An Adaptive Edge Router Enabling Internet of Things

Mirjami Jutila
VTT Technical Research Centre of Finland, Oulu, Finland

***Abstract*—The vision of future networking is that not only people but also all things, services and media will be connected and integrated, creating an Internet of Everything (IoE). Internet of Things (IoT) systems aim to connect and scale billions of devices in various domains such as transportation, industry, smart home/city, medical services and energy systems. Different wireless and wired technologies link sensors and systems together, through wireless access points, gateways and routers that in turn connect to the web and cloud-based intelligence. IoT architectures make great demands on network control methods for the efficient management of massive amounts of nodes and data. Therefore, some of the cloud's management tasks should be distributed around the edges of networked systems, utilizing fog computing to control and manage e.g. network resources, quality, traffic prioritizations and security. In this work we present adaptive edge computing solutions based on regressive admission control (REAC) and fuzzy weighted queueing (FWQ) that monitor and react to network Quality of Service (QoS) changes within heterogeneous networks, and in a vehicular use case scenario utilizing IEEE 802.11p technology. These adaptive solutions are providing more stable network performance and optimizing the network path and resources.**

***Index Terms*—Internet of Things, fog computing, adaptive queuing, FWQ, fuzzy scheduler, regressive admission control**

I. Introduction

Operators, developers and manufacturers are striving to become part of the Internet of Things (IoT) and Internet of Everything (IoE) revolution, creating new types of products and systems. IoT systems with connected devices and things will cover the whole world and affect all people globally. Hence, networked intelligence will spread to various application domain areas including industry, Intelligent Transportation Systems (ITS), wearables, health, smart homes, offices, buildings, grids and cities. Typical IoT architecture can be categorized broadly into four interconnected systems including things, gateways/routers, networks and clouds. Efficient IoT architecture requires that the things and sensors must be intelligent enough to filter and manage the data that they send to the cloud. However, many of the current sensors were initially not designed to be connected to the Internet and are not capable of processing and sending data to the cloud, although there can be great amounts of data flowing around. For example, a jet engine may produce 10 TeraBytes of data about its performance and conditions in only 30 minutes of flight, according to Cisco [1]. To transfer all the data into a cloud and the response back to the system without any pre-processing would consume not only the scarce bandwidth resources but the time and money of different players in the IoT product chain. In response to this problem, part of the network intelligence and data management should be distributed to gateways and routers in the interconnected systems creating fog [2] and edge computing operations.

A wide variety of IoT scenarios have been addressed within the projects Digile Internet of Things (IoT) [3] and Celtic+ CoMoSeF [4]. In the Digile IoT project, we are developing various sensor networking scenarios and gateway management systems in order to be able to transfer data smoothly from sensors to various applications and services in the cloud. The CoMoSeF project developed and researched various co-operative mobility solutions to support large scale deployment of ITS applications and strategies. In these projects, we have gained valuable insight concerning how the systems can be managed more efficiently and smoothly to increase the potential of various IoT services. Applied networking technologies vary from cellular 3G/4G, ITS-G5 (IEEE 802.11p), Bluetooth Low Energy (BTLE), ZigBee and 6LoWPAN to WiFi. Therefore, routers and gateways must be capable of interfacing and making the systems interoperable in order to apply the intelligent functions in a coherent and efficient way.

This work presents computing intelligence at the network edges for controlling and managing the data and network resources. In our vehicular use case scenario, the vehicles acquire data in real-time via intermediary collectors at Road Side Units (RSUs) which is being disseminated with the best available connectivity. The management capabilities utilizing adaptive edge devices include:

1) real-time communication and high quality message exchange for applications that require low latencies and reliable delivery of information.
2) monitoring solution that collects and measures data periodically from a multitude of data sources to provide flow awareness and Quality of Service (QoS).
3) adaptive edge router solutions for traffic prioritization and bandwidth optimization and management.

Some major challenges related to edge and fog computing that are addressed in our work include QoS, network provisioning and resource management. The management capabilities use fuzzy weighted queueing (FWQ) control mechanism for optimizing traffic path utilizing IEEE 802.11p technology. The FWQ control with a feedback mechanism provides properties related to system stability, short settling times and

M. Jutila, VTT Technical Research Centre of Finland, Oulu, Finland, e-mail: mirjami.jutila@vtt.fi

fast response times. For managing the end-to-end network performance we have introduced a regressive admission control (REAC) method at the network edge, assisted by real-time passive QoS monitoring. Changes in the network QoS determine decision making procedures on the possible flow rejection, marking or support for prioritized users/data, thus bringing cognition to the network path. If the traffic flows attempting to enter the network, e.g. malicious traffic or Denial of Service (DoS) attacks, exceed or are likely to exceed the network capacity, they are treated differently in order to alleviate and minimize the effects of bandwidth attacks.

This paper is organized as follows. Section II presents the challenges and technological enablers for IoT. The computing principles and system specifications at the network edge are described in Section III with a vehicular use case scenario. The adaptive traffic management methods and system description are presented in Section V. Results with discussion are presented in Section V. Conclusions are described in Section VII.

II. Challenges and technology enablers for IoT

The IoT challenges are very versatile concerning various demands including scalability, energy efficiency, intelligence, communication, integration, dependability, semantics, manufacturing and standards. The new techniques and concepts must also be easily integrated to enhance existing technologies. The cognitive approaches exploiting heterogeneous network resources should support seamless connectivity between different access technologies especially in harsh mobile environments such as vehicular communications and other Intelligent Transportation Systems (ITS) and in smart environments such as cities, homes and industrials. The system enablers especially from the networking perspective should provide the following technological characteristics [5]:

1) *Reliability*: Support for management of network mobility and heterogeneity in order to guarantee reliable communications and system operations. Reliable energy-efficient communications must be configured to ensure dependability when billions of heterogeneous devices are connected.

2) *Scalability*: The system must be robust, providing high performance and scalable algorithms and protocols capable of handling varying number of devices, workload levels and heterogeneous networks.

