

RESEARCH

Open Access



# Hybrid wireless networks with FIB-based Named Data Networking

Zhuo Li<sup>1,2,3\*</sup> , Kaihua Liu<sup>4</sup>, Deliang Liu<sup>5</sup>, Heping Shi<sup>6</sup> and Yutong Chen<sup>3</sup>

## Abstract

In Named Data Networking (NDN) architecture, packets carry data names rather than source or destination addresses. To realize this paradigm, each NDN router maintains three data structures CS, PIT, FIB in the forwarding plane. Designing a quick enough forwarding plane with high capacity is a major challenge within the overall NDN research area. In this paper, we present a novel implementation of FIB called MaFIB. The evaluations indicate that the excellent data structure and the efficient lookup algorithm of MaFIB can effectively reduce the memory cost and can accelerate the lookup process. Its outstanding performance in probability of false positive also makes MaFIB meet current network requirements.

**Keywords:** Named Data Networking (NDN), Forwarding information base (FIB), Mapping Bloom filter (MBF), Bloom filter

## 1 Introduction

In terms of wireless network topology, two basic types of wireless networks are available: infrastructure-based cellular network and infrastructureless ad hoc network [1, 2]. Hybrid wireless networks are combining these two above technologies. Thus, a cellular network can be extended into regions where no base station is reachable. The base stations can also then provide access to other networks. Therefore, a hybrid wireless network takes advantage of both the long distance communication property of the infrastructure-based cellular network and the peer-to-peer communication property of the infrastructureless ad hoc network. Hybrid wireless network is considered as a promising candidate for the future network [3, 4]. In this paper, we study hybrid wireless networks with Named Data Networking (NDN) in the infrastructure mode.

NDN [5] is a recently proposal of future internet architecture [6]. Among many interesting features, NDN is a receiver-driven, data-centric communication protocol, which focuses on the content rather than where the content is from. All communications in NDN are performed by using two distinct types of packets: Interest and Data,

both of which carry a *content name* [5] that uniquely identifies a piece of data. To receive data, a consumer sends out an Interest packet. A router remembers the interface from which the Interest packet comes and then forwards the Interest packet through retrieving the longest matching prefix of content name in its forwarding information base (FIB). In the FIB of NDN router, an entry contains the following fields: <name prefix, stale time, interface ID, routing preference, RTT, status, rate limit > [5, 7]. As a key component in the NDN router, FIB is similar to the FIB in an IP router except that it contains name prefixes instead of IP address prefixes, and it may show multiple interfaces for a given name prefix. Meanwhile, the variable-length and hierarchical content name in NDN brings a significant cost, both in router storage and in packet processing [5, 7]. In NDN, the content name of packet like a URL is more complex than the IP address, so the router of NDN calls for a much more memory space. If FIB has to be stored in large and slow memories, it is really a heavy burden for the routers to complete fast forwarding. Consequently, there are two open questions: how to reduce the size of FIB to store the forwarding information efficiently on the routers, and how to fast operate on FIB [7].

In the forwarding plane of NDN, FIB applies the method mentioned in [5], and the longest prefix matching (LPM) [8] as the matching algorithm. This traditional structure of FIB can effectively reduce the number of

\* Correspondence: zli@tju.edu.cn

<sup>1</sup>Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, Tianjin Normal University, 300387 Tianjin, China

<sup>2</sup>College of Electronic and Communication Engineering, Tianjin Normal University, 300387 Tianjin, China

Full list of author information is available at the end of the article

In view of the deficiency of FIB in NDN router, this paper proposes an enhanced FIB, namely, MaFIB, which realizes variable-length and hierarchical content name-oriented lookup. Meanwhile, the use of Mapping Bloom filter (MBF) [12] and multi-level hardware memory further improve the lookup speed in MaFIB, reduces the frequency of access to memory and realizes significant reduction in memory consumption. Our evaluations indicate that MaFIB can minimize the on-chip memory consumption to 4.194 MB and can accelerate the lookup process in the memory. Meanwhile, the probability of false positive is under 1% for 2 million names. It allows MaFIB to use SRAM as on-chip memory and satisfies the current network requirements.

