET and link availability. The rewriting of forwarding strategy was realized by modifying the function *fw::DoPropogateInterest()* and replacing the *PropagateInterest ()* with the codes associated with specific realization of the process mentioned above. We also choose to write two new documents-CustomStrategy.cc and CustomStrategy.h based on the source code of *best-route.cc* to realize our own modular strategy.

6.2 Parameters settings

Our simulation is run at ndnSIM2.0 in NS-3, and we choose a simple Sprint network topology [31] and configure RSBLQ strategy at each node. The maximum communication distance is 50 m (here is not 500, because our simulation area is limited). We also build a small traffic scene graph and simple road topology by VanetMobiSim [32]. The motion model of vehicles is IDM_LC (Intelligent Driver Model with Lane Changes), in which way the cars can change lane automatically, and the setting of specific parameters is shown in Table 6.

We set the requests of consumer nodes to satisfy Poisson characteristics, and within a 100-s simulation time, they send 10 Interest packages per second in average. Meanwhile, considering the effect of content popularity, and in order to make the simulation closer to the real scene, the type of the request content satisfies the zipf law, which means that 80% of internet users are more interested in only 20% content in the network. The replacement algorithm of CS is LRU (Least Recently Used) [33].

6.3 Simulation results analysis

To analyze the netwrok performance of RSBLQ, we would have a simulation results comparison with TCP, CCVN [14], and V-NDN [16] in terms of the following aspects.

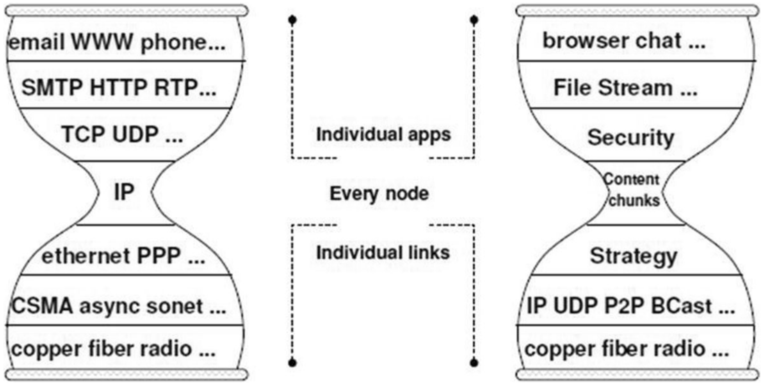

Fig. 7 ICN architecture

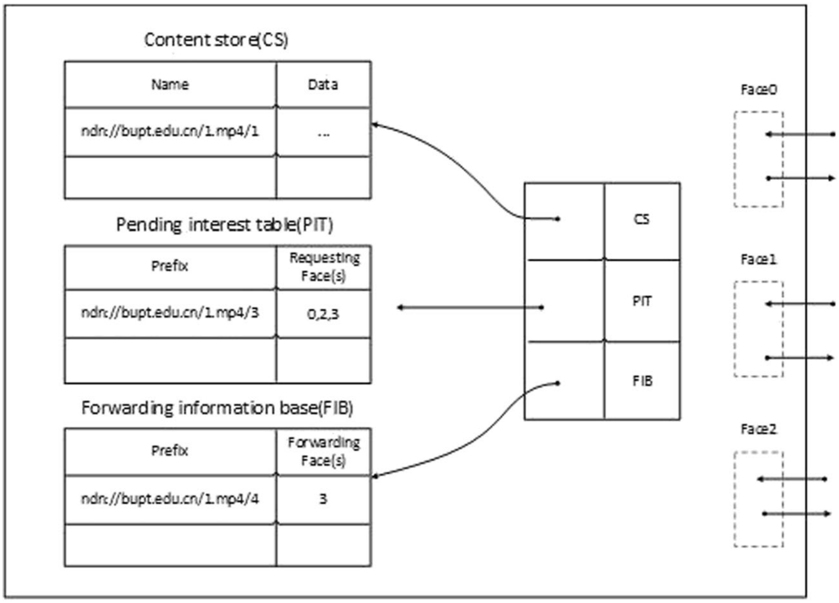
Content store(CS)
Name
Data
ndn://bupt.edu.cn/1.mp4/1
...
Pending interest table(PIT)
Prefix
Requesting Face(s)
ndn://bupt.edu.cn/1.mp4/3
0,2,3
Forwarding information base(FIB)
Prefix
Forwarding Face(s)
ndn://bupt.edu.cn/1.mp4/4
3
CS
PIT
FIB
Face0
Face1
Face2

Fig. 8 Nodes' structure

(1) Transmission efficiency. We use the average finish time of the same tasks to represent the transmission efficiency of each scheme, and it can reflect the routing performance directly.

