Providing overlay-based multicast in data center networks with optional millimeter wavelength links

Yi Wang, Bingquan Wang, Chen Tian & Fu Xiao

Telecommunication Systems
Modelling, Analysis, Design and Management

ISSN 1018-4864
Volume 73
Number 1

Telecommun Syst (2020) 73:95-104
DOI 10.1007/s11235-019-00601-8

Springer

Telecommunication Systems (2020) 73:95–104
https://doi.org/10.1007/s11235-019-00601-8

Providing overlay-based multicast in data center networks with optional millimeter wavelength links

Yi Wang[1] · Bingquan Wang[2] · Chen Tian[2] · Fu Xiao[3]

Published online: 26 July 2019

Abstract
Efficient multicast support is very important in data center (DC) networks, as many big data systems in data centers are bandwidth-hungry. Multicast implementations in DC networks are usually overlay-based. The major challenges are how can we accurately infer the topology of DC networks with both wired and wireless millimeter wavelength (MMW) links and how to design multicast algorithm for these topologies. In this paper, we first use hierarchical clustering to accurately infer the topology. Then, we propose the Inter-Rack First Multicast (IRFM) algorithm to match the fan-in nature of DC topologies. Evaluation results demonstrate that IRFM is 3.7−11.2× faster than naive multicast implementations in the pure wired case, and 4.8−14.6× faster in the case of MMW enhancement.

Keywords Data center networks · Millimeter wave · 60 GHz wireless · Multicast

1 Introduction

Efficient multicast (i.e., one-to-many communication) support is very important in data center (DC) networks [1–3]. In nowadays big data systems, more and more applications need to multicast massive amounts of data to other nodes. These applications include: publish-subscribe services for data dissemination [4], web cache updates [5], scatter-gather by iterative optimization algorithms [6] as well as fragment-replicate joins in Hadoop [7]. Therefore, efficient multicast algorithm can significantly improve the performance of applications.

✉ Chen Tian
tianchen@nju.edu.cn

Yi Wang
151485321@qq.com

Bingquan Wang
wangbingquan@smail.nju.edu.cn

Fu Xiao
xiaof@njupt.edu.cn

1 School of Modern Posts, Nanjing University of Posts and Telecommunications, Nanjing 210046, China

2 State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

3 School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210046, China

Multicast implementations in DC networks are usually overlay-based. As we know, L2 or L3 multicast supports are usually disabled for management reasons in DC networks [8]. So, we target overlay-based multicast algorithm. An intuitive option is to mimic BitTorrent-like protocols [9] which accelerate the dissemination process by split the data to chunks and perform chunk-level scheduling [10]. This approach severely increases CPU load which could have been used for more important computations in DC [11]. Instead, we let the data transferred as a whole from one node to another.

There are two major challenges. First, how can we accurately infer the topology of DC networks? The topology of DC networks is very important for overlay-based multicast algorithms. Common wired architectures are tree [12], leaf-spine [13] and fat-trees [14] or Clos networks [15,16]. However, Millimeter wavelength wireless technology (MMW) [17–19] is used to further accelerate DC networks [20–22] in recent years. These wireless links constantly change by Micro-Electro-Mechanical Systems (MEMS). As a result, The network topology becomes so dynamic that how to accurately infer the topology of DC networks is a challenge. Second, how to design multicast algorithms for these topologies.

In this paper, we target overlay-based multicast for big data systems. The scheduling objective of our multicast algorithm is to minimize the average *multicast completion time* (MCCT). We make three contributions. First, we use

hierarchical clustering to accurately infer the topology in DC networks even in the case of a hybrid of wired and wireless. Second, we propose the Inter-Rack First Multicast (IRFM) algorithm to match the fan-in nature of DC topologies and derive optimal parameter settings for two different scenarios. Third, we conduct large-scale NS3 simulations to evaluate the performance of our algorithm. The results show that our multicast scheme (IRFM) is 3.7−11.2× faster than naive multicast implementations in the pure wired case, and 4.8−14.6× faster in the case of millimeter wavelength wireless technology enhancement.

The rest of this paper is organized as follows. Section 2 discusses background and related works. Section 3 presents the design of our system. Then we discuss topology inference and Inter-Rack First Multicast (IRFM) in Sects. 4 and 5 respectively. Finally, we conclude in Sect. 6.

2 Background and intuition

2.1 60 GHz links in the DC networks

In recent years, the millimeter wavelength wireless technology (MMW) has gradually matured. Spectrum between 57 and 64 GHz, colloquially known as the 60 GHz band, is available world-wide for unlicensed use [23]. There are more and more DC adopt 60 GHz wireless technologies [24–26]. The 60 GHz wireless links can afford multi-Gbps data rates for DC. Furthermore, 60 GHz devices with directional antennas can be deployed densely which is suitable for DC networks, because the signal attenuates rapidly due to the high frequency [23]. And the most important things are the low power characteristics and adjustable direction of wireless links [27,28]. Wireless links not only reduce a lot of the hassle of wiring, but they are also more flexible [29]. MMW communication technology is complementary to traditional wired DC networks [30].

2.2 Overlay-based multicast in DC networks

Dan et al. [31] focus on routing strategy to prevent multicast loops. However, our goal is to minimize the average *multicast completion time* (MCCT). And we need to take note of the fact that most DC operators do not enable multicast features in their networks for scalability and stability reasons.

Orchestra manages data transfers in computer clusters [32]. For multicast transfers, Orchestra implements an optimized BitTorrent-like protocol called Cornet, augmented by an adaptive clustering algorithm to take advantage of the hierarchical network topology in many data centers. BitTorrent-like protocol splits data into blocks and subdivides blocks into small chunks. This approach adds burden on CPU when transfer large data simultaneously. In addition,

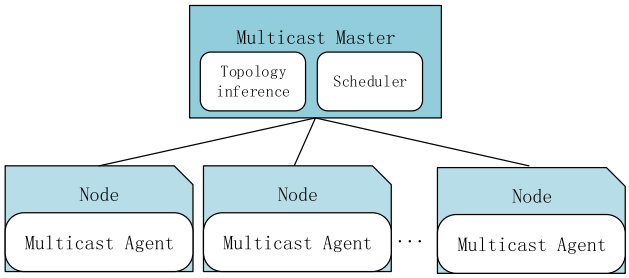

Fig. 1 System architecture

orchestra did not consider the architecture of the wired and wireless hybrid DC networks in which the changing wireless antenna direction may affect network topology seriously.

