

Compute Node Communication in the Fog: Survey and Research Challenges

Vasileios Karagiannis

Distributed Systems Group, TU Wien, Austria

v.karagiannis@infosys.tuwien.ac.at

Abstract—Fog and edge computing have given rise to applications that utilize cloud services at the edge of the network. To enable services at the edge, compute nodes are provisioned at strategic positions throughout the network in order to avoid bottlenecks and reduce latency. A lot of research has been conducted in this context, resulting in multiple computing platforms which organize the compute nodes using different communication types, i.e., hierarchical, peer to peer or hybrid. To better understand the role communication plays in fog and edge computing, this paper reviews key literature on fog computing platforms and identifies research challenges that emerge. The outcomes of this review also suggest that the communication type of a computing platform affects the functionality of the final applications.

Index Terms—Edge computing, Literature review

I. INTRODUCTION

The compute resources of a fog/edge computing system are heterogeneous and geographically distributed [1]. These resources are typically represented by physical or virtual compute nodes which integrate logic to communicate with the nodes in proximity and outsource workloads towards each other [2], [3]. Outsourcing workloads towards compute nodes in proximity (rather than towards remote clouds) hinders the formation of bottlenecks and reduces latency, which can be useful for coping with the increased traffic from the Internet of Things (IoT) [4], [5]. Thus, hereinafter the fog is considered to consist of multiple compute nodes which may reside in the cloud, the core network or the edge and share workloads in order to facilitate applications, e.g., for the IoT [6].

Notable advances in the field of fog/edge computing describe the development of fog computing platforms [7]. However, since the focus of these platforms has been to demonstrate the runtime environment, the internal communication types are often not analyzed or evaluated. Even though all fog computing platforms share similar goals (e.g., to facilitate IoT applications), the compute nodes within them are organized differently [8]. As a consequence, each platform offers different functionality to the applications. For this reason, it is important to understand how the compute nodes are organized in the fog, which are the alternatives and which is the preferred communication type based on the requirements of the final application in mind.

The contributions of this paper are: *i)* The three main types of communication in fog computing are identified, namely: hierarchical, peer to peer (P2P) and hybrid. *ii)* A review

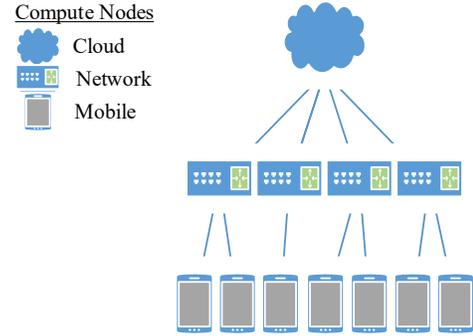


Fig. 1: Hierarchical communication in the fog.

of fog computing platforms is conducted, which focuses on the communication among the compute nodes and maps each platform to the respective communication type. *iii)* A set of criteria based on requirements of fog computing is proposed for comparing the reviewed computing platforms and communication types. *iv)* Research challenges in the communication among the compute nodes of the fog are identified by examining which criteria are met by each communication type. Even though current literature recognizes and briefly discusses communication types in the fog (e.g., [8] and [9]), the paper at hand provides a first attempt at showing that the communication type of a computing platform has an impact on the functionality of the final applications.

The rest of this paper is structured as follows. Sections II, III and IV present fog computing platforms from the literature and categorize them as hierarchical, P2P or hybrid, respectively. Afterwards, Section V provides a comparison of the discussed literature and identifies research challenges that emerge. Finally, Section VI concludes the paper.

II. HIERARCHICAL COMMUNICATION IN THE FOG

The hierarchical type refers to the communication among compute nodes which form a tree-like topology as shown in Fig. 1. In this type, each compute node implements functionality to reach the nodes lower/higher in the hierarchy. Usually, the compute nodes in the highest level of the hierarchy, reside in the cloud which is responsible for global coordination [9]. The following paragraphs present three representative platforms which operate using the hierarchical communication type. Other similar approaches can be found in [10]–[14].

Chekired and Khoukhi [15] propose a hierarchical multi-layer fog computing architecture for IoT applications. To realize this architecture, the authors suggest organizing the compute nodes of the network in a tree hierarchy which represents the fog and use the fog in order to minimize latency and processing delay. The basic idea of this approach is to aggregate the peak workloads that exceed the capabilities of low layer compute nodes to nodes with higher resource capacities (i.e., higher in the hierarchy). This enables the handling of larger amounts of IoT devices, especially during peak hours. Along with the architecture, the authors also develop an algorithm for distributing workloads over the layers of the hierarchy. This algorithm takes into account the amount of compute resources which are provisioned to handle each workload and solves an optimization problem in order to minimize the average execution delay.

