Software-defined Content Dissemination Scheme for Internet of Healthcare Vehicles in COVID like Scenarios

Amuleen Gulati*, Gagangeet Singh Aujla†, *Senior Member, IEEE*, Neeraj Kumar‡, *Senior Member, IEEE*,
Sahil Garg§, *Member, IEEE*, Georges Kaddoum§, *Member, IEEE*
*Ericsson, Ottawa, Ontario, Canada
†Department of Computer Science, Durham University, Durham, United Kingdom
‡Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, India
§ Ecole de Technologie Sup´erieure, Universit´e du Qu´ebec, Montr´eal, Canada
(E-mail: amuleengulati13@gmail.com, gagangeet.s.aujla@durham.ac.uk, neeraj.kumar@thapar.edu,
sahil.garg@ieee.org, georges.Kaddoum@etsmtl.ca)

Abstract

The exponential increase in the number of connected devices in the network led to a paradigm shift from the traditional host-centric IP-based network architecture to a content-based paradigm, which eliminates the address-content bindings in traditional IP-based architectures. In content-centric networking (CCN) architecture, contents are accessed by name rather than the physical location where these are stored which results in more flexible and robust content-based architecture. However, the broadcast nature of devices in CCN causes network congestion and latency. Internet of Healthcare Vehicles (IoHV) is as evolving paradigm which rely on the smart transportation involving ambulances and other healthcare vehicles (for rapid COVID testing) to connect with each other through the Internet specially in case like COVID 19 testing and contact tracing. However, the applications of IoHV are not similar to other networks and therefore bring new technical challenges due to the high mobility and rapid changes of topology. Therefore, to handle these challenges, in this paper, we propose a novel scheme which finds an optimal path. Using the optimal path, the interest packet is forwarded among healthcare vehicles in a smart city scenario. The proposed scheme maximizes the probability of finding the requested data while minimizing latency. As the proposed scheme does not involve broadcasting of interest packets, the problem of network congestion in CCN can be solved to a large extent. In order to facilitate the inter-interoperability of heterogeneous healthcare devices in the network, and to improve the routing flexibility, an SDN controller is deployed in IoHV scenario. It maintains the global information of the network in its flow tables. Moreover, to achieve faster response time and lower latency in IoHV environment, edge devices are utilized instead of central cloud. Keeping in view the memory constraints of edge devices, deletable bloom filters are used for storage and caching. Finally, the open issues and challenges related to the proposed solution for IoHV scenario are also presented.

Index Terms

Bloom filter, Content centric networking, COVID 19, Data dissemination, Edge caching, Internet of healthcare vehicles, Software defined networking

1 Introduction

In COVID-like situations, the testing and contact tracing is the most important part of handling the pandemic effectively. This leads to the movement of sensitive healthcare data across different healthcare devices and vehicles. For example, a COVID testing van deployed in a high-risk area collects a large number of samples and sends the information related to the tested person through some healthcare application running over the Cloud. This process requires an open mechanism for smooth transmission of data from one healthcare vehicle to another remote sample processing centre. Despite of various developments in the communication paradigms, the core network mechanisms still depend on the address-content binding. A user, who wishes to request data must provide the location (in terms of IP/MAC addresses) of the node from which data is to be retrieved. But with the ever-increasing amount of Internet content (multiplied in COVID times), and the distributed nature of the dedicated servers, this address-content binding induces latency in communication. It also incurs additional overhead due to multiple DNS lookups and name resolution services which the protocol needs to invoke to get the requested content. But, the reliance on such traditional address-content binding based network architectures does not suite such healthcare scenarios.

To overcome these issues, content-centric networking (CCN) emerged in the recent years [1]. In CCN, instead of focusing on the physical location of the data which is to be retrieved, the main emphasis is given to the content which is required. So, instead of accessing a data item by specifying the IP address of the host on which it is stored, the users can simply fetch the data by name. This is done by sending the query in the form of named data packets, also called Interest packets, in the network to fetch the desired content. When the data matching the queried content is found, it is returned to the requester of the data in the form of data packets, by following the reverse path of the query.

Owing to its content-centric approach, in-network caching and intrinsic security support due to the use of data name instead of IP-addresses to access data, information-centric approach has found applications in many emerging fields as a replacement of traditional networking architecture. The content-centric model has been applied in many scenarios including Internet of Things [2], Smart Grids [3], Vehicular Ad Hoc Networks [4] and Wireless Sensor Networks [5]. However, the emergence of these scenarios has introduced heterogeneity in the network. Today's networks consists of many different devices, including smart phones, laptops, vehicles, traffic lights and virtual machines. All these devices having different propriety firmware and different configurations must be integrated into the same network for effective communication and resource sharing. Moreover, with this growth in the number of devices and increase in network heterogeneity, the network traffic is increasing rapidly but at the same time, the traffic pattern has become unpredictable. Therefore, there arises a need to manage the traffic flow in the network to optimize performance by minimizing network traffic and to maximize resource utilization. Hence, to address these issues, software defined networking approach (SDN) was proposed in the literature [6]–[8].

SDN works by segmenting the network control and forwarding functions. In this scenario, an SDN controller segregates the network into two planes: control plane and data plane. The control plane is responsible for making routing decisions for the data packets and the data plane actually moves the packets from the source to the intended destination [9]. This architecture makes the control plane directly programmable through SDN controller which allows the application of globally aware software-based control at the edges of the network. Another advantage offered by SDN is its flexibility. SDN controller uses OpenFlow standard which allows integration of multiple propriety firmware into a single network [10]. This enables the edge devices to access the network switches and routers using propriety firmware, thus eliminating the possibility of vendor lock-in of firmware. Therefore, by using SDN approach, software-based routing can be achieved in a heterogeneous network, thereby reducing the problem of network congestion to a large extent.

2 Contributions

The above vision could be very promising in context of Internet of Vehicles specifically Internet of Healthcare Vehicles (IoHV) that are predominantly used as mobile testing centers in COVID 19 or related pandemic scenarios. Therefore, keeping in mind the requirements of such critical scenarios, the contributions of this paper are as follows:

- An IoHV scenario is presented in a smart city. In this scenario, an SDN-based controller is deployed to improve the flexibility of routing and interoperability of heterogeneous network devices.
- A cluster formation and cluster head selection scheme has been designed for optimal clustering of devices. In this scheme, the mobility, connectivity and inter-vehicular distance of healthcare vehicles are used as parameters for cluster formation.
- A deletable bloom filter-based cache structure is proposed for maintaining the flow tables in FIB and caches at each node in IoHV environment.
- A content announcement and data dissemination scheme has been presented to determine the optimal path for routing interest packets in order to maximize hit ratio and to minimize latency.