2.3 Other related

FreeFlow [33] implements a container networking library to make communication more efficient no matter which network API set is used. The authors consider the scenario that both sender and receiver containers reside on the same host. FreeFlow uses shared memory to accelerate the data transmission. Shared memory is a particularly efficient way to transfer data between two nodes on the same host. Although it is possible that the multicast source node and the destination node may be on the same host during the multicast process. In this paper, we only consider that sender and receiver reside on different hosts and leave the above scenario to future work.

Although the topologies vary with the orientation of the MMW wireless antenna in wired and wireless hybrid DC, we can still infer almost real-time topology information. Based on the topology of DC networks, we target a more efficient overlay-based multicast.

3 System design

To manage and optimize data transfers in the whole multicast process, we design the multicast system as shown in Fig. 1. The key idea of the system is to separate the multicast control plane from the data plane, including the multicast master control node and the slave data transmission node. Each slave node has a multicast agent. The multicast API is *Bool multicast(source node, multicast group, data)*. An application simply calls the API. The multicast agent sends the request to a centralized multicast master node. The master node has a topology inference module and a multicast scheduler module. Since the direction of the MMW wireless antenna changes constantly with the change of the DC workload, so the topology inference module infers the DC networks topology among nodes via historical transfer throughput among nodes periodically (Sect. 4.1).

The scheduler coordinates the whole multicast process. Based on the network topology, it instructs the source node to send all data to one or several destination nodes first. When a destination node receives all data, it becomes a *seed*. The scheduler then instructs the seed to transfer data to other destination nodes on behalf of the original source. When the source or a seed finish one transfer, it might be scheduled with another destination. So on and so forth, until all destination nodes receive the data. The scheduling algorithm of the scheduler is shown in Sect. 5.

4 Topology inference

4.1 Design philosophy

Many data centers employ hierarchical network topologies with oversubscription [15]. For the same amount of data, the transfer time between two nodes in different racks (i.e., inter-rack communication) is much higher than that of two nodes in the same rack (i.e., intra-rack communication).

The MMW wireless antenna on the top of the rack can alleviate the hot spots between different racks to a certain extent because MMW wireless links increase the bandwidth between the racks. However, the MMW wireless links do not affect the wired topology of the DC networks. In this paper, we call the wired topology the physical topology (rack level). Without loss generality, we assume there is a wireless link between rack 1 and rack 2, then the bandwidth between rack 1 and rack 2 will be increased. From a hierarchical topology perspective, these two racks can be viewed as a *super-rack*. We can use hierarchical clustering to accurately infer the physical topology (rack level) and the *super-rack* topology (*super-rack* level).

With topology information, the scheduler module can coordinate the multicast process more efficient [34]. For this reason, we implement a topology inference module, similar to previous work [32], but we added hierarchical clustering to adapt to the MMW wireless environment between racks.

The steps for topology inference are as follows:

- We use (historical) node to node transfer throughput records to construct an $n \times n$ sparse distance matrix D, where n is the number of nodes in the topology, and the entries are the median transfer throughput between two nodes [35,36].
- Missing entries are inferred in the distance matrix using non-negative matrix factorization procedure of Mao and Saul [37]. Non-negative matrix factorization (NMF) is linear dimensionality reduction that can be applied to the distance matrix D.
- After completing the matrix D, we project the nodes onto a two-dimensional space using non-metric multidimensional scaling, or NMDS [38]. The goal of NMDS is to collapse information from multiple dimensions into just a few, so that they can be visualized and interpreted.
- Finally, we cluster nodes using a mixture of spherical Gaussian. By setting different thresholds, we can get the physical topology (rack level) and the *super-rack* topology (*super-rack* level).

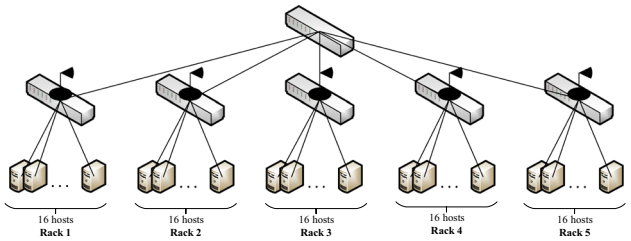

Fig. 2 The simulated data center topology

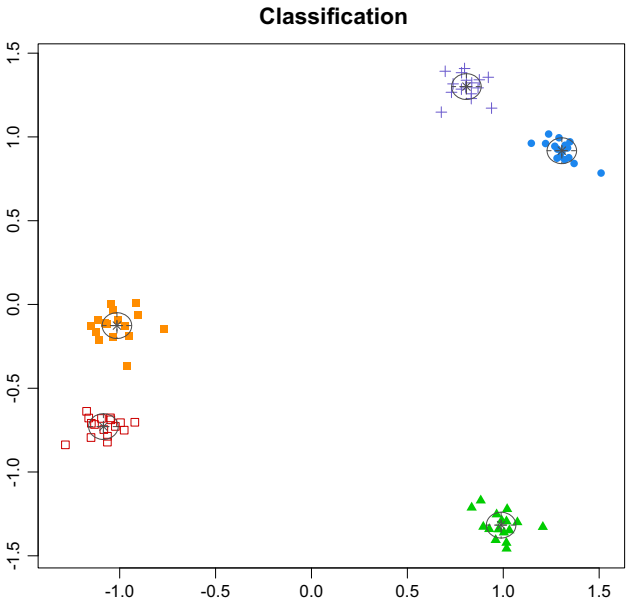

Fig. 3 The classification of the nodes in the data center networks without background flows from the perspective of rack level

With enough training data, the above topology inference algorithm can infer the network topology accurately, as we will show in Sect. 4.2. We expect this implementation can be easily extended to topologies with more than two switch layers.