3) *Security*: It is necessary to provide security by embedding and provisioning new secure data keying material during the manufacturing of the device and when in operation, by establishing access control policies to heterogeneous networks and services, developing processes for secure software development and updates, and by efficient cryptographic primitives [6].

4) *Intelligence*: Software and algorithms for distributed problem-solving and decision-making to various management parts of the IoT systems. Efficient proliferation of intelligence is needed to save resources and energy.

5) *Self-management*: To gain simpler and more intelligent systems capable of self-adaptive, self-configuration and self-healing features to save energy required by the complex methods and algorithms.

6) *Virtualization*: Network virtualization techniques are among the important enablers to ensure an evolutionary and modular path for the deployment of IoT applications with assured QoS. The common virtualization techniques include cloud function virtualization, Software Defined Networking (SDN) and Network Function Virtualization (NFV). SDN and NFV techniques are proposed to ease the implementation and management of networks in many aspects of fog and edge computing, thus e.g. reducing costs and easing the resource allocation and traffic monitoring. Currently utilized methods for virtualizing different parts of the network include e.g. OpenFlow [7] for router and switch operations and OpenStack [8] for cloud virtualization functions.

Open standards are key enablers for IoT technologies and for any kind of Machine-to-Machine (M2M) communication. Without globally recognized and interoperable standards the expansion of IoT and IoE systems and services cannot reach a global scale in many industrial sectors. In order to manage the large amount of data coming from different data sources, a separate sensor middleware is often required. Current IoT enabling standards include various middleware solutions for defining the requirements for a unique global identification, namely data fusion, scalability and interoperability to support all-IP based communications. The current oneM2M standard [9] focuses on providing an interoperable platform and technical specifications that can be readily embedded with various hardware and software modules providing a common service and application development framework. 3GPP is standardizing LTE-M for M2M applications. LTE-M offers the benefits of wide spectrum and low cost cellular systems for M2M communications also enabling a long battery life with enhanced coverage for large numbers of devices.

Due to the complex and diverse nature of IoT technologies a single interoperability solution may not be adequate and integration is therefore required. Interoperability of IoT technologies will always be a complex topic requiring research effort to address the new challenges raised. This might be achieved by increased embedded intelligence and virtualization with cognitive capabilities. Not all the technological enablers presented here are vital building blocks as such, but rather affect how much operational fluency, savings, advantage and revenue these solutions will bring, depending on the different players and their motivation in the IoT field. The IoT framework is still an open playground for various actors as long as there are many resolvable issues including pricing, scalability, resource usage, billing options, and how to divide the services and with what granularity.

III. Traffic computing and management at the network edge

The concept of fog computing has been introduced as an intelligent bridge between IoT devices and remote data centers in the cloud. Terms mobile edge computing (MEC) [10] by ETSI and fog computing are often used interchangeably. MEC allows content, services and applications development to be accelerated, while increasing responsiveness from the edge based on insight into the radio and network conditions. The

MEC includes features such as proximity of the information source with low latency operations, location awareness and providing network context information. Cisco considers fog computing as an extension to cloud computing switched from the core of the network to the network edge. With fog computing, some of the load processing and management, data storage and other enabling features presented in the previous section can be handled by transferring part of the computing to edge devices, that are becoming increasingly smart and sophisticated. In a typical fog computing model, the cloud retains its central role of analyzing data and orchestrating the operations and management. When there are no resource constraints and the connectivity among multiple data sources is adequate, pure centralized cloud computing makes more sense. However, the cloud can also delegate some tasks to the smart edge devices in order to localize part of the data analysis and decision making. Typically, the mission of the intelligent edge devices is not to carry out in-depth data analysis, but to actively filter local data and selectively relay data to the cloud. The success of fog computing relies on the ability of these intelligent edge solutions to speed up the deployment, cost-effective scalability and ease of management with limited resources. The fog computing world becomes a critical issue, representing part of the intelligent processing of data at the network edges that enables people to manage their daily lives, from locking their homes to checking their children's location. Therefore, security and privacy issues should be in the forefront when designing new operations and services.

According to a recent survey on fog computing [11], the potential important issues that must be considered include fog networking, QoS, interfacing and programming model, computation offloading and load balancing, accounting, billing and monitoring, provisioning, resource management, and security and privacy. *Fog networking* handles the connectivity and mobility of various heterogeneous networks utilized by the IoT system. Emerging SDN and NFV techniques are proposed to be the key enablers in fog networking. However, SDN and NFV features in fog computing and MEC are currently under standardization. For example when utilizing NFV in fog computing, the performance of virtualized network appliances is still the first concern relating to throughput and delay requirements [12]. Another challenge is how to achieve efficient instantiation, placement and migration of virtual appliances in a dynamic network. *QoS metrics* can be divided into connectivity, reliability, capacity and delay. In our work we have especially concentrated on network capacity issues, and how to achieve the best bandwidth usage with largest number of satisfied customers while preserving fairness. Measuring the delay is an important factor for keeping the delay-sensitive real-time services running smoothly. QoS metrics also affect the network provisioning and resource management very heavily. For example in order to meet the QoS requirements, provisioning is required to prepare resources for service mobility. Common *interfacing and programming models* need to be defined in order to help developers transfer the applications to fog computing platforms. *Computation offloading* can overcome the resource and energy constraints on mobile devices requiring high performance computations to save storage and battery lifetime.

Our adaptive edge solution provides the following requirement specifications presented in Table I. The listed requirements are among those presented above for fog computing that are mapped to the requirement specifications presented in this paper. More detailed specifications are presented with the system description in the next Section.

TABLE I
CONSTRAINTS AND SPECIFICATIONS FOR THE ADAPTIVE EDGE ROUTER.

Requirements	Defined Solution
Offloading	Monitoring the performance of available interfaces and adaptively share resources.
Provisioning	Measuring, monitoring and controlling available networks.
Resource management	Sharing resources fairly (FWQ, REAC) among the entering flows.
Quality of Service	Service classification and support for flow prioritizations (FWQ, REAC).