3 Background

The traditional IP-based networking model is based on client-server communication between the data requesters and a data host. This model is based on TCP/IP protocol and it is widely used to transfer the data between nodes for different applications (e.g., smart transportation, e-health, online social network etc.). Each node in the network is assigned an IP-address and the source of requested data is made accessible through this unique IP-address. This architecture enabled a general-purpose node in the network, running a specialized server operating system, to extend its capabilities by using shared resources of other nodes in the network and thus serve as a centralized server. However, this model did not make it mandatory for the server to have more resources than the clients in the network. This model is used by the centralized computing architecture, which explicitly allocated a large number of network resources to the servers. The client invoked services of the server may have a request-response TCP/IP protocol to request data or resources. But in a large network scenario, the server became a performance bottleneck due to overburdening or overloading. This issue was resolved by using multiple servers in the network.

However, the exponential increase in the number of Internet users [11] made the generation of massive Internet content inevitable. As a result of this, there was a shift from the traditional client-sever architecture to a distributed and more sophisticated peer-to-peer networking model which reduced the burden on server by facilitating the source and destination to communicate among themselves to share content and resources. A new peer-to-peer protocol (P2PP) came into existence to maintain heterogeneous connectivity in the network and to enable resource-publishing and look up. It was implemented as a request-response protocol but TCP/IP and UDP still remained the underlying protocols to exchange request/response messages in P2PP. Different routing schemes were implemented by this model viz., recursive routing and iterative routing, for unicast and broadcast, respectively. In this way, this model was able to reduce the cost of implementation and configuration and eliminated the use of dedicated servers. But, despite of its advantages, it gave birth to new issues. As there was no centralized control over data sharing, it resulted in security breaches of data. Further, unauthorized use of even a single node in the network could lead to compromise of the entire

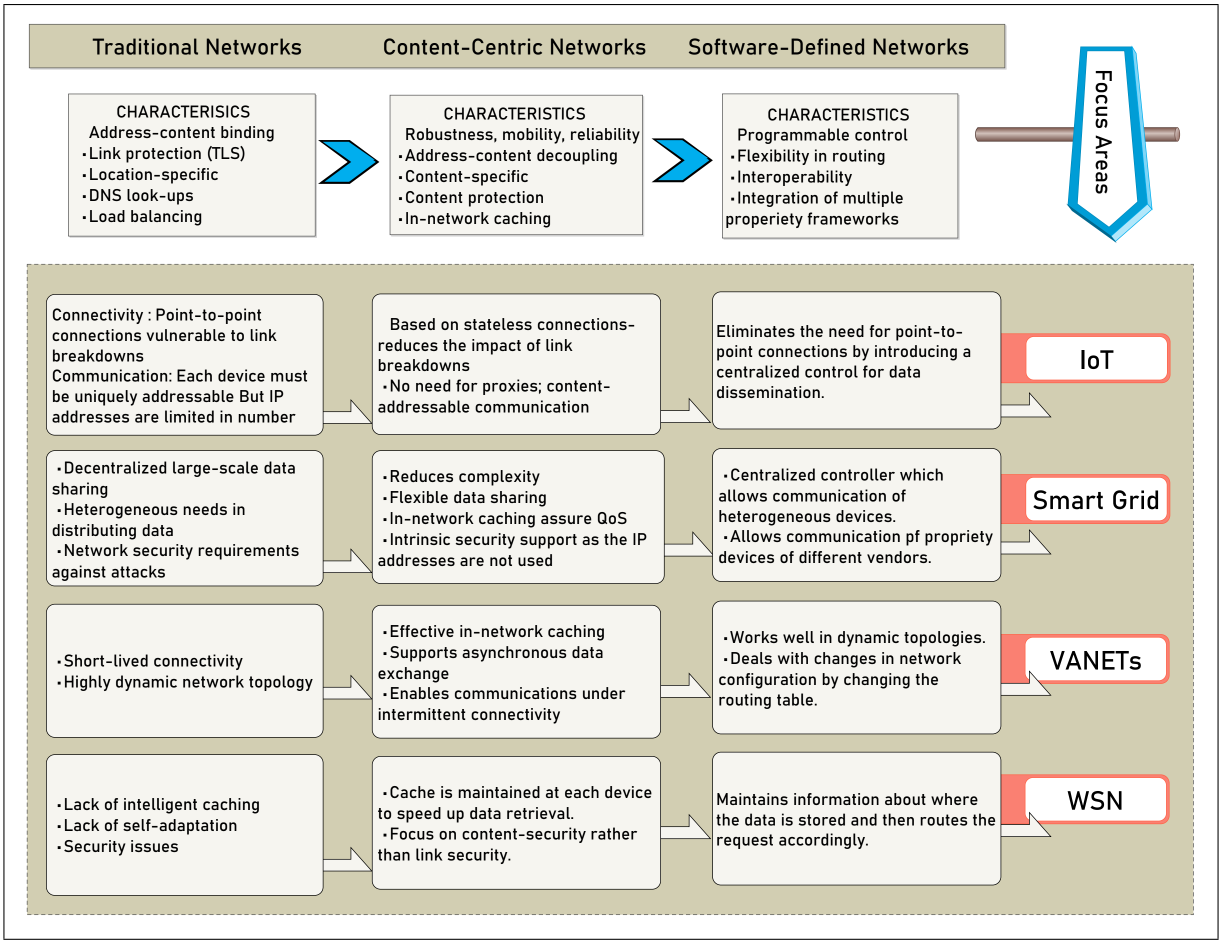

Traditional Networks
Content-Centric Networks
Software-Defined Networks
Focus Areas
CHARACTERISICS
Address-content binding
•Link protection (TLS)
•Location-specific
•DNS look-ups
•Load balancing
CHARACTERISTICS
Robustness, mobility, reliability
•Address-content decoupling
•Content-specific
•Content protection
•In-network caching
CHARACTERISTICS
Programmable control
•Flexibility in routing
•Interoperability
•Integration of multiple properiety frameworks
Connectivity : Point-to-point connections vulnerable to link breakdowns
Communication: Each device must be uniquely addressable But IP addresses are limited in number
Based on stateless connections- reduces the impact of link breakdowns
•No need for proxies; content-addressable communication
Eliminates the need for point-to-point connections by introducing a centralized control for data dissemination.
IoT
•Decentralized large-scale data sharing
•Heterogeneous needs in distributing data
•Network security requirements against attacks
•Reduces complexity
•Flexible data sharing
•In-network caching assure QoS
•Intrinsic security support as the IP addresses are not used
•Centralized controller which allows communication of heterogeneous devices.
•Allows communication pf propriety devices of different vendors.
Smart Grid
•Short-lived connectivity
•Highly dynamic network topology
•Effective in-network caching
•Supports asynchronous data exchange
•Enables communications under intermittent connectivity
•Works well in dynamic topologies.
•Deals with changes in network configuration by changing the routing table.
VANETs
•Lack of intelligent caching
•Lack of self-adaptation
•Security issues
•Cache is maintained at each device to speed up data retrieval.
•Focus on content-security rather than link security.
Maintains information about where the data is stored and then routes the request accordingly.
WSN

Figure 1: Evolution of networking architectures

network. Therefore, there arose a need for a centralized control over the data sharing.

