Chapter 6
Urban IoT Edge Analytics

Aakanksha Chowdhery, Marco Levorato, Igor Burago, and Sabur Baidya

6.1 Introduction

The application of the Internet of Things (IoT) paradigm to the urban environment is of particular relevance as it responds to important societal needs and trends [1]. The push to provide solutions toward a functional and efficient Smart City architecture is demonstrated by the large number of academic and industrial endeavors, as well as initiatives from city administrations. For instance, IBM, Siemens, Cisco, ABB, Alcatel-Lucent, Toshiba, and Google have undergoing projects that aim at the development of smart interconnected systems, as well as established city-wide endeavors involving cities in the USA, Europe, and Asia [2, 3].

Current IoT architectures rely on two extremes. On the one hand, enterprise computing largely relies on hauling all the data to the cloud to leverage the cost-benefits and efficiency of a high-capacity storage and compute platform in the data centers [4, 5]. On the other hand, mission-critical applications, such as self-driving cars and autonomous robots, largely rely on local computation for their decision-making because of stringent low latency requirements. In the urban IoT and Smart City scenarios, a city-wide deployment of IoT technologies poses several inherent conceptual and technical challenges that are not resolved by those two extreme architectures. For instance, the transportation of raw streams of data from personal mobile sensors, video surveillance systems, traffic monitoring systems, and other relevant systems to city-scale data centers would require an enormous amount of bandwidth, and energy drain from mobile devices, and would likely result in service

A. Chowdhery
Princeton University, Princeton, NJ, USA
e-mail: aakanksha@princeton.edu

M. Levorato (✉) • I. Burago • S. Baidya
University of California, Irvine, CA, USA
e-mail: levorato@uci.edu

A.M. Rahmani et al. (eds.), *Fog Computing in the Internet of Things*,
DOI 10.1007/978-3-319-57639-8_6

disruption at the wireless edge of the network. Similarly, in a full-scale and mature urban IoT scenario, centralized real-time processing of this large and heterogeneous set of data streams is not feasible.

Edge computing is an architecture that uses end-user clients and one or more near-user edge devices collaboratively to store substantial amounts of data, process compute-intensive tasks, communicate jointly to reduce interference, and carry out management tasks cooperatively to improve application performance. In such edge computing architectures, any device with compute, storage, and networking capabilities can serve as a near-user edge device. The end-user clients and various edge devices can exist in a hierarchy alongside the existing cloud-based architecture to improve the overall system performance. This notion of edge computing is also referred to as "Edge analytics" in [6] or "Fog computing" in [7]. Edge computing solves four key challenges by executing tasks near the edge of local access networks. First, it pools underutilized resources of edge devices in terms of storage or compute capabilities and minimizes the network overhead of hauling data to the cloud. Second, it provides context awareness as application level details are available near the client at the network edge. Thirdly, it enables real-time response with latency in the order of tens of milliseconds by processing near the network edge instead of relying on the cloud, where the multi-hop architecture of the network core may result in undesirable delays. Finally, the software stacks on edge devices can be upgraded in an agile manner without modifying the software stacks in the cloud or core network.

Innovative city-wide architectures should make the best use of those new paradigms, which have the potential to lead to a significant advancement in how data are acquired, transported, and processed over large-scale systems. In particular, there should be a strong interconnection between information acquisition, data communication, and processing across the many geographical and system scales involved. This interconnection can dramatically reduce network load, while significantly improve the quality of Smart City services and reduce response latency. For instance, data fusion and processing performed within or at the edge of local wireless networks can inform data filtering and resource allocation strategies (Fig. 6.1).

The rest of the chapter is organized as follows. Section 6.2 further discusses the design challenges. Section 6.3 presents the architecture and discusses its main components, namely information acquisition and compression, content-aware networking, and information availability. Section 6.5 concludes the chapter.

6.2 Design Challenges

Large cities face many challenges, including traffic congestion, public safety concerns, high energy use, sanitation, public internet connectivity, and providing baseline municipal services. A major issue in establishing smart cities is availability of ubiquitous broadband bandwidth and connectivity. While most modern cities

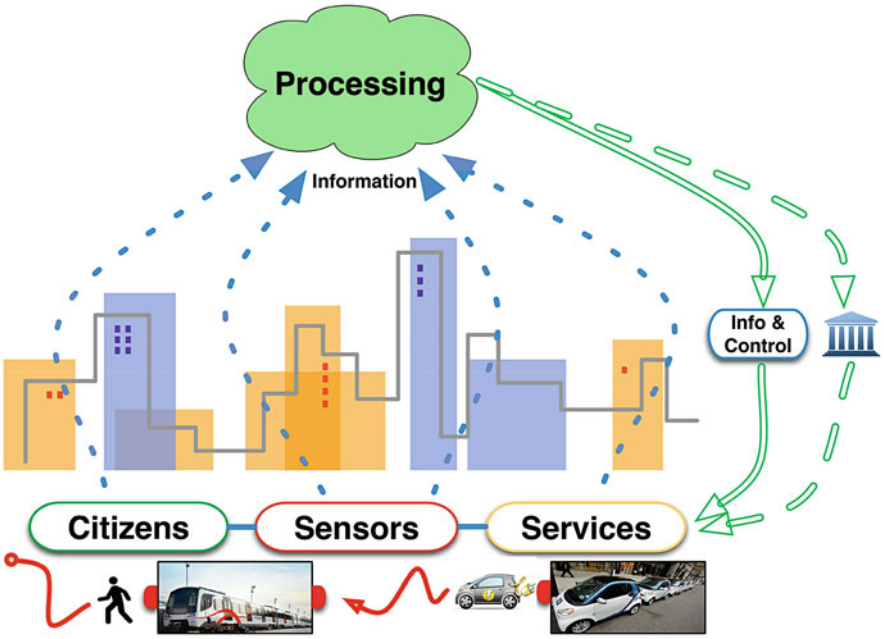

Fig. 6.1 The urban IoT interconnects systems and citizens to provide innovative services

have one or more cellular networks providing adequate coverage, these networks are often designed to have capacity and peak bandwidth limits that just meet the needs of their existing subscribers. This leaves a relatively small, and time-varying, amount of bandwidth for the advanced municipal services envisioned in the Smart City paradigm.

A critical need in modern cities, which is also the focus of Smart city efforts, is safety and security. In the Smart City, this need is addressed by a large and distributed network of sensors and systems. Municipal networks may carry sensitive data (i.e., police dispatches) and operate life-critical systems (e.g., smart transportation, collision avoidance applications, first responder communications, *etc.*), and therefore must be both secure and reliable. Traffic monitoring applications require constant traffic flow updates at each road and intersection to manage road congestion and diverting traffic flows from accident areas.