(2) Response time. It is represented by the time from which a node sends an Interest package until it receives the corresponding Data packet. The average response time also can be used to judge the efficiency of a forwarding strategy.

(3) Cache hit ratio. It is the percent number of Data packets received from the cache but not the source nodes, and it is used to determine the effectiveness of a caching policy.

(4) Network traffic. It is the total number of Interest packets and Data packets in network, which is used to determine the network load.

6.3.1 Transmission efficiency

We randomly choose different clients and servers as a group and use the average finish time of the same tasks to represent the transmission efficiency. As shown in Figs. 10, 11, and 12, slots on the black lines mean an equal transmission efficiency to our scheme. We can see TCP/IP has the worst results and its finish time is in a range of 18 to 30 s. CCVN is slightly better than V-NDN, and they are in a range of 16 to 28 s and 15 to

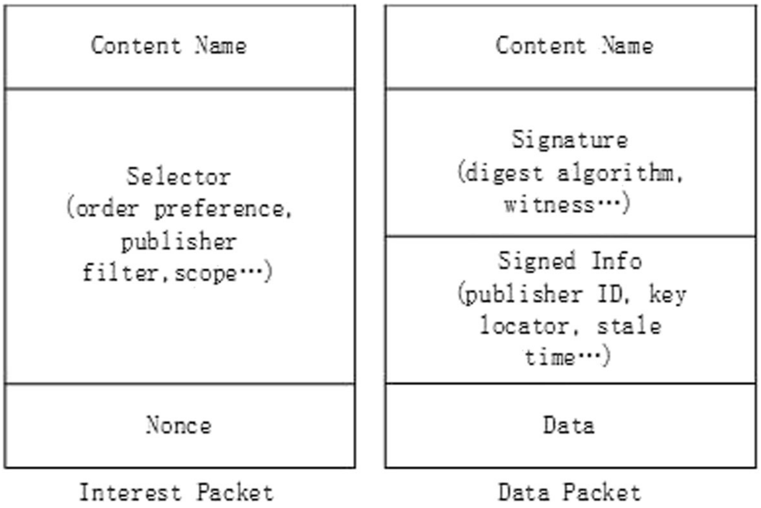

Fig. 9 Packets' structure

Table 6 Simulation parameter settings

Parameters	Value
Simulation platform	ndnSIM2.0 in NS-3
Network topology	Sprint network topology
Cache size	20
Bandwidth	1 Mbps
The communication distance	50 m
Scope of the environment	1000(m) × 1000(m)
The data link type	DLT_IEEE802_11_RADIO
Packet size	1040 B
Motion model	IDM_LC
Wireless propagation model	ConstantSpeedPropagationDelayModel
Signal attenuation model	FriisPropagationLossModel
RxGain	−10 db
The simulation time	100 s
(α, β)	(0.5, 0.5)

26 s separately. V-NDN is closer to our RSBLQ. We think it is because the inflexible and fixed communication method in TCP/IP takes too much time to find the destination; however, the other three schemes do not care where the hosts are. And complete flooding in CCVN is easy to result in broadcast storm, V-NDN can decrease the useless traffic by listening to the channel, and then V-NDN plays better than CCVN. Our RSBLQ avoids broadcasting and selects optimal outgoing interfaces in FIB, so it has the least finish time.

6.3.2 Response time

We use the average response time to represent the network performance, and it is calculated by the following formula:

