Software-defined and Value-based Information Processing and Dissemination in IoT Applications

Mauro Tortonesi[1], James Michaelis[2], Niranjan Suri[2,3], Michael Baker[2]
[1]Department of Engineering, University of Ferrara
Ferrara, Italy
mauro.tortonesi@unife.it
[2]United States Army Research Lab
Adelphi, MD, USA
james.r.michaelis2.civ@mail.mil, niranjan.suri.ctr@mail.mil, michael.a.baker.mil@mail.mil
[3]Florida Institute for Human and Machine Cognition
Pensacola, FL, USA
nsuri@ihmc.us

Abstract — **In the near term, a multitude of IoT applications are expected, each taking advantage of heterogeneous device collections ranging from environmental sensors to smartphones. However, approaches taken in many IoT systems – based on the paradigm of Cloud computing – face challenges of both high latency and network utilization. A clear demand now exists for new paradigms to facilitate IoT application usage of computational resources at the edge of the network for data analysis purposes, as well as smart dissemination solutions to deliver information to consumers. This paper presents *SPF* (Sieve, Process, and Forward), a Software Defined Networking (SDN) solution for creating and managing IoT applications and services. By leveraging programmable information processors deployed at the Internet/IoT edge, the SDN approach introduced by SPF represents a promising architecture for future urban computing applications.**

Keywords—Internet of Things; Software Defined Networking; Information Dissemination; Value of Information.

I. Introduction

In coming years, a multitude of Internet of Things (IoT) applications are expected to be deployed, each designed to process data from heterogeneous devices ranging from environmental sensors to smartphones. In-line with multiple visions for smart city infrastructure [1], IoT systems are increasingly being researched and tested. However, significant computational and networking challenges unique to urban computing environments remain to be addressed [2].

From an IoT application and infrastructure engineering perspective, management of data deluge is an issue that impacts most aspects of development. Design approaches based on Cloud computing – the transmission of IoT data to Cloud-based services for processing – present several drawbacks. These include potentially high latency in data transmission to and from Cloud services, high storage requirements in the Cloud itself, as well as overall heavy network utilization.

Additionally, given that IoT-based ecosystems can be viewed as an example of a “Big-Data” application, they may be susceptible to the “diamonds in the raw data” fallacy [3]. Rather than attempt to store and process *all data* available in an IoT ecosystem, in hopes that *most data* will reveal important insights, more practical approaches will likely rely on s*elective filtering* to aid in data storage and dissemination.

This paper introduces *SPF* (Sieve, Process, and Forward), a proposal to support IoT applications based on a Software Defined Networking (SDN) approach, on the adoption of robust and disruption-tolerant information dissemination solutions that can naturally take advantage of the opportunities presented by heterogeneous networks and Device-to-Device (D2D) communications, and on the adoption of Value-of-Information (VoI) [4] concepts for the prioritized transmission of critical information.

By leveraging programmable information processors deployed at the Internet/IoT edge, VoI based prioritization, and disruption tolerant information dissemination solutions, the SDN approach introduced by SPF allows for easy development, deployment, and management of IoT applications in urban computing environments.

II. Scenarios in Urban Computing Environments

For IoT systems and applications, challenges in communications management will emerge from complexity of mobile devices as well as heterogeneous and dynamic wireless networking environments. In those networks, applications need specific middleware support to automatically take advantage of mobile offloading opportunities, e.g., switching from 4G/LTE to WiFi communications in case an open hotspot is detected [5].

Device-to-device (D2D) communications represent a very important aspect of this scenario [6]. In cellular networks, peer-to-peer connections between mobile devices can be a very effective way to bypass the potential bottleneck at the eNB base station, also enabling spectrum reuse, energy savings, and extended network coverage [7]. However, even with D2D

978-1-5090-0223-8/16/$31.00 ©2016 IEEE

communications, generated data will still put a significant burden on networks. In particular, networks will have to accommodate larger quantities of data, paving the way both to mobile crowdsensing [8] and anticipatory mobile computing [9].