Sinaeepourfard et al. [16] develop a fog computing architecture for data preservation in the IoT. This architecture exploits the compute nodes of the fog in order to reduce latency while using the cloud for resource demanding computations. The compute nodes in this architecture are organized hierarchically in a multi-layer structure. The top layer of this structure represents the compute nodes in the cloud while the other layers represent the compute nodes of the fog. The number of layers in the fog depends on the amount of participating compute nodes. In this architecture, the authors design a data preservation mechanism for the IoT. According to this mechanism, the data is collected by the nodes of the lowest layer, which reside at the edge. If this data is requested by an application, it is available in real time because of the close proximity. However, since the storage capacity at the edge is limited, the least recent data is periodically transferred upwards the hierarchy.

Skarlat et al. [17] propose FogFrame, a framework for enabling the execution of workloads in the fog. FogFrame organizes the participating resources into a hierarchy of compute nodes. At the top of the hierarchy, there is a cloud compute node. Below the cloud, the compute nodes of the fog follow a tree structure with the IoT devices as the leaves. Between the cloud and the fog, there is a centralized component called the fog controller. This component is responsible for provisioning virtual resources to the compute nodes. Moreover, the fog controller provides an interface for new nodes that want to join the system. In this architecture, a workload can be submitted for execution to any of the compute nodes in the fog. After submission, the workload is distributed among multiple compute nodes at the edge of the network, unless there are not enough compute resources. In this case, the workload is outsourced to the cloud.

In general, based on the hierarchical communication type, the compute nodes of the fog are organized in layers according to resource capacities. This can be beneficial to fog computing platforms, especially for the resource constrained IoT devices. These devices do not have the resources to implement complex mechanisms for executing or forwarding workloads [18]. For this reason, the resource constrained devices are configured

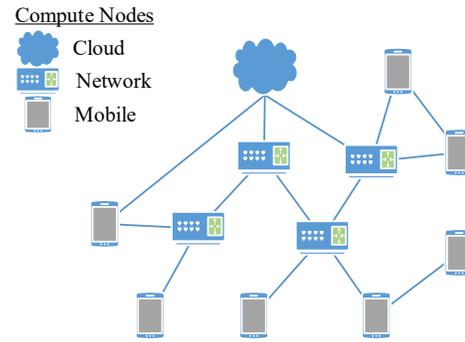


Fig. 2: Peer to peer communication in the fog.

to occupy the leaves of the tree hierarchy acting only as data sources and thus, they are not required to perform resource demanding activities.

III. PEER TO PEER COMMUNICATION IN THE FOG

The P2P communication type refers to organizing the compute nodes of the fog in a P2P manner as shown in Fig. 2. This means that the compute nodes communicate with each other by having a partial view of the network [19]. P2P has shown potential for handling fog infrastructures in a scalable manner and for this reason, P2P mechanisms have found use in fog computing platforms [20]. The following paragraphs present platforms that use the P2P type to organize the compute nodes within the fog. Similar P2P approaches can be found in [21]–[23].

Santos et al. [24] propose a resource discovery service based on Distributed Hash Tables (DHTs) which can be used for dynamic resource provisioning in fog computing. A DHT is used for providing a mechanism to organize the compute nodes in a ring structure, because of the provable performance metrics (e.g., logarithmic performance of node lookup operations). Thus, by using DHTs, the nodes exchange provisioning information about the available compute resources and service level information about the workloads (e.g., allocated amount of bandwidth). Mechanisms for proximity awareness and fault tolerance (which can be useful for fog computing scenarios) when using DHTs have also been proposed in the literature [25]. However, even though this approach provides a messaging system for exchanging provisioning information, no actual provisioning mechanisms are discussed. To assess the proposed approach, the authors examine three alternative P2P protocols (which are based on DHTs), namely: Chord, Kademlia and Pastry.

