SPECIAL SECTION ON SMART CACHING, COMMUNICATIONS, COMPUTING AND
CYBERSECURITY FOR INFORMATION-CENTRIC INTERNET OF THINGS

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 26, 2018, accepted March 22, 2018, date of publication April 23, 2018, date of current version May 24, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2829141

Virtualization for Distributed Ledger Technology (vDLT)

**FEI RICHARD YU[1], (Fellow, IEEE), JIANMIN LIU[1], YING HE[1], (Student Member, IEEE),
PENGBO SI[2], (Senior Member, IEEE), AND YANHUA ZHANG[2]**
[1]Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada
[2]Beijing Advanced Innovation Center for Future Internet Technology, Beijing University of Technology, Beijing 100022, China

Corresponding author: Richard Yu (richard.yu@carleton.ca)

ABSTRACT Recently, with the tremendous development of crypto-currencies, distributed ledger technology (DLT) (e.g., blockchain) has attracted significant attention. The traditional Internet was originally design to handle the exchange of information. With DLT, we will have the Internet of value. Although, DLT has a great potential to create new foundations for our economic and social systems, the existing DLT has a number of drawbacks (e.g., scalability) that prevent it from being used as a generic platform for distributed ledger across the globe. In this paper, we present a novel virtualization approach to address the challenges in the existing DLT systems. Specifically, in the proposed virtualization for DLT (vDLT), the underlying resources (e.g., hardware, compute, storage, network, and so on) are abstracted. By providing a logical view of resources, vDLT can significantly improve the performance, facilitate system evolution, and simplify DLT management and configuration. Several use cases of vDLT are presented to illustrate the effectiveness of the proposed vDLT.

INDEX TERMS Distributed ledger technology (DLT), blockchain, directed acyclic graph (DAG), virtualization.

I. INTRODUCTION

Since ancient times, ledgers have been at the heart of economic activities - to record assets, payments, contracts or buy-sell deals. They have moved from being recorded on clay tablets to papyrus, vellum and paper. Although the invention of computers and the Internet provides the process of record keeping with great convenience, the basic principle has not been changed - ledgers are usually *centralized*. Recently, with the tremendous development of crypto-currencies (e.g., Bitcoin [1]), the underlying *distributed ledger technology* (DLT) has attracted significant attention [2].

A distributed ledger is essentially a consensus of replicated, shared and synchronized data geographically spread across a network of multiple nodes. There is no central administrator or centralized data storage. Using a consensus algorithm, any changes to the ledger are reflected in the copies. The security and accuracy of the the ledger are maintained cryptographically according to rules agreed by the network. One form of distributed ledger design is the *blockchain*, which is at the heart of Bitcoin. Blockchain is a continuously growing list of records, called blocks, linked and secured using cryptography. Nevertheless, not all distributed ledgers have to necessarily employ a chain of blocks to successfully provide secure and valid achievement of distributed consensus: a blockchain is only one type of data structure considered to be a distributed ledger. Besides blockchain, there are other data structures to implement DLT, such as *directed acyclic graph (DAG)*, which is at the heart of IOTA [3]. The transactions issued by nodes constitute the site set of the DAG, which is the distributed ledger for storing transactions.

DLT has great potential to create new foundations for our economic and social systems by efficiently establishing trust among people and machines, reducing cost, and increasing utilization of resources. DLT is similar to another fundational technology, distributed computer networking technology TCP/IP (transmission control protocol/Internet protocol), which laid the groundwork for the development of the Internet [4]. DLT is making waves in industries such as finance; music and entertainment; diamond and precious assets; artwork; supply chains of various commodities; and more. While DLT has multiple advantages, it is in a nascent stage and still being explored to adopt in the best possible ways.

The existing DLT has a number of drawbacks that prevent it from being used as a generic platform for distributed ledger across the globe. One notable drawback is the scalability issue. Most of existing public blockchain consensus protocols (e.g, Bitcoin [1], Ethereum [5] and Ripple [6]) require that every fully participating node in the network needs to process every transaction, which results in low transaction throughput and high network traffic [7]. Although some possible approaches (e.g., SegWit [8], Lightning Network [9], Raiden Network [10], Plasma [11], Cardano [12], etc.) have been proposed to address this issue, there is no systematic solution to this issue, and the approach proposed for one blockchian may not be suitable for another blockchain. In addition, this is related to another important issue of existing DLT, ossification, meaning that it is difficult to make changes after a DLT is deployed. Furthermore, most existing DLT platforms are specialized platforms, each with a specialized application (e.g., Bitcion for crypto-currency and IOTA for the Internet of Things (IoT)), which makes interoperability among these platforms a difficult task.

It is not surprising that the current DLT has these challenges. From the history of information technology (IT), we have seen similar challenges faced by other technologies in their early stage. For example, in the 1960s, most computers were specialized for specific applications, such as word processing, CAD, weather data acquisition [13], etc., and they were not easily inter-workable and upgradeable [14]. Another example occurred in the networking field. In the 1990s, specialized networks were used for different applications, e.g., public switched telephone networks (PSTNs) for voice, IP networks for data, and cable networks for TV.

Virtualization has been playing a central role in addressing these challenges in the IT landscape [15]. Essentially, virtualization refers to technologies designed to provide abstraction of underlying resources (e.g., hardware, compute, storage, network, etc.). By providing a logical view of resources, rather than a physical view, virtualization can significantly improve the performance, facilitate system evolution, and simplify system management and configuration, and reduce cost. Indeed, virtualization is one of the major enabling technologies behind recent advances of IT, including cloud computing [16], edge computing [17], and network function virtulization (NFV) [18].

In this paper, we present a novel virtualization approach to address the challenges in the existing DLT systems. The contributions of this work are summarized as follows.

- We survey the virtualization approaches in the IT landscape, including computing, storage, OS, software, and networks. Then, we show that virtualization will be naturally the next step to address the challenges in the existing DLT systems.
- We review the layered model for the existing DLT systems, and present the architecture of virtualization for DLT (vDLT). The application programming interface (API) of vDLT is described.