4.2 Topology inference evaluation

The simulated topology we adopt is shown in Fig. 2. There are total 80 hosts, with 16 hosts in each rack. All 5 racks switches connect to 1 spine switch. Also, there is a directional antenna on the top of every rack, which can provide 5 Gbps bandwidth on a 60 GHz carrier wave. Every wired link has a bandwidth

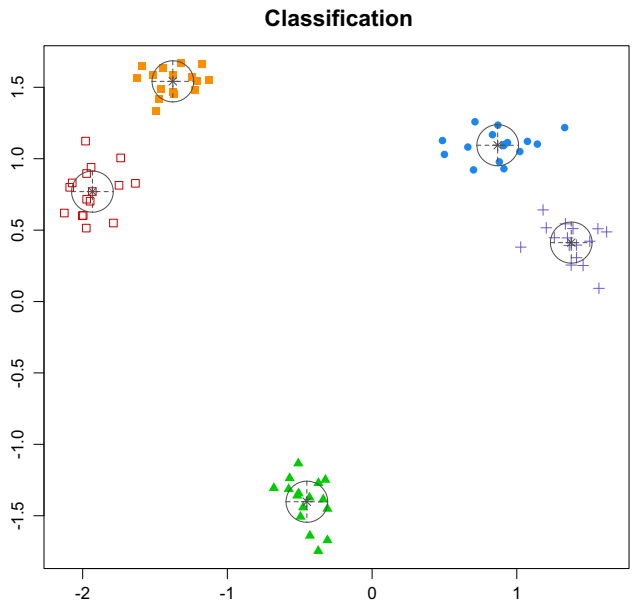

Fig. 4 The classification of the nodes in the data center networks with 80 random background flows from the perspective of rack level

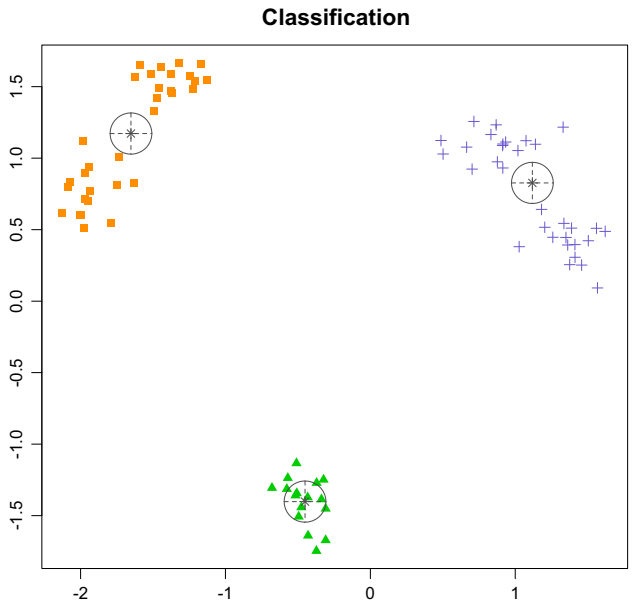

Fig. 6 The classification of the nodes in the data center networks with 80 random background flows from the perspective of *super-rack* level

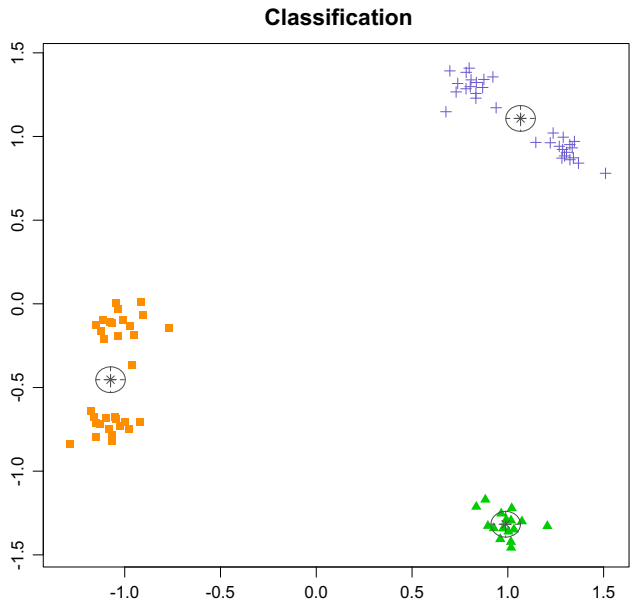

Fig. 5 The classification of the nodes in the data center networks without background flows from the perspective of *super-rack* level

of 10 Gbps and the link delay is 1 us. The direction of the MMW wireless antenna varies with the DC workload so it is out of our control.

Without loss of generality, we assume that rack 1 and rack 2, rack 3 and rack 4 are connected by wireless links. Rack 5 does not turn on wireless. We randomly generate $80 \times 80 \times 0.5$ data flows, and the size of each flow is 1 MB. We record the transfer time of each flow, so that matrix D is 50% full. As mentioned before, we infer the missing data in the matrix using the non-negative matrix factorization. We project the nodes onto a two-dimensional space [38] using the *isoMDS* function of the MASS library in R. Here we use *mclust* which is an R package for normal mixture modeling via EM, model-based clustering, classification, and density estimation [39,40].

The results, without and with concurrent background flows, are shown in Figs. 3 and 4 respectively. The ellipses in the figure represent the inferred clusters. The different shapes in the figure represent different racks. From Figs. 3 and 4, we can see that our topology inference algorithm can accurately infer the rack level network topology. However, as shown in Figs. 5 and 6, we found that the two racks (rack 1 and rack 2) in the bottom left corner of the figure can be classified into one cluster and regarded as a *super-rack* by changing the threshold for higher level clustering. The other two racks (rack 3 and rack 4) in the top right corner is also a *super-rack* and the rack 5 in the bottom right corner is a *super-rack* on its own. Through hierarchical clustering, we can get the topologies of different levels in the DC networks which reflects the MMW wireless links between the racks. From Figs. 5 and 6, we can see there are 3 *super-racks* in the topology compared to the 5 racks in Figs. 3 and 4.