Related to safety and security, illustrative of the technical challenges of building city-scale systems is video monitoring and surveillance. Smart cities, retail stores, public transport, and enterprises increasingly rely on camera sensors to improve safety and security, identify unauthorized access, and increase reliability of their infrastructure. Local processing does not lend itself to successful deployment, whereas the sheer bandwidth of data being collected over a large-scale network makes it impractical to transport all the data to the cloud to obtain real-time insights. City-scale deployments (such as on traffic lights) and remote areas do not have

enough bandwidth to upload high-data rate video streams. Many applications such as real-time tracking and detection of intruders pose strict latency constraints on such infrastructure. Additionally, privacy constraints must be maintained so that the video does not reveal a person's identity to any unauthorized parties. Advanced distributed analytics provides the opportunity to build real-time, latency-sensitive distributed surveillance systems that maintains privacy. We can leverage nearby nodes to intelligently partition video processing between edge devices colocated with cameras and the cloud so as to enable real-time tracking, anomaly detection, and interesting insights from data collected over long time intervals.

Finally, a smart city can also capitalize on the crowdsourced information collected from its citizens using their mobile sensors. For instance, crowdsourced information can be used to estimate parking availability, neighborhood security, wireless signal strength, and congestion in public spaces.

We summarize the key challenges in building a city-wide infrastructure that can capitalize on large sensor systems installed in the city as well as the crowdsourced sensors: *(a) scarce wireless bandwidth*—Available wireless bandwidth is scarce for multiple sensors to coexist with existing wireless services while a wired infrastructure requires heavy investments, *(b) low latency*—Low response latency is critical to applications such as traffic monitoring, where hauling all the data to the cloud to obtain insights can take several minutes to an hour, *(c) efficiency*—A city would require petabytes of storage if it were to transport all the sensor data streams to a single data center, where most of the data would not be useful except to obtain summaries or detect abnormal events. Energy efficiency also favors local computation because the radio transmit power required to continuously communicate collected data often drains the sensors and mobile devices in crowdsourced environments, and *(d) privacy*—Local storage and computation can maintain privacy of the individual sensor streams collected from different entities compared to a centralized solution that aggregates data in a single place. Finally, it is much easier to maintain context of the information closer to the sensor than in a centralized place.

6.3 Edge-Assisted Architecture

Based on the challenges described in the previous section, we contend that edge computing is a key component of such architecture, as it interconnects information acquisition, communication, and computation systems to create a flexible multiscale architecture, where all the components interoperate to maximize efficiency in terms of trade-off between latency, network utilization, energy constraints, and system performance. Intelligence, then, can permeate all the scales of the communication and computation system to enable flexible and adaptive operations targeted to city-wide tasks. The main features of the architecture are:

- *Computation-Aware Information Acquisition*—Edge computing will be the main engine of a distributed intelligent system for adaptive information acquisition. The objective is to preselect and compress data sources across sensor systems to minimize network load and energy expense. The key is to develop algorithms capable of locally removing information which is not needed to accomplish the global computational objective. Note that a similar rationale is used in current algorithms for the compression of multimedia streams, where "information" not usable by humans is removed. In the Smart City context, information irrelevant to the application algorithms can be eliminated. However, different from the former case, in the latter case the needed information is dependent on time-varying parameters such as the computational objective, and the state of the observed system.
- *Content- and Processing-Aware Networking*—In Smart City systems, the objective is to maximize the rate of usable data delivered to the computational resources performing the processing. Edge devices connected to local base stations and access points will enable the implementation of content and processing-aware resource allocation techniques and interference control mechanisms. Interference control can take the form of transmission power and rate control, or channel access control. In both cases, the network manager must be aware of the needs of the application algorithms.
- *Effective information availability*—Edge computing will place data at the edge of the network, thus improving local availability and searchability over a distributed and heterogeneous infrastructure. To this aim, portable semantic structures need to be maintained at all the scales of the system.

In this section, we present the components of the architecture and discuss our preliminary results.

6.3.1 Information Acquisition and Compression

The low-latency link between the edge resources and local sensorial systems enables system-wide messaging determined at the local network scale. The proposed edge-assisted architecture uses this messaging to empower the Urban IoT system with the ability to intelligently and adaptively select relevant data. We contend that the cooperation of devices at different scales is critical to achieve this objective. In fact, whereas IoT devices have all their individual data available, the edge processor may have compressed and corrupted versions of data from a multitude of individual sensors. Thus, the architecture needs to implement messaging to provide contextual information to individual sensors, which will evaluate the relevance of their local data and determine transmission decisions and data compression rate. The data selection and compression system can be seen as a distributed processing system, where heterogeneous agents cooperate to provide critical information to the final application. We logically divide this part of the architecture into adaptive compression and distributed computing components.

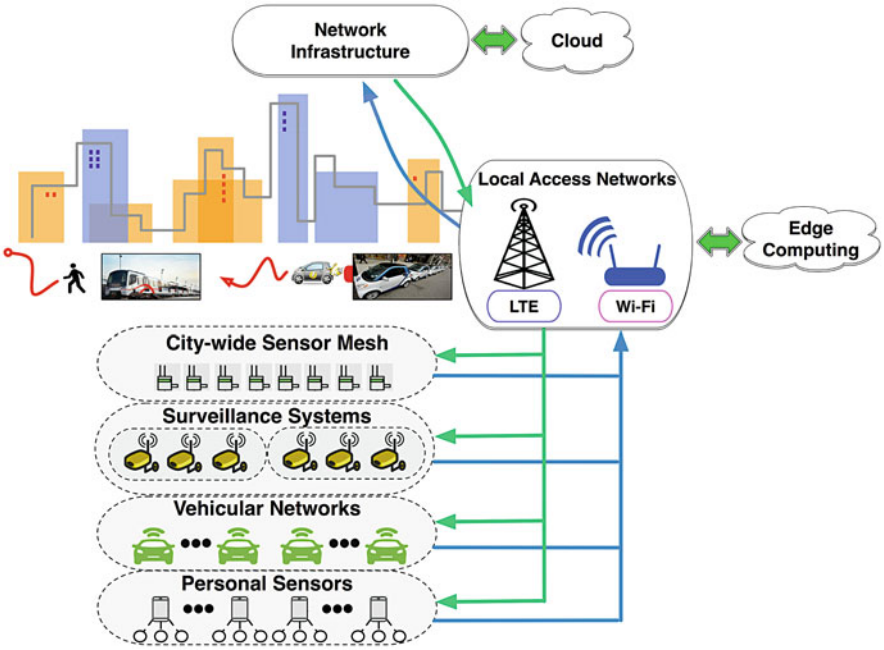

Fig. 6.2 The edge-assisted architecture interconnects information acquisition, communication infrastructures, and processing resources at multiple scales. *Blue* and *green arrows* represent the bidirectional exchange of data and control

Adaptive Compression: One of the key observations to make is that in a sensor-rich, and large-scale, environment where applications have specific computational tasks (e.g., detecting an event), not all the information is needed at the final controller. Conversely, in order to minimize network load and energy expense in the mobile and low-power devices, only necessary information should be pushed through the communication infrastructure (Fig. 6.2).

