RAMPS: Next Generation Platform for Real Time and Resilient IoT Analytics using MmWave and Programmable Switches

Vishal Shrivastav
Purdue University

Dimitrios Koutsonikolas
Northeastern University

Saurabh Bagchi
Purdue University

ABSTRACT

Real time IoT analytics remains a challenging problem due to the distributed nature of the analytics platform (comprising sensors, edge server(s), and actuators), which raises three fundamental challenges of (i) how to map computations to a distributed and heterogeneous compute fabric, (ii) how to communicate multi-Gbps of data wirelessly between sensors and edge servers for high analytics accuracy, and (iii) how to effectively share the communication channel between multiple sensor-edge network streams. To meet these challenges, we envision an analytics platform that will tightly couple the application stack, the network stack, and emerging networking technologies, namely mmWave wireless and programmable switches, to meet both the computation and communication demands for real time IoT analytics.

CCS CONCEPTS

• **Computing methodologies** → **Distributed computing methodologies**; • **Networks** → **Network architectures**;

KEYWORDS

Distributed IoT Analytics; mmWave Wireless; Programmable Switches

ACM Reference Format:
Vishal Shrivastav, Dimitrios Koutsonikolas, and Saurabh Bagchi. 2021. RAMPS: Next Generation Platform for Real Time and Resilient IoT Analytics using MmWave and Programmable Switches. In *Fifth Workshop on Distributed Infrastructures for Deep Learning (DIDL) 2021 (DIDL '21), December 6, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3493652.3505631

DIDL '21, December 6, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-9173-3/21/12.
https://doi.org/10.1145/3493652.3505631

1 INTRODUCTION

Autonomous Internet-of-Things (IoT) systems comprising sensors, such as cameras, and actuators have become ubiquitous, capturing and generating large amounts of data every second. However, analyzing all that data and generating appropriate response in real time remains a challenging problem. To ground this discussion in a specific context (which we detail in § 2.1), consider two application contexts. In the first, we are generating high bandwidth video that will be used in an AR/VR setting while in the second, we are getting a hybrid mix of sensor data from manufacturing equipment and in aggregation, the sensor stream consumes high bandwidth. In both cases, the analytics system has to process the sensor streams and provide results under strict latency bounds. In the first, the result is to show different scenes to the user; in the second, the result could be to idle the machine to prevent a breakdown.

The challenge to supporting these application scenarios is that sensors are typically computation and power constrained, and hence one has to offload a part or whole of the analysis onto servers; we will consider "edge servers" that are closer to the sensor data, rather than traditional servers in datacenters. This, in turn, means that for real time analytics, the analytics platform must run very fast "sense-communicate-actuate" loops between the sensors, the edge servers, and the actuators. However, there are three fundamental challenges in designing such a platform.

First, the distributed nature of the computation between the sensors and the edge servers means that the network is on the critical path. Further, the communication channel needs to be shared between multiple network streams originating at different sensors. These streams have different bandwidth, latency, and reliability requirements, and more crucially, these requirements change dynamically. Also, the bandwidth at the edge is typically constrained, with each edge server serving streams from multiple sensors, resulting in many-to-one communication, also called Incast, which is known to cause network congestion that is very hard to handle, often resulting in throughput collapse [7].

Second, the need for very high analytics accuracy requires multi-Gbps of data to be communicated from the sensors to the edge servers [18, 21]. Millimeter-wave (mmWave)

wireless has emerged as the prime candidate technology for providing multi-Gbps data rates in wireless networks. However, communication at mmWave frequencies faces fundamental resiliency challenges due to the high propagation and penetration loss and the use of directional transmissions makes links susceptible to disruption by human blockage. As a result, all today's commercial off-the-shelf (COTS) devices that are equipped with mmWave radios are *dual-band*, featuring a second radio operating in the sub-6 GHz frequency band. We envision a similar dual-band architecture for our sensors. However, this results in a fundamental challenge of how to schedule packets belonging to data streams with different latency and reliability requirements between multiple frequency bands, and that too in a power-aware manner.