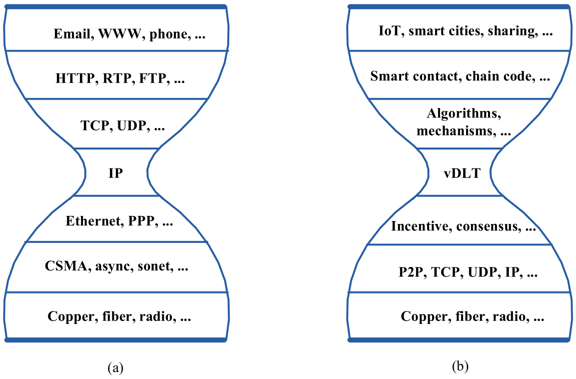


FIGURE 1. **The architectures of (a) Internet of information and (b) Internet of value.**

- Several use cases of vDLT are presented to illustrate the effectiveness of the proposed approach.

The rest of this paper is organized as follows. The related work is presented in Section II. Section III describes different layers in DLT, and presents the architecture of the proposed vDLT and the API. Several use cases of vDLT are presented in Section IV. Finally, we conclude this work in Section V.

II. RELATED WORK

In this section, we first present the transition from the Internet of Information to the Internet of value. Then, we briefly review the the journey of virtualization for computing and storage, followed by a description of recent advances in network function virtualization.

A. FROM THE INTERNET OF INFORMATION TO THE INTERNET OF VALUE

The traditional Internet was originally design to to handle the exchange of information, e.g., using emails and websites. It was never designed to handle the exchange of actual value. Anyone transferring money online is not actually moving the value directly. Instead, she/he is transferring instructions to an intermediary - whether through a band, a credit card company, Western Union or PayPal - to pass on the value. The involvement of such third parties in the exchange of value naturally comes at a significant cost. By eliminating intermediary and enabling direct trust, DLT makes the breakthrough that the Internet has been looking for and never had. The Internet of value enabled by DLT is the second era of the Internet. For the last 40 years we have had the Internet of information; now, with DLT, we will get the Internet of value [19].

In order to successful implement the Internet of value, we can take a look at the successful architecture of the Internet of information. Figure 1-(a) shows the hourglass architecture centering on the universal network layer (i.e., IP), which implements the basic functionality necessary for global interconnectivity. By allowing both lower and upper layer technologies to innovate independently, this ‘‘thin waist’’ principle has successfully enabled the explosive growth of the Internet of information [20].

Similarly, we envision a "thin-waist" architecture of the Internet of value, which is shown in Figure 1-(b). The thin waist in this architecture is called virtualization for DLT (vDLT) (or virtualized DLT) that abstracts the functionality necessary for distributed ledgering. More details about vDLT will be described in Section III.

B. VIRTUALIZATION FOR COMPUTING AND STORAGE

Virtualization has been revolutionizing the ways in which IT is developed, and traces its root all the way back to the 1960s. Hypervisor software, a term coined in 1966, was the original conceptualization of what would become virtualization. A hypervisor is software that creates and runs virtual machines. The word is portmanteau of the prefix "hyper," meaning "above," and "supervisor," the primitive operating systems in use at the time. IBM Cambridge Scientific Centers CP-40 was the first OS to use complete virtualization. It supported 14 machines simultaneously. The main advantages of using virtual machines vs a time-sharing operating system (OS) was more efficient use of the system since virtual machines were able to share the overall resources of the mainframe, instead of having the resources split equally between all users.

Unix developed during the 1970s is an example of virtualization at the OS Level. Since Unix was written in C, only small parts of the OS had to be customized for a given hardware platform, and the rest of the OS could easily be re-compiled for each hardware platform with little or no changes.

The development of desktop computing and ×86 architecture changed the direction of computer development, requiring the development of new virtualization options. In 1987, Locus Computing Corporation released Merge, a rudimentary virtualiation program, which ran MS-DOS on a SCO-UNIX environment. In 1997, Connectix released Virtual PC, offering virtualization to Mac users. In 1998, a company called VMWare was established, and began selling a product similar to Virtual PC called VMWare workstation.

Through the use of Unix and C compilers, a user could run just about any program on any platform, but it still required users to compile all the software on the platform they wished to run on. For true portability of software, you needed some sort of software virtualization. Java was developed by Sun Microsystems in the 1990s. Java allowed you to write an application once, then run the application on any computer with the Java Run-time Environment (JRE) installed.

Virtualization is also one of the major enabling technologies behind recent advances cloud computing [16] and edge computing [17]. Cloud computing has become very popular to enable access to a shared pool of computing and storage resources [21]. As the distance between the cloud and the end device is usually large, cloud computing services may not provide guarantees to low latency applications. To address these issues, *Edge computing* [22], [23] has been studied to deploy computing resources closer to end users, which can efficiently improve the quality of service (QoS) for applications that require intensive computations and low latency [24], [25].

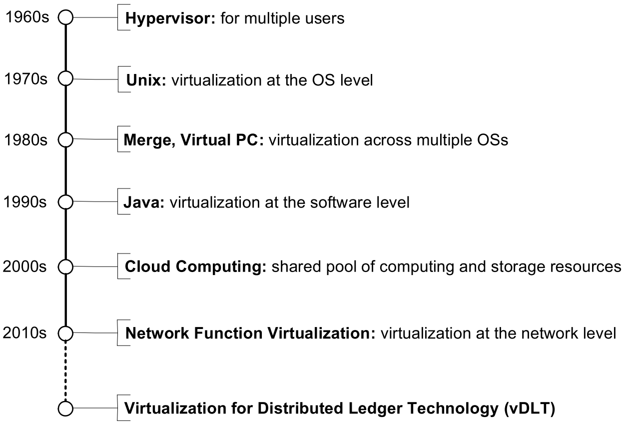


FIGURE 2. **A brief journey of virtualization.**

Figure 2 shows a brief journey of virtualization. We can see that virtualization has been playing an important role to abstract the underlying resources, so that people can focus on the things they care the most. Now, we can have access to hardware, OS, software, storage, and network via virtualization (network virtualization will be described in the next subsection). Therefore, we believe that virtualization will be naturally the next step for DLT.