An alternative strategy, termed "Fog Computing" [10], involves pushing computational resources towards the edge of the network, thus enabling low-level data processing via available devices such as event detection and data aggregation. However, the quick evolution of hardware enables even more agile, ad hoc solutions – essentially based on the elastic allocation of virtual resources, i.e., VMs, in a Cloud data center. Modern hardware solutions like neuromorphic processors (such as IBM's True North Chip[1]) and hybrid CPU/manycore (such as Adapteva's Parallela board[2]) or CPU/FPGA architectures (such as Xilinx's Zynq-7000 SoC[3]) enable the deployment of energy efficient computational power capable of supporting data aggregation and processing tasks. Nonetheless, leveraging those resources will require more dynamic approaches.

Our proposal follows a different approach, based on the realization that not all raw data have the same importance and that implementing dynamic data filtering at the Internet/IoT edge presents interesting opportunities [11] and building on existing visions for smart city applications [1]. In this context, we now take a look at a possible near future scenario in which IoT applications are used to support participants and workers at a large street music festival. The music festival is an event involving a relatively large number of street artists performing in an urban environment, attracting a significant number of people. The aim of this scenario is twofold: First, to highlight IoT infrastructure challenges expected in urban environments; and second, to frame discussion on how these challenges can be addressed.

At the music festival, four groups of people are in attendance: *Participants*, *Vendors*, *EMS Personnel*, and *Police*. Participants are regular attendees that will be interested in performances as well as vending options for both refreshments and merchandise. Likewise, vendors will want to track varying kinds of participant activity to plan their sales activities, like having food cooked once a nearby performance lets out. Both EMS personnel and police will be concerned with maintaining safety and security of the event. More specifically, EMS personnel may be interested in tracking where large gatherings occur, to ensure appropriate resource allocation (e.g., deployment of water, cooling centers, and staff, and management of emergency requests). Likewise, police personnel will need to be aware of criminal incidents on site, such as public drunkenness and assaults. Additionally, police could be interested in incident indicators, such as large vehicles with tinted windows which could conceal weapons or explosives.

Through the requirements of this scenario, as well as technical challenges raised for IoT infrastructures in urban computing environments, we see Software Defined Networking (SDN) approaches as representing a very promising research direction.

III. The SPF Solution

To address requirements of the street music festival, as well as similar IoT applications, we propose SPF (Sieve, Process, and Forward), an SDN-based solution that enables information filtering and dissemination. The SPF architecture, depicted in Fig. 1, is based on a centralized controller and on the deployment of SPF Programmable IoT Gateways (PIGs) at the 6LoWPAN/Internet edge of the network.

SPF centers on *services*, or functions implemented through the processing and dissemination of IoT data. Services can be activated on-demand by users, and can include audio/video analysis, as well as object tracking and/or counting. In turn, *IoT applications* are defined as collections of related services with the same priority and the same target users.

The primary function of the SPF Controller is to receive service requests and deploy corresponding services. Additionally, the SPF Controller allows administrators (or *managers*) to define IoT applications that provide a set of services. Multiple IoT applications can be supported simultaneously, each having different priority levels and catering to different users. In turn, PIGs enable information processing and dissemination strategies to be deployed based on developer-generated routines.

In order to filter information objects, SPF uses a minimum content difference threshold for new IoT data to be considered for processing and dissemination. In addition, SPF prioritizes dissemination of critical information by ranking objects according to their corresponding Value of Information (VoI) metric [4].

PIGs can be deployed directly on the gateway nodes that connect 6LoWPAN networks to the Internet or on dedicated hardware placed in the gateway nodes' proximity. For simplicity, the rest of this paper will assume the usage of reasonably powerful gateway nodes, each capable of hosting an SPF PIG software component.