Finally, the compute fabric is both distributed and heterogeneous. This happens because today's edge deployments, and those in the anticipated near-term future, are composed of nodes with different compute capabilities. This makes mapping the analytic computations to the compute fabric challenging. Further, the resource availability at each device changes dynamically. Hence, the mapping needs to be adaptive and dynamic, as well as resilient to mispredictions in resource availability and performance models.

Our proposed architecture: RAMPS

To meet these challenges, we envision an analytics platform that will tightly couple the application stack, the network stack, and emerging networking technologies, namely mmWave wireless [38] and programmable switches [32]. First, while mapping computation to the compute fabric, one can leverage the wireless connectivity among the devices and corresponding chances of overhearing neighborhood communication, to estimate the state of the resource availability at the edge servers and the network routers to dynamically adapt the mapping. Second, one can use application-level knowledge of bandwidth, latency, and reliability requirements for network streams, to guide the scheduling and bandwidth allocation decisions at both the network and the link layer. One can also tightly couple the transport and the link layers, for example, by using real time bandwidth estimation techniques for mmWave wireless channels to signal the transport layer of the available bandwidth at the link layer at very fine timescales. Finally, one can leverage emerging *programmable* network switches [32] to put intelligence into the network, by implementing custom in-network scheduling and load balancing policies for efficient network sharing. Programmable switches can also be leveraged to offload certain analytic computations to reduce the load on the edge servers and to conserve the edge bandwidth. This is a challenging problem because programmable switches are suitable for limited kinds of computation and they should run such computation only if their core capability, namely routing packets at line speed, is not affected.

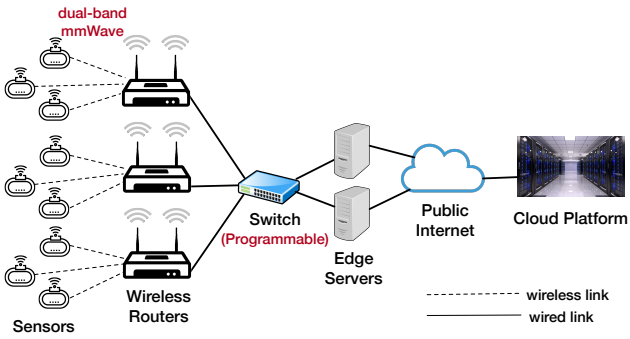

Figure 1: RAMPS architecture. The analytics on the high bandwidth streams occur in the network switches as well as on the edge servers. The availability of mmWave radio links enables the low latency, high bandwidth transport. Multiple network streams from different sensors, with varying reliability and latency requirements, compete for the communication and the computation resources.

2 APPLICATIONS AND ARCHITECTURE

2.1 Real time analytics on sensor data from smart manufacturing systems

In a smart manufacturing system, various sensors (e.g., vibration, ultrasonic, pressure sensors) are used for process control, automation, production plans, and equipment maintenance. The data from these sensors need to be analyzed in a streaming, real time manner to fill a critical role in predictive maintenance tasks, through anomaly detection and anomaly localization processes. These often need to be accomplished in real time to prevent damage to equipment, or in a small subset of cases, to prevent harm to the humans operating on the shop floor.

The sensor devices (vibration, ultrasonic, or pressure sensors) are typically computation and power constrained, hence the analytics on the sensor data from these applications needs to be distributed between the local sensor nodes and some remote analytics platform. Next, the network streams between the sensors and the remote analytics platform require **high bandwidth** to transmit high bit-rate video frames or full spectrum vibration time series data. The high fidelity video streams consume bandwidth of upwards of 5 Gbps for each stream (uncompressed 1920x1080 stream at 60 fps, the kind needed to make critical decisions based on video such as imperfections in the product coming out of a machine) [12]. Regarding vibration data, the leading edge commercial sensors [14] sense at up to 25.6 KHz and the entire vibration spectrum from 10 Hz to 1 KHz. These streams also require **low latency** to support real time analytics. Also, these network streams have **non-binary reliability** requirements,

e.g., several of the algorithms used for video analytics and anomaly detection are robust to some (but not all) packet losses [5, 9, 30, 31]. Finally, the network requirements (bandwidth, latency, reliability) of these streams **change dynamically** over time. For example, a machine vibration sensor generating normal vibration data might suddenly start to capture some abnormal data, which would require immediately changing the network requirements for those streams to high bandwidth, low latency, and high reliability.