To combat these issues, cloud computing model was proposed in late 90's. This model is a combination of characteristics of all the previous models including client-server model, grid computing, and peer-to-peer computing. It provides location independence-based data collection and analytics which allows its users to access cloud services, including softwares, infrastructures, platforms, networks, storage facilities, and so on, using their web browsers irrespective of the location. Virtualization mechanism, which is the crux of cloud computing model, further improves the resource utilization by separating a physical device into a number of independent virtual machines. In addition, this model provides centralization of network data which makes data access mechanisms more efficient and reduces the risk of security breaches on data. However, the distribution of data over such a large number of devices and networks increases the complexity of security in the model and rigorous attempts are being made by the research community to enhance the data security of cloud data.

Despite of these developments in the communication paradigms, the core network mechanisms still depends on address-content binding. A user, who wishes to request data must provide the location (in terms of IP/MAC addresses) of the node from which data is to be retrieved. But with the ever increasing amount of Internet content, and the distributed nature of the dedicated servers, this address-content binding induces latency in communication. It also incurs additional overhead due to multiple DNS look ups and name resolution services which the protocol needs to invoke to get the requested content.

Also, due to the increasing number of content providers and consumers, the number of IP-addresses available are depleting at a rapid pace. Further, due to an unprecedented increase in the number of smart devices and hence the amount of consumer generated data, the traditional IP-based networks are no longer sufficient to handle the content requests from the user due to the address-content binding mechanisms used in IP-based protocols. Such protocols incur high overhead in delivering content due to host-to-host session-based data delivery architectures. A brief description of the challenges faced by traditional networks and how the content-centric approach and software-defined networking can solve these issues are summarized in Fig. 1.

4 CCN Model for Internet of healthcare Vehicles

Fig. 2 shows the proposed CCN-based IoHV scenario in a smart city. In this scenario, we have considered a special case where the healthcare vehicles (such as ambulances, COVID testing vans, COVID drive through centers, etc) are

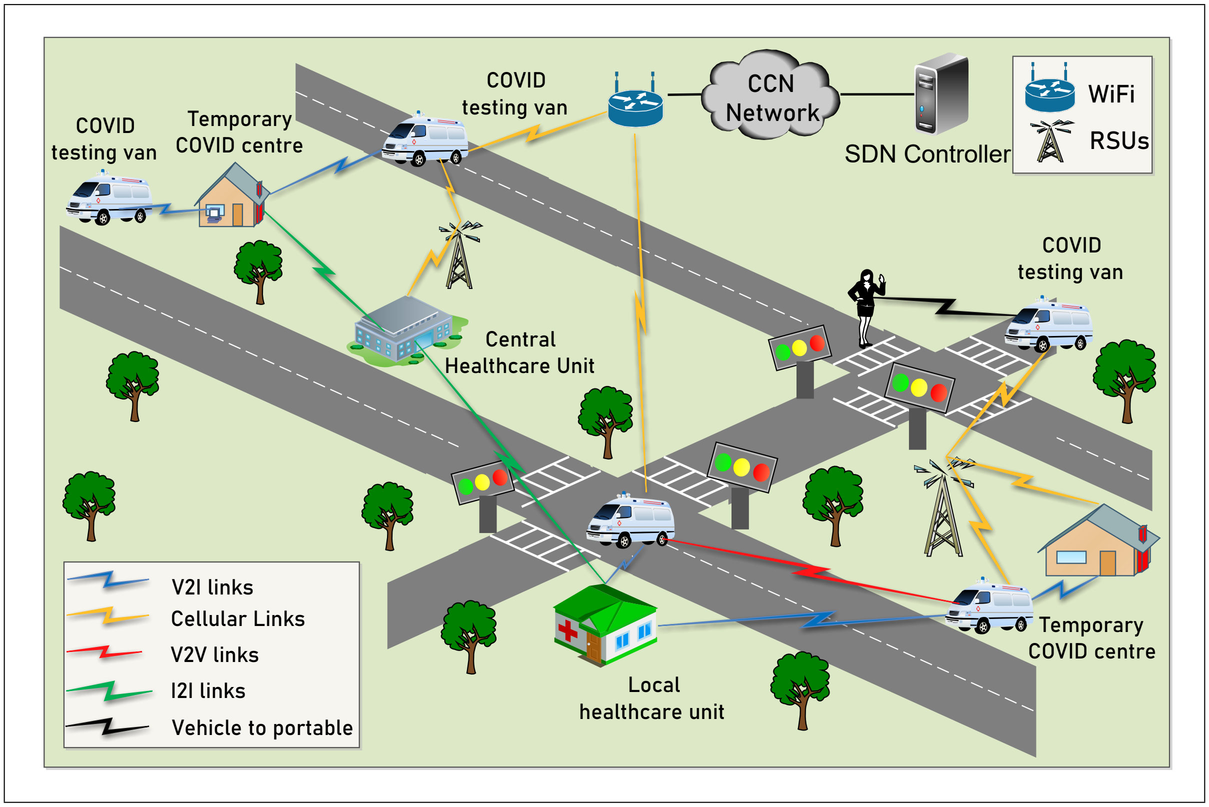


Figure 2: Problem IoHV scenario

deployed across a smart city setup. All healthcare vehicles in the network are connected to each other through different types of links, viz., V2I, I2I, V2V, V2P and cellular links [12]. An SDN controller is deployed to handle the flow scheduling among various devices. These devices may be any edge device including vehicles, road signs, portable devices like laptops, mobile phones, PDAs, any building like offices, homes and so on. Every device in the network consists of the three data structures storing different types of data as follows:

1) Content Store (CS) - The CS is a cache maintained at every node which serves to decrease the network traffic by caching the content based upon some caching policy intrinsic to each network. Every node contains some data that is stored in the CS. It stores the data along with its $content\_id$ for indexing. The data is named according to a naming scheme adopted by the network. The CS keeps track of the names of all types of data stored in it.
2) Pending Interest Table (PIT) - The PIT at each node stores the requests or interest packets that are received by it from other nodes. It contains the $node\_id$ of the requesting node along with the $content\_id$ of the requested data. It keeps record of similar requests which have been passed on by this node but whose reply has not yet been received. Upon receiving a request, a node checks its PIT. If a similar entry already exists in its PIT, then the interest packet is not forwarded. It just creates a new entry for the request. When the node receives the data, it sends out the data packets to all the nodes whose entries exist in its PIT. This structure helps in reducing the network congestion as the requests which want to fetch the same data objects are not forwarded again in to the network.
3) Forwarding Information Base (FIB) - The FIB is responsible for storing the routing information at each node. It contains information regarding the nodes to which the interest packets must be forwarded to reach the node containing the requested data. It keeps track of the $cluster\_id$ to which the node belongs, the $content\_id$ of data and the $next\_hop$ to which interest packet must be forwarded to retrieve the data with a $content\_id$ from the device with corresponding $cluster\_id$.