However, such selection and compression is challenging, due to the scale and heterogeneity of the system, which induces a mismatch between the information available to the individual sensors and the final controller. Being close to the network edge, edge computing can bridge these two extreme scales and support efficient and informed local data selection and compression.

Adaptive compression compresses the sensor streams based on their relevance or importance to the application goal. The incoming sensor stream data is filtered and compressed adaptively. The sensors themselves might lack the compute capability or the storage for prior trained models. However, they extract useful features from sensor streams and communicate them to an edge device. The edge device uses prior trained models to analyze the relevance or priority of the content and signals it back to the sensors for adaptive compression.

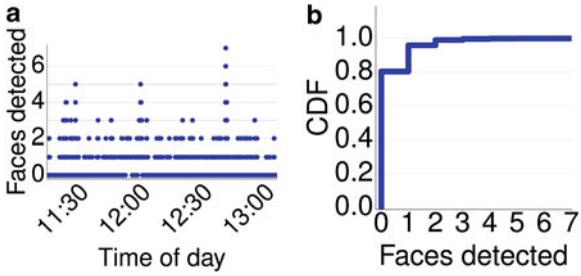

Fig. 6.3 Time series and cumulative distribution function (CDF) of a count of the number of faces a state-of-the-art vision algorithm detects during a busy period at an office building. (**a**) Face count time series. (**b**) Face count CDF

A system, called Vigil [8], illustrates this concept for video monitoring and real-time video surveillance applications. The application goal is to detect the times and detected faces in a busy office hall while the network bandwidth is constrained from the camera device to the central processor. Figure 6.3a shows the number of faces detected at the peak lunch hour, while Fig. 6.3b shows that less than 20% of the collected video in a busy office hall contains relevant information (e.g., moving objects), thus rendering its transportation to the central processor useless. This system uses this insight to implement adaptive compression where the edge device in conjunction with the camera nodes prioritizes the frames with detected faces while sending them over the network.

Note that adaptive compression schemes provide additional gains over MPEG-4 video compression schemes because they are content aware. While standard MPEG-4 video compressions work well for streaming to the web or television, they are not effective for application goals requiring computer vision because artifacts due to spatial and temporal compression impair the effectiveness of algorithms (e.g., object detection, classification, and tracking). Importantly, the activation of the sensor could depend on the state of the observed environment, where more information may be needed if contextual information changes (e.g., a partially hidden moving object is detected). Furthermore, edge computing can be used to interconnect heterogeneous sensor systems, so that bandwidth-demanding sensors (e.g., video capture) are activated based on information from low-bandwidth sensors (e.g., acoustic and motion sensors).

Additionally, the design of adaptive data representation and compression schemes is needed to make them more robust and resilient to the impairments of the wireless channel. This design should be driven by the final objective of processing, where more relevant features, determined based on contextual and local information, are more protected.

Distributed computing among edge devices: The information selection and compression architecture is based on the notion of distributed intelligence. The flexibility granted by edge computing architectures can play an important role in the realization of the envisioned architecture.

In edge computing, a network of sensors and edge devices can be leveraged to allocate computing functions based on their capabilities without requiring the individual sensors to share their sensor streams. This can be extremely valuable

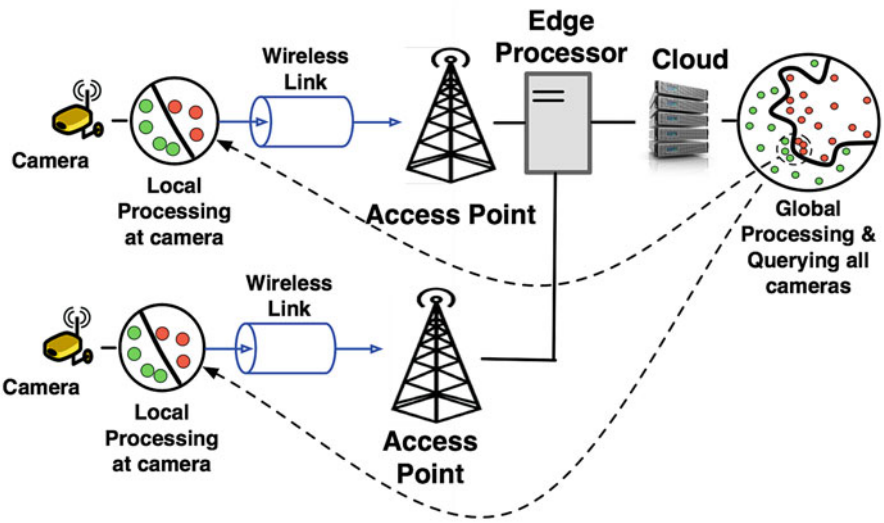

Fig. 6.4 Multi-camera system with distributed classifiers for context-aware local data filtering or selection

in scenarios where the local context can be better established from a group of edge devices and where the storage can be offloaded to a nearby device. The key advantage is that it eliminates the need of using the backhaul network to collect the state information.

At the same time, it brings two new aspects that often come up in distributed computing: *(a)* synchronization of different sensor streams to fuse or correlate them together when collected from devices with different clocks, and *(b)* the granularity of temporal scales to share the information between the edge devices. In the video monitoring example, the video streams largely need to be matched only with respect to detected events or objects when the goal is identifying anomalies or drawing useful insights.