We can conclude that whether or not there are background flows, the algorithm can infer the physical topology and the *super-rack* topology of the network accurately. Since the nodes in a *super-rack* are very close to each other, the *super-rack* is then taken as a whole and called rack in the following for the sake of simplicity.

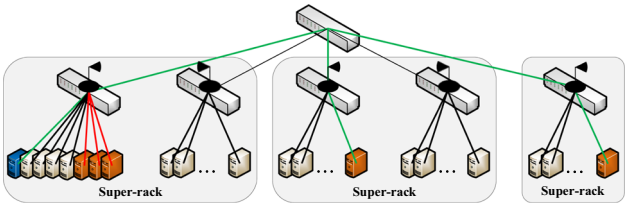

Fig. 7 The Inter-Rack First Multicast (IRFM) algorithm

5 Multicast algorithm

5.1 Inter-Rack First Multicast

Inter-Rack First Multicast (IRFM) leverages the fact that the transfer times between two nodes in different racks is significantly higher than between nodes in the same rack. At the beginning of multicast transmission, IRFM first transmit data to the nodes in other racks. This approach lets each rack to have a copy of data within the shortest time. Then the data can be transferred inside each rack, which not only increases parallelism and can reduce MCCT greatly, but also free up the bandwidth between racks which can do something more valuable.

We also need to fully utilize the bandwidth of uploading nodes. Inter-rack links can be congested. We propose two solutions to solve this problem. First, we use the MMW wireless antenna on the top of racks to increase the bandwidth between the racks. Second, when we perform inter-rack transfer, we should transfer data to intra-rack nodes at the same time. This scheme can exploit the remaining upload bandwidth of source/seeds.

The Inter-Rack First Multicast (IRFM) algorithm is shown in Fig. 7, and the details of the IRFM is shown in Algorithm 1.

Algorithm 1 Inter-Rack First Multicast

Step 1. At the beginning of multicast transmission, the multicast source node (blue) transfers data to m intra-rack (orange in the same *super-rack*) and n inter-rack (orange in the different *super-rack*) destination nodes at the same time.
Step 2. As long as there has a node which have completed the data transmission, it becomes a seed and performs the same action as Step 1 until all racks have at least one copy of the data.
Step 3. Multicast source node and all seeds only transfer data to intra-rack nodes until the whole multicast process is completed.

5.2 Guidelines for choosing parameters

In this section, we discuss the parameter selection under different conditions. The first is *Exclusive*, where a single multicast can use the whole network bandwidth. There is no concurrent multicast sessions and background flows. This scenario is for analysis and comparison purpose only. The second is *Realistic*, where multicast sessions can be parallel and there exist other background flows. In all scenarios, we give inter-rack traffic higher priority than intra-rack traffic.

Table 1 Different multicast algorithms

Multicast algorithms	Description
Collateral multicast (CM)	The multicast source node transfer data to multicast destination nodes simultaneously
Sequential multicast (SM)	The multicast source node transfer data to multicast destination nodes sequentially
Random multicast (RM)	The multicast source node transfer data to a multicast destination node randomly, then the node transfer data to another node randomly when the node complete, no matter source or destination node
Inter-Rack First Multicast (IRFM)	The multicast source node transfer data to m intra-rack and n inter-rack destination nodes at the same time in the pure wired case
Inter-Rack First Multicast with Wireless (IRFMW)	The multicast source node transfer data to m intra-rack and n inter-rack destination nodes at the same time in the case of MMW enhancement

5.2.1 Exclusive scenario

In this scenario, we prove that the optimal parameter setting is that the values of m and n are both 1. We prove it as following.

Without loss generality, we assume the transfer proceeds in rounds. We assume that the multicast source node transfers data to k other racks at the first round. We assume that the time between two nodes' transmission is 1 time unit if without competition. As there are no background flows, transfer to k nodes will share the bandwidth equally and the transmission time is k. When the first round completes, there are $k+1$ nodes have the data, i.e., one source and k seeds. After the second round, there are $(k+1)^2$ nodes have the data, and so on and so forth. Thus, if we want to multicast data to N different rack in the shortest time, we can get the equation:

$$(k+1)^t = N, \tag{1}$$

where t represent the total transfer rounds we need. Hence, to multicast data to all N nodes we need

$$T = \log_{k+1} N \cdot k. \tag{2}$$

From Eq. 2, we can discover that when the total multicast nodes N is a constant value, the larger the number of concurrent transmission k, the longer the *multicast completion*