A. Vehicular use case scenario

Intelligent Transportation Systems and Services (ITS) are an important part of the IoT framework, while increasing numbers of people live in cities, using various means for commuting and utilizing different Internet services. ITS and vehicular networking solutions aim at for traffic safety, fluency and informatics that require high quality connections for vehicular-to-vehicular (V2V) and vehicular-to-infrastructure (V2I) communications. Currently, ITS-G5 (IEEE 802.11p) [13] can be considered as the most mature standard for short-range vehicular communications requiring rapid message exchange. The latest IEEE 802.11p devices have very good performance in terms of delay and line-of-sight range. However, due to the high frequency of 5.9 GHz for ITS-G5, the physical obstacles on the link path can be problematic for the signal propagation [14]. LTE could be an interesting solution for vehicular networks in cases when the frequency is out of the IEEE 802.11p range, and when the application requirements are not very time-sensitive. Another benefit is that a wide deployment of LTE infrastructure is already available in many countries.

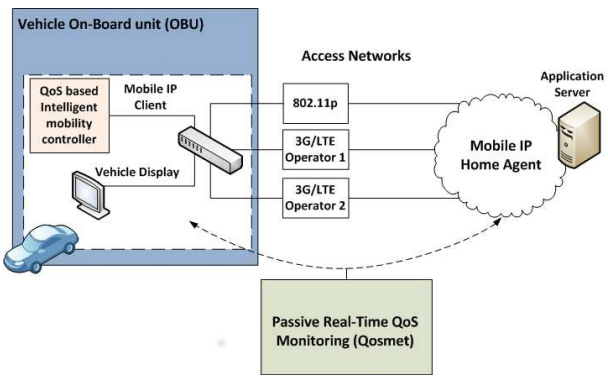

Fig. 1. Vehicular use case with intelligent handover.

In order to utilize available networks in the most efficient way, and to deliver accurate real-time co-operative ITS (C-ITS) messages, it is necessary that the networks are monitored, performance indicators are measured and mobility is controlled. The CoMoSeF project researched C-ITS messaging, solutions, devices and applications feasible for large scale ITS deployment. One of the research test pilot cases is shown in Fig. 1 including solutions from the adaptive traffic management methods presented more detailed in Section V.

In the test scenario, the vehicular was equipped with an on-board unit (OBU) including Mobile IP and Qosmet measurement clients prioritized to utilize the primary local area network with IEEE 802.11p, but whenever the measured signal strength and QoS decreased, it smoothly handed over the connection to commercial cellular 3G/LTE communication offered by two operators. Our system utilized Mobile IP for carrying out the vertical handover between IEEE 802.11p and 3G/LTE based on the passive real-time QoS monitoring with Qosmet [15] and measuring RSSI (Received Signal Strength Indication) values. This test case showed that by providing status information of the networks and managing the traffic in a dynamic way at the edge of the vehicular network, the performance, reliability and capacity of traffic networks and services can be improved.

IV. Adaptive traffic management methods

The architectural framework shown in Fig. 2 positions the communicating adaptive entities with REAC [16] and FWQ [17], [18] that control and schedule information from the application server through the edge router to the RSU(s), and along to mobile vehiculars. Providing the QoS-aware path in the core network has been tested with REAC algorithm in a test-bed setup explained more detailed in Section V-A. For testing the adaptive traffic scheduling with FWQ at the edge router over IEEE 802.11p, we have simulated a similar topology shown in Fig. 2. The simulation model for FWQ is shown more detailed in Section V-B.

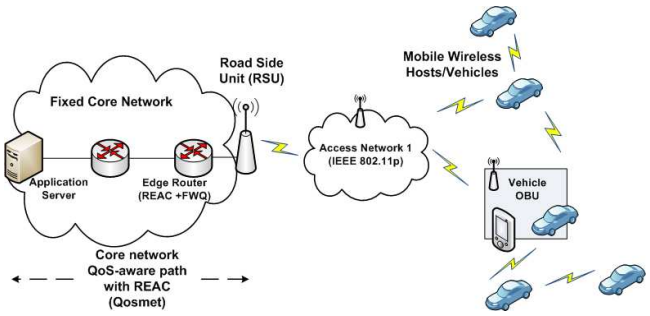

Fig. 2. Architectural framework.

The overall system description and operational flow with REAC and FWQ algorithms are shown in Fig. 3. The functional steps include the following operations:

1) network traffic is monitored. In addition, all flows are registered in the controller component (data base) enabling the specifications of the flows such as QoS requirements, scheduling and packet information. We have tested the traffic monitoring part in various heterogeneous environments. The latest work relates to real-time vehicular scenarios [14], [19] utilizing ITS-G5 (IEEE 802.11p) and cellular 3G/LTE networks to assess the suitability of these technologies for time-sensitive vehicular communications as described in the previous Section III-A.

2) flows are identified and classified into two stages, during the connection establishment and at a later point, in order to improve the classification accuracy. A flow entry contains information such as source/destination IP address/port, IP address of the next hop, and the new destination IP address and status information concerning whether the flow is active or inactive after a certain period of time.

3) the controller exploits a REAC algorithm to make the QoS level estimation and network admission decision for the flows. The REAC controller utilizes the information conveyed by Qosmet and the traffic classifier to implement the admission control (AC) logic. The Quality of Service Measurement Communications Protocol (QMCP) allows full remote control of measurements. The purpose of the logic is to monitor the QoS-level variation by measuring the delay, and to estimate when the QoS-level decrease affects the quality of the high priority applications.

4) the system first traces the suspect flow(s), and then either drops packets of these flows (REACdrop), or decreases the flow priority using e.g. Differentiated Services Code Point (DSCP) marks depending on the network capabilities and support for prioritization (REACmark).