## 2 Characteristics of FIB in NDN router

In an NDN router, the FIB is similar to the FIB of an IP router in many ways [13]. For example, both of them need to use the LPM algorithm for lookup; the forwarding information recorded is basically the same [5]. But, the content name-based packet naming method of NDN makes the characteristics of FIB change dramatically [7]. Therefore, this section will elaborate on the characteristics of FIB in the NDN.

## 2.1 Variable-length and hierarchical names

In an NDN router, all lookup operations reply upon finding the longest matching prefix for a given content name of packet, which differs from IP's LPM in two substantial ways [5]. First, the content names of NDN are explicitly hierarchical, consisting of a series of

delimited components while IP addresses can match a prefix at any bit position [8]. Second, IP addresses are fixed length, while the content name of NDN are variable and unbounded which leads to obvious differences in designing FIB. For example, in the FIB of IP router, a group of Bloom filter is used to determine whether the longest matching prefix is in FIB. As IP has a fixed length, the quantity of Bloom filter can be determined when designing the FIB. However, in the FIB of NDN router, as the content name is unbounded, there is no way to determine the specific number of matching prefix. As a result, the quantity of Bloom filter cannot be determined directly when designing the FIB [7].

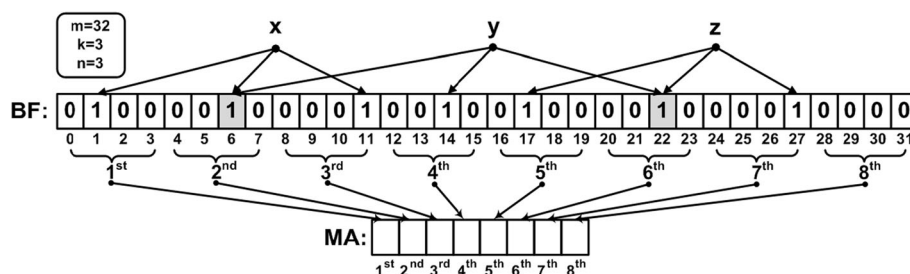
## 2.2 Routing information

Usually, there is only one next hop information in each record of FIB. However, in order to achieve a more efficient intelligent forwarding, each record in the FIB of NDN router will contain multiple interfaces of next hop information [7]. Besides, FIB in the IP router only records the information of next hop corresponded to the IP address prefix. But, in the FIB of NDN router, the record also includes a series of routing and control information [7]. For example, the records in FIB of NDN contain the field of *stale time*.