Tato et al. [26] propose Koala, an overlay network which can be used for organizing all the compute nodes. This overlay targets the decentralization of the cloud by assuming an environment of many small geographically distributed data centers. Koala lowers the protocol overhead by eliminating periodic messages typically used for detecting node failures. Instead, failures are discovered by detecting the nodes that do not respond to the application traffic. Upon the discovery of such nodes, the routing information of the responsive nodes is

updated in order to account for the failures. This mechanism is used for providing fault tolerance to the overlay network. Moreover, this overlay provides proximity aware routing. For this, all compute nodes are organized in a ring structure and each node maintains only a partial view of the system. This view is used for routing by choosing the next hop of each message based on a trade-off between hop count and latency.

Cabrera et al. [27] propose a system model and a P2P implementation for fog computing storage. This storage aims at coping with the increased amount of generated data from the IoT devices by managing the data of the different devices in the fog. To this end, the compute nodes of the fog are organized into a P2P network in order to provide storage with low latency and high throughput. Each compute node communicates with the nodes in proximity using wireless means. In order to provide fault tolerance, the authors propose a mechanism to account for compute nodes that leave/disconnect from the network unexpectedly. According to this mechanism, the data is stored in a distributed manner among the compute nodes of the fog. If one of these nodes becomes unresponsive, the responsive nodes select a leader which gathers the chunks of the data from all the other nodes. Then, the leader reassembles the data using forward error correction codes and redistributes chunks of the data to the remaining nodes. This way, the proposed solution becomes fault tolerant and the data remains available even when compute nodes fail.

P2P is a well established communication type for computing at the edge of the network [28]. For this reason, integrating P2P communication in fog computing platforms ensures tolerance to node failure. However, the organization of P2P networks, assumes that all nodes are equal and does not account for resource heterogeneity.

IV. HYBRID COMMUNICATION IN THE FOG

The hybrid communication type refers to organizing the compute nodes of the fog using both the hierarchical and the P2P types as shown in Fig. 3. In this case, the compute nodes are still organized in layers. However, the nodes of each layer may communicate with each other using P2P or device to device connectivity, i.e., without requiring the presence of an intermediate node [29]. This can be useful for the case that different parts of the fog have different requirements. For instance, dynamic compute nodes at the edge may have different requirements for fault tolerance than the stable compute nodes of the cloud. The following paragraphs present fog computing platforms which implement the hybrid communication type. Similar approaches can be found in [30]–[33].

Lee et al. [34] propose a framework for forming fog networks in order to outsource workloads from IoT applications. To this end, the authors formulate an optimization problem with the objective to select the neighbor nodes of each compute node dynamically, so that outsourcing workloads to the neighbors results in minimum computational latency. To solve this problem, the compute nodes of the fog form connections with each other and communicate in a device to device manner. However, the overall architecture of the system is organized

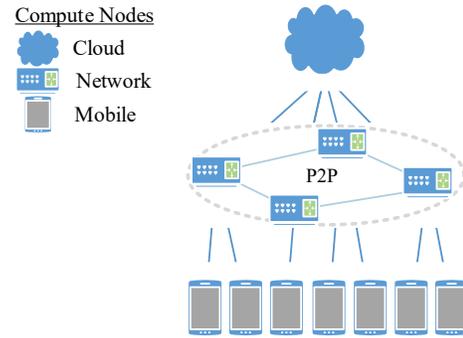


Fig. 3: Hybrid communication in the fog.

hierarchically using three layers. The bottom layer contains resource constrained IoT devices which outsource workloads to the layer higher in the hierarchy, i.e., the fog layer. Each compute node of the fog layer shares the workload with the neighbor nodes in proximity, but can also outsource workloads to the cloud layer which is located at the top of the hierarchy.

Chen et al. [35] present a framework for outsourcing workloads from mobile applications towards the cloud. This framework offers the flexibility of selecting whether a workload should be: *i)* executed locally on the mobile compute node, *ii)* outsourced and shared among compute nodes in proximity or *iii)* outsourced towards compute nodes in the cloud. Therefore, this framework is based on a hierarchical three-layer architecture with the mobile devices at the bottom, a layer of fog compute nodes in the middle and the cloud at the top. However, the workloads are outsourced based on an optimization problem that minimizes the execution cost. According to the solution of this problem, workloads can be outsourced on the path of the hierarchy towards the cloud or on a path among the compute nodes of the fog, which communicate with each other in a device to device manner. For solving this optimization problem, the authors develop two heuristics (based on the random/greedy concept) which show efficient outsourcing of workloads in fog computing environments.