5) flows are scheduled and bandwidth weights are assigned according to fuzzy weighted queueing (FWQ). We have tested the fuzzy scheduling part for LAN and wireless IEEE 802.11b using the Network Simulator NS-2. Here the model is expanded to IEEE 802.11p environment. The developed FWQ model is more stable and reacts faster to different traffic states in order to prioritize e.g. delay-sensitive or critical C-ITS traffic.

A. Regressive admission control

Our regressive admission control (REAC) part of the system accepts flows a priori in the network, without any end-to-end negotiations. In this sense, our work presents a novel approach among traditional measurement-based (MBAC) and parameter-based (PBAC) admission control methods. The conceptual difference between REAC and traditional AC is shown in Figure 4. Traditional AC enforces an admission decision upon each client arrival, as depicted in Figure 4a. This implies that the traditional methods perform an end-to-end negotiation for the AC decision, also adding some extra delay. Figure 4b shows that the REAC decision is not tied to the arrival event. The AC logic monitors the QoS-level variations and estimates when the QoS-level decrease affects the quality of the high priority applications. Whenever the measurement indicates that the quality drops below a given threshold, a decision-making process is initiated to drop or mark the excessive low priority flows.

The REAC method focuses on the temporal variation, e.g. of a delay parameter, averaged over a defined measurement

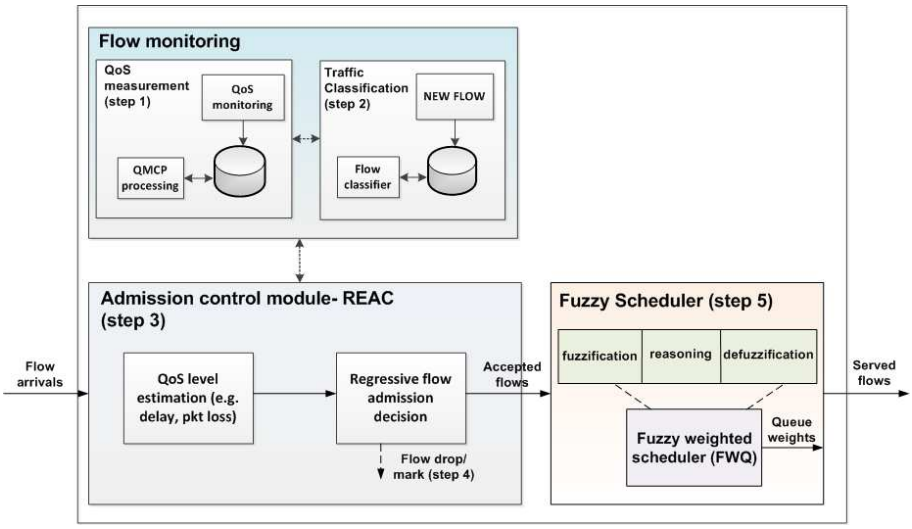

Fig. 3. Overall system description.

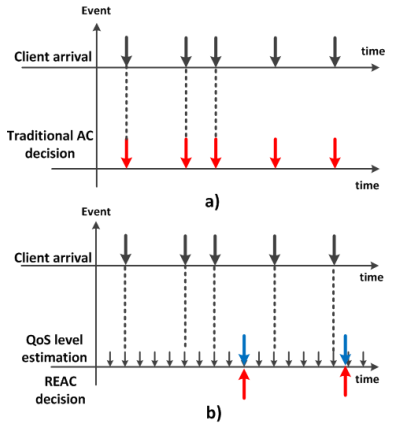

Fig. 4. AC decision triggering a) Traditional MBAC and PBAC b) REAC

window. Other QoS parameters can also be utilized as congestion indicators, such as packet loss rate or queue size. In fact, a combination of metrics could be used, or even metrics that perform pseudo-subjective analysis of the quality. An example of such metrics is PSQA (Pseudo-Subjective Quality Assessment) [20].

The REAC logic is implemented by a Finite State Machine (FSM) having five states, namely: SETUP, NORMAL, ALERT, PREDICT, and ACTION shown in Fig. 5. The FSM operation is synchronized to the arrival and processing of the QoS samples. The system operation starts with a SETUP phase by filling its repositories to collect a sufficient amount of QoS history values. In its NORMAL state the system calculates the mean delay value and compares it to the previous value. We allow for up to exponential growth before entering the ALERT state denoted by the system's increased alerts counter. When a number of maximum alerts *(max_alerts)* is reached, the system resets the counter and enters a new state, namely PREDICT. The system transits to the PREDICT state because the mean delay has kept increasing for at least *max_alerts* interval. Instead of just comparing the mean delay to a system-dependent maximum value for a delay tolerance, REAC makes a projection into the future of its current behavior. The prediction assumes that, since the mean delay has been increasing in an exponential way during the last updating intervals, it will maintain the same tendency. The purpose of such a prediction is to give the system some self-knowledge, and the possibility to diagnose whether or not its current state is progressive. If the mean delay stops increasing exponentially, the system activates a false alarms counter. It allows for the maximum number of predictions to occur before considering that the shock period is finished and reverting back to the NORMAL state. On the other hand, if the prediction shows that in the next interval the system will have surpassed the ceiling, it immediately enters the ACTION state to drop or mark excessive flows.

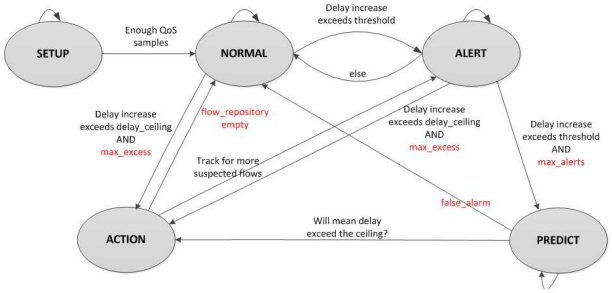

Fig. 5. Operational work flow of REAC algorithm.