Local computational resources can effectively interconnect individual sensors (e.g., cameras in multi-camera systems), enabling the pruning of these high-bandwidth data streams when only a subset of them is sufficient to perform the tasks dictated by the city applications. Vigil [8], a real-time distributed wireless surveillance, deploys distributed computing across cameras in addition to adaptive compression to improve the system performance. Figure 6.4 illustrates the general architecture. Vigil runs an intra-cluster algorithm across different cameras overlooking the same geographical area to determine the most valuable frames from cameras within a cluster and to eliminate redundant observations, capturing the same objects, to minimize communication bandwidth without actually exchanging the redundant frames. Figure 6.5 illustrates the gains of such an approach. Note that the bandwidth required at low activity level (at most 16 Kbit/s) is lower than the available per-camera wireless capacity and therefore, both Vigil and *Round-Robin* achieve more

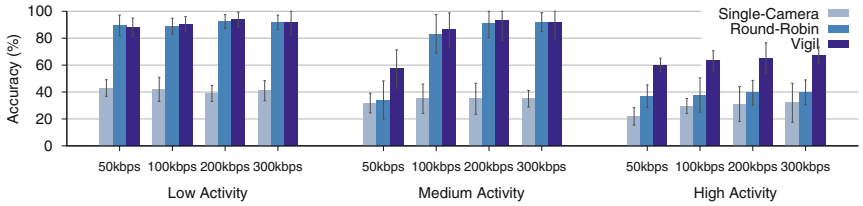

Fig. 6.5 Accuracy of intra-cluster frame selection in system name relative to a single-camera system and a multi-camera system with round-robin scheduling. Error bars show standard deviation of the experiment in varying wireless conditions

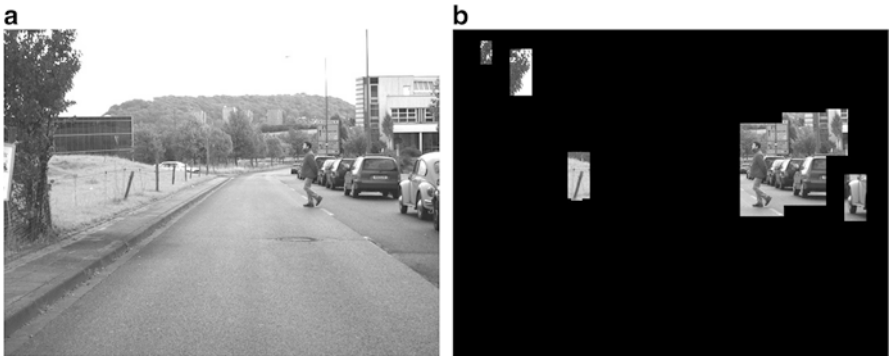

Fig. 6.6 Example of a frame filtered with a Haar feature-based pedestrian classifier. (**a**) Input frame. (**b**) Filtered frame

than 90% accuracy, while the single camera suffers because of lack of sufficient spatial coverage. Similar results are observed for medium activity level, except Vigil outperforms other approaches when the available per-camera wireless capacity 50 kbps is lower than the bandwidth required for medium activity level (at most 80 kbps). Finally at high activity level, the bandwidth required is much higher than the available per-camera wireless capacity and we observe 23–30% gains for Vigil compared to *Round-Robin* because Vigil prioritizes those frames across cameras that maximize the application accuracy.

Extending this reasoning, we studied the energy-bandwidth trade-off in an edge-assisted system, where the video acquisition device is capable of running a simple classification algorithm to eliminate redundant information within each frame. The objective, then, is to transmit to the edge processor only regions within the frames necessary to the global data analysis goal. In the considered setup, the device implements a cascade classifier [9] to select portions of individual frames containing objects of interest (e.g., pedestrians).

Figure 6.6 shows an example of frame before and after the classifier is applied. It can be seen that the classifier mistakenly locates pedestrians in portions containing other objects. This is due to the need to keep the classifier as simple as possible to run on devices with limited computational power while preserving the frame

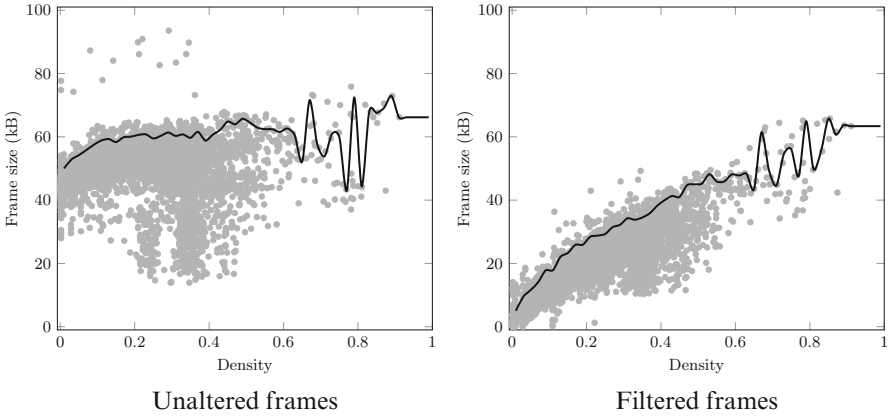

Fig. 6.7 Size of unaltered (**a**) and filtered (**b**) frames in compressed video stream as a function of their object density. *Black lines* depict the corresponding 95% quantiles

output rate. The false positives increase the bitrate requirements when the classifier is activated but do not harm the performance of the remote video processor, which will eventually exclude them by using a more powerful classifier. However, the classifier correctly includes the pedestrian in the picture.

Figure 6.7 depicts the size of the output frames after compression when the classifier is active and inactive. This measure is instrumental to the adaptation framework, as it allows individual sensors to estimate the bandwidth required by the two actions. It can be observed that when the density of objects in the picture is small, the output frame size is much smaller compared to the case when the classifier is inactive. In principle, a perfect classifier would filter out the entire picture when pedestrians are not present. In the practical classifier implemented in our study, when density is smaller than 0.2, the frame size is reduced by a factor of 2 to 10. When the density is large, the activation of the classifier does not correspond to a benefit in terms of output data rate, while increasing energy consumption. The solid lines correspond to the 95% quantile which is used for activation/deactivation decisions.

However, as shown in Fig. 6.8, the video streaming pipeline from the end-device including the classifier requires more power for a longer time with respect to the pipeline without the classifier. There is, thus, an important trade-off between energy spent by edge devices and bandwidth required to stream necessary information to edge and cloud resources. Object density also influences the number of operations necessary to process frames. The energy trade-off is obviously important when mobile devices are considered. However, the overall power consumed by sensors is certainly a central issue in city-scale architectures, where thousands of sensors are deployed.

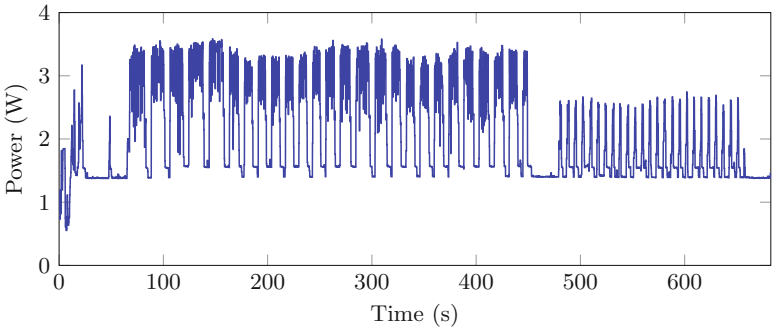

Fig. 6.8 Power consumed by a Raspberry Pi when streaming the video with and without classifier. The clusters of increased power corresponds to group of 60 frames. The first sequence of clusters corresponds to the case where the classifier is used, which increases both energy and execution time

Figure 6.9 depicts exemplar traces for the proposed bandwidth adaptive technique for the two loads. Red lines show the maximum bandwidth available to the end-device in each case, black lines show the actual bandwidth used by the end-device, and gray lines show the bandwidth necessary to transmit unfiltered stream. The strides where the end-device made decision to filter frames are highlighted in green. The effect of bandwidth variations on the output rate are apparent, where a bandwidth insufficient to support the predicted output rate for the current density leads to the activation of the classifier. This action causes visible reduction of the output rates in the plots.