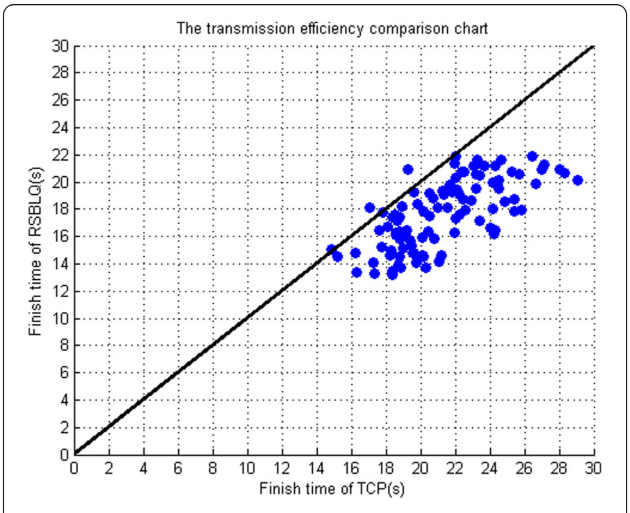

Fig. 10 The transmission efficiency comparison with TCP

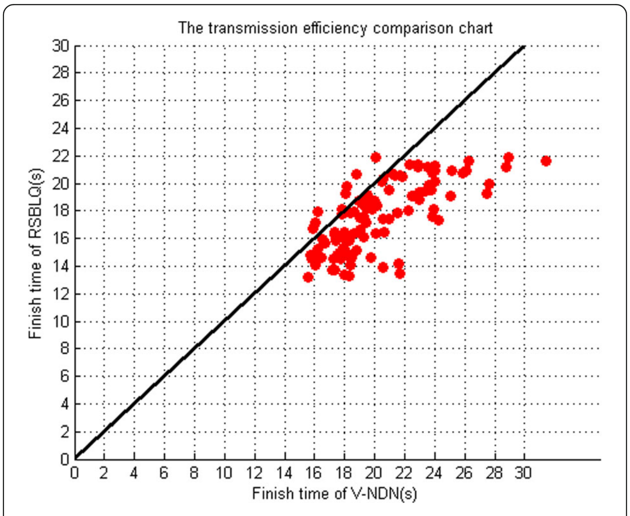

Fig. 11 The transmission efficiency comparison with V-NDN

$$\mathrm{ART}_i=\frac{\sum_{j=1}^{i}\mathrm{Time}_j}{i} \tag{26}$$

where the ART_i (average response time) is the average response time within the first i seconds. *Time*$_j$ is the response time at the j_{th}. s

As shown in Fig. 13, our RSBLQ plays better than the other schemes. The LCE (Leave Copy Everywhere) is used in NDN and CCNx; thus, there is no doubt that the average response time for V-NDN and CCVN are both superior than TCP, which makes no use of any cache. And CCVN has a closer result with our RSBLQ: before 60 s, RSBLQ is better on the average response time, because it carries out the content routing time prediction, it makes the cache more easily to be hit and also reduces the time delay, but after 60 s, we can find out that RSBLQ and CCVN have the same average response

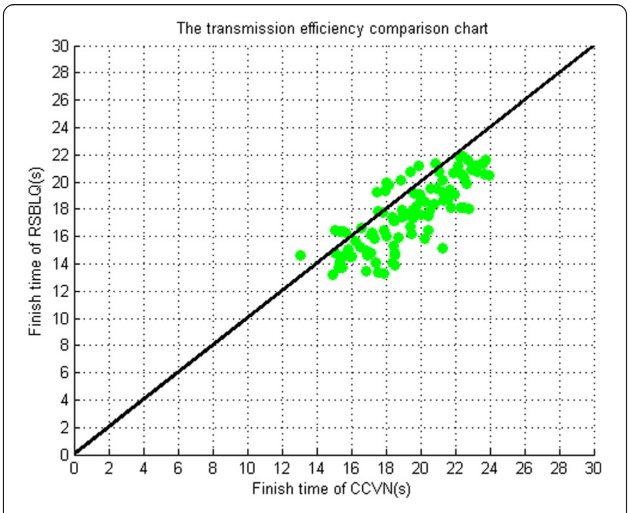

Fig. 12 The transmission efficiency comparison with CCVN

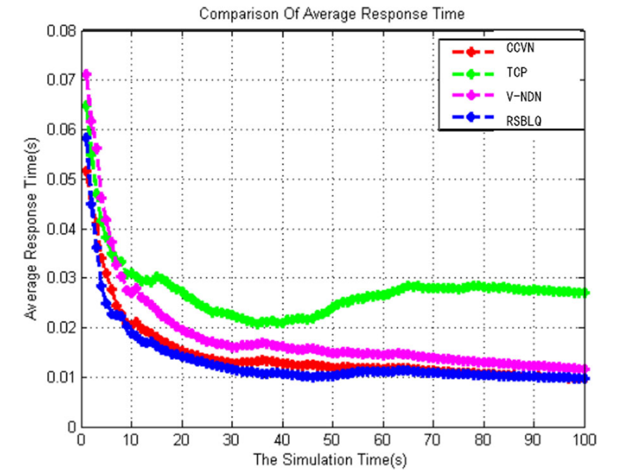

Fig. 13 The comparison of average response time

time. As a result, RSBLQ is much better than the others, because in our scheme, we chose the link with a longer LET and bigger link available probability, and the selected paths must be the sub-paths for CCVN and V-NDN; thus, their average response time are both bigger than RSBLQ in general.

6.3.3 Cache hit ratio

We have a comparison of the number of cache hits and misses, then get the cache hit ratio in Fig. 14.