2.2 Architecture

Figure 1 shows our system architecture. The architecture has five key components:

(1) **Sensors.** These include high-definition cameras and machine sensors (e.g., vibration, ultrasonic, pressure sensors). Each sensor is dual-band, i.e., equipped with both mmWave wireless (for high bandwidth communication) and sub-6 GHz radios. There can be few 10s to few 100s of such sensors.
(2) **Wireless Routers.** The sensors directly communicate with wireless routers. There are multiple wireless routers to support the aggregate data rate from all the sensors.
(3) **Programmable Switch.** The wireless routers connect to a switch using wired links. We assume the switch to be *programmable*, i.e., it can be programmed with custom packet processing logic. A reference programmable switch is Intel Tofino [32], which can be programmed using a domain-specific language called P4 [6], and can achieve orders of magnitude better processing throughput and latency compared to a commodity server [34].
(4) **Edge servers.** The programmable switch connects to one or more edge servers that perform real time analytics that cannot be performed locally at the sensor nodes.
(5) **Cloud Platform.** Finally, our architecture uses public cloud, such as Amazon AWS, to perform computationally heavy analytics that does not require real time response.

3 CHALLENGE 1: DISTRIBUTED ANALYTIC COMPUTATIONS

With an increase in the number of latency-sensitive applications that need to be executed based on sensor data, it is attractive to run them on a distributed computation fabric comprising of provisioned edge devices and network routers. These applications themselves are not monolithic, but rather comprise of a DAG of ML functions, with the functions having different latency requirements, resource consumption, and reliability requirements. The computational fabric is heterogeneous spanning from sensor nodes with very limited compute capacity, to edge devices that do not have GPUs (like Raspberry Pis), to those that do have GPUs (like the Jetson series of devices), to emerging programmable switches. Further, there is many-to-one mapping of the sensor streams to the compute nodes leading to oversubscription and transient unavailability of these devices; in other words, we are dealing with an environment where the reliability is below 100%. There is some element of resource availability prediction that is possible for the edge devices, to varying degrees of lookahead, while with the network elements, their resource availability depends on the exogenous factor of how much data traffic is currently in the system. In the context of the above challenges, our problem formulation is simple — to map the different functions of an application DAG to the heterogeneous mix of these devices, and to make remapping decisions as the dynamic conditions change.

Research Directions

We can take inspiration from the line of work on mapping ML tasks to heterogeneous computing environments. Such work has shown the ability to perform this mapping in an agile manner, and optimizing the different metrics of interest in different papers, such as, utilization of the computer cluster [15], latency [35], or availability [27]. Some works have considered a two-class heterogeneity in that there is a sensor device which is generating the data stream and an edge device and a partitioning of the task is done between the two [8, 33]. This line of work has not considered DAGs of applications, which introduce the challenge that the end-to-end performance may have a complex dependency on the performance of each function, an aspect that is well understood in the distributed systems and performance modeling literature [11, 17].

The solution approach that we are pursuing has three key insights and correspondingly, three design ideas. *First*, it is possible to expose some tuning knobs for many ML functions which allow us to play in the tradeoff space of accuracy versus latency or resource consumption, *e.g.,* in our work on embedded object classification [36] and object detection [37]. *Second*, in our environment with wireless connectivity among the devices and corresponding chances of overhearing neighborhood communication, it is possible to estimate the state of the resource availability at the edge devices and the network routers. *Third*, it is possible to build incremental models for the end-to-end performance in a DAG that can be dynamically instantiated and queried in a prompt manner as dynamic conditions change [19].

