# An Agent-Based Network Resource Management Concept for Smart City Services

Doruk Sahinel[1], Cem Akpolat[1], Fikret Sivrikaya[1], and Sahin Albayrak[2]

[1]GT-ARC gemeinnützige GmbH, Berlin, Germany
[2]DAI Labor, Technische Universität, Berlin, Germany
*doruk.sahinel@gt-arc.com*

***Abstract*— Massive data generation with the introduction of IoT devices and new technologies in smart cities require a flexible network management architecture to meet the dynamic demands of smart city services. In project ISCO (Intelligent Framework for Service Discovery and Composition), we envision a smart city network operator that is tenant to multiple smart city services, and it can provide network resources and network functions on-demand to satisfy the connectivity requirements of these services. To explain this structure, we present an augmented sightseeing scenario that reflects various smart city service requirements. Then, we give an insight into the technological enablers of ISCO network management platform and explain how these technologies can satisfy the needs of the smart city scenario. Finally, we briefly present our ISCO network architecture and network resource management approach, which uses a distributed game-theory based learning algorithm to solve a multi-agent decision problem, and discuss how autonomous network management concept can be used to optimize resource allocation in our smart city communication platform.**

***Index Terms*—IoT, Smart City, Software Defined Networking, Fog Computing, Autonomous Network Management**

## I. Introduction

As 5G standardization efforts of industry and research efforts in academia for the next generation mobile networks speed up, more and more works that aim to bring novel smart city concepts into life emerge. ISCO project is part of these research efforts, and its main motivation is to establish a smart city framework that connects different Internet of Things (IoT) services with seamless service orchestration, and matching mechanisms based on each service's defined optimality criteria. By taking network-as-a-service concept into account, network is considered as another service among the many services in ISCO, and represented with an agent. This network agent communicates with other smart city services to provide them network slices on-demand. As defined in [1], network slicing divides a physical network into logical networks, and network resources can be dynamically allocated to logical network slices based on their Quality of Service (QoS) demands. Built on an agent-based distributed platform and a game-theory based resource allocation algorithm, ISCO enables other smart city services to become actively involved in resource allocation decisions.

Network capabilities targeted by ISCO framework will be implemented with the main aim of enabling the communications between smart city objects and service endpoints. Establishing such a communication platform requires an analysis of novel challenges that smart city IoT use-cases are likely to bring to network infrastructures. Heterogeneous wireless networks with dense small cells and wireless IoT sensor networks are also expected to bring new challenges to smart city network management. ISCO network management architecture needs to tailor network solutions that tackle the problem of building an efficient resource allocation algorithm by considering these challenges.

The rest of the paper is organized as follows: In Section 2, we describe a smart city scenario to exemplify the requirements of network management in smart cities. We give an insight into how network slicing concept can realize such scenarios by connecting novel technologies such as Java-based Intelligent Agent Componentware (JIAC), Software Defined Networking (SDN), Network Function Virtualization (NFV) and Fog Computing to respond to these challenges in Section 3. We then thoroughly explain our ISCO network management framework and we identify how an autonomous network management framework can address the challenges of smart cities, with the aim of bringing the right level of abstraction to parameter optimization in Section 4. The proposed resource management algorithm is briefly explained in Section 5, and the paper is concluded in Section 6.

## II. Augmented Sightseeing Use Case

Before explaining our proposed ISCO management framework, we present a smart city scenario inspired by METIS-II service types [2] that reflects various aspects of service requirements. As expected from a smart city infrastructure in the 5G era, we highlight the need to support a wide range of services such as extreme mobile broadband (xMBB), massive machine type communication (mMTC), and to enable their coexistence with on-demand resource provisioning and cooperative decision-making.