Under the traditional TCP/IP environment, there is no caching mechanism, so the cache hit ratio loses its meaning. From the results, we can see the cache hit ratio of RSBLQ is about 40%, CCVN has a 33% result, and V-NDN has a worse performance, which is only about 28%. We think it is because the selecting process in RSBLQ gets the stable routing paths with a bigger time tolerance, so it can make full use of the network cache; as V-NDN applies a complete flooding way, which would have a bad impact on the network performance, this may be the reason for its worse result.

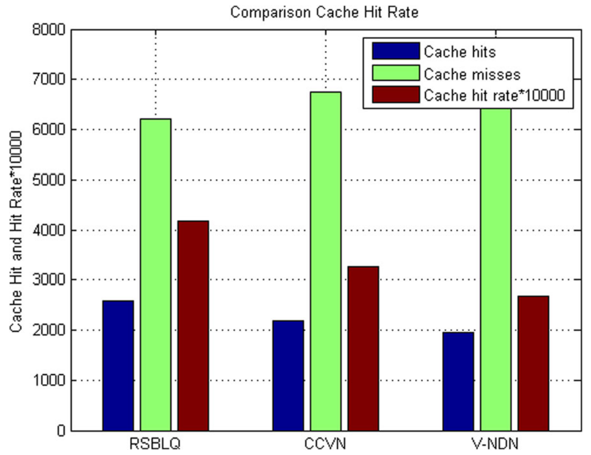

Fig. 14 Cache hit ratio

6.3.4 Network traffic

We compare the network traffic from the following two aspects: instantaneous network flow and average network traffic. The instantaneous network traffic is the sum of flow rate within per unit time, and the average traffic is calculated by the following formula:

$$\mathrm{ANT}_i = \frac{\sum_{j=1}^{i} nt_j}{i} \tag{27}$$

where ANT_i is the average network traffic within the first i seconds. nt_j is the network traffic at the j_{th} s.

The comparison result of instantaneous flow for the four strategies is shown in Fig. 15. The horizontal axis is the simulation time, and the vertical axis is the sum of the network flow that all the nodes generate within a unit time. We can see that the results at the start stage are close. But with the increasing of time, the curves begin to fluctuate, the volatility of the V-NDN is the biggest, and at 49 s, its instantaneous flow is close to 1100 kB/s. The instantaneous flow of TCP/IP tends to increase, but the others show the decreasing trend. Moreover, our RSBLQ has a more significant decline compared to V-NDN and CCVN; in terms of the smooth of traffic flow, RSBLQ and CCVN are much better than V-NDN. It is no doubt that the broadcasting way in CCVN and V-NDN makes more traffic than others, CCVN has the biggest fluctuation for unstable channel, the results of TCP/IP are caused by its redundant paths, and the selecting algorithm in RSBLQ decreases the useless traffic to a large extent.

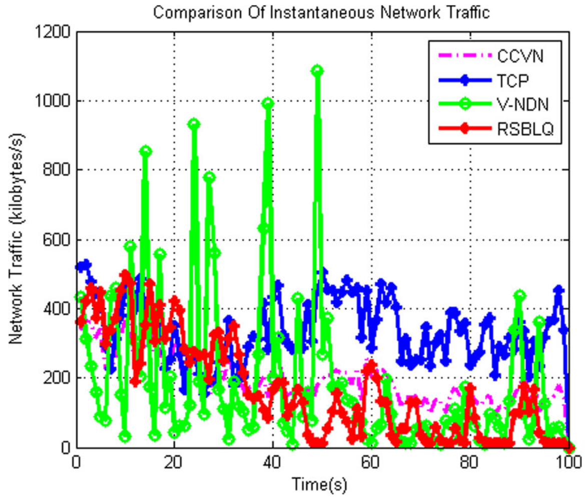

Fig. 15 The instantaneous network traffic contrast

Figure 16 describes the average network traffic of the four schemes. It is not hard to see that the average network traffic of TCP is the highest; our RSBLQ has a rapid convergence, and after the first 50 s, the curves of RSBLQ and CCVN are very close. It also shows the RSBLQ strategy has a better performance than the others totally.