B. Adaptive flow scheduling

For complementing the edge network QoS we utilize intelligent scheduling and queueing algorithms. Queueing algorithms participate in congestion control and prevention and in allocating resources. Efficient resource allocation to individual traffic flows requires choosing the right kind of packet scheduler. If there is a situation in which network resources cannot serve all flows, queues will start to build up in the routers. A packet scheduler has an important role in dequeing the packets and keeping track of the network

resources and preserving fairness among the different flows. However, conventional scheduling and queueing methods provide a rather weak form of resource reservation and cannot guarantee QoS, because weights are only indirectly related to the bandwidth which the flow receives. Another problem of these methods and their modifications is that they are rather static in their operations. The latest development of scheduling methods is directed to the dynamic adaptation of scheduling parameters which gives better overall performance [21]–[24]. In our previous publications, we have considered fuzzy expert systems for adaptive weighted fair queueing (AWFQ) with fixed connections [17] and for FWQ with wireless connections [18].

The fuzzy scheduler calculates an adaptive weight coefficient that determines the bandwidth share, e.g. for delay-sensitive real-time and C-ITS applications with UDP traffic and best-effort TCP flows (see Fig. 6). Other flow sharing and prioritization primitives such as gold, silver and bronze user group labeling can also be specified, but the fuzzy rule base has to be tuned accordingly. Another solution to treat certain classes inside a flow classification is *e.g.,* to utilize a cascaded fuzzy system model.

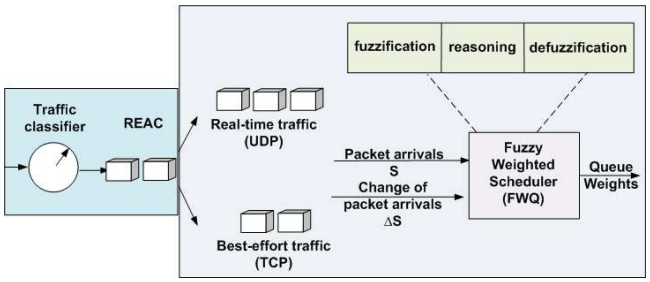

Fig. 6. Traffic management at the adaptive edge router.

The weight update requires two inputs. The first is the share of the UDP and TCP input traffic data rate (*S*), which is calculated in the following way [17]:

$$S = \frac{Q_{UDP}}{Q_{TCP} + Q_{UDP}} \tag{1}$$

where Q_{UDP} is queue length of the UDP traffic queue and Q_{TCP} is queue length of the TCP traffic queue. It can be seen from the Equation 1 that *S* is higher than 0.5 when Q_{UDP} exceeds Q_{TCP}. The other input is the change of the share of received packets (ΔS) calculated as follows [17]:

$$\Delta S = \frac{CQ_{UDP}}{CQ_{TCP} + CQ_{UDP}} \tag{2}$$

where CQ_{UDP} is change of received UDP packets and CQ_{TCP} is change of received TCP packets. In this case also, ΔS is higher than 0.5 when the CQ_{UDP} exceeds CQ_{TCP}. The fuzzy weight update model (located at a router, Fig. 6) has three modules: fuzzification module, reasoning module and defuzzification module. The logic or rule base of the model was composed by analyzing the QoS requirements, the dynamics of input traffic to routers, *i.e.*, traffic density (number of incoming datagrams/time unit), delays and jitter of incoming datagrams as well as transient responses and steady-state properties of the system. The size of the rule base is 25 rules.

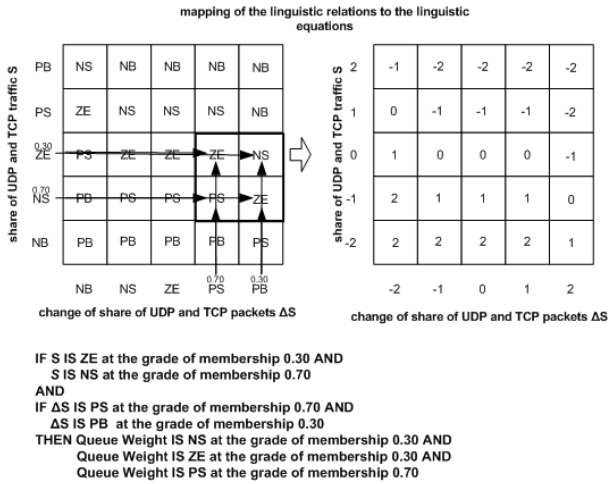

Fig. 7. The mapping of linguistic relations to linguistic equations.

A linguistic model of a system was described by a group of linguistic relations (rules) that can be converted into numerical equations. Suppose, as an example, that X_{ij}, *i=1,2; j = 1, 3,..., m* (*j* is uneven number), is a linguistic level (*e.g.,negative big, negative small, zero, positive small, positive big*) for a variable X_i. The linguistic levels are replaced by integers $\frac{-(j-1)}{2}, ..., -2, -1, 0, 1, 2, ..., \frac{(j-1)}{2}$. The direction of the interaction between fuzzy sets is presented by coefficients A_{ij}=$\{-1, 0, 1\}$, *i=1,2; j = 1,..., m*. This means that the directions of the changes in the output variable decrease or increase depending on the directions of the changes in the input variables [25]. Thus a compact equation for the output $\mathbf{Z_{ij}}$ is:

$$\sum_{j=1}^{m}\sum_{i=1}^{m} A_{ij}X_{ij} = Z_{i,j}. \tag{3}$$

The rule base includes a control policy, which is usually presented with linguistic conditional statements, *i.e.*, if-then rules. In this application, a linguistic model of a system was described by linguistic relations. The linguistic relations form a rule base that was converted into numerical equations to decrease the computation load of the controller. The mapping of linguistic relations to linguistic equations for this application is described in Figure 7. The linguistic weight value is then transformed back into the physical domain to find the crisp output value for the *weight value* using the center of area method (CoA) [17]. Detailed reasoning examples are provided by, for example, in [18] and [26].