Berlin attracts millions of tourists each year and city bus tours are popular among them. However, sometimes it might be difficult for these tourists to envision how exactly historical events took place in the city. Suppose that, with a "Time Travel" city tourism application, Berlin smart city allows tourists to create a time travel illusion around their environment through augmented reality glasses, which displays the

ISBN 978-3-903176-02-7 

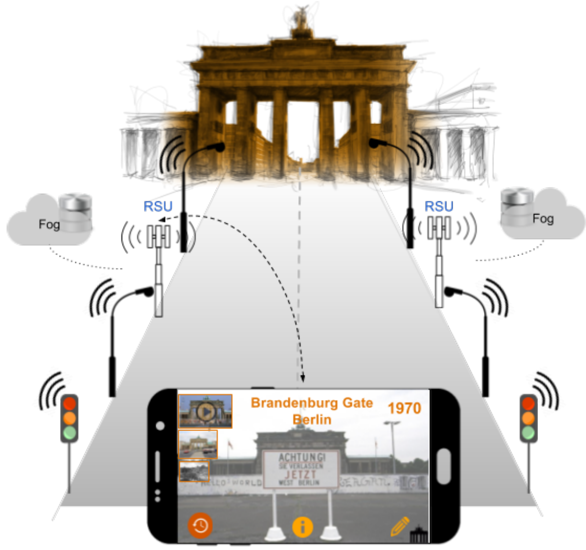


Figure 1: Augmented Sightseeing Scenario for Berlin Smart City

main sights in vicinity with images, video and text in real-time from any desired time period in history. The application is visualized in Figure 1.

To realize this on-demand tourist experience, various network requirements must be satisfied. Firstly, the capacity provided to each application user must significantly increase. In addition, for example, in a case when the information given for a monument is not placed exactly on top of this monument but somewhere else, user experience will be highly degraded, meaning that the latency must be at the ultra-low level. All these requirements mean that augmented reality application can be considered as a xMBB use case, and it needs a network slice that consumes many resources with excellent QoS demands. This can be interpreted as a stress test for increasing demand over the network management architecture. We assume that ISCO network infrastructure belongs to the city and services are run on top of this infrastructure, meaning that we have a single operator use-case.

Imagine that other mMTC services such as air quality sensors, parking sensors, etc. are active in the region as separate services. Reliable connections are also needed for a large number of widespread low-powered nodes and their slices should also be managed. Now we have a two-sided problem in this scenario. On the one hand, a guaranteed QoS must be provided to a bandwidth-hungry and delay sensitive application that request a network slice on-demand. On the other hand, QoS of all other smart city services in the same physical location must not be affected from the huge increase in demand. We believe that using intelligent agents for smart city services in a distributed fashion can help them implement their own policies and optimization algorithms and give them the capability to adjust themselves under varying demand conditions based on the QoS feedback from their users. For example, a smart city service can dynamically monitor the network and send only prioritized data packets to the data center when the demand in the area is high, and send the whole data packets when the demand is low. The service can also communicate with other agents to receive their bandwidth data and ask the network operator to change its data flow path if other services are using more bandwidth.

## III. ISCO Network Layer Technological Enablers

To respond to the requirements emphasized in Section II, here we introduce the available tools, technologies and protocols that we employ in our concept of IoT communication in smart cities.

*1) JIAC:* In [3], a service is defined as "a component that performs a granular and self-contained function that can be invoked by external clients through interfaces." JIAC is an open source multi-agent framework that can serve as the base for a service-oriented distributed architecture and provide the needed functions via agent beans and the well-defined interfaces for communication between ISCO smart city services [4]. For example, in our use case, if the bandwidth provided to a service is low, JIAC service agent can communicate with other agents to compare its utility function with them.

*2) Software Defined Networking:* The complexity and dynamic topology changes in smart city networks require quicker network reconfiguration, and SDN can respond to this by separating control and data planes and by dynamically adjusting flow rules with programmable network elements [5]. Thanks to this decoupling, SDN can change the network behavior for services by sending control messages to the devices (routers, switches) without interrupting the end-users data flow. For this reason, SDN-based approaches are common in literature to meet on-demand requirements of smart city services.