6.4 Simulation results based on different node number and speed

In this section, our simulation results will be compared with the performance of V-NDN [16] and the Strategy Based on Hops [14]. The comparison includes the following four parts:

1. Average transmission delay refers to the total average time from which an Interest was sent until the corresponding Data was returned, which is an important network index to reflect the routing performance.
2. Average length of the path is calculated by the average hops of the routing paths; it can tell us whether the routing we have chosen is the better or not.
3. Data delivery ratio (DDR) is equal to the ratio between the number of the Data packets received and the total number of the Interest packets sent out, and it is an important parameter to reflect the reliability of a routing mechanism.
4. Network cache is the whole number of a content cached in the network, and here, we use the percentage between the caching nodes and the whole number of the nodes to represent it. For ICN, the number of the caching can show the performance of a routing protocol at a certain degree, and what we want is to try best to have a lower network caching under a same DDR request.

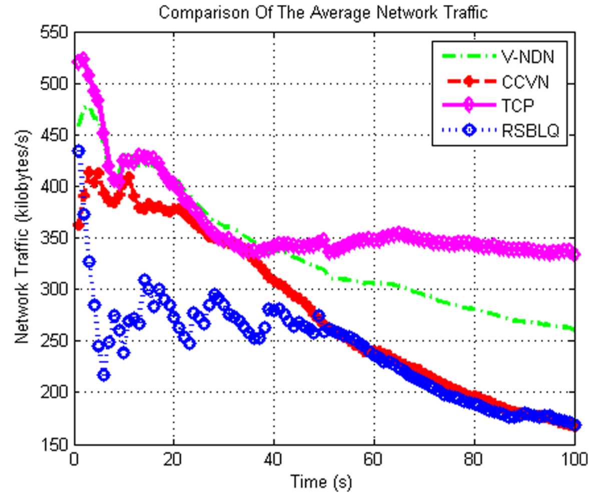

Fig. 16 The average network traffic

6.4.1 Average transmission delay

There may be more hops of the flooding with a big node density, so that the delay in V-NDN and the Strategy Based on Hops are increased sharply with the increase of the nodes' number, but our strategy has just a very little change as the more routing nodes there are, the more choices RSBLQ will have, and the delay rises in the calculating process but not too much (Figs. 17 and 18).

However, in the environment with a different speed, our strategy performs better than the another two schemes within a range of the speed and even has a little decrease of the average delay, because the proper increase of the speed will be helpful to the packet delivery in RSBLQ, but if the speed is too great to result in lots of link failure, the link delay will increase quickly and even bigger than V-NDN and the Strategy Based on Hops.

6.4.2 Average hops

As same as the analysis of the average delay, the more nodes may cause more hops in V-NDN and the Strategy Based on Hops, so that the average hops of the two strategies is rising with the nodes' increase. The Strategy Based on Hops is better than the V-DND for the hop limitation function of the threshold *T*; the strategy of creating routing paths of RSBLQ is constant, it has some little fluctuations but basically in a formal state (Figs. 19 and 20).

Like in Fig. 18, the proper increase does good to the performance of RSBLQ, but once the speed is great enough, the link failure probability will become bigger.

6.4.3 Network cache

With the same request of DDR, the smaller network cache means the better performance of a network routing protocol. As shown in Fig. 21, the vertical axis