V. Results and Discussion

The primary aim of the developed QoS-aware methods was to present an adaptive resource management solution in edge routers also suitable for IoT use cases.

A. Admission Control

The REAC method operates at the network edges, and has been tested e.g. in a scenario, where the network is pushed to its extreme by introducing more high-priority flows than the network can handle. Having the network working at its extreme, the delay starts to fluctuate heavily and all the flows suffer from poor quality. This means that extra flows ruin the performance of all the users. In the REAC case, our AC module ensures that most of the users will maintain good quality, even when the total offered traffic load exceeds the capacity. When the capacity is exceeded, the Qosmet monitoring module quickly notices the QoS degradation, and the AC module controls the flow admittance, resulting in no damage noticeable to the existing users in the network. Upon such an indication, the system first traces the suspect flow(s), and then either drops packets of these flows (REACdrop) or decreases the flow priority using e.g. Differentiated Services Code Point (DSCP) marks (REACmark). Parameters for the test-bed setup with LAN connections illustrated in the overall architecture in Fig. 2 are shown in Table II. Performance results for the scenario are presented in Table III when the input data rate $>>$ output capacity. Many QoE models provide quantitative evaluation of the perceived mapped quality, for example, in the Mean Opinion Scale (MOS) range from 1 to 5 where the numbers present a verbal counterpart of the perceived quality. We have used Absolute Category Rating (ACR) where 5 stands for "excellent", and 1 for "unusable" quality. Typically, value 3 presents a threshold value, where the quality is on the average fair but impairments are already slightly annoying being not suitable for long time use. Utilization, delay and control overhead are also important performance metrics shown in Table III, more details about the performance metrics can be found in [16].

TABLE II
REAC TEST-BED PARAMETERS.

Parameter	Value
Protocols	LAN with IP/UDP and TCP
Flow arrivals	8 with random interval [10,100]s
Data rates	500 kbps-4 Mbps
Bottleneck bandwidth	10 Mbps

TABLE III
REAC PERFORMANCE WHEN INPUT DATA RATE $>>$ OUTPUT.

	pureBE	**REAC-drop**	**REAC-mark**
Subjective score (MOS)	3.6	4.8	4.8
% of time with good quality	52.6	92.0	92.1
% of time with poor quality	15.9	1.8	1.6
Delay[ms]	49.9	16.5	21.3
Max. utilization	95.0%	67.1%	96.7%
Congestion delay [ms]	75.9	19.6	27.2
Control overhead	-	1.4%	1.1%

The subjective quality is poor only less than 2% of the time in both REAC packet dropping and marking methods, whereas the normal best-effort (BE) method without REAC suffers from bad quality approximately 16% of the time. Furthermore, REAC cases allow the quality to be at a good level more than 92% of time, whereas in the BE case the corresponding figure is only about 50%. The results show that REAC can be used to give quality guarantees. REAC marking shows a very good performance, with a maximum utilization of approximately 97%. That is a high value when taking into account the fact that many AC schemes suffer from low utilization. The biggest differences appear in the congestion delay, where REAC paths achieve clearly the lowest values, meaning that REAC is able to resist congestions. AC mechanisms always bring some overhead with the methods they are using, e.g. for measurements and control policies, and this is one of their drawbacks. In the REAC scheme, however, the information about network conditions is rendered by an external entity, i.e. the measurement tool, and no other control overhead is needed. Another drawback of some AC schemes is the decision delay, which in the REAC case is zero because of the regressive operation for the flows, which will be allowed to continue as high-priority flows. The control overhead is between 1.0% and 1.4% over all the REAC cases, which can be considered as quite reasonable.

B. Fuzzy service classifier

In this work the FWQ algorithm was applied to topology shown in Fig. 2 over IEEE 802.11p technology. It is considered that the six best-effort TCP (packet size of 256 bytes) traffic sources with link bandwidth 200 kbps (1200 kbps altogether) and six delay-sensitive UDP (packet size of 512 bytes) traffic sources with link bandwidth 350 kbps (2100 kbps altogether) are connected to the RSU through LAN in the core network. The RSU including the FWQ logic is connected to the mobile vehicular. In the fuzzy scheduler, the flows are treated as aggregates respectively for UDP and TCP. The number of datagrams of the incoming traffic was assumed to be Pareto[1] distributed. The transmission delay of each packet was assumed to be normally distributed. The burst nature of the traffic was enhanced by increasing Pareto distributed datagram bursts randomly to the simulated link.

TABLE IV
PARAMETERS FOR THE IEEE 802.11P SIMULATIONS.

Parameter	Value
Simulation time	500 s
Protocols	IP/UDP and TCP
MAC	CSMA/CA
Carrier frequency	5.9 GHz
RSU transmitter power	200 mW
Channel bandwidth	10 MHz
Vehicle speed	0-60 km/h
Simulation area	600 m x 600 m

The adaptive FWQ scheduler is operating at the network edge as shown in Fig. 2 between fixed infrastructure and wireless IEEE 802.11p network. The wireless receiving vehicle for

[1]Data network traffic has self-similar and long-range dependent nature, which is known to obey Pareto distribution with Pareto distributed interval times.

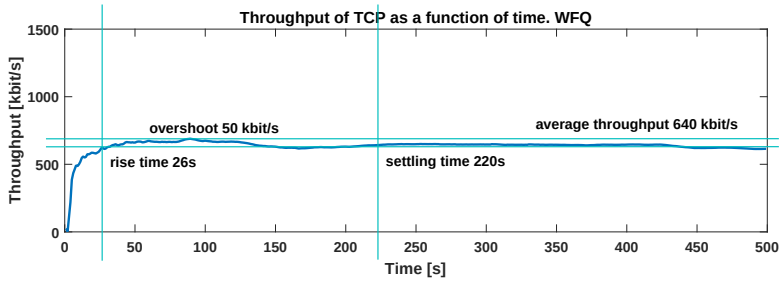

Fig. 8. Throughput of TCP traffic as a function of time when WFQ was used and input data rate >> output.