*3) Fog Computing:* Fog computing is an architecture that applies cloud features such as compute, communication, and storage at the edge, addressing the vast amount of data generation by IoT devices. Pushing cloud functionality to the network edges enables faster processing time and local decision-making. In addition to these functionalities, data caching will increase the efficiency of data processing at the edge and overcome the response time and reliability constraints [6].

*4) Network Function Virtualization and Placement:* In [7], ETSI defines NFV as decoupling network functions' software from the computation, storage, and networking resources. This gives the network operator the opportunity to customize data flows of smart city services with chained virtual network function (VNF) middleboxes to optimize their QoS with customized data processing, as introduced in [3]. This paper proposes a service-oriented approach for NFV orchestration by using SDN-based traffic management, and is used as an example for ISCO network management architecture. In our use-case, VNF chaining limits the scope of possible flows for a service requesting a path with higher bandwidth, as all the flows need to pass through the VNF node. This is a constraint while computing an optimal flow for a service. Alternatively, network orchestrator can also decide to change the node on which the VNF server is implemented to get a better bandwidth for multiple services using the VNF.

ISBN 978-3-903176-02-7 © 2018 IFIP

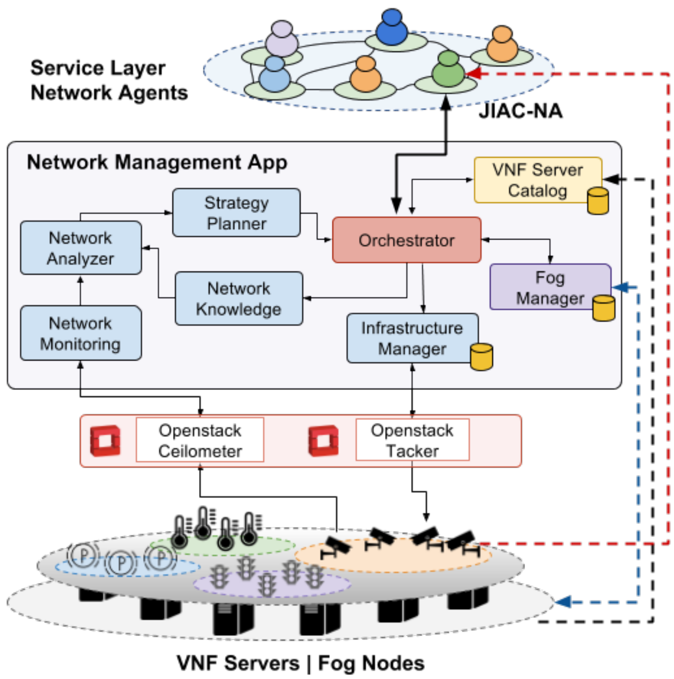


Figure 2: Smart City Autonomous Network Architecture

## IV. ISCO Network Architecture: A Solution for Dynamic Smart City Networks

ISCO platform is a JIAC based service-oriented middleware, on which IoT services operate their tasks. The interaction between IoT services and network devices in ISCO platform are managed through ISCO network layer, which is composed of three essential blocks: Network Management Application (NMA), JIAC Network Agent (JIAC-NA), and OpenStack tools.

### A. Network Management Application (NMA)

NMA provides a generic concept to orchestrate the underlying network resources. It adopts the approach of Monitor-Analyze-Plan-Execute over a shared Knowledge (MAPE-K) and its decision making components are designed accordingly. Orchestrator is responsible for network resource distribution in the smart city network. It also optimizes resource management by using Network Analyzer, Network Monitoring with OpenStack Ceilometer, and Strategy Planner components. Infrastructure Manager (IM) allocates nodes and links in the physical network to network slices, VNF servers and Fog servers. VNFs are deployed to requested nodes by using OpenStack Tacker. When a VNF server is deployed, then this server is added to the VNF Server Catalog so that smart city services can decide whether to add them to their network slices. Fog Manager provides the overview of the deployed fog server resources to the orchestrator.