Assistance by the edge processor, then, optimizes the energy-bandwidth tradeoff. In the simplest setup, the available bandwidth reported by the local network managers can be used as a constraint, which determines the activation and deactivation of the classifier at the end-device, and how many stages are used. This decision is assisted by the edge, which reports to the end-device the current object density—which can be computed only if the full classifier is used—to enable the prediction of future bandwidth and energy associated with the number of stages performed locally. Note that the local classifier can be re-programmed online to adapt to time-varying application objectives.

6.3.2 Content-Aware Wireless Networking

As discussed in the previous section, an efficient information acquisition architecture is a key component of the urban IoT architecture. However, the transportation of relevant information to the edge computing resource can be a challenging and delicate task due to the complexity and heterogeneity of modern communication infrastructures. Importantly, the urban IoT traffic will share the same network

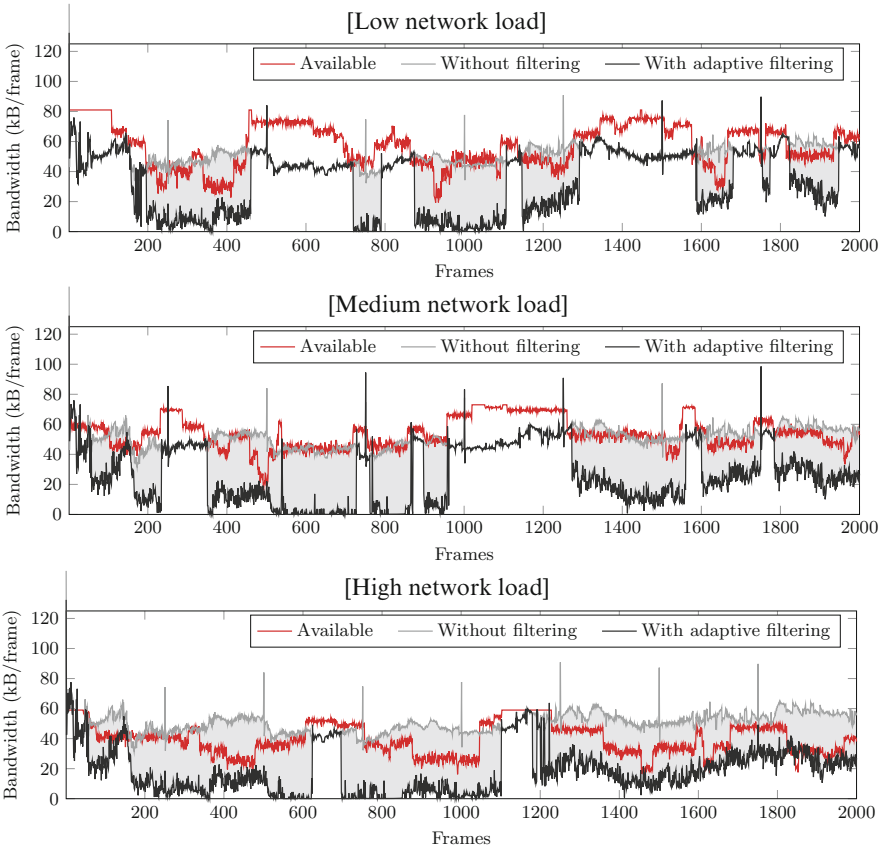

Fig. 6.9 Exemplars of bandwidth traces for streaming adaptively filtered frames in the scenarios of low (**a**), medium (**b**), and high (**c**) network load. *Red lines* show the maximum bandwidth available to the end-device in each case, *black lines* show the actual bandwidth used by the end-device, and *gray lines* show the bandwidth necessary to transmit unfiltered stream. The strides where the end-device decided to filter frames are highlighted in *light gray*

resource with that generated from traditional applications and services. Thus, the communication resources available to the urban IoT may vary over time and be scarce in peak hours. Furthermore, the coexistence of these data streams over a heterogeneous network sharing the same channel resource will make interference control difficult in the physical layer. Hence, we need smart and adaptive network management and aggregation techniques to effectively handle this difficult coexistence.

Mutual interference between information streams can be mitigated by designing access protocols generating specific *content-based* interference patterns. The main idea behind our approach is to make networking and transmission protocols *aware*

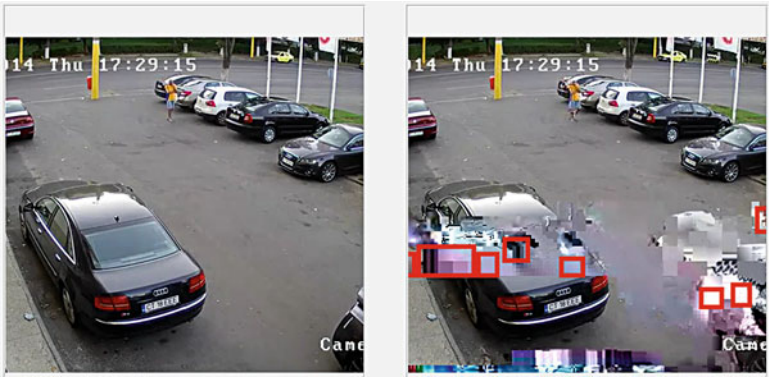

Fig. 6.10 In city-scale video processing, artifacts induced by spatial and temporal compression can severely impair the performance of detection and tracking algorithms

of the content being transported and the structural properties of its encoding. This minimizes the loss of relevant information to the processing algorithms given the current state of the observed system.

Our recent work [10, 11] demonstrated the effectiveness of information-centric techniques in coexistence scenarios, where Wi-Fi Device-to-Device (D2D) communications coexist with Frequency Division Duplex (FDD) Long-Term Evolution (LTE) cellular communications on the same bandwidth [12]. The application scenario is city monitoring, where video data streams from surveillance camera systems are processed by real-time edge computational resources to perform object detection and activity recognition [13]. Interference may cause artifacts that would significantly impair the performance of detection and tracking objects (see Fig. 6.10).