4 CHALLENGE 2: ADAPTIVE NETWORK SHARING

The network channel between the sensors and the edge servers is shared by multiple network streams originating at different sensors. These streams have heterogeneous network requirements in terms of bandwidth, latency, and reliability. For example, some streams carry more critical data than others, and hence demand lower response latency and

bigger share of bandwidth between the sensors and the edge server. Similarly, different streams also have different (non-binary) reliability requirements. For example, if a machine vibration sensor is continually generating the same normal vibration data, loss of some of that data would not affect the actuation response. However, if it generates abnormal data, it must be delivered with 100% reliability. More crucially, the network requirements for each stream can change dynamically over time, and hence, the network must be *adaptive* to meet these dynamically changing heterogeneous network requirements. This problem is exacerbated even further in larger deployments with 100s of sensors, where the edge bandwidth becomes a bottleneck and the network needs to handle many-to-one communication between the sensors and the edge server. Such communication patterns, called Incast, are extremely hard to handle and are known to cause throughput collapse [7]. And while there have been several works [10, 13, 20] to handle Incast, they have all focused on datacenter-like environment and do not directly apply to our environment with a mixture of wired and wireless links. Also existing solutions only reduce the chances of throughput collapse, instead of completely eliminating it.

Research Directions

To meet these challenges, we envision a platform that tightly integrates the application, transport, and data link layers of the network stack, and further leverages switch programmability to make adaptive and intelligent scheduling and congestion control decisions. First, in our recent work [3], we show that by using carefully designed neural networks, we can accurately predict the available wireless link bandwidth at very fine timescales, which we can then communicate to the transport layer to guide optimal bandwidth allocation for network streams. Second, we can use switch programmability to communicate custom congestion signals to the transport layer at the sensors in real time for effective congestion control under scenarios such as Incast. Finally, we can tightly couple the application and the network stack, by making applications communicate their custom network requirements, such as bandwidth and latency requirements, priority of a packet, required degree of reliability, etc. to the network stack, and letting the network enforce those requirements. In that regard, we can again leverage the programmability of switches [25, 26] to implement custom scheduling policies inside the switch to enforce global network requirements across network streams from multiple sensors.

5 CHALLENGE 3: USING MMWAVE WITH OTHER NETWORK MODALITIES

A key research challenge in dual-band devices, equipped with both mmWave and sub-6 GHz radios, is to determine which radio to use at any given time. Selecting the right radio is challenging because mmWave and sub-6 GHz radios have very different features including supported data rates, range, link stability, reliability, and power consumption. First, mmWave technologies (e.g., 802.11ad) support multi-Gbps data rates but only at ranges up to a few 10s of ft [1, 23]. Second, the data rate of mmWave links fluctuates rapidly, even under static, line-of-sight (LOS) conditions, whereas the data rate of sub-6 GHz WiFi (e.g., 802.11ac/ax) is much much more stable [1, 23, 28]. Hence, for applications with strict timing guarantees it may be preferable to use sub-6 GHz WiFi for communication even at the cost of reduced throughput. Third, mmWave signals are easily blocked by objects and human bodies [1, 28, 29], resulting in temporary link outages. Given that even optimized blockage detection and interface switching solutions [28] can take up to 100 ms in the worst case and algorithms on COTS devices take several seconds [22, 28], simultaneously utilizing both interfaces might be preferable for high-priority streams. Finally, the two technologies have very different power profiles [2, 24]. While for sub-6 GHz WiFi the Tx power for a given data rate is much higher than the Rx power, as is the case for most wireless technologies, the trend is reversed for 802.11ad: the Tx power is lower than the Rx power and much lower than the 802.11ac/802.11ax Tx power even for very high data rates.

Research Directions

In the following, we outline three research directions based on the key idea in the design of Ramps– tight coupling of the application and network stacks – that seek to balance three (often conflicting) goals: satisfy the application requirements (data rate, reliability, latency, analytics accuracy), achieve energy efficiency, and provide resiliency against the rapidly fluctuating and unpredictable mmWave channel. First, given the low latency requirements of certain applications and the rapid throughput fluctuation and sensitivity to blockage of mmWave links, we envision simultaneous use of mmWave and sub-6 GHz radios for communication via transport [22] or link layer solutions [4]. For example, in our recent work [22], we showed for first time that it is possible to effectively bundle the two radios by designing MuSher, a throughput-ratio-aware MPTCP scheduler that assigns packets to the two interfaces at a ratio equal to the ratio of the bandwidths of the two interfaces, with the goal of maximizing throughput. Second, we will consider multihop mmWave networking to extend the communication range for bandwidth-demanding applications. For example, a recent work [18] proposed Spider, a dual-band multihop network architecture for video analytics, consisting of a mmWave data plane and a separate WiFi control plane. We also envision a unified control-data plane architecture that would allow different types of control and data packets to use either of the two frequency bands, depending on availability and needs. Finally, we emphasize that the designs of both transport/link layer packet schedulers and multihop solutions have to be