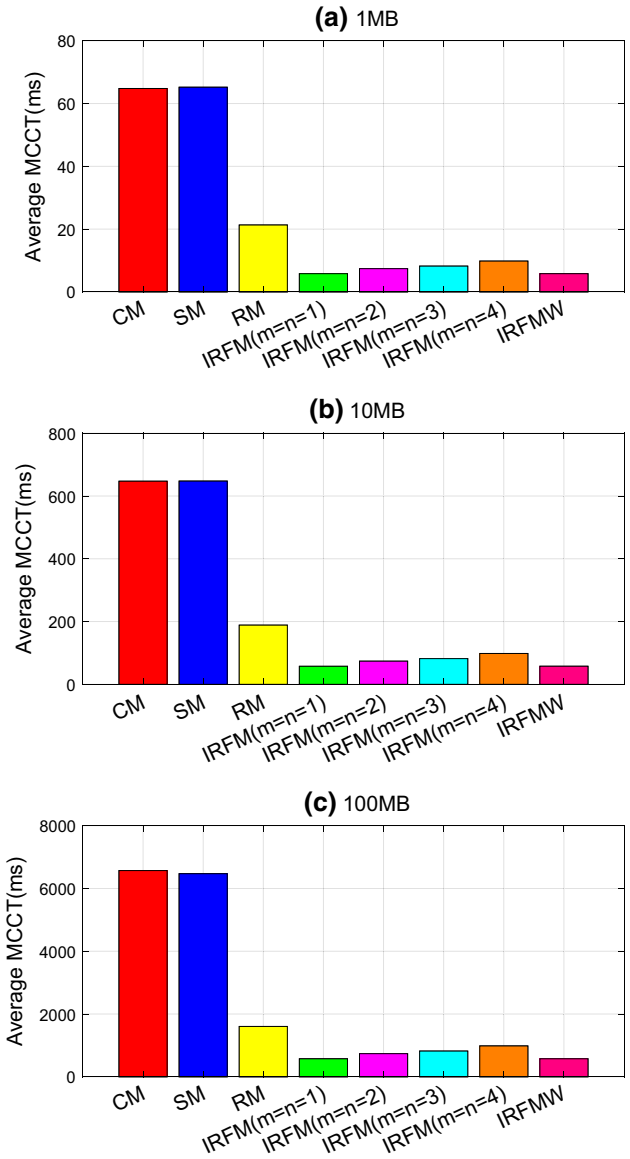

Fig. 8 The average MCCT without background flows with a single multicast

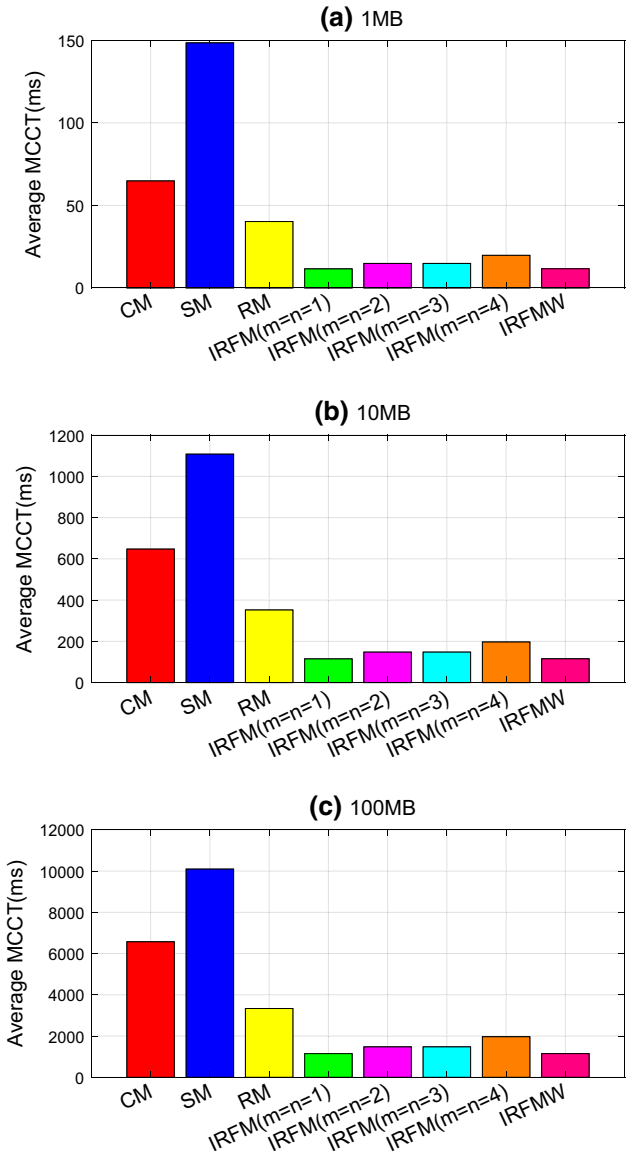

Fig. 9 The average MCCT without background flows with two concurrent multicasts

time (MCCT) T. It is trivial to extend the analysis to intra-rack transfers. In conclusion, when there are no background flows, the optimal parameters for IRFM is that the values of the m and n are both 1.

5.2.2 Realistic scenario

In this case, we assume that all background flows and multicast flows share the bandwidth. Since now background flows and inter-rack multicast flows are equally sharing the bandwidth, we choose the value of n as large as possible to get bandwidth as much as possible. We make n equal to the number of the multicast destination nodes racks minus 1 (i.e., except the rack with the source node).

For intra-rack parameter m, we adopt an adaptive strategy. A source/seed monitors its uplink bandwidth utilization periodically. Once it found there are idle bandwidth, which results from the congested inter-rack links, the host increases one multicast transfer to another node in the same rack.

5.3 Inter rack first multicast (IRFM) evaluation

In this section, we conduct large-scale NS3 simulations to evaluate the performance of IRFM. The topology of the network we used in NS3 simulations is shown in Fig. 2. The network settings in simulation are the same as in Sect. 4.2.

Several heuristic algorithms are also evaluated for comparison, as shown in Table 1. We implement the above