### B. JIAC Network Agent (JIAC-NA)

JIAC-NA establishes a bi-directional communication between service-layer and network layer. Each smart city service on the ISCO platform can communicate with JIAC-NA to manage their network slices. Forming a new network slice with a specific QoS class, the measurement of users' quality of experience, and the direct interaction with NMA to obtain the available network resources are all carried out through JIAC-NA. Furthermore, service decisions on using Fog and/or VNF servers can be sent to the network management application through JIAC-NA.

### C. Openstack Tools

OpenStack [8] is an open source cloud computing platform that provides various tools and technologies to hide the complexity of the underlying infrastructure by abstracting virtual and physical network resources. OpenStack offers various ready-to-use applications such as Tacker, Ceilometer, etc. to manage those resources. Tacker aims at creating a Generic VNF Manager and NVF Orchestrator in order to install VNFs and network services on an NFV platform and to establish the data flow channels through an Opendaylight SDN Controller. ISCO network manager has the capability of monitoring network slices through OpenStack APIs and its applications, namely by OpenStack Ceilometer.

### D. Interaction between the Components

The aforementioned components in ISCO network layer are displayed in Figure 2. The interaction flow of these components from network slice request to its optimization can be explained as follows: Each smart city service requests a network slice with a QoS class (excellent, good, fair) based on some parameters such as bandwidth, latency, and price. Smart city service also sends its VNF and Fog server requests, if there are any. These information are then transmitted to JIAC-NA with JIAC communication interface. The extracted values from the requested QoS class are then shared with the NMA. NMA first searches for suitable VNF and Fog servers with respect to the given network parameters and send a list of available servers to the smart city service agent. Service agent selects the suitable VNF and Fog servers based on its utility function, and then sends this information back to the NMA. Once the orchestrator inside the NMA gets this message, it initiates the creation of new network slice and then binds the servers to the slice with OpenStack Tacker. Fog server node selection and its connection with the current VNF servers are handled in cooperation with Network Orchestrator and Fog Manager. NMA analyzes the aggregate QoS parameters of all services to either apply a new strategy such as VNF service replacement or keep the current network state. In the mean time, service providers measure their service-related QoS parameters and send new requests to NMA in case they decide that they can optimize their network slice.

In the following section, we clarify how a game-theory approach can realize a distributed resource allocation for smart city services.

## V. Distributed Resource Allocation for Smart City Services

Using mobile intelligent agents in a distributed fashion for resource allocation enables smart city services to implement their own policies based on their own objective functions. The fact that all services have QoS characteristics, priorities and user profiles that they are likely to serve means that they require more personalized utility functions. Besides, in the architecture defined above, NMA can implement its own policies on behalf of smart city network operator by measuring aggregate values such as bandwidth and delay.

Inspired by [9], this multi-agent resource allocation decision problem is defined as an Stackelberg Game among VNF and Fog servers and an Evolutionary Game among users. The players in these games are: ISCO smart city services, smart city network operator, VNF servers and Fog servers. Smart city services can interact with other services in JIAC agent platform, thereby taking part in an evolutionary game and being able to imitate other services' strategies with replicator dynamics to maximize their utilities. OpenStack monitoring provides services the measurement data regarding their data flows. VNF and Fog servers are third party actors who rent their infrastructure for revenue and exponential reinforcing learning algorithm is used to reach the equilibrium of the game among them. Smart city network operator aims to maximize the number of satisfied ISCO services (JIAC agents) and uses aggregate values instead of adjusting each parameter individually.

We extend the referenced work [9] on two main aspects. First, we extend the utility function of smart city services by using the utility function explained in [10] to involve associated independent attributes utility. The utility function of user $i$ in our resource allocation algorithm is denoted with $U_i(b_{k,c}, S_{k,c}, \pi_{k,c})$. $U_i$ is a function of available bandwidth $b_{k,c}$, satisfaction attributes $S_{k,c}$, and service price $\pi_{k,c}$, where $k$ characterizes the user type (excellent, good, fair), and $c$ the service class provided to the user. The details of the utility function can be found in [10]. The main advantages of this utility function is the ability to define different user types, which makes sense when we consider the variety of smart city use cases explained in Section II, and giving smart city services to communicate with other services and gather information about the VNF or Fog server about its reputation, security level, etc. by introducing network independent QoS parameters inside the satisfaction attributes.