Current standards prescribe simple techniques to regulate coexistence in the unlicensed and licensed band. For instance, Wi-Fi and LTE coexistence is realized by implementing listen-before-talk mechanism, where one of the two technologies is prioritized by forcing the other idle when the former is active [14–17]. We contend that more flexible strategies are needed to support the operations of the urban IoT and facilitate coexistence with existing services. For the coexistence in licensed bands, recent work proposes scheduling and interference control strategies that limit the Signal-to-Interference-plus-Noise-Ratio (SINR) at the cellular base stations [18–20]. However, these techniques often require instantaneous channel knowledge and may result in packet loss when coordination between networks is not perfect. Our design revolves around the notion of utility of data within the stream, where utility variables are computed, exchanged, and processed by the cloud, edge, and end-device resources. The colocation of the edge computation resource and network controllers such as base stations and access points allows to establish a direct exchange of information between them. The edge, then, processes and communicates the utility to associated network controller, which determines

channel allocation and transmission strategy of the connected end-devices based on the current network state. However, the operations of this class of protocols require information about content being transmitted and processing state. To this aim, content and state information should be shared with network transmitters and resource management units to create content-specific interference patterns and resource allocation.

Conforming with the 3rd Generation Partnership Project (3GPP) standard for proximity services [21], we choose a topology where an end host is transmitting real-time data on the uplink of LTE to the Internet for computation and processing, and two mobile devices in proximity are connected with each other with network-assisted D2D communications. The LTE end-user is streaming real-time video on uplink by Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) toward the edge resource. The LTE base station (element Node B—eNodeB) scheduler allocates resource blocks for data transmission and assigns modulation and power based on channel quality and interference [22]. When the D2D communication interferes with the LTE uplink, the channel quality degrades and the LTE receiver will more likely fail to decode the packet. Note that interference from the D2D link may influence the modulation and transmission power of the User Equipment (UE).

Video compression techniques exploit spatial and temporal similarities in individual frames and across video. In the most efficient compression standard, H.264 creates Group of Pictures (GoP) composed of reference (I-Frame) and differential (P- and B-Frames) frames. Reference frames transport the whole picture, whereas differential frames encode differences with respect to the reference. When an encoded frame is damaged, due to spatial compression, it affects the transform coefficients which leads to the corruption in the decoded image. The spatial propagation of errors may create artifacts that are detected as objects or impair the ability of the algorithm to detect existing objects. If a reference frame is corrupted, the effect propagates through the entire GoP. When a differential frame is damaged, the effect is smaller compared to loosing part of a reference frame, as key features in the following frames may be recovered using the reference frame. In the proposed framework, we use a simple notion of utility based on frame class, where the edge decompresses the video stream prior to processing and signals to the eNodeB when a reference or differential frame begins. Based on this information and channel statistics, the eNodeB determines the transmission probability of the D2D link. Thus, channel access in the local network is regulated based on the transmitted data, and based on the feedback from the computation algorithm consuming the data stream.

Figure 6.11 shows object detection probability as a function of the throughput of the D2D link for fast and slow fading channels, where the data stream transports a video from a parking lot surveillance camera. The content-based transmission probability scheme (Frame Determined Transmission Probability—FDTP) is compared with the case where the D2D transmits with Fixed Probability (FP). In the considered case, video transmission consumes the entire LTE bandwidth, and the D2D link would starve if listen-before-talk is used. The lines are obtained by varying the transmission probabilities. For comparable object detection performance, the

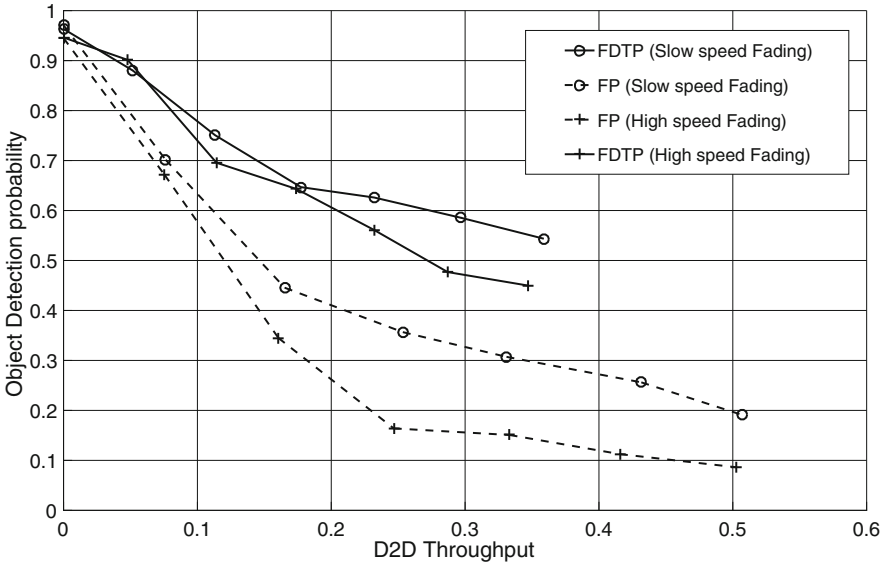

Fig. 6.11 Object detection probability vs D2D throughput in low speed and high speed fading scenario

FDTP scheme grants significant throughput increase to the D2D link, thus enabling coexistence on the same resource. Our study in [10] shows that the efficiency of coexistence, measured as application performance over throughput of the D2D link, is maximum in specific transmission power and probability regions.

6.3.3 Information Availability

While the sensors themselves can use the data they collect to make intelligent decisions, the edge devices and the cloud have access to a larger pool of sensor data from multiple sensors and at multiple timescales. Thus, the cloud and the edge devices can assist in making intelligent decisions that individual sensors might be incapable of making. Further, since the cloud has longer range information in temporal scales, it can understand the traffic patterns and other unexpected events such as roadwork or accidents, to plan more efficient paths in traffic monitoring scenarios.

Existing literature relies on aggregating the data in the cloud or a cluster of thousands of servers that can be indexed to enable structured or nonstructured data queries. The cloud-based models are heavily used for applications such as web search, advertising, social networking, and photo repositories to enable users to query data or draw insights. These models largely rely on distributed dataflow systems and programming models, e.g., Map-Reduce [23] and Spark [24]. Sensor

data and video data can be especially challenging to search in, for example, existing methods to analyze the video feeds in real time or post facto do not scale and are error-prone. The vision pipelines must be carefully handcrafted for cloud execution by the engineers requiring their focus on nitty gritty details such as how to parallelize, in which order to execute the modules, etc. Similarly, existing dataflow systems such as Spark require analogous handcrafting of user-defined modules as they lack query optimization. Supporting ad hoc queries or post facto analysis on stored video or scaling to a large number of cameras remain key open problems.