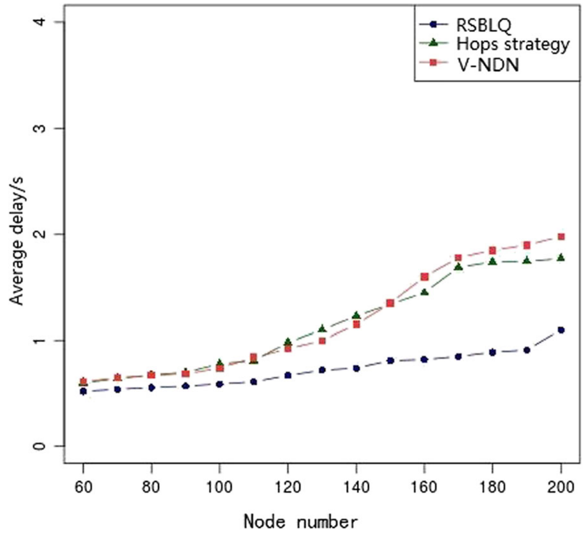

Fig. 17 Average delay with different nodes

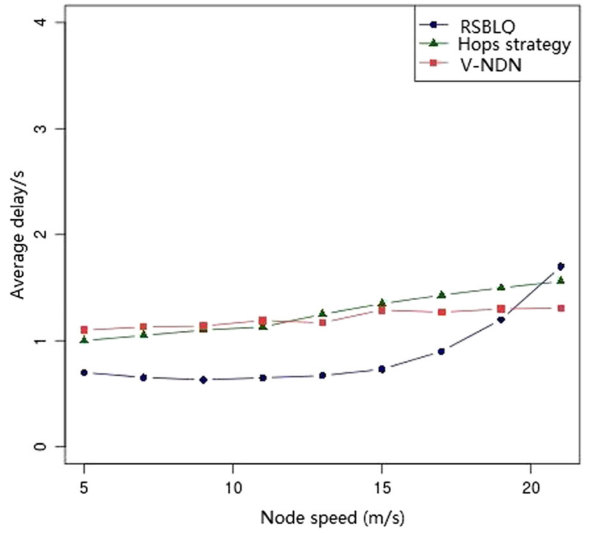

Fig. 18 Average delay with different speed

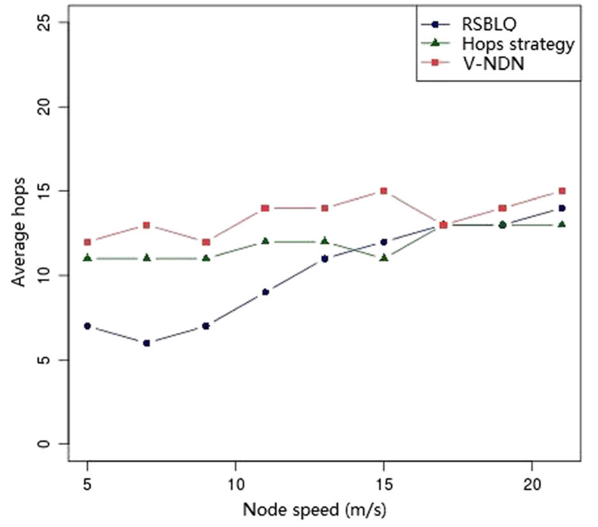

Fig. 20 Average hops with different speed

represents the percentage of the caching node in the whole network and the horizontal axis is the DDR; we can conclude that RSBLQ has a bigger advantage than the another two strategies, due to the process of FIB selecting, which chooses more stable routings, and it does not only ensure the successful return of the Data packet but also reduce the useless network traffic.

7 Conclusions

In this paper, considering traditional TCP/IP network mode could not adapt well to the high dynamic feature of urban VANET, we studied the content routing forwarding strategies in several schemes combining ICN with VANET. In order to get more stable and reliable ICN routing paths and avoid the useless traffic caused by flooding, we propose our RSBLQ based on link quality, when the Interest packets need to be forwarded by FIB, our scheme would calculate the LET and link available probability as the metrics for each link associated with the FIB interface, then it selects the FIB interface with the biggest *Final Value* as the optimal choice to forward the Interests. From the simulation results, we can conclude that our RSBLQ gets a better performance than TCP/IP, CCVN, and V-NDN in the transmission efficiency, response time, cache hit ratio, and network traffic.

Although RSBLQ is more efficient than some former approaches, there are also more works that need to be