Autonomic networking requires monitoring the network status and adapting the network environment by making use of these observations. This is much needed in autonomic network management and policy execution, for which we aim to use the framework explained in [11]. In ISCO, we aim to achieve autonomic behavior in a management system by implementing learning approaches inside the network management application. Self-optimization is based on aggregate values of the network parameters obtained from OpenStack and the QoS related parameter optimization based on network operator's global objective function. Network management application is the intelligent agent that implements the policies for the network operator. Global policy of the network operator can be maximizing the number of satisfied users, achieving fairness by distributing the resources as equal as possible, reserving some physical resources to be resilient against network failures, or a mix of strategies, depending on the use case.

## VI. Conclusions

Integration of flexible management frameworks is essential to make use of the potential of smart city services in IoT era. ISCO is responsible for providing network slices to all smart city services running on its operator network by ensuring QoS class requirements of all services, and enabling technologies should address problems such as scaling the network dynamically based on service demand. ISCO project aims to implement JIAC agents for distributed decision-making by smart city services over resource allocation, SDN and NFV technologies to enable network slices specialized for each service, OpenStack for network monitoring and network function placement, and fog computing to extend the capabilities of these network slices. For network resource allocation in ISCO, we model the resource allocation problem as an evolutionary game where smart city services act independently and request a network slice with a QoS class from the network operator to maximize their utility functions. Service provisioning with short-term contracts is foreseen for smart cities in this architecture. Smart city network operator, on the other hand, can decide on its own strategy for optimizing resource allocation in the entire network topology. The validation use case and the achieved performance with the test results will be presented in our future works.

## Acknowledgment

The work of authors at GT-ARC is supported by the German Federal Ministry of Education and Research within ISCO project with grant number 16KI50580.

## References

[1] H. Zhang, N. Liu, X. Chu, K. Long, A. H. Aghvami, and V. C. M. Leung, "Network slicing based 5g and future mobile networks: Mobility, resource management, and challenges," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138–145, 2017.

[2] S. E. Elayoubi, M. Fallgren, P. Spapis, G. Zimmermann, D. Martin-Sacristan, C. Yang, S. Jeux, P. Agyapong, L. Campoy, Y. Qi, and S. Singh, "5g service requirements and operational use cases: Analysis and metis ii vision," in *2016 European Conference on Networks and Communications (EuCNC)*, pp. 158–162, June 2016.

[3] B. Martini and F. Paganelli, "A service-oriented approach for dynamic chaining of virtual network functions over multi-provider software-defined networks," *Future Internet*, vol. 8, no. 2, 2016.

[4] "Java-based Intelligent Agent Componentware." http://www.jiac.de/agent-frameworks/jiac-v/ (accessed: 2017-11-29).

[5] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 3, pp. 1617–1634, 2014.

[6] H. Kim, J. Park, M. Bennis, S. Kim, and M. Debbah, "Ultra-dense edge caching under spatio-temporal demand and network dynamics," *CoRR*, vol. abs/1703.01038, 2017.

[7] E. T. S. Institute, "Network functions virtualisation (nfv); management and orchestration," 2014.

[8] "OpenStack Telemetry." https://wiki.openstack.org/wiki/Telemetry (accessed: 2017-11-29).

[9] S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, "A game theoretic approach for distributed resource allocation and orchestration of softwarized networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, pp. 721–735, March 2017.

[10] M. A. Khan and U. Toseef, "User utility function as quality of experience(qoe)," *ICN 2011 : The Tenth International Conference on Networks*, 2011.

[11] M. A. Khan and H. Tembine, "Meta-learning for realizing self-x management of future networks," *IEEE Access*, vol. 5, pp. 19072–19083, 2017.