All these components together serve the purpose of extracting named content while minimizing the number of packets disseminated in the network.

All devices can share their data with other devices in the network, where each device belongs to a cluster. The devices are clustered based upon several metrics and a cluster head is selected for each cluster. The cluster head performs the task of content announcement which informs devices about the routes through which interest packets must be sent. These routes are stored by each device in its FIB which is updated periodically.

With this scenario, when a device wants to read a data object, it looks into its FIB table to determine the next node to which it has to forward the interest packet. It then sends out the interest packet to the designated node and eventually, the interest packet reaches the cluster head of the cluster whose devices contain the requested data

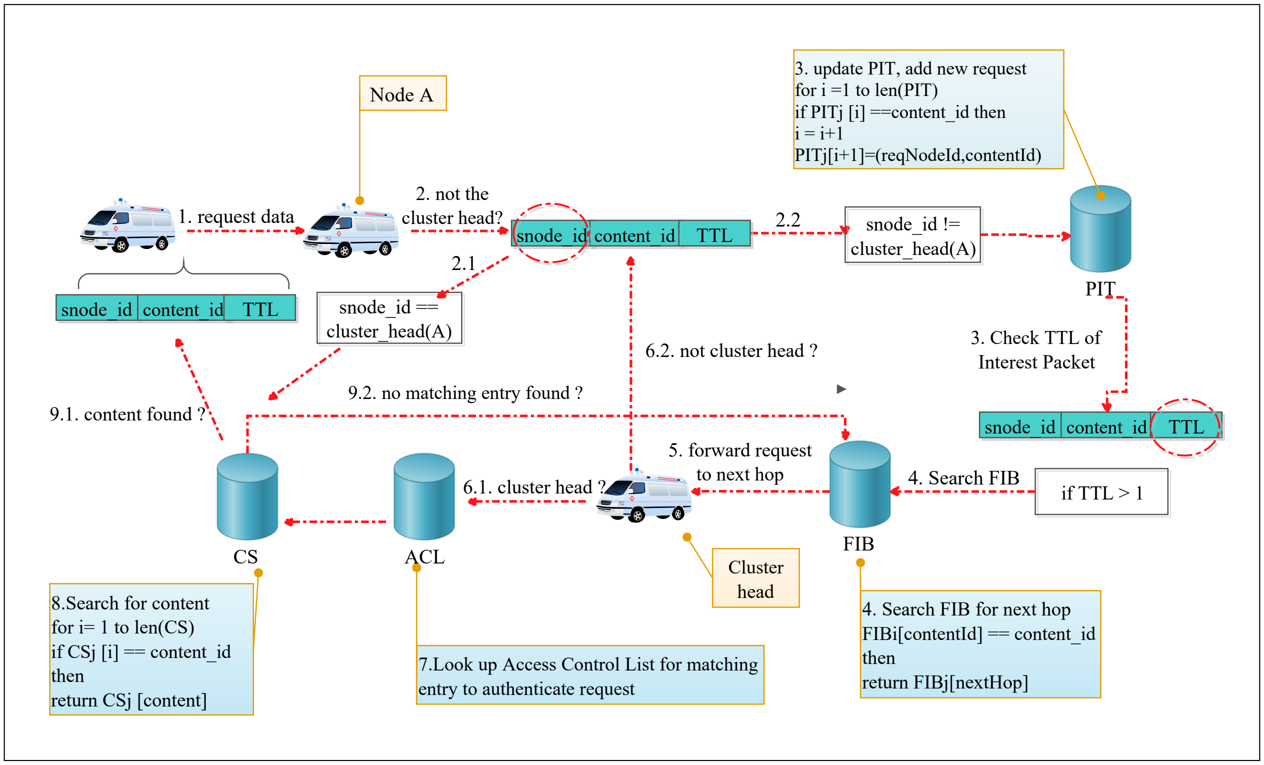

Node A
3. update PIT, add new request
for i =1 to len(PIT)
if PITj [i] ==content_id then
i = i+1
PITj[i+1]=(reqNodeId,contentId)
1. request data
2. not the cluster head?
snode_id
content_id
TTL
2.2
snode_id != cluster_head(A)
PIT
2.1
snode_id == cluster_head(A)
snode_id
content_id
TTL
6.2. not cluster head ?
3. Check TTL of Interest Packet
9.2. no matching entry found ?
9.1. content found ?
snode_id
content_id
TTL
5. forward request to next hop
4. Search FIB
if TTL > 1
6.1. cluster head ?
CS
ACL
FIB
Cluster head
8.Search for content
for i= 1 to len(CS)
if CSj [i] == content_id
then
return CSj [content]
7.Look up Access Control List for matching entry to authenticate request
4. Search FIB for next hop
FIBi[contentId] == content_id
then
return FIBj[nextHop]

Figure 3: Proposed scheme

packets. On receiving the interest packet, the cluster head first checks the Access Control List (ACL) to check if the requesting node has the authority to access and read the requested data. If the corresponding entry exists in the ACL of cluster head, the cluster head then looks up into its FIB to determine which devices in its cluster are holding the requested data and forwards the interest packet to those devices. The interest packet contains the $node\_id$ of the cluster head. Each device, before responding with the data packets matches the requester $node\_id$ in the interest packet with the $cluster\_id$ stored in its ACL. Once the cluster head is authenticated by the node, it sends the requested data to the cluster head which in turn, combines the responses of all interest packets sent out by it and sends it back through the reverse path of the interest packet.

In this way, the data packet reaches the requesting device. This scheme ensures that only that device in the network can access data which has the authority to read the data. So any possibility of illegitimate data access is minimized. Also, devices holding the similar data are grouped together into clusters, which reduces the duplicate data packets in the network, thereby reducing network traffic. Also, as the SDN controller has a global view of the network, the underlying routing and clustering decision are be programmed to optimize network performance.