energy-aware. For example, in our recent work [16], we outlined the design of a simple energy-aware MPTCP scheduler, based on the insights from [2]. Such simple designs have to be revisited in more complex scenarios (e.g., involving blockage) as well as in the case of multihop dual-band networks, where communication involves two IoT devices, one acting as a transmitter – consuming low (high) power in the case of 802.11ad (802.11ac) and one as a receiver, where the power consumption relationship is reversed.

6 TAKEAWAYS

We have laid out an architecture for high-bandwidth IoT scenarios where low latency sense-compute-actuate loops are required. Real time IoT analytics remains a challenging problem due to the distributed nature of the analytics platform, comprising sensors, edge server(s), and actuators. We argue for an analytics platform that tightly couples the application stack, the network stack, and the emerging wireless networking technologies, namely mmWave wireless and programmable switches.

REFERENCES

[1] Shivang Aggarwal, Moinak Ghoshal, Piyali Banerjee, and Dimitrios Koutsonikolas. 2021. An Experimental Study of the Performance of IEEE 802.11ad in Smartphones. *Elsevier Computer Communications* 169 (2021), 220–231.

[2] Shivang Aggarwal, Moinak Ghoshal, Piyali Banerjee, Dimitrios Koutsonikolas, and Joerg Widmer. 2021. 802.11ad in Smartphones: Energy Efficiency, Spatial Reuse, and Impact on Applications. In *Proc. of IEEE INFOCOM*.

[3] Shivang Aggarwal, Zhaoning Kong, Moinak Ghoshal, Y. Charlie Hu, and Dimitrios Koutsonikolas. 2021. Throughput Prediction on 60 GHz Mobile Devices for High-Bandwidth, Latency-Sensitive Applications. In *Proc. of the Passive and Active Measurement Conference (PAM)*.

[4] Ghufran Baig, Jian He, Mubashir Adnan Qureshi, Lili Qiu, Guohai Chen, Peng Chen, and Yinliang Hu. 2019. Jigsaw: Robust Live 4K Video Streaming. In *Proc. of ACM MobiCom*.

[5] Christophe Bertero, Matthieu Roy, Carla Sauvanaud, and Gilles Trédan. 2017. Experience report: Log mining using natural language processing and application to anomaly detection. In *2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 351–360.

[6] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker. 2014. *P4: Programming Protocol-Independent Packet Processors*. SIGCOMM CCR.

[7] Yanpei Chen, Rean Griffith, Junda Liu, Randy H. Katz, and Anthony D. Joseph. 2009. Understanding TCP Incast Throughput Collapse in Datacenter Networks *(WREN '09)*. Association for Computing Machinery, New York, NY, USA, 73–82. https://doi.org/10.1145/1592681.1592693

[8] Vittorio Cozzolino, Aaron Yi Ding, and Jörg Ott. 2017. Fades: Fine-grained edge offloading with unikernels. In *Proceedings of the Workshop on Hot Topics in Container Networking and Networked Systems*. 36–41.

[9] Mingjian Cui, Jianhui Wang, and Meng Yue. 2019. Machine learning-based anomaly detection for load forecasting under cyberattacks. *IEEE Transactions on Smart Grid* 10, 5 (2019), 5724–5734.