Fu et al. [36] design a framework for providing reliable storage based on fog computing in order to deal with the increased amount of data stemming from the IoT. This framework integrates an architecture with different kinds of compute nodes, i.e., edge servers, proxy servers and cloud servers. In this architecture, the data from the IoT is first sent to the edge servers. The edge servers transform the data into a unified representation and fuse them at the feature level. Then, the data is sent to the proxy servers which encrypt the data and make it suitable for storing in the cloud. To access the data from the cloud, the user sends the trapdoor of the query to the cloud server which uses the trapdoor to search for the encrypted data. This way, secure cloud storage is achieved by outsourcing the necessary encryption workloads to the compute nodes of the fog. Notably, even though each compute node performs specific operations, the edge servers also communicate with each other, making it possible to share the workloads in a

P2P or similar manner.

The hybrid communication type attempts to combine hierarchical organization with P2P elements. To this end, this communication type supports resource heterogeneous compute nodes due to the inherent hierarchical structure. However, fault tolerance mechanisms are no longer supported because the hybrid approaches use simple P2P connections instead of fully fledged P2P protocols.

V. DISCUSSION AND RESEARCH CHALLENGES

After presenting representative fog computing platforms for each communication type (cf. Sections II–IV), this section discusses differences and research challenges based on the reviewed literature. To compare the types of communication in the fog, a set of criteria is proposed, based on requirements of fog computing. Specifically, these criteria are:

- **Resource heterogeneity.** The fog includes a plethora of resource heterogeneous devices that need to be configured and maintained [37]. This criterion denotes if a platform considers this heterogeneity in the organization of the fog.
- **Provisioning mechanism.** This criterion shows if a platform includes a mechanism to provision virtual resources for providing automatic resource management [38].
- **Duplex outsourcing.** Fog computing platforms implement scheduling algorithms for outsourcing workloads on compute nodes across the system [7]. This criterion denotes if a platform allows duplex communication for potentially enabling compute nodes to outsource workloads in a duplex manner, e.g., from the edge to the cloud and also from the cloud to the edge.
- **Proximity awareness.** Fog computing uses the proximity between data source and compute node to reduce latency [39]. This criterion shows if a platform integrates a measure (e.g., hop count, latency) to determine proximity.
- **Fault tolerance.** This criterion shows if a platform is able to maintain a fog network connected when compute nodes fail unexpectedly. This is a serious challenge in fog computing because faults may occur at any time [40].

According to these criteria, Table I is presented in order to show the differences among the reviewed platforms. Based on this table, the hierarchical communication copes well with heterogeneous devices. This happens because the hierarchical type utilizes different layers for different devices. Moreover, the hierarchical type is preferable for developing provisioning mechanisms. This may occur due to the simplicity of the hierarchical design whereby a compute node provisions virtual resources on the predecessors/successors. However, due to the assumption that the nodes of the immediate layers are in close proximity, additional proximity measures are often not utilized. Fault tolerance mechanisms are also neglected because the IoT devices that provide unstable resources often reside at the leaves of the tree hierarchy. Thus, having nodes that fail as leaves, does not divide a tree into disjoint parts.

Regarding the P2P communication type, there is inherent support (from P2P systems) for fault tolerance and proximity awareness. However, since the P2P approach considers all

TABLE I: Comparison of reviewed literature.

	Resource heterogeneity	Provisioning mechanism	Duplex outsourcing	Proximity awareness	Fault tolerance
<i>Hierarchical</i>					
Chekired et al. [15]	✓	✓			
Sinaeep et al. [16]	✓				
Skarlat et al. [17]	✓	✓		✓	
<i>Peer to peer</i>					
Santos et al. [24]			✓	✓	✓
Tato et al. [26]			✓	✓	✓
Cabrera et al. [27]				✓	✓
<i>Hybrid</i>					
Lee et al. [34]	✓		✓	✓	
Chen et al. [35]	✓			✓	
Fu et al. [36]	✓				

nodes as equal, node heterogeneity is often not supported. On the contrary, the hybrid communication which attempts to combine the other two types, is usually able to support the resource heterogeneity of the nodes because of the layered structure that aligns with the hierarchical. In addition, the hybrid type may enable proximity awareness because the devices within a layer communicate with each other in a P2P or device to device manner.