Therefore, the major goal of the proposed scheme is to maximize the value of the objective function ψ as,

$$\begin{aligned} \psi = max\{P_i[C_j] - Z(D(i,j)/|v_i - v_j|)\} \\ \forall i,j \in \{1,2,3,...,n\}; \\ subject\ to\ constraints \\ |v_i - v_j| > 0; \\ D(i,j) \leq TX_{CH}; \end{aligned}$$

where $P_i[C_j]$ is the probability of finding content requested by device i in the cache of device j, $D(i,j)$ denotes the distance between devices *i* and *j*, and $|v_i - v_j|$ is the relative velocity of node *i* and *j*, $Z(D(i,j))/|v_i - v_j|)$ is the value of latency $D(i,j))/|v_i - v_j|)$ normalized between *0* and *1*. ψ tends to maximize the probability $P_i[C_j]$ while minimizing the latency incurred for retrieving the requested content.

The proposed work is designed to find an optimal route to send the interest packet to the requesting node to maximize ψ.

5 Proposed Scheme

The proposed scheme (Fig. 3) uses deletable bloom filters for storing information in CS, PIT, FIB and ACL at each node as well as for maintaining the flow tables. The devices in the network are clustered based upon several device metrics collected by RSUs and a cluster head for each cluster is selected. Fig. 4 shows the sequence diagram of the proposed scheme. The overall scheme follows the steps given below:

1) In step 1, the cluster head (when newly elected), sends a probe message to each device in its cluster, asking for the type of data that each device holds.

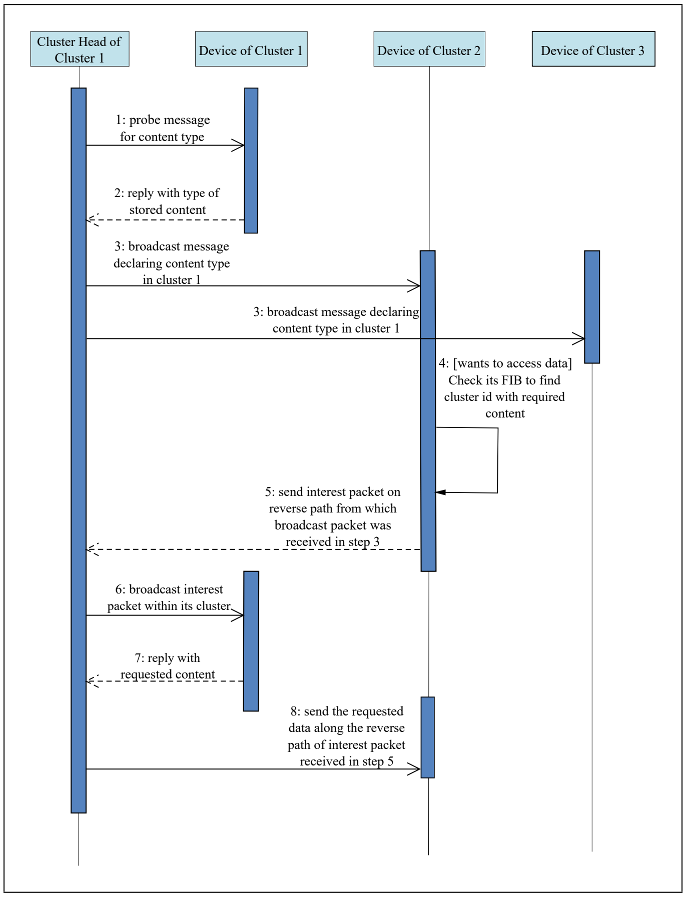

Cluster Head of Cluster 1
Device of Cluster 1
Device of Cluster 2
Device of Cluster 3
1: probe message for content type
2: reply with type of stored content
3: broadcast message declaring content type in cluster 1
3: broadcast message declaring content type in cluster 1
4: [wants to access data] Check its FIB to find cluster id with required content
5: send interest packet on reverse path from which broadcast packet was received in step 3
6: broadcast interest packet within its cluster
7: reply with requested content
8: send the requested data along the reverse path of interest packet received in step 5

Figure 4: Sequence diagram

2) In step 2, each device replies with the type of content it is holding by sending a content announcement (CA) packet as a response to the probe message.
3) In step 3, the cluster head combines all the received CA packets and sends out a single CA packet to all devices in the network (except its own cluster), indicating the type of content in its cluster. Each device upon receiving this CA packet stores the path through which it has received the packet along with the $cluster\_id$ in its FIB.
4) In step 4, when a device wants to access data, it looks in its FIB to find the next hop to which it must send interest packet to reach the cluster holding the required content. In case, there is no matching FIB entry, then the device broadcasts the interest packet.
5) In step 5, the device sends out the interest packet to the next hop(s) found in the FIB.
6) Upon receiving the interest packet, the cluster head changes the id of source node $snode\_id$ to its own $cluster\_id$ and broadcasts the packet in its cluster.
7) In step 7, when the devices receive the interest packet from the cluster head, they perform a CS look up to check if they have the requested data. The devices holding the requested data send the data packet to the cluster head.
8) Finally in step 8, the cluster head combines all received data packets into a single packet and sends it along the reverse path of the interest packet. Every intermediate node updates its PIT and CS and finally the data packet reaches its requester.

5.1 Deletable Bloom Filter-based Storage

To speed up the look up operations of flow tables in SDN and the CS, PIT, FIB and ACL in devices and to make the operation more efficient, we have used a probabilistic data structure- Bloom Filter (BF) [13], [14]. The insertion operation is done by passing the input through *k* hash functions h_1, h_2,...,h_k to produce *k* indices. The bits at the corresponding indices in the bloom filter are then set to *1*. Later, if a data item *x* is searched, then *x* is passed through the same hash functions and *k* indices are produced. If all *k* bits are set to *1*, it indicates the presence of *x* in the BF. The insertion and look up operation is demonstrated in Fig. 5.

The BF makes a trade-off between space efficiency and accuracy, to maintain this information. However, it completely removes the possibility of detecting false negatives. Therefore, the effect on accuracy can be neglected for improving the space efficiency. Thus, using BFs for detecting data availability in a data structure can significantly speed up data access while also improving space efficiency.