It is expected that PIGs will have access to multiple communication modalities. For urban computing applications, these will likely be Wi-Fi, 4G/LTE, and IEEE 802.15.4, but occasionally also Bluetooth-LE and NFC. Of these, 4G/LTE is the only infrastructure-based modality available, allowing reliable communications with the SPF Controller for reconfiguration purposes. However, 4G/LTE can be resource intensive and hence not ideal for battery-operated devices (e.g., deployed PIGs). To address tradeoffs between communications reliability and resource cost, SPF enables information dissemination over short-range device-to-device communications when possible which, albeit less reliable, represents a solution particularly well suited for urban computing environments with high node density and mobility.

A current open-source prototype implementation of SPF, written in Ruby and released under the MIT license, is available for download on the GitHub website at: `https://github.com/mtortonesi/spf`

[1] http://www.research.ibm.com/articles/brain-chip.shtml

[2] http://www.adapteva.com/parallella-board/

[3] http://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html

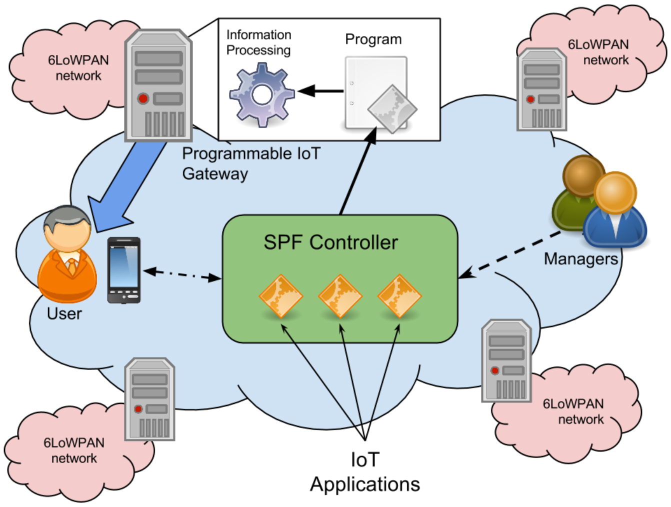

Fig. 1. The SPF Architecture.

IV. SPF Controller and IoT Application Definition

The SPF Controller contains a dedicated Orchestrator component for each of the IoT applications. Each of the Orchestrators receives input from administrators (*application configuration*) and users (*service requests*). In case one of the Orchestrator reports a state change, e.g., due to a new information processing policy issued by the managers or a spike in user requests for a specific service, the SPF Controller reprograms the Programmable IoT Gateways (PIGs) accordingly.

A. Definition of IoT Applications and Services

The SPF Controller enables managers to define IoT applications and the corresponding services they implement using a dedicated Domain Specific Language (DSL). Each application has several aspects that can be configured, such as: a *name*, a *priority level* (between 1 and 100), a set of *allowed service types* provided to the users, and a set of *service configurations* and *dissemination policies*. SPF enables IoT applications to receive specific priority levels in order to differentiate between critical and best-effort applications. Service configurations control how the application deals with user service requests of the corresponding type. Dissemination policies instead control all the configurable aspects involved in the information dissemination process, such as: the dissemination channel, the maximum transmission frequency, the maximum number of retransmissions, the obsolescence management policy for the information objects managed by the application, etc.

For instance, in the Street Music Festival scenario from Section II, the applications could be configured as shown in Fig. 2. This figure presents the configuration of 3 different IoT applications, to support the needs of festival participants, EMS personnel, and police forces respectively. The SPF Controller receives service requests from users and instantiates the corresponding services accordingly. This allows for dynamic behavior that triggers the execution of information processing and dissemination only when they are actually needed.

```
application “participants”, priority: 50.0,
  allow_services: [ :find, :listen ],
  service_policy: {
    find: {
      uninstall_after: 2.minutes
      distance_decay: { type: :exponential,
                        max: 1.km },
      filtering_threshold: 0.05 },
    listen: {
      require_processing: :identify_song,
      time_decay: { type: :linear,
                    max: 2.minutes } } },
  dissemination_policy: {
    subscription: “participants”,
    retries: 1, wait: 30.seconds,
    on_update: :overwrite,
    allow_channels: :WiFi }

application “ems”, priority: 90.0,
  allow_services: :all,
  dissemination_policy: {
    subscription: “ems”,
    retries: 3, wait: 20.seconds,
    on_update: :overwrite
    allow_channels: [ :WiFi , :cellular ],
    channel_policy: {
      WiFi: {
        transmission_frequency: 100.0 },
      cellular: {
        transmission_frequency: 20.0 } } }

application “police”, priority: 100.0,
  allow_services: :all,
  dissemination_policy: {
    subscription: “police”,
    retries: 60, wait: 10.seconds,
    on_update: :remind_old_value,
    allow_channels: [ :WiFi , :cellular ] }
```