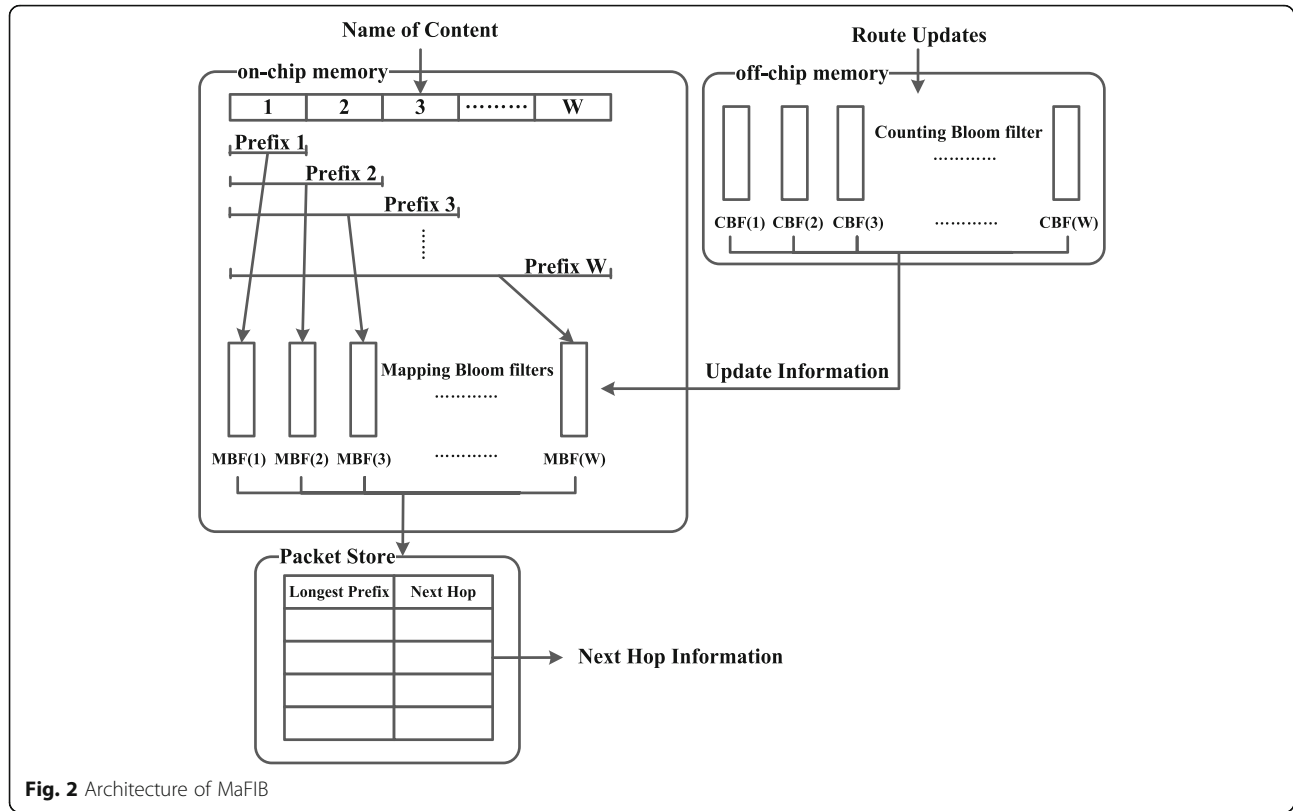
### 3 Mapping Bloom filter

In this section, we illustrate the data structure called MBF which is proposed in [12]. The MBE, which is an improvement of BF, supports querying and mapping the set elements in the memory and lowers on-chip memory (summary) consumption.

The MBF consists of two structures: a regular BF and a Mapping Array (MA), which are bit arrays in the on-chip memory, as illustrated in Fig. 1. With the BF, the elements are verified whether they are in the MBF or not. To access the record in the memory, the value of MA is utilized as the offset address of packet store. To achieve this, the BF is divided into  $j$  equivalent parts, and each part corresponds to one bit of  $j$  bits MA. Meanwhile, the initial state of every bit in the MA is 0 for every incoming element. When an element is



**Fig. 1** Architecture of MBF



**Fig. 2** Architecture of MaFIB

inserted into MBF, some bits of MA are set to 1 according to the  $k$  hash values of this element in the BF. That is, for this incoming element, if any bit position is hashed in the  $i$ th ( $i = 1, 2, \dots, j$ ) part of BF, the value of the  $i$ th bit in MA is 1. Otherwise, the value of the  $i$ th bit in MA is 0. Finally, the value of MA is the offset address of this element in the memory.

The false positive probability [9] of MBF is derived in [12]. Assume that each hash function selects an array position with equal probability. The probability of false positive  $P$  is defined as

$$P = P_1 + P_2, \quad (1)$$

where  $P_1$  is the false positive probability of BF, and  $P_2$  is the false positive probability of MA. According to [12], the false positive probability of MBF can be represented as

$$P \approx \sum_{i=0}^k A_k^i (1-\rho)^{k-i} \cdot (\rho\alpha)^i \quad (2)$$

#### 4 MaFIB

In view of the characteristics of FIB in the NDN router, we propose the MaFIB which is an enhanced FIB architecture with the model MBF. Having the advantages of traditional FIB in NDN, MaFIB can not only meet the

communication needs of NDN completely, but also accelerate the lookup process further. This section will introduce the architecture and proceeding algorithm of MaFIB in details.

##### 4.1 Architecture

MaFIB consists of an on-chip memory and an off-chip memory. Among them, the on-chip memory uses high-speed memory, such as SRAM, and multiple MBFs with the function of summary are deployed on it. The off-chip memory uses low-speed memory, such as DRAM, and multiple Counting Bloom filters (CBF) [9] corresponded to the MBFs in the on-chip memory are deployed on it. Meanwhile a static storage called packet store is implemented too for storing the forwarding information of actual packets, as illustrated in Fig. 2.