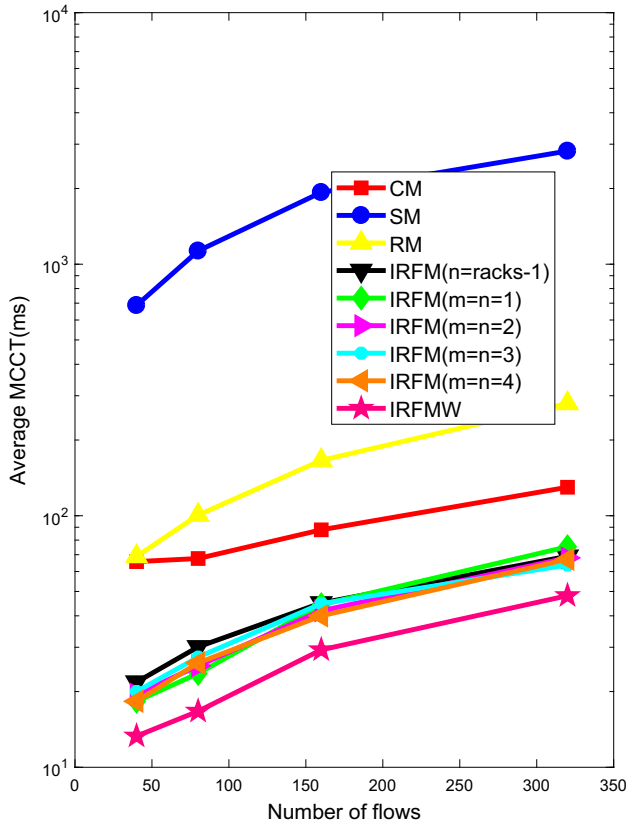

Fig. 10 The average MCCT with background flows, 1 MB, 1 multicast

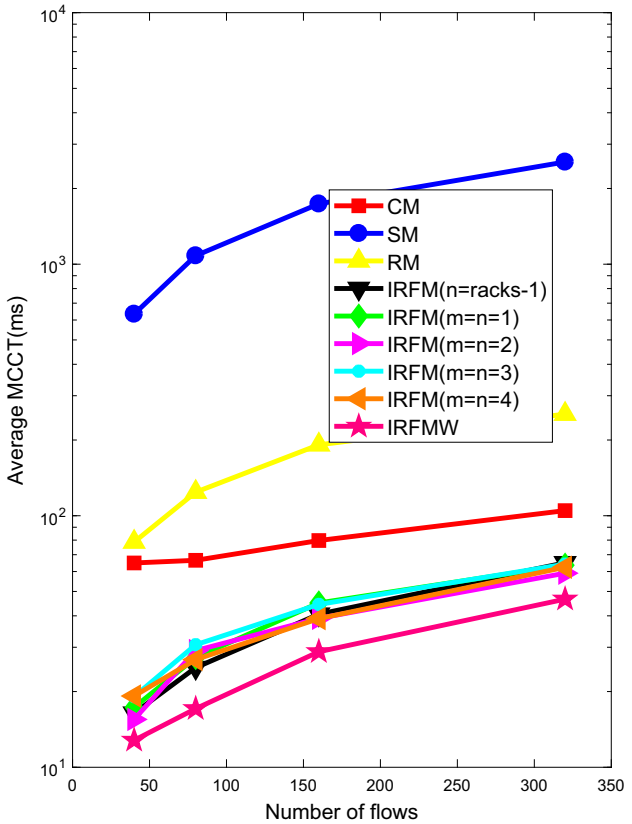

Fig. 11 The average MCCT with background flows, 1 MB, 2 multicasts

algorithms in NS3 simulations under exclusive and realistic scenarios discussed in Sect. 5.2. The multicast settings(e.g., multicast source node and multicast destination nodes) are the same in those algorithms. We compare the MCCT of different multicast algorithms.

5.3.1 Multicast in exclusive case

When there is only one host multicast data to other 79 hosts, Fig. 8 shows the *multicast completion time* (MCCT) of different multicast algorithms to transfer 1 MB (Fig. 8a), 10 MB (Fig. 8b) and 100 MB (Fig. 8c), respectively.

As shown in Fig. 8, IRFM is superior to other multicast algorithms. IRFM is 3.7−11.2× faster compared with other algorithms. For IRFM, the parameters m and n are both 1, as we proved in Sect. 5.2. This setting is 1.3−1.7× faster compared with other parameters. Because the bandwidth between the racks is not the bottleneck in exclusive case, so the results under MMW enhancement (IRFMW) is the same as it was when we didn't use it.

We also test when there are two hosts in different racks and each would multicast data to other 79 hosts. The results are shown in Fig. 9. It's similar to Fig. 8 except the average MCCT of CM. The reason is that the only bottleneck of CM is the uplink of the source node. Because two multicast source hosts are in different racks, so they are independent to each other and the average CM MCCT is the same as that in Fig. 8. For other multicast algorithms, the two nodes multicast data by sharing bandwidth, so the average MCCT is doubled.

5.3.2 Multicast in realistic case

Figure 10 shows that the proposed multicast algorithm IRFM can reduce MCCT greatly even when there exist background flows. IRFM is 2.0−44.2× faster compared with other algorithms when transfer 1 MB data. In addition, MMW links between racks can further improve multicast performance because of the bottlenecks between the racks. IRFMW is 1.3× faster compared with IRFM and 2.7−58.6× faster compared with other naive algorithms. And we also test when there are two concurrent multicast transfers (Fig. 11). We can conclude that each multicast algorithm is quite stable. By using millimeter wavelength wireless technology (MMW) in traditional wired DC, we further improve the performance of multicast in realistic case.

There is no significant difference for choosing any parameters for m of IRFM in realistic case. The reason is that IRFM detects the bandwidth utilization of the node, and adaptively change the number of concurrent transfers. The small data size of transfer data alleviates the advantage of this