In general, Table I shows that each communication type meets different criteria and thus, the communication type of a platform affects the functionality of the final applications. However, since the proposed criteria are based on generic fog/edge computing requirements, all platforms can benefit from additional mechanisms to satisfy the unmet criteria. For this reason, the following research challenges are identified:

- 1) *How to integrate fault tolerance mechanisms from P2P into the hierarchical communication type?*
- 2) *How to extend the P2P communication type to account for resource heterogeneous compute nodes?*
- 3) *How to design provisioning mechanisms for the hybrid and the P2P communication types?*

These challenges refer to combining the hierarchical type with P2P communication in an efficient manner. Even though P2P was not originally designed for fog computing, this type can aid in realizing the vision of the fog because many P2P mechanisms are also applicable to fog computing [20], [37].

VI. CONCLUSION

The emergence of fog and edge computing have given rise to many computing platforms with compute nodes that communicate with each other using different communication types. For this reason, this paper conducts a review of key literature on fog computing platforms, which focuses on compute node communication. Moreover, the paper at hand proposes a set of criteria based on fog computing requirements, which can be applied to the computing platforms. By applying the criteria to the reviewed literature, the following conclusions are drawn:

- i) The communication type of a fog computing platform should be determined according to the requirements of the final application because each type meets different criteria.

ii) Based on the criteria met by each communication type, research challenges that may advance future fog computing platforms can be identified.