However, a major drawback of using BFs is that it provides no function for deleting data, i.e., the traditional BF is non-deletable. In case some data is to be deleted, the BF needs to be constructed again from scratch. This process can be costly and time consuming as the cache information needs to undergo regular insertions and deletions to keep the cache up-to-date. Therefore, in our scheme, a Deletable Bloom Filter (DBF) [15], an enhancement over the traditional BF is used. DBF provides the following operations:

- Find(*x*): This function returns *1* if all *k* bits are set to *1* and *0* otherwise.
- Insert(*x*): This function inserts *x* into the cache by setting the corresponding bits to *1* in DBF. If the bit is already set in DBF, then the region is marked as non-deletable by setting its *r* bit to *1*.
- Delete(*x*): This function deletes *x* from the cache by resetting the corresponding bits to *0* in DBF. Only the bits which lie in the collision-free zone are reset.

DBF is based on the fact that even if one bit of an element *x* is deleted, then *x* will be deleted from DBF. To implement deletion, an *m*-bit DBF is divided into *r* regions, each of (m'/r) bits where $m' = m - r$. The *r* bits correspond to the *r* regions and are used for encoding the region collision information. In a DBF, collision is said to occur, if more than one element has the same bit index in DBF. While inserting an element, if the corresponding bit is already set in DBF, then the region is marked as non-deletable (set to *1*) by setting the corresponding *r*-bit to *1*. Hence, only those bits can be deleted which lies in collision-free regions. The entire process is illustrated in Fig. 5.

5.2 Clustering in IoHV scenario

The clustering process in IoHV scenario is divided into two phases as described below.

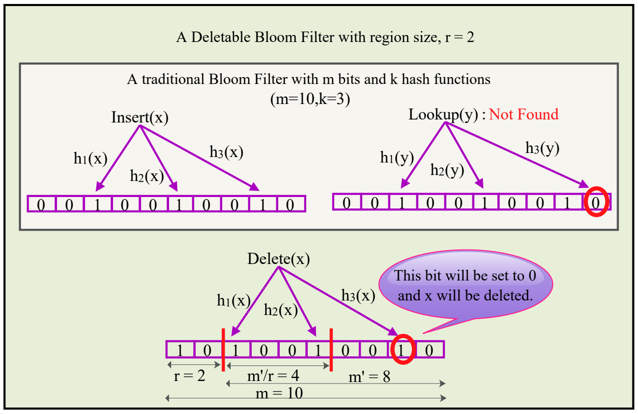


Figure 5: Deletable Bloom Filter

5.2.1 Cluster Formation

In the proposed work, various edge devices are grouped into clusters. Clustering devices in this way enables the cluster head (CH) to maintain a local view of its cluster, which in turn, improves the manageability of heterogeneous network devices. Each device in the network periodically sends their velocity, connectivity and position information to the nearby RSU. Hence, each RSU have a local view of the network topology and consequently, the SDN controller have a global view of the network topology.

This information sent out by each device is used to calculate their respective mobility M_i using Euclidean distance formula and the Inter-Vehicular distance of each device as $D(i, j)$ denoting the distance of device i from device j. Based upon these metrics of devices in a road segment L, with in the range of an RSU, clusters are formed consisting of healthcare vehicles with comparable mobility and inter-vehicular distance. These clusters are updated periodically as devices move in and out of the network.

5.2.2 Cluster Head Selection

Each cluster independently chooses a CH. The task of CH is two-fold:

- To keep track of the type of data that is stored in the devices and healthcare vehicles lying in its cluster.
- To inform other devices in the network about type of content in its cluster through content declaration.

The CH of a cluster is selected based upon two-factors: mobility and connectivity. The device with minimum mobility, min(M_1, M_2, M_3,..., M_n) and maximum connectivity, i.e., the node which has the maximum number of one-hop neighbors is selected as the CH. Meanwhile, another back-up CH is also elected to serve as CH in case the CH fails or moves out of the cluster range.

5.3 Content Announcement

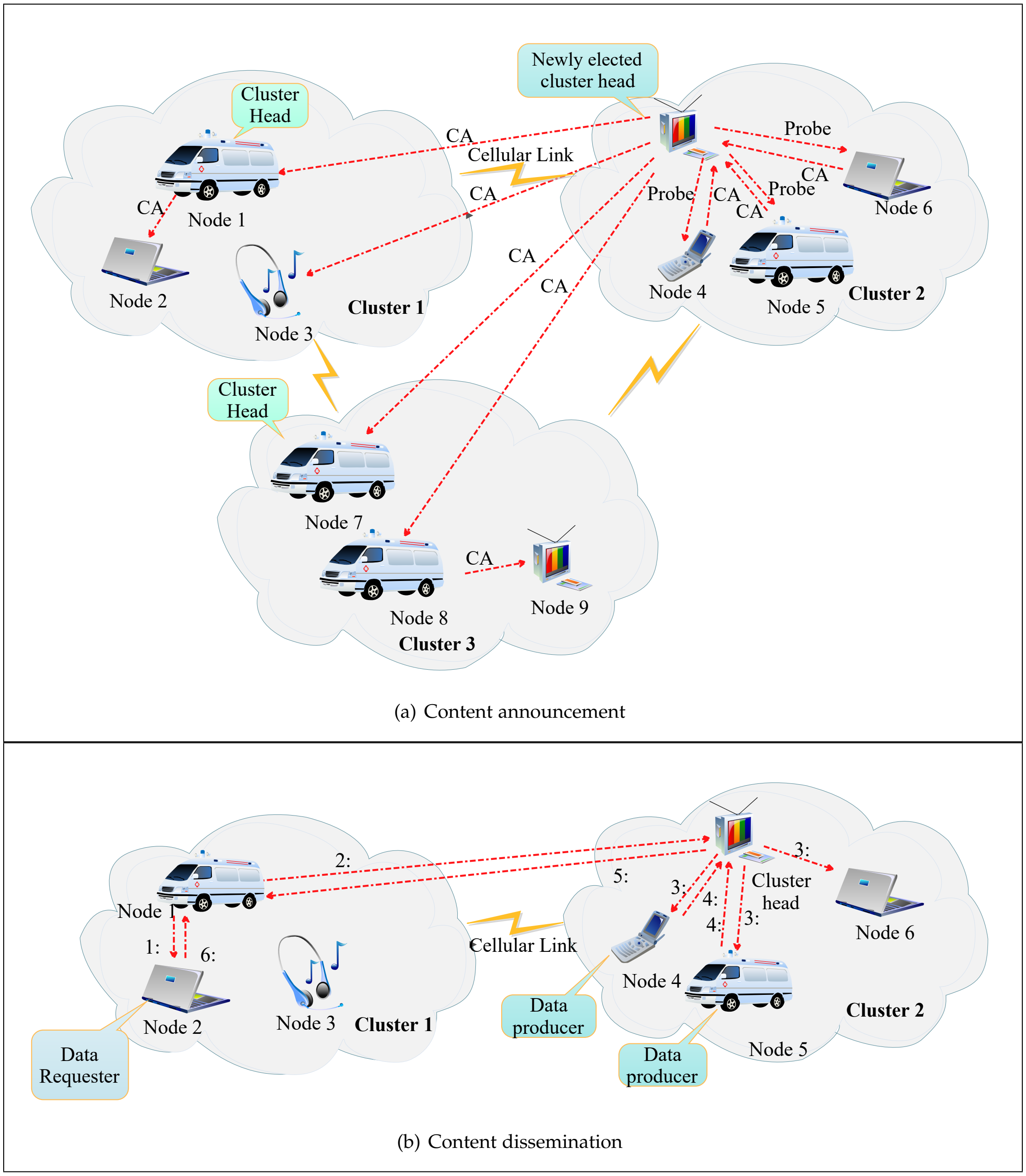
The aim of this phase is to make all the devices aware of the various types of data available in different clusters. Fig. 6(a) shows the content announcement scenario.