C. VIRTUALIZATION FOR NETWORKS

With the tremendous growth in the Internet traffic and services, it is natural to extend the success of virtualization from computing and storage to networks. Recently, network virtualization has been actively used in Internet research testbeds, such as G-Lab [26] and 4WARD [27]. It aims to overcome the resistance of the current Internet to fundamental architecture changes. Network virtualization has been considered as one of the most promising technologies for the future Internet [28]. Particularly, the NFV concept was presented by a group of network service providers in 2012. These service providers wanted to simplify and speed up the process of adding new network functions or applications. The European Telecommunications Standards Institute (ETSI) Industry Specification Group for Network Functions Virtualization proceeded to spearhead NFV development and standards [18].

In traditional networks, network services are run on proprietary, dedicated hardware. With NFV, functions like routing, load balancing and firewalls are packaged as virtual machines (VMs) on commodity hardware. Individual virtual network functions (VNFs), are an essential component of NFV architecture. Because NFV architecture virtualizes network functions and eliminates specific hardware, network managers can add, move or change network functions at the server level in a simplified provisioning process. Figure 3 show the comparison between the traditional network appliance approach and the NFV approach.

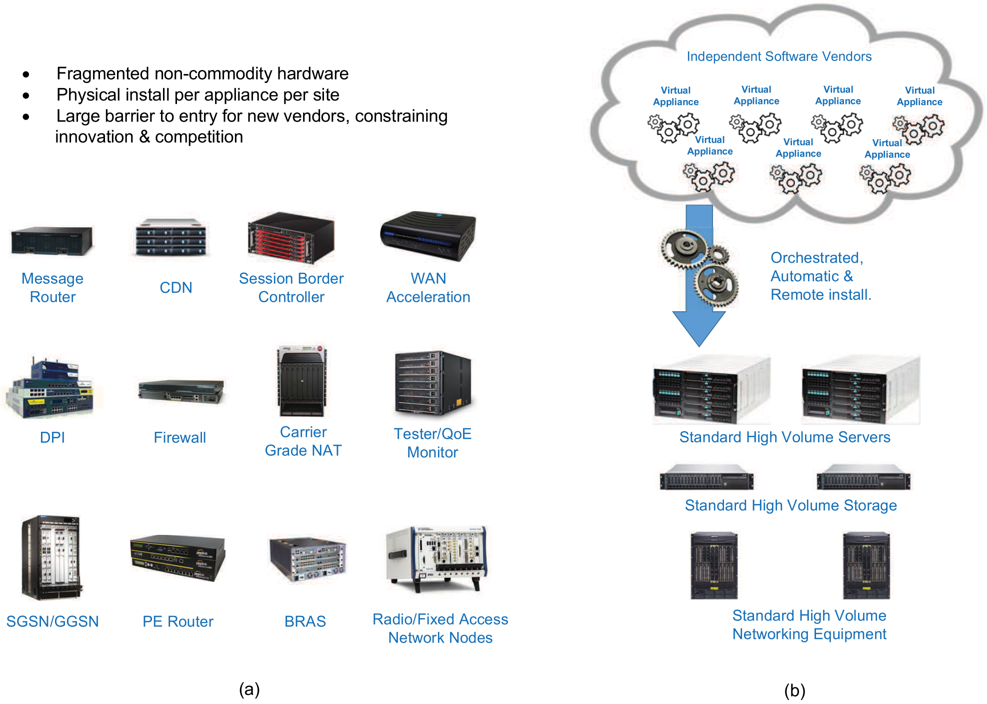


FIGURE 3. Comparison between (a) classical network appliance approach and (b) network function virtualization (NFV) approach.

III. VIRTUALIZATION FOR DISTRIBUTED LEDGER TECHNOLOGY (vDLT)

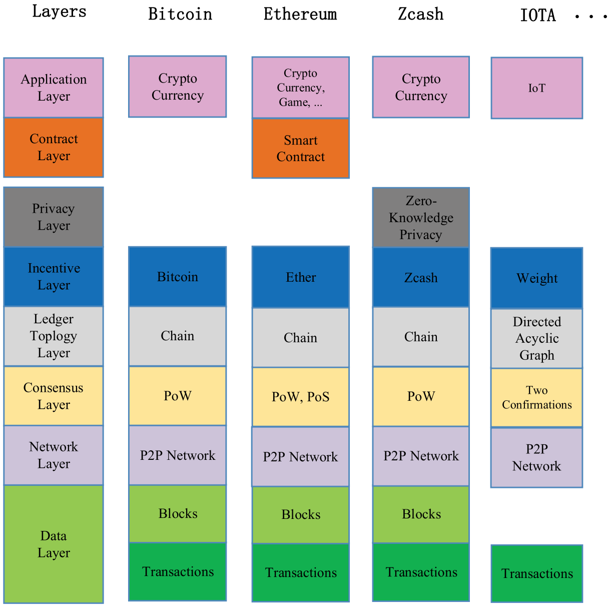
In this section, we first describe different layers in DLT. Then, we present the architecture of vDLT, followed by the application programming interface (API) of vDLT.

A. DIFFERENT LAYERS IN DISTRIBUTED LEDGER TECHNOLOGY (DLT)

The existing DLT systems can be divided into several layers, including data layer, network layer, consensus layer, ledger topology layer, incentive layer, privacy layer, contract layer, and application layer, as shown in Fig. 4.

The data layer in the DLT architecture encapsulates the data generated from different applications. In the blockchain form of DLT, each block contains a number of transactions, and is "chained" back to the previous block, resulting an ordered list of blocks [29]. There are mainly two parts in each block: the block header and the block body. The block header specifies the metadata, including hash of previous block, hash of current block, timestamp, Nonce and Merkle root. The block body stores the verified transactions [30]. In the DAG form of DLT, instead of blocks, data are added directly to a graph of transactions, which reference previous transactions.

The network layer defines the networking mechanism used in DLT. The goal of this layer is to distribute, forward and verify data generated from the data layer. The network can be generally modeled as a P2P network, where peers are participants. Once data is generated, it will be sent to the peer nodes using the networking mechanism.