A recent research paper proposes a system Optasia [25] that brings together advances from two areas: machine vision and big data analytics systems. This convergence leads to an efficient query answering system over many cameras. The system demonstrates a modularized approach to building vision processing components for applications such as classifying vehicles by color and type, reidentifying vehicles across cameras, tracking lane changes, identifying license plates, etc. This modularized implementation allows the dataflow system to de-duplicate and parallelize the processing.

To address the challenge of scaling to a rich set of ad hoc queries and to many cameras, Optasia casts the problem as an application of a relational parallel dataflow system and wraps the above-described vision modules inside some well-defined interfaces (processors, reducers, and combiners). This makes querying efficient and fast for a city-wide deployment of cameras by decomposing the vision analytic tasks. Each vision module is expressed as a composition of the corresponding relational operators (select, project, aggregate, and Cartesian product). End-users simply declare their queries over the modules in a modified form of SQL. Then, the query optimizer reuses optimization rules and translates user queries into appropriate parallel plans over several different vision modules. The primary advantages of this combination are: (1) ease-of-use for end-users, (2) decoupling of roles between end-users and the vision engineers: the vision engineers can ignore pipeline construction and need only focus on efficiency and accuracy of specific modules, and (3) automatic generation of appropriate cloud execution plans that, among other things, de-duplicate similar work across queries and parallelize appropriately.

Evaluation on traffic videos from a large city on complex vision queries shows high accuracy for Optasia with many fold improvements in query completion time and resource usage relative to existing systems. Figure 6.12a plots the ratio of the query completion time for Optasia with query optimization against a version of Optasia that has no query optimization, for single queries on a parking garage video feed. We see that, with query optimization, Optasia is roughly 3× faster. Further, the query completion time of Optasia remains constant as dataset sizes increase illustrating the fact that the query optimization sets the degree-of-parallelism correctly. The large gains arise from de-duplicating the work in the vision modules (e.g., generating histogram-of-gradient features, etc.).

Within this area, several challenges must be solved to develop systems such that make information readily available at city-scale and can answer queries in real-time. In the multiscale edge architecture we proposed, in addition to analyzing sensor

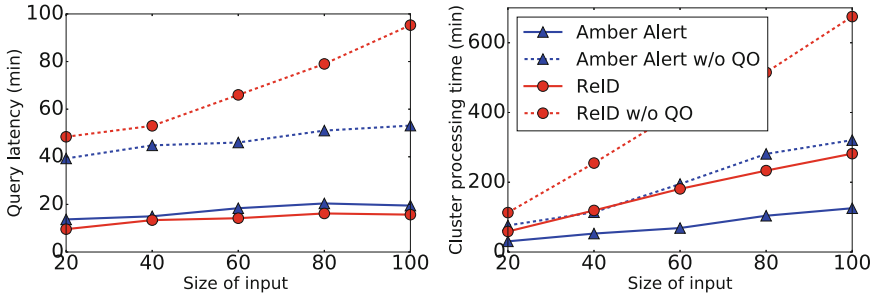

Fig. 6.12 In Optasia [25], Query Optimization reduces the query completion time significantly for both amber alert and Re-ID (**a**) as the number of input videos increases for each query. Further, query optimization ensures the most efficient cluster resource utilization in terms of processing time (**b**)

streams, the key challenge is that we need to search, identify anomalies, trigger alerts, and draw insights from the data collected over a multitude of edge nodes. To enable efficient ways to search over a multitude of distributed sensors, the key questions we must answer are: (a) How do we represent different spatial regions in the environment at the urban scale?, (b) How does the cloud obtain information about regions at different resolutions?, and (c) How do intermittently connected sensors transmit sensor information from these regions in a loss-resilient manner over the wireless channel?. One approach is similar to the compression approach used by Graphics community, where they use "Octree" to compress 3D content, especially point clouds to represent 2D spatial grids. Recent works [26] have shown that the approach extends to collecting and querying the sensory data collected by the self-driving cars. An alternate approach relies on Named Data Networking to name information objects and make them easy to query by other objects.

An additional challenge that arises in drawing insights from a multitude of sensors is to ensure appropriate access control mechanisms and respect data privacy where needed. Thus, computation techniques that are privacy-preserving, such as secure multiparty computation [27] or differential privacy [28], can be extremely useful in data aggregation over a variety of sources.

6.4 Related Work

Several architectures have been proposed which process Smart City data in the cloud [4]. Related to the application case discussed herein, in [29], an adaptive architecture to discover the topology of a distributed multi-camera system is presented. Mitton et al. [5] proposes a general cloud-based architecture for distributed sensor systems in the Smart City.

Edge and fog computing techniques have been considered to effectively reduce the load generated to the central communication and computation infrastructure [6, 7, 30, 31]. However, the solutions explored so far solely focus on data processing, without an in-depth analysis of information acquisition, representation, and transportation solutions needed to increase efficiency and achieve a sustainable technology.

Another line of work [32, 33] aims to reduce the latency of uploading data to the cloud, by partitioning computation tasks between mobile sensors and the cloud for personal mobile devices. Odessa [32] supports interactive perception applications by dynamically offloading parts of computation tasks from mobile devices to the cloud. A recent system Gabriel [33] targets a similar class of augmented reality applications based on a cloudlet architecture, which comprises computation devices located at the edge of network to reduce network latency.

Sensor selection in microscale sensor networks, e.g., see [34], and in-network compression, e.g., [35], have been the focus of intense work. However, we contend that these approaches do not directly apply to the urban IoT scenario. Although these works provide solid theoretical and system design basis, the involved multisystem, multiscale urban IoT architecture requires significant conceptual and practical advancements.

6.5 Conclusions

In this chapter, we proposed a novel architecture supporting urban IoT operations based on our prior results and experimental experience. One of the main contributions is the notion that the information acquisition, networking, and computation logical components of the urban IoT should be interconnected and conjointly operate to make city-wide applications feasible. The proposed architecture, then, is based on a notion of intelligence that pervades all the layers and devices operating in the urban IoT and uses edge computing as a key element to bridge the local fine-time scale of sensors to the coarser topological and temporal operations of the cloud. We introduced the notion of context and computation-aware data selection and compression to maximize the efficiency of the communicated data for specific applications and processing tasks. We introduced the concept of content-aware networking protocols that tune channel access and transmission based on the representation and relevance of the data travelling over the network. Finally, we argued that the presented layered architecture will facilitate information search and improve its availability.