Fig. 2. Example of IoT application configuration sent from the SPF controller to the IoT programmable information processors for the “Street music festival” reference scenario.

Upon receipt of service requests from users, the SPF Controller updates the state of the service and consequently contacts the PIGs to update their configuration. More specifically, the SPF Controller provides updated information about the number of requests for a given service and the location of the service user in closest proximity to the gateway, and changes the configuration of the information dissemination channels in response to a sudden spike (or decrease) in the number of service users.

Fig. 3 shows examples of service requests for the Street Music Festival scenario. Here, participants are interested in finding information about where they can purchase water and which songs are being played in their proximity. Likewise, EMS personnel are interested in counting the number of participants going towards the festival location, in order to plan for the needed amount of resources. Finally, the police force wants to count the number of large vehicles going towards the festival location. Once these requests reach the SPF Controller, they are processed and trigger a consequent reprogramming of the Programmable IoT Gateways.

```
find "water"

listen sample: 5.seconds, every: 3.minutes,
  processing: :identify_song
```

```
count_objects type: :person,
  direction: :inbound
```

```
track_objects type: :vehicle,
  size: :large, direction: :inbound
```

Fig. 3. Example of services for the Street Music Festival reference scenario, for the "participants", "ems", and "police" applications respectively.

V. Value of Information-based Prioritization

To help guide information dissemination – both within and across IoT applications – SPF leverages concepts from Value of Information (VoI) research to aid in content filtering and prioritization in IoT applications. Specific VoI strategies SPF borrows from associate a dynamic and recipient-specific value to each piece of information under consideration for dissemination (e.g., [4]).

As an example from the Street Music Festival scenario, consider a high-quality (e.g., high resolution) image of a van with tinted windows pulling up near a performance stage. While this image may have low value to participants, it could have a much higher value for police and security forces. Since the quality of this image can impact its ultimate value, QoI has been considered by [4] as a key part of establishing VoI.

SPF calculates VoI according to different factors, such as the priority of applications, the number of requests for a specific type of information, the timeliness relevance of the information object with respect to the request origination time, and the proximity relevance of an information source location with respect to the requestor.

VI. Information Processing and Dissemination

The Programmable IoT Gateway (PIG) is the component of SPF in charge of information processing and dissemination. The PIG behavior is fully programmable according to the instructions received by the SPF Controller. The rest of this Section describes the internal operations carried on by the three main software modules of a PIG: the Information Processing, Information Dissemination, and Programmable Controller components.

The processing of data is performed by several *information processing pipelines* that are set up by the Programmable Controller, according to the instructions received by the SPF Controller. Pipelines can be shared between services and have the purpose of analyzing raw data to obtain higher-level information. Such services could include object tracking and/or counting, optical character recognition, speech-to-text, and so on.

After the pipeline processing, the message containing raw data is transformed into an Information Object (IO). The Information Processing component calculates the VoI of the IO for each service and hands the IO and the corresponding VoI to the Information Dissemination component.

Once they are ready for dissemination, IOs are transferred to the corresponding *service dispatcher*. Service dispatchers are entities that take charge of the information dissemination process for a specific service. Each service dispatcher maintains a transmission queue, where IOs are ordered according to their corresponding VoI score. IOs with higher VoI scores are transmitted first and those with lower VoI scores are transmitted later.