In the on-chip memory, each MBF corresponds to the name prefix of one length, so as to quickly verify whether a name prefix is recorded in MaFIB or not. It is available to realize the lookup of forwarding information

**Table 1** Dataset of domain names

Dataset	# of names	# of total components	# of size (MB)
DMOZ	1,000,000	5,071,400	42.593
Alexa	1,000,000	2,185,221	15.833
Total	2,000,000	7,256,621	58.426

**Table 2** Number of domains with different components' number

Dataset	Number of components													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
DMOZ	0	0	55103	776725	150537	16204	1403	24	3	0	1	0	0	0
Alexa	0	0	177	870108	117468	10646	1328	147	86	33	4	3	0	0
Total	0	0	55280	1646833	268005	26850	2731	171	89	33	5	3	0	0
Ratio	0	0	2.76%	82.34%	13.40%	1.34%	0.13%	0.008%	0.004%	0.001%	0.0003%	0.0002%	0	0

in the packet store through using the value of MA in the MBF. With the help of periodical synchronization between the CBF and the MBF, MaFIB can support removing.

Considering the characteristic mentioned in Section II that the content name of NDN are variable and unbounded, 2 million name prefixes from DMOZ [14] and Alexa [15] are analyzed to determine the quantity of MBF used in on-chip memory. And the analysis result is shown in Tables 1 and 2.

It can be found from Table 1 that, for these domain names, the number of components is in concentrated distribution. Among the 2 million names, the number of names with 4 components is 1,646,833 accounting for 82.34%. Followed by the name with 5 components, of which the number is 268,005, accounting for 13.4%. In general, the names with 4 components are in the majority, and the number of the names with other components only accounts for 17.66%.

According to the distribution of domain names in Table 2, it only needs to deploy two MBFs as the summary in the on-chip memory that the requirements of NDN can be achieved. Among them, the first MBF

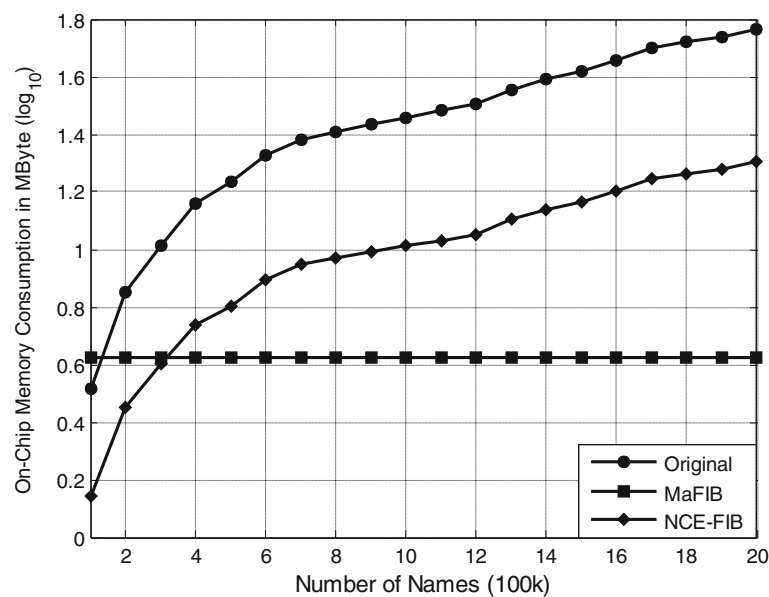
corresponds to the names with the number of component being 4, and the second MBF corresponds to the names with other numbers of component.

According to the design principles of BF in [9] and (2), there are two MBFs with the same parameters in the on-chip memory. In each MBF, the size of BF is  $2^{24}$  and is divided into 30 equal parts. So the size of MA is set to be 30 bits. To improve the processing speed, the longest prefix of content name is encoded by MD5 and SHA1 [10], which generates 288 bits binary number, and every 24 bits are used as the key of a hash function [10]. So the number of hash functions is 12 in the proposed MaFIB.