References

1. A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of things for smart cities. IEEE Internet Things J. **1**(1), 22–32 (2014)
2. P. Neirotti, A.D. Marco, A. Cagliano, G. Mangano, F. Scorrano, Current trends in smart city initiatives: some stylised facts. Cities **38**, 25–36 (2014)
3. M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, R. Morris, Smarter cities and their innovation challenges. Computer **44**(6), 32–39 (2011)
4. M. Rahimi, J. Ren, C. Liu, A. Vasilakos, N. Venkatasubramanian, Mobile cloud computing: a survey, state of art and future directions. Mobile Netw. Appl. **19**(2), 133–143 (2013)
5. N. Mitton, S. Papavassiliou, A. Puliafito, K. Trivedi, Combining cloud and sensors in a smart city environment. EURASIP J. Wirel. Commun. Netw. **2012**(1), 1–10 (2012)
6. M. Satyanarayanan, The emergence of edge computing. Computer **50**(1), 30–39 (2017)
7. Openfog reference architecture for fog computing, produced by the openfog consortium architecture working group. [Online]. Available: https://www.openfogconsortium.org/ra/
8. T. Zhang, A. Chowdhery, V. Bahl, K. Jamieson, S. Banerjee, The design and implementation of a wireless video surveillance system, in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking* (ACM, New York, 2015), pp. 426–438
9. K.-D. Lee, M.Y. Nam, K.-Y. Chung, Y.-H. Lee, U.-G. Kang, Context and profile based cascade classifier for efficient people detection and safety care system. Multimed. Tools Appl. **63**(1), 27–44 (2013)
10. S. Baidya, M. Levorato, Content-based cognitive interference control for city monitoring applications in the urban IoT. IEEE Globecom 2016, Dec 4–8, Washington, DC, 2016
11. S. Baidya, M. Levorato, Content-based interference management for video transmission in d2d communications underlaying LTE, in *IEEE ICNC 2017*, Jan 26–29, Silicon Valley, 2016
12. K. Doppler, M. Rinne, C. Wijting, C.B. Ribeiro, K. Hugl, Device-to-device communication as an underlay to LTE-advanced networks. IEEE Commun. Mag. **47**(12), 42–49 (2009)
13. S. Hengstler, D. Prashanth, S. Fong, H. Aghajan, Mesheye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance, in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks* (ACM, New York, 2007), pp. 360–369
14. J. Jeon, H. Niu, Q. Li, A. Papathanassiou, G. Wu, LTE with listen-before-talk in unlicensed spectrum, in *2015 IEEE International Conference on Communication Workshop (ICCW)* (IEEE, New York, 2015), pp. 2320–2324
15. R. Ratasuk, N. Mangalvedhe, A. Ghosh, LTE in unlicensed spectrum using licensed-assisted access, in *2014 IEEE Globecom Workshops (GC Workshops)* (IEEE, New York, 2014), pp. 746–751
16. A. Mukherjee, J.-F. Cheng, S. Falahati, L. Falconetti, A. Furuskär, B. Godana, H. Koorapaty, D. Larsson, Y. Yang et al., System architecture and coexistence evaluation of licensed-assisted access LTE with IEEE 802.11, in *2015 IEEE International Conference on Communication Workshop (ICCW)* (IEEE, New York, 2015), pp. 2350–2355
17. R. Ratasuk, M.A. Uusitalo, N. Mangalvedhe, A. Sorri, S. Iraji, C. Wijting, A. Ghosh, License-exempt LTE deployment in heterogeneous network, in *2012 International Symposium on Wireless Communication Systems (ISWCS)* (IEEE, New York, 2012), pp. 246–250
18. P. Phunchongharn, E. Hossain, D. Kim, Resource allocation for device-to-device communications underlaying LTE-advanced networks. IEEE Wirel. Commun. **20**(4), 91–100 (2013)
19. C. Yu, O. Tirkkonen, K. Doppler, C. Ribeiro, On the performance of device-to-device underlay communication with simple power control, in *IEEE 69th Vehicular Technology Conference*, pp. 1–5, 2009
20. Y. Wen-Bin, M. Souryal, D. Griffith, LTE uplink performance with interference from in-band device-to-device (D2D) communications, in *IEEE Wireless Communications and Networking Conference*, pp. 669–674, March 2015

21. 3GPP TR 36.843 feasibility study on LTE device to device proximity services - radio aspects (2014)
22. European Telecommunications Standards Institute, E-UTRA physical layer procedures, Generation Partnership Project Technical Specification (3GPP TS) 36.213, V.10, 2011
23. D. Jiang, B. Ooi, L. Shi, S. Wu, The performance of mapreduce: an in-depth study. Proc. VLDB Endow. **3**(1), 472–483 (2010)
24. M. Armbrust et al., Spark SQL: relational data processing in spark. in *SIGMOD* (2015)
25. Y. Lu, A. Chowdhery, S. Kandula, Visflow: a relational platform for efficient large-scale video analytics, in *ACM Symposium on Cloud Computing (SoCC)* (ACM, New York 2016)
26. S. Kumar, L. Shi, N. Ahmed, S. Gil, D. Katabi, D. Rus, Carspeak: a content-centric network for autonomous driving. SIGCOMM Comput. Commun. Rev. **42**(4), 259–270 (2012) [Online]. Available: http://doi.acm.org/10.1145/2377677.2377724
27. C.-T. Chu, J. Jung, Z. Liu, R. Mahajan, sTrack: secure tracking in community surveillance, in *Proceedings of the 22nd ACM International Conference on Multimedia*. MM '14, pp. 837–840, 2014
28. C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, M. Naor, Our data, ourselves: privacy via distributed noise generation, in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT'06, 2006
29. Y. Wen, X. Yang, Y. Xu, Cloud-computing-based framework for multi-camera topology inference in smart city sensing system, in *Proceedings of the 2010 ACM Multimedia Workshop on Mobile Cloud Media Computing* (ACM, New York, 2010), pp. 65–70
30. M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, B. Amos, Edge analytics in the internet of things. IEEE Pervasive Comput. **14**(2), 24–31 (2015)
31. F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*. MCC '12, pp. 13–16, 2012
32. M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, R. Govindan, Odessa: enabling interactive perception applications on mobile devices, in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*. MobiSys '11 (ACM, New York, NY, 2011), pp. 43–56. [Online]. Available: http://doi.acm.org/10.1145/1999995.2000000
33. K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, M. Satyanarayanan, Towards wearable cognitive assistance, in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys '14, 2014, pp. 68–81
34. U. Mitra, B. Emken, S. Lee, M. Li, V. Rozgic, G. Thatte, H. Vathsangam, D. Zois, M. Annavaram, S. Narayanan et al., Knowme: a case study in wireless body area sensor network design. IEEE Commun. Mag. **50**(5), 116–125 (2012)
35. G. Quer, R. Masiero, G. Pillonetto, M. Rossi, M. Zorzi, Sensing, compression, and recovery for WSNs: sparse signal modeling and monitoring framework. IEEE Trans. Wirel. Commun. **11**(10), 3447–3461 (2012)