1) The CH floods its cluster with probe packets.
2) Each node, upon receiving the probe packet, replies with a content announcement (CA), packet which indicates the types of content stored at that device.
3) The CH then combines all the CA packet and sends a single CA packet, indicating all types of content available in that cluster, to all the devices and healthcare vehicles in the network.
4) Each device, upon receiving CA packet stores the information, including cluster id, type of content in that cluster and the path, consisting of intermediate nodes, through which an interest packet must be sent to that cluster, in the FIB and later, can access this type of data through this stored path.
5) The FIB is updated periodically by the received content declaration packets. In this way, the changing network topology can be accommodated in the FIB.

5.4 Content Dissemination

This phase is invoked when a device or healthcare vehicle wants to access content in the network. Fig. 6(b) shows the flow of events that take place in this scenario.

1) In the first step, the device looks up into its FIB to detect the entry for the corresponding data. This look up will return the device id of the device to which it must forward the interest packet. In case, no FIB entry is found, then it will broadcast the packet to all devices within one hop distance. This can be done by setting the TTL field of interest packet to *1*.
2) Upon receiving an interest packet, if the receiving node is not the cluster head, it first checks the device id to determine whether the packet has been sent by its cluster head or some other node in the network. In the former case, it first checks its CS for data. If it holds the requested content, then it forwards it along the reverse path of the query.
3) If the content is not found, then it follows the steps of the latter case, step 2, in which it looks up into its PIT to check for duplicate requests. Then, it checks the TTL field of the received packet. If it is zero, then the packet is discarded. Otherwise, it invokes an FIB look up to determine the next hop to which it must transmit the interest packet.
4) In case, the node receiving an interest packet is the cluster head, then it will look up in the ACL to ensure the requesting device id has the permission to access the data. If matching entry does not exist, then access to content is denied. This is done to ensure only authenticated data access.
5) If matching entry is found, then the device looks up in its CS. If it has the requested content, then it returns the content through the reverse path of the query. Otherwise, in step 3, it looks up into its FIB to determine the next hop to transmit the interest packet.
6) Point 2 is then recursively called for each device in the cluster, and the devices holding the requested content returns it in the form of packets in step 4.



(a) Content announcement

(b) Content dissemination

Figure 6: Content announcement and dissemination

7) The cluster head then combines all the content packets received and send it through the reverse path of the query in steps 5 and 6.

6 Future Challenges and Open Issues

In this paper, an SDN based content announcement and data dissemination approach is presented for a heterogeneous IoHV network scenario. In the proposed scheme, each node maintains the flow tables and cache data using deletable bloom filter data structure which facilitates the interoperability of network devices. Initially, the SDN controller gains the global information of network devices including position, velocity and connectivity through local RSUs and then use this information for optimal cluster formation. In the next stage, a cluster head is selected which serves as a gateway for entry and exit, to and from the cluster. In the next step, the cluster head performs the task of content announcement using probe and CA packets. Finally, the data dissemination takes place through stored paths in FIB. The presented approach aims to maximize the hit ratio by sending interest packets through those paths where the probability of finding the corresponding data packets is high. As the interest and data packets are not broadcast, this approach reduces the network traffic and latency. However, still some open issues and challenges exists. Such challenges are listed as below:

1) **Controller Placement Problem (CPP):** In a large network scenario, the placement position of the SDN controller is crucial and can incur significant delays in the network. Therefore, deciding the num-

ber of RSUs covered by an SDN controller is a major challenge.

2) **Energy Availability:** By using the multiple SDN controllers, the number of OpenFlow switches increases which in turn increase the energy consumption of the devices. Therefore, the energy of devices is a concern.
3) **Security and Integrity:** As each node can serve as a data provider as well as data consumer, the integrity of the data is a major concern. ACL can be maintained at each device for authentication of source. This aspect needs to be explored further.
4) **Fault Tolerance:** The Cluster Head can move out of the network range which can handicap the cluster. Therefore, dual cluster head selection and handover to new cluster in case of failure is still unaddressed.
5) **Flow Table Management:** As the SDN controller has a limited memory capacity, accommodating the high cost and power consumption of the flow table management is a challenging task.

Acknowledgement

This work is partially supported through Startup Fund provided by the Durham University, UK.

References

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, 2012.

[2] O. Waltari and J. Kangasharju, "Content-centric networking in the internet of things," in *13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2016, pp. 73–78.

[3] K. Katsaros, W. Chai, N. Wang, G. Pavlou, H. Bontius, and M. Paolone, "Information-centric networking for machine-to-machine data delivery: a case study in smart grid applications," *IEEE Network*, vol. 28, no. 3, pp. 58–64, 2014.

[4] M. Amadeo, C. Campolo, and A. Molinaro, "CRoWN: Content-centric networking in vehicular ad hoc networks," *IEEE Communications Letters*, vol. 16, no. 9, pp. 1380–1383, 2012.

[5] Z. Ren, M. A. Hail, and H. Hellbrück, "CCN-WSN-a lightweight, flexible content-centric networking protocol for wireless sensor networks," in *IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2013, pp. 123–128.

[6] G. S. Aujla, N. Kumar, A. Y. Zomaya, and R. Rajan, "Optimal decision making for big data processing at edge-cloud environment: An SDN perspective," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, Feb 2018.

[7] G. S. Aujla, M. Singh, A. Bose, N. Kumar, G. Han, and R. Buyya, "Blocksdn: Blockchain-as-a-service for software defined networking in smart city applications," *IEEE Network*, vol. 34, no. 2, pp. 83–91, 2020.