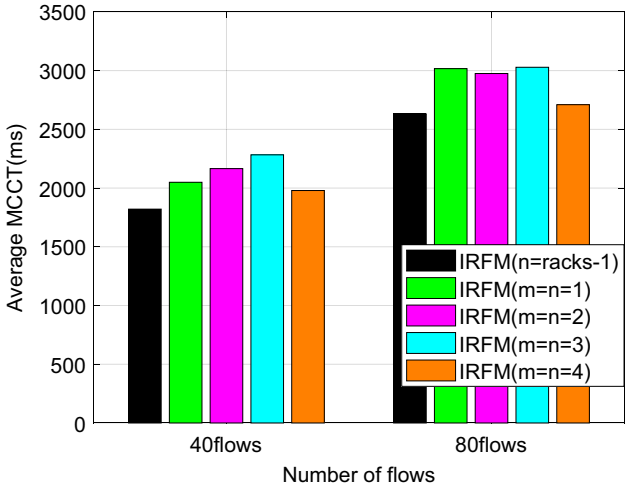

Fig. 12 The average MCCT with background flows, 100 MB, 1 multicast

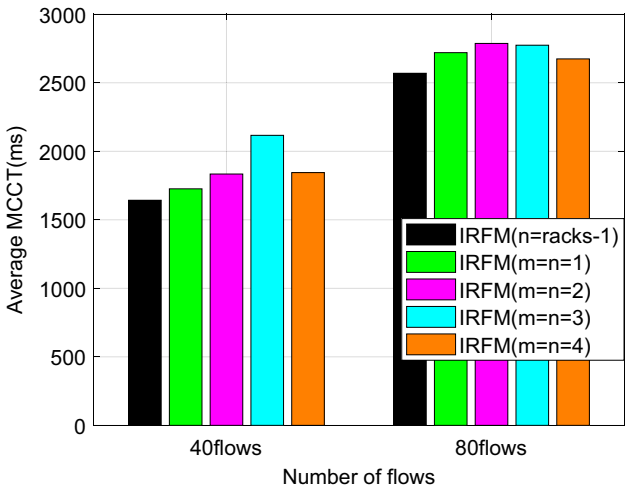

Fig. 13 The average MCCT with background flows, 100 MB, 2 multicasts

algorithm. Therefore, we multicast mass of data (100 M) to evaluate the performance of IRFM. As shown in Figs. 12 and 13, we select parameters n equals to the number of $rack - 1$, as recommended in Sect. 5.2. With the above results, we can conclude that the experiments are in accordance with the theory.

6 Conclusions

In this paper, we not only proposed a multicast algorithm Inter-Rack First Multicast (IRFM) for big data systems in DC networks. But also, we take advantage of millimeter wavelength wireless technology (MMW) to further accelerate DC networks. Our work was motivated by multicast in DC networks, which is widespread in many applications (e.g., HDFS [41]). We implement the multicast API in the application layer. The multicast algorithm IRFM make full use of the bandwidth and the topology in the data center networks. Our experiments show that our multicast algorithm IRFM is 3.7−11.2× faster than the naive multicast implementations in the pure wired case, and 4.8−14.6× faster in the case of MMW enhancement.

Acknowledgements The authors would like to thank anonymous reviewers for their valuable comments. This research is supported by the National Key R&D Program of China 2018YFB1003505, the National Natural Science Foundation of China under Grant Nos. 61602194, 61772265, and 61802172, the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Jiangsu Innovation and Entrepreneurship (Shuangchuang) Program.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Li, D., Cui, H., Hu, Y., Xia, Y., & Wang, X. (2011). Scalable data center multicast using multi-class bloom filter. In *2011 19th IEEE international conference on network protocols* (pp. 266–275). New York: IEEE.
2. Li, D., Li, Y., Wu, J., Su, S., & Esm, J. Y. (2012). Efficient and scalable data center multicast routing. *IEEE/ACM Transactions on Networking (TON)*, *20*(3), 944–955.
3. Li, X., & Freedman, M. J. (2013) Scaling IP multicast on datacenter topologies. In *Proceedings of the 9th ACM conference on emerging networking experiments and technologies* (pp. 61–72). New York: ACM.
4. Data Distribution Service. http://portals.omg.org/dds/.
5. Fan, L., Cao, P., Almeida, J., & Broder, A. Z. (2000). Summary cache: A scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking (TON)*, *8*(3), 281–293.
6. Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008). Large-scale parallel collaborative filtering for the netflix prize. In *International conference on algorithmic applications in management* (pp. 337–348). Berlin: Springer.
7. Gates, A. F., Natkovich, O., Chopra, S., Kamath, P., Narayanamurthy, S. M., Olston, C., et al. (2009). Building a high-level dataflow system on top of map-reduce: The pig experience. *Proceedings of the VLDB Endowment*, *2*(2), 1414–1425.
8. McBride, M., & Liu, H. (2012). *Multicast in the data center overview*. https://datatracker.ietf.org/doc/draft-ietf-mboned-dc-deploy/.
9. Chokkalingam, A., & Riyaz, F. (2004). *BitTorrent protocol specification v1. 0*. CSI 5321.
10. Qiu, D., & Srikant, R. (2004). Modeling and performance analysis of BitTorrent-like peer-to-peer networks. In *ACM SIGCOMM computer communication review* (Vol. 34, pp. 367–378). New York: ACM.
11. Beloglazov, A., & Buyya, R. (2010). Energy efficient resource management in virtualized cloud data centers. In *Proceedings of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing* (pp. 826–831). New York: IEEE Computer Society.