[10] Peter X. Gao, Akshay Narayan, Gautam Kumar, Rachit Agarwal, Sylvia Ratnasamy, and Scott Shenker. 2015. PHost: Distributed near-Optimal Datacenter Transport over Commodity Network Fabric. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT '15)*. Association for Computing Machinery, New York, NY, USA, Article 1, 12 pages. https://doi.org/10.1145/2716281.2836086

[11] Giovanni Paolo Gibilisco, Min Li, Li Zhang, and Danilo Ardagna. 2016. Stage aware performance modeling of dag based in memory analytic platforms. In *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*. IEEE, 188–195.

[12] Bo Gowan. 2018. How much bandwidth does Broadcast HD video use? https://www.ciena.com/insights/articles/How-much-bandwidth-does-Broadcast-HD-video-use_prx.html. Last accessed: August-11-2021.

[13] Mark Handley, Costin Raiciu, Alexandru Agache, Andrei Voinescu, Andrew Moore, Gianni Antichi, and Marcin Wojcik. 2017. *Re-architecting datacenter networks and stacks for low latency and high performance*. SIGCOMM.

[14] Fluke Inc. 2021. Fluke 3561/3502 FC Vibration Sensor Starter Kit with Software. https://www.fluke-direct.com/. Last accessed: August-11-2021.

[15] Myeongjae Jeon, Shivaram Venkataraman, Amar Phanishayee, Junjie Qian, Wencong Xiao, and Fan Yang. 2019. Analysis of large-scale multi-tenant {GPU} clusters for {DNN} training workloads. In *2019 {USENIX} Annual Technical Conference (USENIX ATC)*. 947–960.

[16] Imran Khan, Moinak Ghoshal, Shivang Aggarwal, Dimitrios Koutsonikolas, and Joerg Widmer. 2021. Multipath TCP in Smartphones Equipped with Millimeter Wave Radios. In *WiNTECH*. ACM.

[17] Dzmitry Kliazovich, Johnatan E Pecero, Andrei Tchernykh, Pascal Bouvry, Samee U Khan, and Albert Y Zomaya. 2016. CA-DAG: Modeling communication-aware applications for scheduling in cloud computing. *Journal of Grid Computing* 14, 1 (2016), 23–39.

[18] Zhuqi Li, Yuanchao Shu, Ganesh Ananthanarayanan, Longfei Shangguan, Kyle Jamieson, and Victor Bahl. 2020. *Spider: Next generation Live Video Analytics over Millimeter-Wave Networks*. Technical Report. Microsoft.

[19] Ashraf Mahgoub, Paul Wood, Alexander Medoff, Subrata Mitra, Folker Meyer, Somali Chaterji, and Saurabh Bagchi. 2019. SOPHIA: Online Reconfiguration of Clustered NoSQL Databases for Time-Varying Workloads. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*. USENIX Association, Renton, WA, 223–240. https://www.usenix.org/conference/atc19/presentation/mahgoub

[20] Behnam Montazeri, Yilong Li, Mohammad Alizadeh, and John Ousterhout. 2018. Homa: A Receiver-Driven Low-Latency Transport Protocol Using Network Priorities. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '18)*. Association for Computing Machinery, New York, NY, USA, 221–235. https://doi.org/10.1145/3230543.3230564

[21] Vit Ruzicka and Franz Franchetti. 2018. Fast and accurate object detection in high resolution 4K and 8K video using GPUs. In *Proc. of IEEE HPEC*.

[22] Swetank Kumar Saha, Shivang Aggarwal, Rohan Pathak, Dimitrios Koutsonikolas, and Joerg Widmer. 2019. MuSher: An Agile Multipath-TCP Scheduler for Dual-Band 802.11ad/ac Wireless LANs. In *Proc. of the 25th Annual International Conference on Mobile Computing and Networking (MobiCom)*.

[23] Swetank Kumar Saha, Hany Assasa, Adrian Loch, Naveen Muralidhar Prakash, Roshan Shyamsunder Anantharamakrishna, Shivang Aggarwal, Daniel Steinmetzer, Dimitrios Koutsonikolas, Joerg Widmer, and Matthias Hollick. 2018. Fast and Infuriating: Performance and Pitfalls of 60 GHz WLANs Based on Consumer-Grade Hardware. In *Proc. of IEEE SECON*.