Layers	Bitcoin	Ethereum	Zcash	IOTA · · ·
Application Layer	Crypto Currency	Crypto Currency, Game, ...	Crypto Currency	IoT
Contract Layer		Smart Contract		
Privacy Layer			Zero-Knowledge Privacy	
Incentive Layer	Bitcoin	Ether	Zcash	Weight
Ledger Topology Layer	Chain	Chain	Chain	Directed Acyclic Graph
Consensus Layer	PoW	PoW, PoS	PoW	Two Confirmations
Network Layer	P2P Network	P2P Network	P2P Network	P2P Network
Data Layer	Blocks Transactions	Blocks Transactions	Blocks Transactions	Transactions

FIGURE 4. Different layers in some existing distributed ledger technology (DLT) systems.

The consensus layer determines the consensus algorithm. In decentralized environments, how to reach consensus efficiently among the untrustworthy nodes is an important issue [31]. In the existing DLT systems, there are four major consensus mechanisms: Proof of Work (PoW) [32], Proof of Stake (PoS) [33], Practical Byzantine Fault

Tolerance (PBFT) [34], and Delegated Proof of Stake (DPoS) [35]. In PoW, nodes repeatedly run hashing functions to find a nonce value, which is difficult to produce but easy for other nodes to validate. Compared to PoW, PoS is an energy-saving mechanism. The creator of the next block in PoS is chosen via various combinations of random selection and wealth or age (i.e., the stake). PBFT is a replication algorithm to tolerate Byzantine faults. DPoS is similar to PoS. The major difference between PoS and DPoS is that PoS is direct democratic while DPoS is representative democratic. In addition, there are other consensus mechanisms, including Ripple [36], Tendermint [37], Proof of Bandwidth (PoB) [38], Proof of Elapsed Time (PoET) [39], Proof of Authority (PoA) [40], and Proof of Retrievability [41].

The ledger topology layer defines the ledger topology for storing the data. There are mainly two topologies in the existing DLT systems, blockchain and DAG. Blockchain is a continuously growing list of blocks, linked and secured using cryptography. By contrast, the DAG builds a graph of transactions. Since transactions do not need to wait to be included in blocks, transactions can be quickly confirmed in DAG when they are received by a node.

The incentive layer in DLT integrates economic incentives to motivate the nodes to contribute their efforts to DLT systems. Specifically, once new data is added, some economic incentives (e.g., digital currencies) will be issued as reward to corresponding nodes according to their contributions.

The privacy layer provides privacy guarantee in DLT systems. In most public blockchain systems, transactions are publicly visible by all network participants, and information such as the sender, recipient, amount, etc. is are open to scrutiny, or at least the addresses are publicly traceable. For example, in order to transact with someone on a blockchain, one needs to know at least one of their addresses. So if one sends them some money, he/she can see where that money goes next. Second, if one happens to know something about a participant from the real world (e.g., what types of assets they trade at what time of day), one can search the chains activity for corresponding patterns, and then infer their address with a high level of confidence. To provide privacy, we can encrypt the the information stored in a blockchain, while still gaining the benefits of data provenance, timestamping and immutability. However, encryption of this nature cannot be used by transactions that represent transfers of tokenized assets. If Alice and Bob were to encrypt their transaction, then the assets in question could not be used safely by any other participant in the chain, because nobody else would know where the assets actually are. The assets would cease to have any collective meaning on the chain, which destroys the entire point. Zero knowledge proofs aim to prove the statement ‘‘this transfer of assets is valid’’, without revealing anything important about the transfer itself. Zcash uses a relatively new technique for zero knowledge proofs called Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARKs) to provide strong privacy guarantee [42].

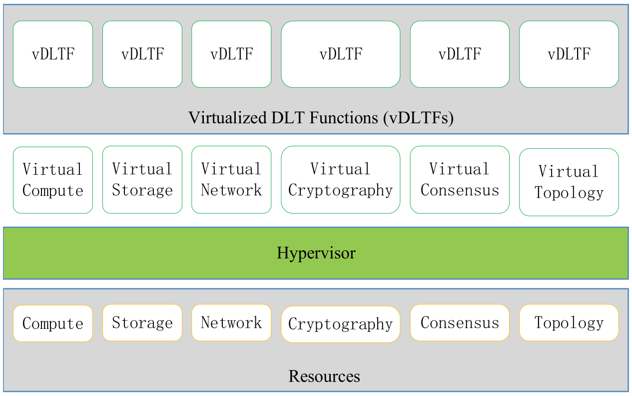


FIGURE 5. **The architecture of virturalization for distributed ledger technology (DLT).**

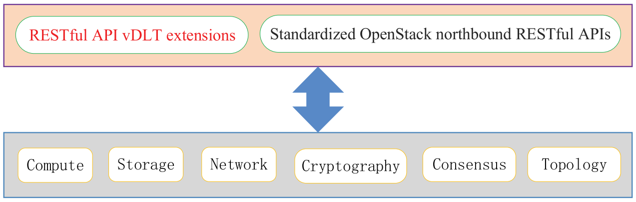


FIGURE 6. **RESTful API vDLT extensions.**

The contract layer brings programmable characteristic into DLT. Various scripts, codes and smart contracts can be used to enable more complex programmable transactions. For example, smart contracts are a group of state-response rules that are securely stored in Ethereum. Smart contracts can control users’ digital assets, express business logic, and formulate the participants’ rights and obligations. When all terms within a smart contract are agreed by two or more participants, the contract will be signed cryptographically and broadcast to the blockchain network for verification [43]. Once the predefined conditions are triggered, the smart contract will execute independently and automatically according to the prescribed rules. Ethereum [44] is the most widely used open-source blockchain platform that supports smart contracts. Ethereum provides decentralized virtual machines to automatically handle various services, applications or contracts created by people [45].

The highest layer in DLT is the applications, including crypto-currency, IoT, games, smart cities, etc. [46]. Although DLT is still in its infancy, academia and industry are trying to apply this promising technology into many areas.