12. Al-Fares, M., Loukissas, A., & Vahdat, A. (2008). A scalable, commodity data center network architecture. In *ACM SIGCOMM computer communication review* (Vol. 38, pp. 63–74). New York: ACM.
13. Alizadeh, M., & Edsall, T. (2013). On the data path performance of leaf-spine datacenter fabrics. In *2013 IEEE 21st annual symposium on high-performance interconnects* (pp. 71–74). New York: IEEE.
14. Petrini, F., & Vanneschi, M. (1997). k-ary n-trees: High performance networks for massively parallel architectures. In *Proceedings 11th international parallel processing symposium* (pp. 87–93). New York: IEEE.
15. Greenberg, A., Hamilton, J. R., Jain, N., Kandula, S., Kim, C., Lahiri, P., et al. (2009). Vl2: A scalable and flexible data center network. In *ACM SIGCOMM*.
16. Singh, A., Ong, J., Agarwal, A., Anderson, G., Armistead, A., Bannon, R., et al. (2015). Jupiter rising: A decade of clos topologies and centralized control in Google's datacenter network. In *Proceeding of the ACM SIGDC 2015* (pp. 183–197). New York: ACM.
17. Adhikari, P. (2008). *Understanding millimeter wave wireless communication*. San Diego: Loea Corporation.
18. Rappaport, T. S., Heath, R. W, Jr., Daniels, R. C., & Murdock, J. N. (2014). *Millimeter wave wireless communications*. London: Pearson Education.
19. Shi, J.-W., Huang, C.-B., & Pan, C.-L. (2011). Millimeter-wave photonic wireless links for very high data rate communication. *NPG Asia Materials*, *3*(4), 41.
20. Bhattacharyya, S., Keshav, S., & Seth, A. (August 3, 2010). *Opportunistic data transfer over heterogeneous wireless networks*. US Patent 7,769,887.
21. Katayama, Y., Takano, K., Kohda, Y., Ohba, N., & Nakano, D. (2011). Wireless data center networking with steered-beam mmwave links. In *2011 IEEE wireless communications and networking conference* (pp. 2179–2184). New York: IEEE.
22. Vardhan, H., Thomas, N., Ryu, S. R., Banerjee, B., & Prakash, R. (2010). Wireless data center with millimeter wave network. In *2010 IEEE global telecommunications conference GLOBECOM 2010* (pp. 1–6). New York: IEEE.
23. Halperin, D., Kandula, S., Padhye, J., Bahl, P., & Wetherall, D. (2011). Augmenting data center networks with multi-gigabit wireless links. In *ACM SIGCOMM computer communication review* (Vol. 41, pp. 38–49). New York: ACM.
24. Hamedazimi, N., Qazi, Z., Gupta, H., Sekar, V., Das, S. R., Longtin, J. P., et al. (2014). Firefly: A reconfigurable wireless data center fabric using free-space optics. In *ACM SIGCOMM computer communication review* (Vol. 44, pp. 319–330). New York: ACM.
25. Wang, X., Kong, L., Kong, F., Qiu, F., Xia, M., Arnon, S., et al. (2018). Millimeter wave communication: A comprehensive survey. *IEEE Communications Surveys and Tutorials*, *20*(3), 1616–1653.
26. Zhou, X., Zhang, Z., Zhu, Y., Li, Y., Kumar, S., Vahdat, A., et al. (2012). Mirror mirror on the ceiling: Flexible wireless links for data centers. *ACM SIGCOMM Computer Communication Review*, *42*(4), 443–454.
27. Jiang, D., Huo, L., Lv, Z., Song, H., & Qin, W. (2018). A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. *IEEE Transactions on Intelligent Transportation Systems*, *99*, 1–15.
28. Jiang, D., Li, W., & Lv, H. (2017). An energy-efficient cooperative multicast routing in multi-hop wireless networks for smart medical applications. *Neurocomputing*, *220*, 160–169.
29. Cui, Y., Wang, H., Cheng, X., Li, D., & Ylä-Jääski, A. (2013). Dynamic scheduling for wireless data center networks. *IEEE Transactions on Parallel and Distributed Systems*, *24*(12), 2365–2374.
30. Yi, X., Liu, F., Liu, J., & Jin, H. (2014). Building a network highway for big data: Architecture and challenges. *IEEE Network*, *28*(4), 5–13.
31. Li, D., Yu, J., Yu, J., & Wu, J. (2011). Exploring efficient and scalable multicast routing in future data center networks. In *INFOCOM, 2011 proceedings IEEE* (pp. 1368–1376). New York: IEEE.
32. Chowdhury, M., Zaharia, M., Ma, J., Jordan, M. I., & Stoica, I. (2011). Managing data transfers in computer clusters with orchestra. In *ACM SIGCOMM computer communication review* (Vol. 41, pp. 98–109). New York: ACM.
33. Yu, T., Noghabi, S. A., Raindel, S., Liu, H., Padhye, J., & Sekar, V. (2016). Freeflow: High performance container networking. In *Proceedings of the 15th ACM workshop on hot topics in networks* (pp. 43–49). New York: ACM.
34. Jiang, D., Huo, L., & Song, H. (2018). Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. *IEEE Transactions on Network Science and Engineering*. (Early Access).
35. Jiang, D., Huo, L., & Li, Y. (2018). Fine-granularity inference and estimations to network traffic for SDN. *PLoS ONE*, *13*(5), e0194302.
36. Nie, L., Jiang, D., & Xu, Z. (2013). A compressive sensing-based reconstruction approach to network traffic. *Computers and Electrical Engineering*, *39*(5), 1422–1432.
37. Mao, Y., & Saul, L. K. (2004). Modeling distances in large-scale networks by matrix factorization. In *Proceedings of the 4th ACM SIGCOMM conference on internet measurement* (pp. 278–287). New York: ACM.
38. Kruskal, J. B., & Wish, M. (1978). *Multidimensional scaling. Quantitative applications in the social*. Beverly Hills: Sage University Papers Series.
39. Mclust. http://www.stat.washington.edu/mclust/.
40. Fraley, C., Raftery, A., Murphy, T., & Scrucca, L. (2012). *mclust version 4 for R: Normal mixture modeling for model-based clustering, classification, and density estimation*. Seattle: University of Washington.
41. HDFS. https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.

Yi Wang received the BS, MS and PhD degrees from the Department of Electronics and Information Engineering at the Huazhong University of Science and Technology, China, in 2000, 2003, and 2009 respectively. She is now a lecturer in the School of Electronics Information and Communications, Huazhong University of Science and Technology, China. Her research interest is Cloud computing.

Bingquan Wang received his BEng degree from School of Electronic Information and Communications, Southeast University, Nanjing, China. He is currently working on completing his Master's degree in Nanjing University of Science and Technology. His research interests include peer-to-peer networks and Networking.

Chen Tian is an associate professor with State Key Laboratory for Novel Software Technology, Nanjing University, China. He was previously an associate professor with School of Electronics Information and Communications, Huazhong University of Science and Technology, China. Dr. Tian received the BS (2000), MS (2003) and Ph.D (2008) degrees at Department of Electronics and Information Engineering from Huazhong University of Science and Technology, China. From 2012 to 2013, he was a postdoctoral researcher with the Department of Computer Science, Yale University. His research interests include data center networks, network function virtualization, distributed systems, Internet streaming and urban computing.

Fu Xiao received the Ph.D. degree in Computer Science and Technology from Nanjing University of Science and Technology, Nanjing, China, in 2007. He is currently a Professor and PhD supervisor with the School of Computer, Nanjing University of Posts and Telecommunications, Nanjing, China. He has published over 30 papers in related international conferences and journals, including IEEE Journal on Selected Areas in Communications, IEEE Transactions on Networking, IEEE Transactions on Mobile Computing, INFOCOM, IPCCC, ICC and so on. His main research interest is Wireless Sensor Networks and Internet of Things. Dr. Xiao is a member of the IEEE Computer Society and the Association for Computing Machinery.