#### 4.2 Proceeding algorithms

In an NDN router, FIB needs to retrieve and to update the forwarding information of the Interest packet. In this subsection, we present the proceeding algorithms of MaFIB at the reception of lookup and update as follows:

- 1) Lookup. An Interest packet that arrives on MaFIB is firstly checked in the MBF group in parallel with the LPM to determine whether the forwarding

**Fig. 3** Consumption of on-chip memory

information of this Interest packet is in MaFIB. If there is a longest matching prefix in MBF, the offset address of this longest matching prefix is obtained in the packet store according to the value of MA in the corresponding MBF. At last, the forwarding information is read according to the offset address. At this point, if there is no matching record in the packet store, it indicates that the records have been deleted when *timeout* [7] happens. Then, corresponding CBF in the off-chip memory will perform deletion operation. Finally, synchronize the corresponding bits of CBF and MBF.

- 2) Update. Firstly, when conducting update in MaFIB, the content name is initially retrieved in the MBF group according to the LPM, and is accessed to the packet store according to the value of MA in the MBF. Secondly, CBF in the off-chip memory that corresponded to the longest matching prefix is updated. If any bit value of CBF changes, synchronize the bit value of MBF corresponding to the on-chip memory, so as to complete the updating of MaFIB.

#### 4.3 Performance analysis

From the operating process mentioned above, it is obvious that the MaFIB achieves identical probability of false positive with the sum of two MBFs. Meanwhile, the operation on the MaFIB is in a constant time. Compared with the traditional FIB of NDN router, MaFIB do not use the additional hash table to map the record in the

**Table 3** Consumption of on-chip memory

Number of names	Original data (MB)	MaFIB (MB)	NCE-FIB (MB)	Compression ratio of MaFIB
500,000	17.101		6.368	75.48%
1,000,000	28.645	4.194	10.265	85.36%
1,500,000	41.612		14.686	89.92%
2,000,000	58.426		20.322	92.82%

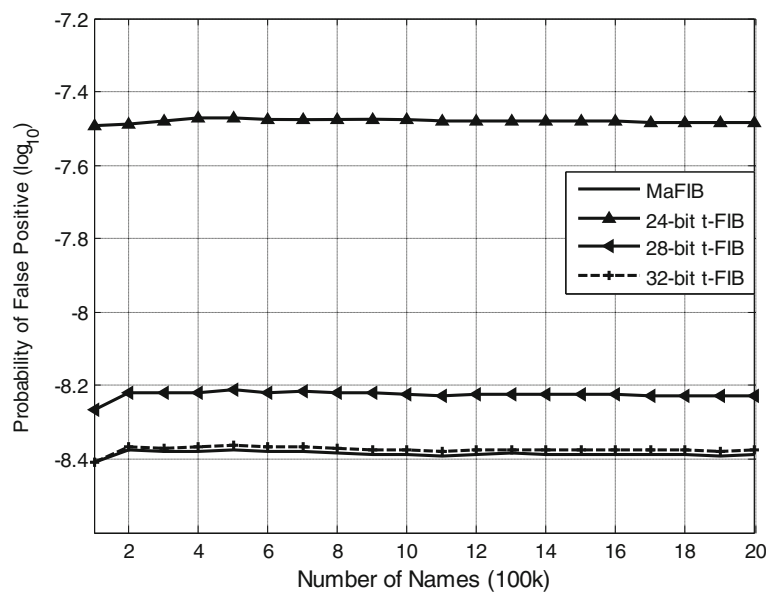
off-chip memory. It means that the work performance of MaFIB is superior to the traditional FIB. In addition, as two MBFs are used as the summary in MaFIB, the on-chip memory required identically equals to  $2 \times (2^{24} + 30) \approx 4.194$  MB. Therefore, the SRAM can be utilized as the on-chip memory.

#### 5 Evaluation and discussion

We evaluate the performance of MaFIB and we compare it with the FIB based on NCE (NCE-FIB) and the traditional FIB of NDN router (t-FIB) in terms of the on-chip memory consumption, probability of false positive, and building time.