Service dispatchers can transmit IOs over different communication channels. In fact, the way the service dispatchers use these devices represents one of the most important policies for IoT service definition.

SPF leverages the disruption-tolerant information dissemination functions provided by the DisService component[4] [12] [5], using a dedicated DisService subscription for each IoT application with tunable parameters.

During the execution of IoT applications and services, the Programmable Controller can (and typically will) receive additional requests for a new service installation or for the reconfiguration of an existing service, say if a much larger pool of users ends up using that service. In this case, the SPF Controller will reconfigure PIGs accordingly, activating new pipelines (and instantiating the corresponding software components that implement them) and deactivating unused ones if necessary.

VII. Conclusions and Future Work

SDN, already very popular in network management, and VoI are concepts that represent an interesting basis for the realization of sophisticated programming and management solutions for IoT applications and services. The approach presented in this paper demonstrated that, paired with programmable information processors deployed at the Internet/IoT edge and on disruption tolerant information dissemination solutions, SDN represents a very promising architecture for future urban computing applications.

Acknowledgments

The research work described in this paper was conducted in large part made while Dr. Tortonesi was at the Adelphi Laborator Center of the United States Army Research Laboratory (ARL) as a participant of the ARL's Open Campus program.

References

[1] J. Gubbi, R. Buyya, S. Marusic, M- Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions", *Future Generation Computer Systems*, Vol. 29, No. 7, pp 1645-1660, September 2013.

[2] Y. Zheng et al., "Urban Computing: Concepts, Methodologies, and Applications", *ACM Transactions on Intelligent Systems and Technologies*, Vol. 5, No. 3, Article 38, September 2014.

[3] N. Silver, "The Signal and the Noise: Why So Many Predictions Fail - But Some Don't", Penguin Press, 2012.

[4] The implementation of DisService is open source and available at https://github.com/ihmc/nomads

[4] N. Suri et al., “Exploring Value of Information-based Approaches to Support Effective Communications in Tactical Networks”, *IEEE Communications Magazine*, Vol. 53, No. 10 (Special Feature on Military Communications), pp. 39-45, October 2015.

[5] G. Benincasa et al., “Agile Communication Middleware for Next-generation Mobile Heterogeneous Networks”, *IEEE Software*, Vol. 31, No. 2 (Special Issue on Next Generation Mobile Computing), pp. 54-61, March-April 2014.

[6] A. Asadi, Q. Wang, V. Mancuso, “A Survey on Device-to-Device Communication in Cellular Networks”, *IEEE Communications Surveys & Tutorials*, Vol.16, No.4, pp.1801-1819, Fourth quarter 2014.

[7] X. Lin, J. Andrews, A. Ghosh, R. Ratasuk, “An overview of 3GPP device-to-device proximity services”, *IEEE Communications Magazine*, Vol.52, No.4, pp.40-48, April 2014.

[8] G. Cardone, A. Corradi, L. Foschini, R. Ianniello, “ParticipAct: a Large-Scale Crowdsensing Platform”, to appear in *IEEE Transactions on Emerging Topics in Computing*, 2016.

[9] V. Pejovic, M. Musolesi, “Anticipatory Mobile Computing: A Survey of the State of the Art and Research Challenges”, *ACM Computing Surveys*, Vol. 47, No. 3, Article 47, April 2015.

[10] 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)*, New York, NY, USA, pp. 13-16.

[11] A. Papageorgiou, B. Cheng, E. Kovacs, “Real-Time Data Reduction at the Network Edge ofInternet-of-Things Systems”, in *Proceedings of 11th International Conference on Network and Service Management (CNSM)*, 2015.

[12] N. Suri et al., “Peer-to-Peer Communications for Tactical Environments: Observations, Requirements, and Experiences”, *IEEE Communications Magazine*, Vol. 48, No. 10 (Special Feature on Military Communications), pp. 60-69, October 2010.