B. THE ARCHITECTURE OF VIRTUALIZATION FOR DISTRIBUTED LEDGER TECHNOLOGY (vDLT)

Fig. 5 shows the architecture of vDLT. The resources consist of networking, compute, and, storage, cryptography,

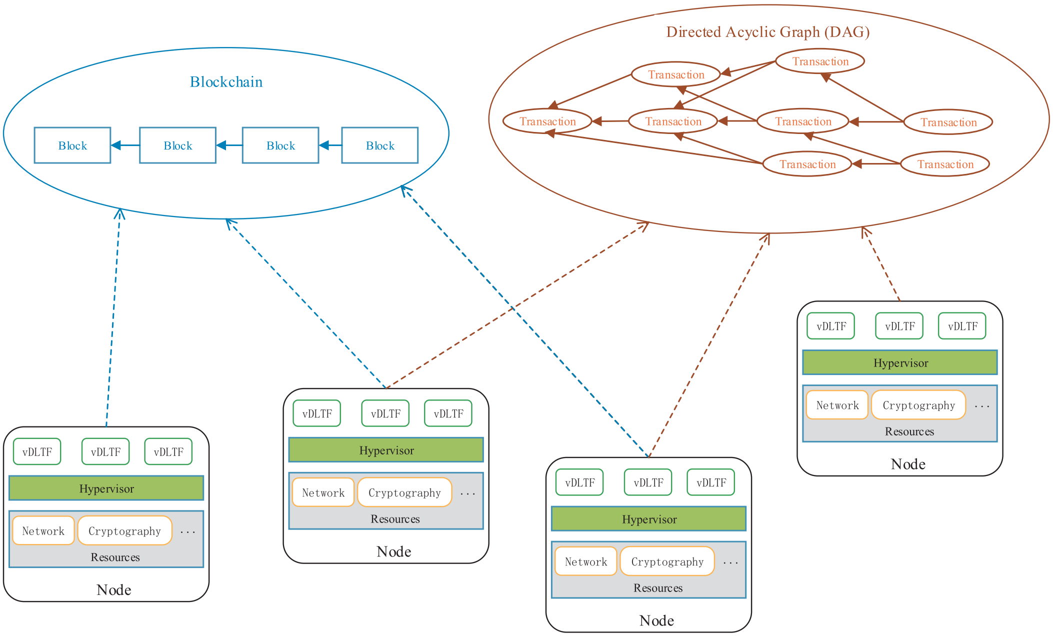


FIGURE 7. Both blockchain and DAG can be configured in vDLT.

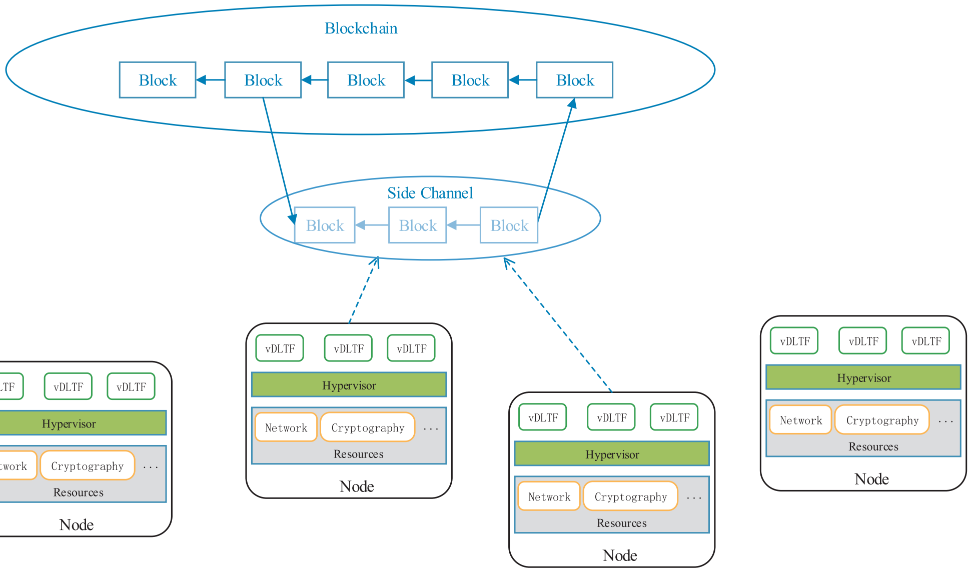


FIGURE 8. Side channel mechanisms (e.g., lighting network) can be viewed as the implementation of a virtual blockchain.

consensus and topology resources. Virtual resources are abstractions of the physical resources. The abstraction is achieved using a virtualization layer (based on a hypervisor), which decouples the virtual resources from the underlying physical resources. The computing and storage resources may be represented in terms of one or more VirtualMachines (VMs), while virtual networks are made up of virtual links and nodes. A virtual node is a software component with either hosting or routing functionality, for example an operating system encapsulated in a VM. A virtual link is a logical interconnection of two virtual nodes, appearing to them as a direct physical link with dynamically changing properties.

A virtualized DLT function (vDLTF) is a functional block within a network infrastructure that has well defined external interfaces and well-defined functional behaviour. Therefore, a vDLTF is an implementation of an NF that is deployed on virtual resources such as a VM or container. A single vDLTF may be composed of multiple internal components, and hence it could be deployed over multiple VMs or containers.

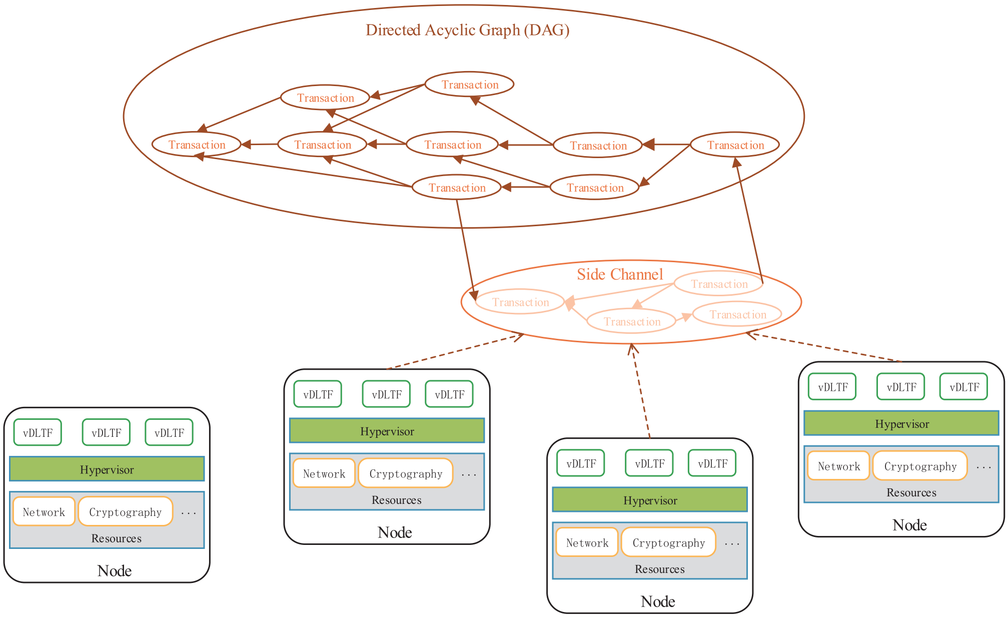


FIGURE 9. Side channel mechanisms in DAG can be viewed as the implementation of a virtual DAG.