REFERENCES

- [1] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric computing: Vision and challenges," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 5, pp. 37–42, 2015.
- [2] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the internet of things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, 2016.
- [3] C. Esposito, A. Castiglione, F. Pop, and K.-K. R. Choo, "Challenges of connecting edge and cloud computing: a security and forensic perspective," *IEEE Cloud Computing*, no. 2, pp. 13–17, 2017.
- [4] V. Karagiannis and A. Papageorgiou, "Network-integrated edge computing orchestrator for application placement," in *13th International Conference on Network and Service Management (CNSM)*, pp. 1–5, IEEE, 2017.
- [5] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," *ICAS Transaction on IoT and Cloud Computing*, vol. 3, no. 1, pp. 11–17, 2015.
- [6] L. Peng, A. R. Dhaini, and P.-H. Ho, "Toward integrated Cloud-Fog networks for efficient IoT provisioning: Key challenges and solutions," *Future Generation Computer Systems*, vol. 88, pp. 606–613, 2018.
- [7] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2017.
- [8] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of everything*, pp. 103–130, Springer, 2018.
- [9] P. Varshney and Y. Simmhan, "Demystifying fog computing: Characterizing architectures, applications and abstractions," in *1st IEEE International Conference on Fog and Edge Computing (ICFEC)*, pp. 115–124, IEEE, 2017.
- [10] V. B. C. d. Souza, W. Ramírez, X. Masip-Bruin, E. Marín-Tordera, G. Ren, and G. Tashakor, "Handling service allocation in combined fog-cloud scenarios," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–5, IEEE, 2016.
- [11] H. Shah-Mansouri and V. W. Wong, "Hierarchical fog-cloud computing for iot systems: A computation offloading game," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3246–3257, 2018.
- [12] D. A. Chekired, L. Khoukhi, and H. T. Mouftah, "Industrial iot data scheduling based on hierarchical fog computing: A key for enabling smart factory," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 4590–4602, 2018.
- [13] C.-L. Tseng and F. J. Lin, "Extending scalability of iot/m2m platforms with fog computing," in *4th IEEE World Forum on Internet of Things (WF-IoT)*, pp. 825–830, IEEE, 2018.
- [14] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner, "Optimized iot service placement in the fog," *Service Oriented Computing and Applications*, vol. 11, no. 4, pp. 427–443, 2017.
- [15] D. A. Chekired and L. Khoukhi, "Multi-tier fog architecture: A new delay-tolerant network for iot data processing," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2018.
- [16] A. Sinaeepourfard, J. Garcia, X. Masip-Bruin, and E. Marín-Tordera, "Data preservation through fog-to-cloud (f2c) data management in smart cities," in *2nd IEEE International Conference on Fog and Edge Computing (ICFEC)*, pp. 1–9, IEEE, 2018.
- [17] O. Skarlat, V. Karagiannis, T. Rausch, K. Bachmann, and S. Schulte, "A framework for optimization, service placement, and runtime operation in the fog," in *11th IEEE/ACM International Conference on Utility and Cloud Computing (UCC)*, pp. 164–173, IEEE, 2018.
- [18] O. Bello, S. Zeadally, and M. Badra, "Network layer inter-operation of device-to-device communication technologies in internet of things (iot)," *Ad Hoc Networks*, vol. 57, pp. 52–62, 2017.
- [19] G. Tato, M. Bertier, and C. Tedeschi, "Designing overlay networks for decentralized clouds," in *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 391–396, IEEE, 2017.
- [20] A. Lebre, J. Pastor, A. Simonet, and F. Desprez, "Revising openstack to operate fog/edge computing infrastructures," in *2017 IEEE International Conference on Cloud Engineering (IC2E)*, pp. 138–148, IEEE, 2017.
- [21] F. Rizzo, G. L. Spoto, P. Brizzi, D. Bonino, G. Di Bella, and P. Castrogiovanni, "Beekup: A distributed and safe p2p storage framework for ioe applications," in *20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, pp. 44–51, IEEE, 2017.
- [22] S. Aditya and R. J. Figueiredo, "Frugal: Building degree-constrained overlay topology from social graphs," in *1st IEEE International Conference on Fog and Edge Computing (ICFEC)*, pp. 11–20, IEEE, 2017.
- [23] L. A. Steffelen and M. K. Pinheiro, "Improving data locality in p2p-based fog computing platforms," *Procedia Computer Science*, vol. 141, pp. 72–79, 2018.
- [24] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Towards dynamic fog resource provisioning for smart city applications," in *14th International Conference on Network and Service Management (CNSM)*, pp. 290–294, IEEE, 2018.
- [25] Y. Zhu and Y. Hu, "Efficient, proximity-aware load balancing for dht-based p2p systems," *IEEE Transactions on parallel and distributed systems*, vol. 16, no. 4, pp. 349–361, 2005.
- [26] G. Tato, M. Bertier, and C. Tedeschi, "Koala: Towards lazy and locality-aware overlays for decentralized clouds," in *2nd IEEE International Conference on Fog and Edge Computing (ICFEC)*, pp. 1–10, IEEE, 2018.
- [27] J. A. Cabrera G, D. E. Lucani R., and F. H. P. Fitzek, "On network coded distributed storage: How to repair in a fog of unreliable peers," in *2016 International Symposium on Wireless Communication Systems (ISWCS)*, pp. 188–193, IEEE, 2016.
- [28] N. M. Gonzalez, W. A. Goya, R. de Fatima Pereira, K. Langona, E. A. Silva, T. C. M. de Brito Carvalho, C. C. Miers, J.-E. Mångs, and A. Sefidcon, "Fog computing: Data analytics and cloud distributed processing on the network edges," in *35th International Conference of the Chilean Computer Science Society (SCCC)*, pp. 1–9, IEEE, 2016.
- [29] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with wi-fi direct: overview and experimentation," *IEEE wireless communications*, vol. 20, no. 3, pp. 96–104, 2013.
- [30] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: Towards minimizing delay in the internet of things," in *2017 IEEE International Conference on Edge Computing (EDGE)*, pp. 17–24, IEEE, 2017.
- [31] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 26–35, 2017.
- [32] A. Kapsalis, P. Kasnesis, I. S. Venieris, D. I. Kaklamani, and C. Z. Patrikakis, "A cooperative fog approach for effective workload balancing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 36–45, 2017.
- [33] D. Nguyen, Z. Shen, J. Jin, and A. Tagami, "Icn-fog: An information-centric fog-to-fog architecture for data communications," in *2017 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2017.
- [34] G. Lee, W. Saad, and M. Bennis, "An online secretary framework for fog network formation with minimal latency," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2017.
- [35] X. Chen and J. Zhang, "When d2d meets cloud: Hybrid mobile task offloadings in fog computing," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2017.
- [36] J. Fu, Y. Liu, H.-C. Chao, B. Bhargava, and Z. Zhang, "Secure data storage and searching for industrial iot by integrating fog computing and cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4519–4528, 2018.
- [37] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.
- [38] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big data and internet of things: A roadmap for smart environments*, pp. 169–186, Springer, 2014.
- [39] B. Cheng, A. Papageorgiou, and M. Bauer, "Geelytics: Enabling on-demand edge analytics over scoped data sources," in *2016 IEEE International Congress on Big Data (BigData Congress)*, pp. 101–108, IEEE, 2016.
- [40] Y. Liu, J. E. Fieldsend, and G. Min, "A framework of fog computing: Architecture, challenges, and optimization," *IEEE Access*, vol. 5, pp. 25445–25454, 2017.