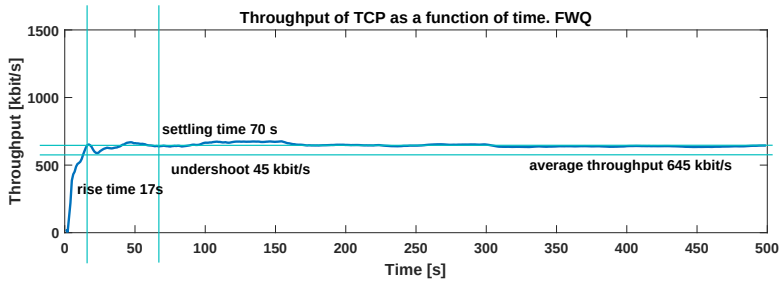

Fig. 9. Throughput of TCP traffic as a function of time when FWQ was used and input data rate >> output.

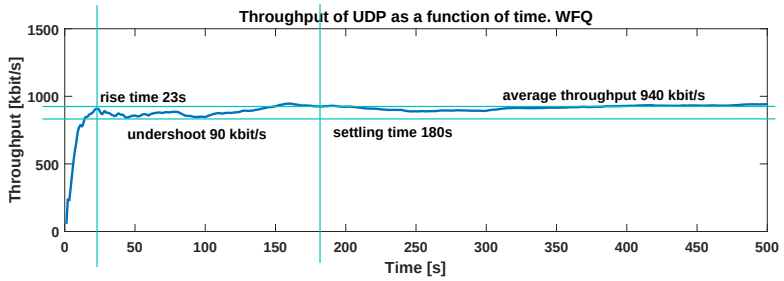

Fig. 10. Throughput of UDP traffic as a function of time when WFQ was used and input data rate >> output.

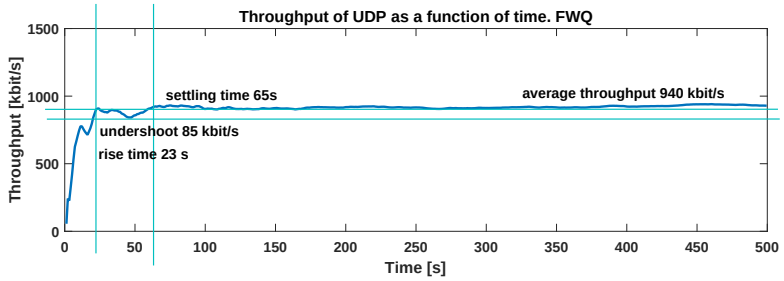

Fig. 11. Throughput of UDP traffic as a function of time when FWQ was used and input data rate >> output.

TABLE V
RISE AND SETTLING TIMES, AVERAGE THROUGHPUTS AND UNDER/OVERSHOOTS WHEN INPUT >> OUTPUT.

Traffic WFQ	Rise time	Settling time	Over-/ undershoot	Throughput	Traffic FWQ	Rise time	Settling time	Over-/ undershoot	Throughput
TCP	26 s	220 s	OS 50 kbps	640 kbps	TCP	17 s	70 s	US 45 kbps	645 kbps
UDP	23 s	180 s	US 90 kbps	940 kbps	UDP	23 s	65 s	US 85 kbps	940 kbps

the TCP and UDP traffic is moving around the 600 meters x 600 meters area with a speed of 0-60 km/h (urban area). There are also four other moving vehicles sending background Constant Bit Rate (CBR) traffic in random intervals [0.01 0.05]. The simulation parameters are shown in Table IV and done with Network Simulator 2 (NS-2). The effect of

increasing the number of vehicles is not considered in this paper. In fact, even with only one vehicle, by increasing the source data rate, we can analyze the FWQ control algorithm functionality that can be reached in IEEE 802.11 networks in similar conditions. Here we are operating at the capacity limits for IEEE 802.11p with one RSU deployment [27] testing a congested situation when input data rate is bigger than the output capacity.

Utilizing FWQ mechanism shorter settling, rise and fall times as well as lower overshoots and undershoots were attained compared to the traditional WFQ algorithm as shown in Figs. 8-11. The developed FWQ scheduler was designed to prioritize delay-sensitive UDP traffic but it can be applied to prioritize e.g. co-operative vehicle applications that are also sensitive to end-to-end delay. The tuned rule base for the fuzzy system anticipates the upcoming traffic and makes it possible to react smoother and faster to prevailing traffic conditions increasing QoS as shown in Table V. For TCP traffic, rise times were 26 s for the WFQ model and 17 s for the FWQ model, whereas for the UDP traffic rise times were 23 s and 23s, respectively. Settling times for the WFQ model were 220 s and 180 s for TCP and UDP traffic. For the FWQ model they were 70 s and 65 s for TCP and UDP traffic. For the WFQ model with TCP and UDP traffic, there was 50 kbits/s overshoot and 90 kbits/s undershoot, respectively. For the FWQ model, there was 45 kbits/s undershoot for TCP and 85 kbits/s undershoot for UDP.

The reason for shorter settling and rise times of the FWQ model may also be that the rule base anticipates the upcoming traffic and makes it possible to react smoother and faster to prevailing traffic conditions. The rule base lets the UDP burst to utilize breaks in TCP flows and vice versa. The rule base has a significant role for the rise and settling times. Hence, the rule base has to be tuned for the overall aim in order to take care of tradeoffs between contradictory subtargets.

VI. Conclusions

This paper proposed adaptive computing methods for IoT networking at the network edges to optimize and control traffic flows and network resources. The fog computing challenges at the edge routers includes e.g. QoS issues, network provisioning and resource management. With the REAC method, the adaptive edge router monitors the link performance to admit the flows to the network in a way which handles congestion and preserves good quality for prioritized users. The QoS scheduling capabilities utilize FWQ to control traffic flows according to the prevailing traffic level in a smooth and fast way in heterogeneous networks.