A service is an offering provided by a TSP that is composed of one or more NFs.

C. APPLICATION PROGRAMMING INTERFACE (API) OF vDLT

Open interfaces and clearly defined reference points are key to accelerate the deployment of vDLT. This approach allows multiple parties to independently develop the building blocks of the vDLT architecture and enables different application to pick the solutions and implementations best suited to their application needs. At the same time, the use of established standards, protocols and interface technologies, e.g. from OpenStack, is highly desirable to shorten time to develop the vDLT framework on a broad basis. For the software implementation, an open framework based on open-source software is the most flexible approach and avoids a lock-in with any particular component. In addition, to ensure carrier grade service and support for provisioning of vDLT services, some extensions to the set of standard APIs are needed, which is shown in Fig. 6.

IV. USE CASES OF vDLT

In this section, we describe several use cases of vDLT, including DAG-based vDLT and blockchain-based vDLT, side channel mechanisms in vDLT, and the separation of control from traffic in vDLT.

A. DAG-BASED vDLT AND BLOCKCHAIN-BASED vDLT

Both blockchain and DAG can be configured easily according to the different requirements of different applications, as shown in Fig. 7. For example, for applications where low latency and low transaction fee are desirable, a DAG-based vDLT can be configured. For applications where latency and transaction fee are not important concerns, a blockchain-based vDLT can be configured, thanks to the flexibility of vDLT.

B. SIDE CHANNEL MECHANISMS IN vDLT

With side channel mechanisms (e.g., Lightning Network [9], Raiden Network [10], Plasma [11], etc.), interactions that could and would normally occur on the distributed ledger get conducted off the distributed ledger. Side channel mechanisms provide significant improvements in the throughput of DLT. A side channel mechanism works as follows. (1). Part of the distributed ledger state is locked. (2). Participants make updates amongst themselves by constructing and cryptographically signing transactions without submitting it to the distributed ledger. (3). At some later point, participants submit the state back to the distributed ledger, which closes the side channel and unlocks the state again. Steps (1) and (3) involve distributed ledger operations that are broadcast to the network, pay fees and wait for confirmations. By contrast, Step (2) does not involve the distributed ledger at all, which can help lifts the burden from the underlying distributed ledger.

A side channel mechanism can be viewed as the implementation of a virtual distributed ledger, e.g., a side channel blockchain from a blockchain (Fig. 8), a side channel DAG from a DAG (Fig. 9), or even a side channel blockchain from a DAG (Fig. 10). With the flexibility of vDLT, side channel mechanisms can be easily implemented.

C. SEPARATION OF CONTROL FROM TRAFFIC IN vDLT

The issue of ossification in the existing DLT has attracted great attentions. Specifically, it is difficult to make changes after a DLT is deployed. Actually, this is related to

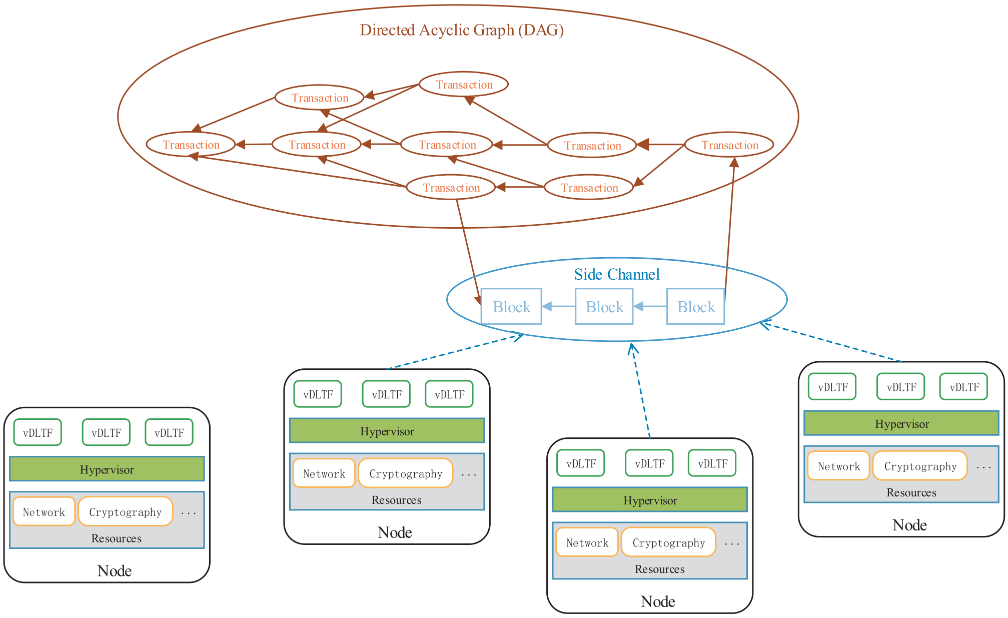


FIGURE 10. **A side channel blockchain from a DAG using vDLT.**

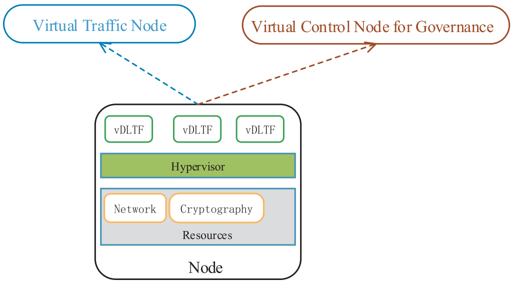


FIGURE 11. **With virtualization, a node can be virtualized to a traffic node or a control node. The control nodes can be used for the governance of vDLT.**

another fundamental issue of existing DLT: governance. The Ethereum and Ethereum Classic split is a classic example of bad governance. The governance issue can be addressed by vDLT. Particularly, a node can be virtualized to a traffic node or a control node, as shown in Fig. 11. The control nodes can be used for the governance of vDLT.