##### 5.1 Experimental setup

These mechanisms are implemented in C++ language and are tested on a PC with an Intel Xeon E5-1650 v2 CPU of 3.50 GHz and DDR3 ECC RDIMM of 32 GB. Generally, NCE-FIB and t-FIB consist of two storages: the on-chip memory acting as a summary and the off-chip memory where the forwarding information of FIB is recorded. For the NCE scheme in our experiment,



**Fig. 4** Probability of false positive



ENPT [11, 16] structure is implemented as the FIB table. Meanwhile, Code Allocation Mechanism (CAM) and State Transition Arrays (STA) techniques [11] are utilized to compress memory size. For the t-FIB, two Bloom filters are utilized as the summary in the on-chip memory, and the additional hash table is implemented to map the space in off-chip memory for each element dynamically. The 24 bits, 28 bits, and 32 bits are generated by MD5 and SHA1 as the hash function in the evaluations.

Given the design essentials of FIB [17], 2 million different domain names are used as the dataset of experiments. The entire domain and URL are collected in July, 2013 [14, 15]. Table 1 shows the details of the dataset and Table 2 shows the domains with different components' number obtained from two datasets.

### 5.2 The on-chip memory consumption

Firstly, we assess the summary space consumption on the on-chip memory about the original data, MaFIB, and NCE-FIB in different numbers of names.

The results are shown in Fig. 3. MaFIB requires the lowest on-chip memory consumption which is 4.194 MB. Furthermore, with two MBFs as the summary in the on-chip memory, the bits per element [10] of MaFIB could be minimized to 16.777 for 2 million elements. This number is much lower than 81.288 which is the number of bits per element with NCE-FIB. Therefore, it can be easily deployed on SRAM. Meanwhile, the memory required by NCE-FIB is 20.322 MB, which could only be deployed on RDRAM or DRAM. More detailed results are given in Table 3. This table shows that both MaFIB and NCE-FIB can realize the data compression, while MaFIB has better compression ratio. And the compression ratios of MaFIB for the 1, 1.5, and 2 million names are 85.36, 89.92, and 92.82%, respectively.

### 5.3 Probability of false positive

We evaluate the probability of false positive between the MaFIB and t-FIB in various numbers of names. Figure 4 illustrates that the false positive probability of MaFIB is much lower than that of 24-bit t-FIB. And the 28-bit and 32-bit t-FIB could achieve better false positive probability due to more off-chip memory space involved. Table 4 shows that the false positive probability of the

**Table 4** Probability of false positive

Number of names	MaFIB	24-bit t-FIB	28-bit t-FIB	32-bit t-FIB
500,000	0.21012%	1.68612%	0.30752%	0.21612%
1,000,000	0.40985%	3.34409%	0.59709%	0.42078%
1,500,000	0.61512%	4.94905%	0.89425%	0.63265%
2,000,000	0.80806%	6.48356%	1.17109%	0.83124%

**Table 5** Building time

Number of names	MaFIB (ms)	NCE-FIB (ms)	t-FIB (ms)
1,000,000	6427	10,281	7706
1,500,000	9547	20,638	11,388
2,000,000	12,792	31,357	15,194

MaFIB approximately equal to 0.41, 0.6, and 0.81% as number of names is 1, 1.5, and 2 million, which meets the current network requirements that the packet loss rate should be under 1% [16, 17].

### 5.4 Building time

In this subsection, we investigate the building time of three structures: MaFIB, NCE-FIB, and t-FIB. To get the building time, which is the execution time of insertion, 2 million names are imported, and the execution time is obtained with 0.5 million names in interval. As illustrated in Table 5, since the NCE-FIB is based on the trie structure, it has the worst performance. Besides, t-FIB requires the additional hash function to map the space in off-chip memory for each element dynamically. Therefore, the MaFIB has the least building time of all. From the above, performance of the MaFIB is acceptable.

### 5.5 Discussion

The above results show that our MaFIB based on MBF can extremely reduce the on-chip memory consumption to 4.194 MB for 2 million names with the probability of false positive under 1%. Meanwhile, its outstanding performance in building time also makes MaFIB use SRAM as the on-chip memory and satisfies the current network requirements.

## 6 Conclusions

In this paper, a novel FIB named MaFIB is proposed to meet the NDN's requirement and makes full use of the current faster memory chips. Our current and future work is to extend this design to build an engine running on multiple parallel threads with real packets.