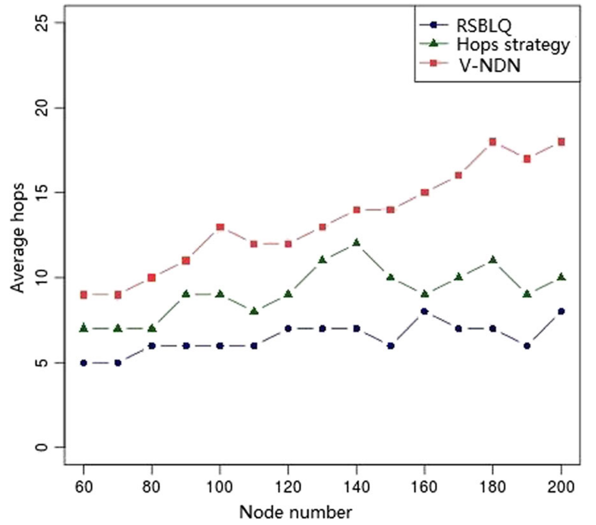

Fig. 19 Average hops with different nodes

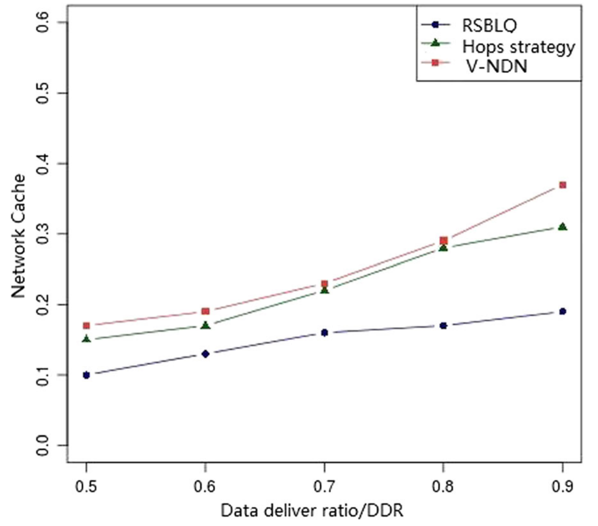

Fig. 21 Network cache with different DDR

done in the future. Firstly, we only consider a constant motion model for vehicles, but in the real environment, the vehicular mobile parameters are always changing; then, in our design, we have simplified many factors in the real scene, such as complex road topology, traffic lights, RSU (road service unit), and so on; thus, it is necessary to have a better routing scheme, which takes all the factors into account; and we also need to think about more useful link quality metrics and define them properly; finally, we would like to combine the caching, naming, and security mechanism with forwarding strategies to make our plan more suitable to VANET in the future works.

Acknowledgements
This manuscript is approved by all authors for publication. On behalf of my co-authors, I would like to declare the work described is original research that has not been published previously and not under consideration for publication elsewhere.

Authors' contributions
LR and XQ contributed to the conception of the study. LR and HG performed the data analyses and wrote the manuscript. RS and HH helped perform the analysis with constructive discussions. All authors read and approved the final manuscript.

Funding
The work is supported by the National Natural Science Foundation of China (61302078, 61372108), Funds for Creative Research Groups of China (61121061), and Beijing Higher Education Young Elite Teacher Project (YETP0476).

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

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

Received: 27 March 2017 Accepted: 15 November 2017
Published online: 08 December 2017