The developed mechanisms are able to react faster to traffic changes and guarantee better quality for prioritized traffic and at the same time preserving fairness to other flows than the traditional control and scheduling methods without adaptive characteristics. The developed overall system reacts to changes in the network QoS by determining decision making procedures on the possible flow rejection, marking, or allowed bandwidth weight assignment, thus bringing cognition to the network path. In future work, the adaptive traffic management methods need to be evaluated and the scalability tested in a large-scale environment for combining the different algorithms optimizing the performance of the IoT applications. Testing these features as SDN and NFV components would also be beneficial for the resource usage optimization.

Acknowledgment

This work was supported by TEKES as part of the Internet of Things program of DIGILE (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT and digital business). The author would like to thank VTT Technical Research Centre of Finland and Tekes for their financial support.

References

[1] M. Enescu, "The three mega trends in cloud and IoT," Cisco blog at http://blogs.cisco.com/cloud, Accessed: 2016-04-01.
[2] Cisco, "Fog Computing, Ecosystem, Architecture and Applications," Category: Experimental, 2013.
[3] "Digile Internet of Things homepage," http://www.internetofthings.fi/proj, Accessed: 2016-04-01.
[4] "Celtic+ Comosef project homepage," http://www.comosef.eu, Accessed: 2016-04-01.
[5] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer *et al.*, "Internet of things strategic research roadmap," *Internet of Things: Global Technological and Societal Trends*, vol. 1, pp. 9–52, 2011.
[6] S. L. Keoh, S. Kumar, and H. Tschofenig, "Securing the internet of things: A standardization perspective," *Internet of Things Journal, IEEE*, vol. 1, no. 3, pp. 265–275, June 2014.
[7] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks." *ACM SIGCOMM Computer 12 Communication Review*, vol. 38, no. 2, pp. 69–74, March 2008.
[8] "OpenStack homepage," http://www.openstack.org, Accessed: 2016-04-01.
[9] "OneM2M homepage," http://www.onem2m.org, Accessed: 2016-04-01.
[10] ETSI, "Mobile-Edge Computing," *White Paper*, 2014.
[11] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, ser. Mobidata'15. New York, NY, USA: ACM, 2015, pp. 37–42.
[12] H. Bo, V. Gopalakrishnan, J. Lusheng, and L. Seungjoon, "Network function virtualization: Challenges and opportunities for innovations," *Communications Magazine, IEEE*, vol. 53, no. 2, pp. 90–97, Feb 2015.
[13] "ETSI EN 302 663. Intelligent Transport Systems (ITS); access layer specification for intelligent transport systems operating in the 5 GHz frequency band." Nov 2012.
[14] M. Jutila, J. Scholliers, M. Valta, and K. Kujanpää, "Assessment of the performance of ITS-G5 for vulnerable road user safety applications," Oct 2015.
[15] J. Prokkola, M. Hanski, M. Jurvansuu, and M. Immonen, "Measuring WCDMA and HSDPA delay characteristics with qosmet," in *Communications, 2007. ICC '07. IEEE International Conference on Communications*, June 2007, pp. 492–498.
[16] M. Jutila, J. Prokkola, and D. Triantafyllidou, "Regressive admission control enabled by realtime qos measurements," *In International Journal of Computer Networks and Communications (IJCNC 2013)*, vol. 5, no. 6, 2013.
[17] T. Frantti and M. Jutila, "Embedded Fuzzy Expert System for Adaptive Weighted Fair Queueing," *Expert Systems with Applications, Elsevier Science*, vol. Vol. 36, No. 8, pp. 11 390–11 397, 2009.
[18] M. Jutila and T.Frantti, "Cognitive Fuzzy Flow Control for Wireless Routers," *Submitted for review in International Journal of Autonomous and Adaptive Communication Systems (IJAACS)*, 2016.
[19] M. Valta, M. Jutila, and J. Jämsä, "IEEE 802.11p and LTE as enablers of cognitive vehicle-to-roadside communication," in *6th IEEE Conference on Cognitive Infocommunications*, 2015.
[20] M. Varela, "Pseudo-subjective quality assessment of multimedia streams and its applications in control," Ph.D. thesis, University, Rennes, France, 2005.

[21] M. F. Horng, W. T. Lee, K. R. Lee, and Y. H. Kuo, "An adaptive approach to weighted fair queue with QoS enhanced on IP," in *IEEE Region 10 International Conference on Electrical and Electronic Technology*, vol. 1, 2001, pp. 181–186.

[22] A. Sayenko, T. Hamalainen, J. Joutsensalo, and L. Kannisto, "Comparison and analysis of the revenue-based adaptive queuing models," *Comput. Netw.*, vol. 50, no. 8, pp. 1040–1058, 2006.

[23] Akashdeep and K. Kahlon, "An embedded fuzzy expert system for adaptive WFQ scheduling of IEEE 802.16 networks," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7621–7629, November 2014.

[24] M. Andrews and L. Zhang, "Rate-adaptive weighted fair queueing for energy-aware scheduling," *Information Processing Letters*, vol. 114, no. 5, pp. 247–251, May 2014.

[25] E. K. Juuso, "Linguistic Simulation in Production Control," in *proceedings of the UKSS'93 Conference of the United Kingdom Simulation Society*, R. Pooley and R. Zobel, Eds., Keswick, UK, 1993, pp. 34–38.

[26] T. Frantti, M. Majanen, and T. Sukuvaara, "Delay Based Packet Size Control in Wireless Local Area Networks," in *proceedings of the Second International Conference on Ubiquitous and Future Networks (ICUFN 2010)*. Jeju Island, Korea: IEEE Communications Society, 2010.

[27] I. Msadaa, P. Cataldi, and F. Filali, "A Comparative Study between 802.11p and Mobile WiMAX-based V2I Communication Networks," *In 4th International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST)*, pp. 186–191, 2010.