[24] Swetank Kumar Saha, Tariq Siddiqui, Dimitrios Koutsonikolas, Adrian Loch, Joerg Widmer, and Ramalingam Sridhar. 2017. A Detailed Look

into Power Consumption of Commodity 60 GHz Devices. In *Proc. of the 25th Annual International Conference on Mobile Computing and Networking (MobiCom)*.

[25] Vishal Shrivastav. 2019. *Fast, Scalable, and Programmable Packet Scheduler in Hardware*. SIGCOMM.

[26] Anirudh Sivaraman, Suvinay Subramanian, Mohammad Alizadeh, Sharad Chole, Shang-Tse Chuang, Anurag Agrawal, Hari Balakrishnan, Tom Edsall, Sachin Katti, and Nick McKeown. 2016. *Programmable Packet Scheduling at Line Rate*. SIGCOMM.

[27] Daniel Smilkov, Nikhil Thorat, Yannick Assogba, Ann Yuan, Nick Kreeger, Ping Yu, Kangyi Zhang, Shanqing Cai, Eric Nielsen, and David Soergel. 2019. Tensorflow. js: Machine learning for the web and beyond. *arXiv preprint arXiv:1901.05350* (2019).

[28] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. 2017. WiFi-Assisted 60 GHz Wireless Networks. In *Proc. of ACM MobiCom*.

[29] Sanjib Sur, Vignesh Venkateswaran, Xinyu Zhang, and Parameswaran Ramanathan. 2015. 60 GHz Indoor Networking through Flexible Beams: A Link-Level Profiling. In *Proc. of ACM SIGMETRICS*.

[30] Shikhar Suryavansh. 2020. *I-Bot: Interference based Orchestration of Tasks for Dynamic Unmanaged Edge Computing*. Ph.D. Dissertation. Purdue University. Advisor: Saurabh Bagchi.

[31] Shikhar Suryavansh, Abu Benna, Chris Guest, and Somali Chaterji. 2021. Ambrosia: Reduction in Data Transfer from Sensor to Server for Increased Lifetime of IoT Sensor Nodes. *arXiv preprint arXiv:2107.05090* (2021).

[32] https://www.intel.com/content/www/us/en/products/network-io/programmable-ethernet-switch/tofino-2-series.html. 2021. *Tofino Switch*. Intel.

[33] Xiaojuan Wei, Shangguang Wang, Ao Zhou, Jinliang Xu, Sen Su, Sathish Kumar, and Fangchun Yang. 2017. MVR: An architecture for computation offloading in mobile edge computing. In *2017 IEEE International Conference on Edge Computing (EDGE)*. IEEE, 232–235.

[34] Xin Jin, Xiaozhou Li, Haoyu Zhang, Nate Foster, Jeongkeun Lee, Robert Soulé, Changhoon Kim, and Ion Stoica. 2018. *NetChain: Scale-Free Sub-RTT Coordination*. NSDI.

[35] Eric P Xing, Qirong Ho, Wei Dai, Jin Kyu Kim, Jinliang Wei, Seunghak Lee, Xun Zheng, Pengtao Xie, Abhimanu Kumar, and Yaoliang Yu. 2015. Petuum: A new platform for distributed machine learning on big data. *IEEE transactions on Big Data* 1, 2 (2015), 49–67.

[36] Ran Xu, Jinkyu Koo, Rakesh Kumar, Pengcheng Wang, Peter Bai, , Ganga Meghanath, Somali Chaterji, Subrata Mitra, and Saurabh Bagchi. 2021. ApproxNet: Content and Contention Aware Video Analytics System for the Edge. *ACM Transactions on Sensor Networks* (2021), 1–27.

[37] Ran Xu, Chen-lin Zhang, Pengcheng Wang, Jayoung Lee, Subrata Mitra, Somali Chaterji, Yin Li, and Saurabh Bagchi. 2020. ApproxDet: content and contention-aware approximate object detection for mobiles. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 449–462.

[38] Xinyu Zhang and Upamanyu Madhow. 2017. Millimeter-Wave Wireless Networking and Sensing: Tutorial at Sigcomm. https://conferences.sigcomm.org/sigcomm/2017/tutorial-mmWave.html.