V. CONCLUSION AND FUTURE WORK

The underlying distributed ledger technology of crypto-currencies has great potential to create new foundations for our economic and social systems. However, the existing DLT has a number of drawbacks, including scalability, ossification and specialization, etc. In this paper, we presented a novel virtualization approach to address the challenges of the existing DLT systems. We first surveyed the virtualization approaches in the IT landscape, then showed that virtualization will be naturally the next step in DLT. We described the architecture of virtualization for DLT, where the underlying physical resources are abstracted as virtual resources. By providing a logical view of resources, vDLT can significantly facilitate DLT evolution and simplify system management and configuration. We presented several use cases of vDLT to illustrate the effectiveness of the proposed vDLT. Future work is in progress to implement the proposed vDLT in different applications, including supply chain, smart cities, etc.

REFERENCES

[1] S. Nakamoto. (Oct. 2018). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[2] *Distributed Ledger Technology: Beyond Block Chain*, UK Government, Office Sci., London, U.K., 2016.
[3] IOTA. Accessed: Apr. 26, 2018. [Online]. Available: https://iota.org
[4] M. Iansiti and K. R. Lakhani, "The truth about blockchain," *Harvard Bus. Rev.*, vol. 95, pp. 118–127, Jan. 2017.
[5] Ethereum. Accessed: Apr. 26, 2018. [Online]. Available: https://www.ethereum.org/
[6] Ripple. Accessed: Apr. 26, 2018. [Online]. Available: https://ripple.com/
[7] J. Pearson. Bitcoin Unlimited' Hopes to Save Bitcoin from Itself. Motherboard. Accessed: Apr. 26, 2018. [Online]. Available: https://motherboard.vice.com/enus/article/wnx7vz/bitcoinunlimited-hopes-to-save-bitcoin-from-itself-block-size
[8] SegWit. Accessed: Apr. 26, 2018. [Online]. Available: https://segwit.org/
[9] *Lighting Network*. Accessed: Apr. 26, 2018. [Online]. Available: https://lightning.network/
[10] *Raiden Network*. Accessed: Apr. 26, 2018. [Online]. Available: https://raiden.network/
[11] *Plasma*. Accessed: Apr. 26, 2018. [Online]. Available: http://plasma.io/
[12] *Cardano*. Accessed: Apr. 26, 2018. [Online]. Available: https://cardano.org
[13] J. A. Cunningham, P. Meissner, and C. A. Kettering, "A computer for weather data acquisition," in *Proc. Int. Workshop Manag. Requirements Knowl.*, New York, NY, USA, Dec. 1960, pp. 57–66.
[14] K. Flamm, *Creating the Computer: Government, Industry, and High Technology*. Washington, DC, USA: Brookings Institution, 1988.
[15] S. T. King and S. W. Smith, "Virtualization and security: Back to the future," *IEEE Secur. Privacy*, vol. 6, no. 5, p. 15, Sep. 2008.
[16] V. Salapura, "Cloud computing: Virtualization and resiliency for data center computing," in *Proc. IEEE 30th Int. Conf. Comput. Design (ICCD)*, Sep. 2012, pp. 1–2.
[17] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, "Consolidate IoT edge computing with lightweight virtualization," *IEEE Netw.*, vol. 32, no. 1, pp. 102–111, Jan. 2018.

[18] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, Feb. 2015.

[19] K. Maclver. *From the Internet of Information to the Internet of Value*. [Online]. Available: https://www.i-cio.com/big-thinkers/don-tapscott/item/from-theinternet-of-information-to-the-internet-of-value

[20] L. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014.

[21] H. Zhang, Q. Zhang, and X. Du, "Toward vehicle-assisted cloud computing for smartphones," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5610–5618, Dec. 2015.

[22] M. Patel *et al.*, "Mobile-edge computing introductory technical white paper," ETSI, Sophia Antipolis, France, ETSI White Paper, Sep. 2014.

[23] C. Liang, Y. He, F. R. Yu, and N. Zhao, "Energy-efficient resource allocation in software-defined mobile networks with mobile edge computing and caching," in *Proc. IEEE INFOCOM Workshops*, May 2017, pp. 121–126.

[24] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[25] N. Kumar, S. Zeadally, and J. J. P. C. Rodrigues, "Vehicular delay-tolerant networks for smart grid data management using mobile edge computing," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 60–66, Oct. 2016.

[26] G-Lab. Accessed: Apr. 26, 2018. [Online]. Available: http://www.german-lab.de/

[27] M. Achemlal *et al.*, "D-3.2.0 virtualisation approach: Concept," The FP7 4WARD Project, Tech. Rep., 2009.

[28] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 358–380, 1st Quart., 2015.

[29] X. Xu *et al.*, "A taxonomy of blockchain-based systems for architecture design," in *Proc. IEEE ICSA*, Gothenburg, Sweden, Apr. 2017, pp. 243–252.

[30] G. Cui, K. Shi, Y. Qin, L. Liu, B. Qi, and B. Li, "Application of block chain in multi-level demand response reliable mechanism," in *Proc. IEEE ICIM*, Chengdu, China, Apr. 2017, pp. 337–341.

[31] T. V. Lakshman and A. K. Agrawala, "Efficient decentralized consensus protocols," *IEEE Trans. Softw. Eng.*, vol. SE-12, no. 5, pp. 600–607, May 1986.

[32] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Tech. Rep., 2008.

[33] D. Larimer. (Nov. 2013). *Transactions as Proof-of-Stake*. [Online]. Available: https://bravenewcoin.com/assets/Uploads/TransactionsAsProofOfStake10.pdf

[34] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. OSDI*, vol. 99. 1999, pp. 173–186.