### Abbreviations

BF: Bloom filter; CAM: Code Allocation Mechanism; CBF: Counting Bloom filters; CS: Content store; FIB: Forwarding information base; LPM: Longest prefix matching; MA: Mapping array; MBF: Mapping Bloom filter; NCE: Name Component Encoding; NDN: Named Data Networking; NPT: Name prefix trie; PIT: Pending Interest Table; STA: State transition arrays

### Acknowledgements

This research was supported the by National Natural Science Foundation of China (61601494, 61602346) and Doctoral Program of Tianjin Normal University (52XB1504).

### 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.

## Author details

<sup>1</sup>Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, Tianjin Normal University, 300387 Tianjin, China. <sup>2</sup>College of Electronic and Communication Engineering, Tianjin Normal University, 300387 Tianjin, China. <sup>3</sup>Department of Computer Science, The University of Arizona, Tucson, AZ 85721, USA. <sup>4</sup>School of Electronic Information Engineering, Tianjin University, 300072 Tianjin, China. <sup>5</sup>Shijiazhuang Mechanical Engineering College, 050003 Shijiazhuang, China. <sup>6</sup>Automotive and Transportation, Tianjin University of Technology and Education, 300222 Tianjin, China.

Received: 10 January 2017 Accepted: 7 March 2017

Published online: 21 March 2017

## References

1. X Wang, Q Liang, On the throughput capacity and performance analysis of hybrid wireless networks over fading channels. *IEEE Trans. Wirel. Commun.* **12**(6), 2930–2940 (2013)
2. X Wang, Q Liang, Scaling laws for hybrid wireless networks over fading channels: outage throughput capacity and performance analysis, *Communications (ICC), 2013 IEEE International Conference on.* (IEEE, 2013), pp. 5573–5577
3. X Wang, Q Liang, On the ergodic throughput capacity of hybrid wireless networks over fast fading channels, *Communications (ICC), 2013 IEEE International Conference on.* (IEEE, 2013) pp. 3333–3337
4. Q Liang, X Cheng, S Huang, D Chen, Opportunistic sensing in wireless sensor networks: theory and applications. *IEEE Trans. Comput.* **63**(8), 2002–2010 (2014)
5. LX Zhang et al., *Named Data Networking (ndn) Project*. Technical Report. NDN-0001 (PARC, Palo Alto, 2010)
6. B Ahlgren et al., A survey of information-centric networking. *IEEE Commun. Mag.* **50**(7), 26–36 (2012)
7. C Yi, A Afanasyev, L Wang, BC Zhang, LX Zhang, Adaptive forwarding in named data networking. *ACM SIGCOMM Computer Communication Review* **42**(3), 62–67 (2012)
8. S Dharmapurikar, P Krishnamurthy, DE Taylor, Longest prefix matching using Bloom filters. *IEEE/ACM Trans. Networking* **14**(2), 397–408 (2006)
9. S Tarkoma, CE Rothenberg, E Lagerspetz, Theory and practice of Bloom filters for distributed systems. *IEEE Commun. Surv. Tutorials* **14**(1), 131–155 (2012)
10. A Kirsch, M Mitzenmacher, G Varghese, *Hash-based techniques for high-speed packet processing*. Algorithms for Next Generation Networks (Springer, London, 2010), pp. 181–218
11. Y Wang, K He, H Dai et al., Scalable name lookup in NDN using effective name component encoding, in *Proceedings of the ICDCS*, 2012, pp. 688–697
12. Z Li, K Liu, Y Zhao et al., MaPIT: an enhanced pending interest table for NDN with Mapping Bloom filter. *IEEE Commun. Lett.* **18**(11), 1915–1918 (2014)
13. T Song et al., Scalable name-based packet forwarding: from millions to billions. *Proceedings of the 2nd International Conference on Information-Centric Networking (ACM, 2015)*, pp. 19–28
14. ODP-open directory project, [Online]. Available: <http://www.dmoz.org/>
15. Alexa the web information company, [Online]. Available: <http://www.alexa.org/>
16. HW Yuan, T Song, P Crowley, Scalable NDN forwarding: concepts, issues and principles, in *Proceedings of ICCCN*, 2012, pp. 1–9
17. D Perino, M Varvello, A reality check for content centric networking, in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, 2011, pp. 44–49

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)