[8] G. S. Aujla, R. Chaudhary, K. Kaur, S. Garg, N. Kumar, and R. Ranjan, "Safe: Sdn-assisted framework for edge–cloud interplay in secure healthcare ecosystem," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 469–480, 2018.

[9] G. S. Aujla, A. Singh, and N. Kumar, "Adaptflow: adaptive flow forwarding scheme for software defined industrial networks," *IEEE Internet of Things Journal*, 2019.

[10] G. S. Aujla, A. Jindal, and N. Kumar, "Evaas: Electric vehicle-as-a-service for energy trading in sdn-enabled smart transportation system," *Computer Networks*, vol. 143, pp. 247–262, 2018.

[11] D. Evans, "The internet of things: How the next evolution of the internet is changing everything. 2011," 2015, [Accessed on: Dec 2017]. [Online]. Available: http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_ 0411FINAL. pdf

[12] N. Kumar, S. Misra, J. J. P. C. Rodrigues, and M. S. Obaidat, "Coalition games for spatio-temporal big data in internet of vehicles environment: A comparative analysis," *IEEE Internet of Things Journal*, vol. 2, no. 4, pp. 310–320, Aug 2015.

[13] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[14] R. Chaudhary, G. S. Aujla, N. Kumar, J. J. P. C. Rodrigues, and A. Vinel, "Optimized big data management across multi-cloud data centers: Software-defined- network-based analysis," *IEEE Communication Magazine*, vol. 56, no. 2, Feb 2018.

[15] C. E. Rothenberg, C. A. Macapuna, F. L. Verdi, and M. F. Magalhaes, "The deletable bloom filter: a new member of the bloom family," *IEEE Communications Letters*, vol. 14, no. 6, 2010.

Amuleen Gulati (amuleengulati13@gmail.com) received her B.Tech. in Computer Science and Engineering from Guru Nanak Dev Engineering College, India in 2016 and her M.Tech. degree in Computer Software Engineering from Thapar University, India in 2018. She completed her Masters of Engineering in Computer Engineering from Carleton University, Canada in 2021. Currently she is working as Software Developer at Ericsson, Canada.

Gagangeet Singh Aujla (S'15, M'18, SM'19) (gagangeet.s.aujla@durham.ac.uk) received the B.Tech. and M.Tech. degrees in computer science and engineering from Punjab Technical University, Jalandhar, India, in 2003 and 2013, respectively, and the Ph.D. degree in computer science and engineering from the Thapar Institute of Engineering and Technology, Patiala, India, in 2018. He is an Assistant Professor of Computer Science at Durham University. Before this, he worked as a post-doctoral research associate at Newcastle University, a research associate at Thapar University (India), a visiting researcher at University of Klagenfurt (Austria) and on various academic positions for more than a decade. He has many research contributions in the area of smart grids, cloud computing, edge computing, vehicular networks, blockchain, Internet of Things, software-defined networks, security, and cryptography. He has over 100+ publications in highly ranked journals and conferences, including 60+ journal papers and 40+ conference articles. He has been awarded 2018 IEEE TCSC Outstanding Ph.D. Dissertation Award and 2021 IEEE System Journal Best Paper Award. He is an Area Editor of Adhoc Networks (Elsevier) and a Senior member of IEEE.

Neeraj Kumar [M'15, SM'18] (neeraj.kumar@thapar.edu) received the Ph.D. degree in CSE from Shri Mata Vaishno Devi University, Katra, Jammu and Kashmir, India, in 2009. He was a Postdoctoral Research Fellow with Coventry University, Coventry, U.K. He is currently a Full Professor with the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology (Deemed to be University), Patiala, Punjab, India. He is also a Visiting Research Fellow with Coventry University, Newcastle University, U.K. He has guided many research scholars leading to Ph.D. and M.E./M.Tech. He has received the 2018 and 2021 Best Paper Award from IEEE Systems Journal and ICC 2018, Kansas city, in 2018. He is an Associate Technical Editor of IEEE Communication Magazine and IEEE Network Magazine. He is an Associate Editor of IEEE Transactions on Network and Service Management, IEEE Transactions on Sustainable Computing, ACM Computing Surveys, IJCS (Wiley), JNCA (Elsevier), Computer Communications (Elsevier), and Security and Communication (Wiley).

Sahil Garg [S'15, M'18] (sahil.garg@ieee.org) received the Ph.D. degree from the Thapar Institute of Engineering and Technology, Patiala, India, in 2018. He is currently a research associate at Resilient Machine Learning Institute (ReMI) in correlation with École de Technologie Supérieure (ÉTS), Montréal. Prior to this, he worked as a postdoctoral research fellow at ÉTS, Montreal, and a MITACS researcher at Ericsson, Montreal. He has many research contributions in the area of machine learning, big data analytics, security and privacy, Internet of Things, and cloud computing. He has over 80 publications in highly ranked journals and conferences, including 50+ top-tier journal papers and 30+ respected conference articles. He has been awarded the 2021 IEEE Systems Journal Best Paper Award; the 2020 IEEE TCSC Award for Excellence in Scalable Computing (Early Career Researcher) and the IEEE ICC best paper award in 2018 at Kansas City, Missouri. He is currently a managing editor of Springer's Human-centric Computing and Information Sciences (HCIS) journal. He is also an associate editor of IEEE Network, IEEE Transactions on Intelligent Transportation Systems, Elsevier's Applied Soft Computing (ASoC), and Wiley's International Journal on Communication Systems (IJCS). In addition, he also serves as the Workshops and Symposia Officer for the IEEE ComSoc Emerging Technology Initiative on Aerial Communications.

Georges Kaddoum [M'11] (georges.kaddoum@etsmtl.ca) received his Bachelor's degree in electrical engineering from the École Nationale Supérieure de Techniques Avancés (ENSTA), France, his M.Sc. degree in telecommunications and signal processing from Telecom Bretagne (ENSTB), Brest, in 2005, and his Ph.D. degree in signal processing and telecommunications from the National Institute of Applied Sciences (INSA), Toulouse, France, in 2009. He is currently an associate professor and Tier 2 Canada Research Chair with the École de Technologie Supérieure, University of Quebec, Montréal, Canada. His recent research activities cover wireless communication networks, resource allocations, security and space communications, and navigation. He was awarded the ÉTS Research Chair in physical layer security for wireless networks in 2014, and the prestigious Tier 2 Canada Research Chair in wireless IoT networks in 2019. He has published over 150+ journal and conference papers and has two pending patents. In addition, he received the Research Excellence Award of the Université du Québec in 2018. In 2019 he received the Research Excellence Award from the ÉTS in recognition of his outstanding research outcomes.