[35] D. Larimer, "Delegated proof-of-stake," White Paper, 2014.

[36] D. Schwartz, N. Youngs, and A. Britto, "The Ripple protocol consensus algorithm," Ripple Labs, San Francisco, CA, USA, White Paper 5, 2014.

[37] J. Kwon, "Tendermint: Consensus without mining," Tech. Rep., 2014.

[38] M. Ghosh, M. Richardson, B. Ford, and R. Jansen, "A TorPath to TorCoin: Proof-of-bandwidth altcoins for compensating relays," Naval Res. Lab, Washington, DC, USA, Tech. Rep., 2014.

[39] (Dec. 2017). *Proof of Elapsed Time (PoET)*. [Online]. Available: http://consensus.readthedocs.io/en/latest/algos/proof-ofelapsed-time.html

[40] (Dec. 2017). *Proof-of-Authority Chains*. [Online]. Available: https://github.com/paritytech/parity/wiki/Proof-of-Authority-Chains

[41] A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz, "Permacoin: Repurposing bitcoin work for data preservation," in *Proc. IEEE SP*, San Jose, CA, USA, May 2014, pp. 475–490.

[42] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Scalable, transparent, and post-quantum secure computational integrity," Cryptol. ePrint Arch., Tech. Rep. 2018/046, 2018. [Online]. Available: https://eprint.iacr.org/2018/046

[43] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proc. IEEE SP*, San Jose, CA, USA, May 2016, pp. 839–858.

[44] (Dec. 2017). *Ethereum*. [Online]. Available: https://www.ethereum.org/

[45] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. Kishigami, "Blockchain contract: Securing a blockchain applied to smart contracts," in *Proc. IEEE ICCE*, Las Vegas, NV, USA, Jan. 2016, pp. 467–468.

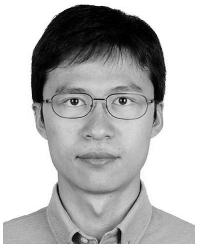
[46] G. Hurlburt, "Might the blockchain outlive bitcoin?" *IT Professional*, vol. 18, no. 2, pp. 12–16, Mar. 2016.

FEI RICHARD YU (S'00–M'04–SM'08–F'18) received the Ph.D. degree in electrical engineering from The University of British Columbia (UBC) in 2003. From 2002 to 2006, he was with Ericsson, Lund, Sweden and a start-up in California, USA. He joined Carleton University in 2007, where he is currently a Professor. His research interests include distributed ledger technology, wireless cyber-physical systems, connected/autonomous vehicles, security, and deep learning. He received the IEEE Outstanding Service Award in 2016, the IEEE Outstanding Leadership Award in 2013, the Carleton Research Achievement Award in 2012, the Ontario Early Researcher Award (formerly Premiers Research Excellence Award) in 2011, the Excellent Contribution Award at the IEEE/IFIP TrustCom 2010, the Leadership Opportunity Fund Award from the Canada Foundation of Innovation in 2009, and the best paper awards at the IEEE GLOBECOM 2012, ICC 2014, VTC 2017 Spring, the IEEE/IFIP TrustCom 2009, and International Conference on Networking in 2005.

Dr. Yu is a registered Professional Engineer in the province of Ontario, Canada, and a fellow of the Institution of Engineering and Technology. He has served as the technical program committee co-chair for numerous conferences. He serves on the editorial boards of several journals, including the Co-Editor-in-Chief for *Ad Hoc & Sensor Wireless Networks* and a Lead Series Editor for the IEEE Transactions on Vehicular Technology, the IEEE Transactions on Green Communications and Networking, and the IEEE Communications Surveys and Tutorials. He is a Distinguished Lecturer, the Vice President (Membership), and an Elected Member of the Board of Governors of the IEEE Vehicular Technology Society.

JIANMIN LIU received the B.Eng. degree in engineering from Tsinghua University in 1989 and the M.Eng. degree in systems engineering from the Beijing University of Aeronautics and Astronautics in 1992. He is currently the CEO of AjaxWeaver. He is also a Research Associate with Carleton University. He has over 20 years of experience with the software industry. His current research interests include distributed ledger technology, network security analytics, and deep learning.

YING HE (S'16) received the B.S. degree in communication and information systems from Dalian Ocean University, Dalian, China, and the M.S. degree in communication and information systems from the Dalian University of Technology, Dalian, in 2011 and 2015, respectively. She is currently pursuing the Ph.D. degree with the Dalian University of Technology and Carleton University. Her current research interests include machine learning, security, blockchain, big data, and wireless networks.



PENGBO SI (SM'15) received the B.E. and Ph.D. degrees in communications engineering and communication and information system from the Beijing University of Posts and Telecommunications, Beijing, China, in 2004 and 2009, respectively. From 2007 to 2008, he was a Visiting Ph.D. Student with Carleton University, Ottawa, Canada. He joined the Beijing University of Technology, Beijing, in 2009, where he is currently an Associate Professor. From 2014 to 2015, he was a Visiting Research Scholar with the University of Florida, Gainesville, FL, USA.

Dr. Si received the Best Paper Runner Up Award from the 2015 International Conference on Wireless Algorithms, Systems, and Applications. He served as the Technical Program Committee Co-Chair of the IEEE ICCC-GMCN'2013 and the Program Vice-Chair of the IEEE GreenCom'2013. He serves as an Associate Editor for the *International Journal on AdHoc Networking Systems*, the Editorial Board Member for *Ad Hoc & Sensor Wireless Networks*, and the Guest Editor for *Advances in Mobile Cloud Computing* and the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING Special Issue on Advances in Mobile Cloud Computing.

YANHUA ZHANG received the B.E. degree from the Xi'an University of Technology, Xi'an, China, in 1982, and the M.S. degree from Lanzhou University, Lanzhou, China, in 1988. From 1982 to 1990, he was with Jiuquan Satellite Launch Center), Jiuquan, China. In 1990s, he was a Visiting Professor with Concordia University, Montreal, Canada. He joined the Beijing University of Technology, Beijing, China, in 1997, where he is currently a Professor. His research interests include QoS-aware networking and radio resource management in wireless networks.

• • •