References

1. M Lister, B Christie, C Lowrance, in *18th World Congress on Intelligent Transport Systems and ITS America Annual Meeting*. Educating tomorrows transport professionals—ITS standards (2011), pp. 6036–6043 https://itswc.confex.com/itswc/WC2011/webprogram/Paper1781.html
2. F Cunha, L Villas, A Boukerche, Data communication in VANETs: protocols, application and challenges. Ad Hoc Netw. **44**, 90–103 (2016). doi: 10.1016/j.adhoc.2016.02.017
3. PK Sahoo, MJ Chiang, SL Wu, SVANET: A smart vehicular *ad hoc* network for efficient data transmission with wireless sensors. Sensors **14**, 22230–22260 (2014). doi: 10.3390/s141222230
4. O Kaiwartya, AH Abdullah, Y Cao, A Altameem, M Prasad, CT Lin, XL Liu, Internet of vehicles: motivation, layered architecture, network model, challenges, and future aspects. IEEE Access **4**, 5357–5373 (2016). doi: 10.1109/ACCESS.2016.2603219
5. L Yaqoob, L Ahmad, E Ahmed, A Gani, M Imran, N Guizani, Overcoming the key challenges to establishing vehicular communication: is SDN the answer? IEEE Commun. Mag. **55**(7), 128–134 (2017). doi: 10.1109/MCOM.2017.1601183
6. R Hussain, J Son, H Eun, S Kim, O Heekuck, in *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*. Rethinking vehicular communications: merging VANET with cloud computing (2012), pp. 606–609. doi: 10.1109/CloudCom.2012.6427481
7. J Li, Z Chen, X Shi, The research of ICN architecture and technology. Netinfo Security **4**, 75–80 (2012). doi: 10.3969/j.issn.1671-1122.2012.04.021
8. M Arifuzzaman, K Yu, NN Quang, T Sato, in *2015 European Conference on Networks and Communications*. Locating the content in the locality: ICN caching and routing strategy revisited (2015), pp. 423–428. doi: 10.1109/EuCNC.2015.7194111
9. CCNx | PARC's implementation of content-centric networking. http://www.ccnx.org/
10. Named Data Networking - A Future Internet Architecture. http://named-data.net
11. SM Jiang, DJ He, JQ Rao, in *Proceeding of 20th Annual Joint Conference of IEEE Computer and Communications Societies*. A prediction-based link availability estimation for mobile ad hoc networks (2001), pp. 1745–1752. doi: 10.1109/INFCOM.2001.916672
12. M Meisel, V Pappas, LX Zhang, in *Fifth ACM International Workshop on Mobility in the Evolving Internet Architecture*. Ad hoc networking via named data (2010), pp. 3–8. doi: 10.1145/1859983.1859986
13. JZ Wang, W Ryuji, LX Zhang, in *2010 IEEE Vehicular Networking Conference*. DMND: collecting data from mobiles using named data (2010), pp. 49–56. doi: 10.1109/VNC.2010.5698270
14. M Amadeo, C Campolo, A Molonaro, Enhancing content-centric networking for vehicular environments. Comput. Netw. **57**(16), 3222–3234 (2013). doi: 10.1016/j.comnet.2013.07.005
15. YT Yu, X Li, G Mario, MY Sanadidi, in *9th International Wireless Communications and Mobile Computing Conference (IWCMC)*. Scalable VANET content routing using hierarchical bloom filters (2013), pp. 1629–1634. doi: 10.1109/IWCMC.2013.6583800
16. G Grassi, D Pesavento, G Pau, V Rama, W Ryuji, LX Zhang, in *2014 IEEE Conference on Computer Communication Workshops*. VANET via named data networking (2014), pp. 410–415. doi: 10.1109/INFCOMW.2014.6849267
17. L Wu, *Research and Implementation of Information Distribution Based on VANET in ICN* (Beijing University of Posts and Telecommunications, Dissertation, 2015)
18. YT Yu, G Mario, in *2016 International Conference on Computing, Networking and Communications (ICNC)*. Information-centric VANETs: a study of content routing design alternatives (2016), pp. 1–5. doi: 10.1109/ICCNC.2016.7440705
19. M Yu, JM Bao, in *2010 IEEE Wireless Communication and Networking Conference*. A ranging-based link availability routing algorithm for MANETs (2010), pp. 1–6. doi: 10.1109/WCNC.2010.5506429
20. M Yu, W Liu, T Xing, in *2011 IEEE International Conference on Networking, Sensing and Control (ICNSC)*. Link availability modeling for routing algorithms to reduce the link break time in MANETs (2011), pp. 115–120. doi: 10.1109/ICNSC.2011.5874913
21. Q Han, Y Bai, L Gong, W Wu, Link availability prediction-based reliable routing for mobile ad hoc networks. IET Commun. **5**(16), 2291–2300 (2011). doi: 10.1049/iet-com.2010.0946
22. S Kumar, N Javaid, Z Yousuf, H Kumar, ZA Khan, U Qasim, in *2012 IEEE Conference on Open Systems (ICOS)*. On link availability probability of routing protocols for urban scenario in VANETs (2012), pp. 1–6. doi: 10.1109/ICOS.2012.6417632
23. DB Johnson, DA Maltz, J Broch, DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. Ad hoc Networking **5**, 139–172 (2001)
24. W Su, M Gerla, *IPv6 Flow Handoff in Ad hoc Wireless Networks Using Mobility Prediction* (Proceedings of the IEEE Global Telecommunications Conference, In, 1999), pp. 271–275. doi: 10.1109/GLOCOM.1999.831647
25. N Meghanathan, Exploring the stability-energy consumption-delay-network lifetime tradeoff of mobile ad hoc network routing protocols. J Networks **3**(2), 17–28 (2007). doi: 10.1109/ICWMC.2007.85
26. N Meghanathan, Path stability based ranking of mobile ad hoc network routing protocols. ISAST Trans J Commun Networking **1**(1), 66–73 (2007)
27. M Natarajan, in *2012 IEEE Consumer Communications and Networking Conference (CCNC)*. A MANET multicast routing protocol for stable trees based on the inverse of Link Expiration Times (2012), pp. 9447–9951. doi: 10.1109/CCNC.2012.6180965
28. H Chiraz, VDB Adrien, I Hanen, V Thierry, AS Leila, in *Global Information Infrastructure Symposium*. Novel link availability aware metrics for routing in wireless mesh networks (2013), pp. 1–7. doi: 10.1109/GIIS.2013.6684364
29. A Izhar, AR Kazi, ET Kemal, in *27th Biennial Symposium on Communications (QBSC)*. A mobility tolerant routing algorithm for vehicular ad hoc networks (2014), pp. 113–117. doi: 10.1109/QBSC.2014.6841196

30. SM Jiang, DJ He, JQ Rao, in *IEEE INFOCOM 2001. Conference on Computer Communications*. A prediction-based link availability estimation for mobile ad hoc networks (2001), pp. 1745–1752. doi: 10.1109/INFCOM.2001.916672
31. N Spring, R Mahajan, D Wetherall, T Anderson, Measuring ISP topologies with rocketfuel. IEEE/ACM Trans Networking (TON) **12**, 2–16 (2004). doi: 10.1109/TNET.2003.822655
32. VanetMobiSim. http://vanet.eurecom.fr/.
33. MZ Weng, YL Shang, Y Tian, in *Proceeding of the 8th International ICST Conference*. The design and implementation of LRU-based web cache (2013), pp. 400–404. doi: 10.1109/ChinaCom.2013.6694629