Intelligent Management for Next Generation Communication Systems

Communication Systems Policies for Delivering Fresh Information

Basel Barakat

School of Engineering
University of Greenwich

This dissertation is submitted for the degree of
Doctor of Philosophy

October 2019

I would like to dedicate this thesis to my loving family ...

Declaration

"I certify that the work contained in this thesis, or any part of it, has not been accepted in substance for any previous degree awarded to me, and is not concurrently being submitted for any degree other than that of Doctor of Philosophy being studied at the University of Greenwich. I also declare that this work is the result of my own investigations, except where otherwise identified by references and that the contents are not the outcome of any form of research misconduct."

Basel Barakat
October 2019

Acknowledgements

"The best investment you can ever do, is the investment in people";
Professor Nizam Barakat, the author parent.

I would like to acknowledge all the efforts by my parents. They had equipped me with all the tools I need to stand where I am at now. With their infinite patience, understanding and trust will always motivate me to contribute to our world.

I want to acknowledge the extraordinary help I received from all my supervisors: Professor Simeon Keates, Professor Peter Kyberd, Professor Colin Hills, Dr Kraim Nasr, Dr Mehdi Baghdadi, Professor Predrag Rapajic and Professor Kamran Arshad.

I greatly appreciate Professor Simeon Keates for his insightful comments that are helping me to see a bigger picture of engineering. His remarks and intelligence helped me to untangle complex problems, simplify them and solve them. I always feel that he is there for me. His consistent bright advise had significantly helped me. His mentality and persistence to build an inclusive technology for the future will always encourage me to contribute in our future. I can easily say he is one of the most inspiring people I know.

I also would like to thank Professor Peter Kyberd for stepping up to help me. Also he had introduced me to a very exciting and inspiring research area, i.e., prosthetic limbs. I am very grateful for Dr Mehdi Bagdadi, for having a significant impact on me and my development. As he gave me one of the most helpful insights about research, i.e., in a PhD, publications are not linear with time. I also would like to thank Dr Karim Nasr, for stepping up to help in my PhD and for giving me the chance to teach.

Many thanks to Professor Kamran Arshad, has always been there for me. His remarks had always been constructive to have a good relation with my colleagues. He has been my backup when things get rough. His inspiring dedication is so inspiring. He would always help even if it means to have a call very late at night.

I greatly appreciate Dr Ian Wassell from the University of Cambridge, for giving me an extraordinary chance to be part of the Digital Technology Research Group. His comments and suggestion had immensely helped me in the research. Also, for his time to read through my work, even when he is extremely busy. His hospitality and understanding is very heartening.

I cannot thank Predrag Rapajic enough for his time and comments; he is one of the people that had an extraordinary impact on me. He had helped me to shift my paradigm and to develop my critical thinking. He had introduced me to countless fascinating concepts. It has always been my pleasure to discuss interesting topics with him. His integrity is so inspiring.

I also would like to acknowledge the academic staff as they had given me a great chance to participate in the teach and involve in the education process. Have the opportunity to help students to understand the several engineering topics had been my absolute pleasure.

I am thankful for Dr Yi Wang and Dr Peter Callaghan for examining me in my MPhil/PhD transfer. There comments had helped me to plan my PhD and reevaluate the direction of my research.

Aistė Steponėnaitė, her attitudes come from the most wonderful heart in the world. Her patience and positive mindset had always been very inspiring. Even in the darkest times, she had been the light.

Lynne Martin, one of the most helpful and considerate individual. She is one of the most emotionally intelligent people I know. The activities she started and run had been helping several other students and me to overcome our stressful life problems. Her exceptional talent and skill in building a safe place to communicate and understand other cutlers are genuinely marvellous.

Nicola Smith and Sharon Wood had always been very helpful in getting things done. They had been there to get advise on almost all the matters in the university.

Abstract

Many Internet of Things (IoT) applications require information to be received and decision-making made in a timely manner. The Age-of-Information (AoI) metric has been proposed to measure and evaluate the freshness of information. Afterwards, a new metric was proposed, named Peak AoI (PA), which represents the worst-case AoI. PA is defined as the maximum time elapsed since the preceding piece of information was generated. The PA metric has a simpler formation and is a more utilisable metric. Consequently, it has gained attention in the literature with various approaches to model it and also to optimise network functions to minimise it.

This thesis answers the following question *'How can we deliver fresh information?'*. To answer this question, several gaps in the current body of knowledge had to be filled. The first gap was *'A method for evaluating the information freshness empirically using experiments'*. Hence, an experimental model is proposed, and validated to evaluate the freshness of information. Using the empirical method, the second research gap, i.e., *the limitation of the policies proposed in the literature*, was examined and proved to be inefficient in some of the real-world scenarios. Thus, the next research gap had been formed, that is *a policy to deliver fresh information in real-world applications*, hence, the author proposed a policy to deliver fresh information and tested it in several real-world scenarios. Afterwards, it was noticed that *to deliver fresh information it is necessary to decrease the throughput*, hence, it a naval policy to deliver freshness information without compromising the throughput was proposed. The *accuracy of queuing models used in the literature* was also investigated.

Table of Contents

List of Figures

List of Tables

List of Acronyms

5G	Fifth Generation of Wireless Communication Systems
ACK	Acknowledgement
AMC	Adaptive Modulation and Coding
AoI	Age-of-Information
ASAP	Adaptive Status Arrivals Policy
ATM	Automated Teller Machine
AWS	Amazon Web Services
BS	Base Station
CAP	Clustered Acknowledgement Policy
CDF	Cumulative Distribution Function
CMMPP	Coupled Markov Modulated Poisson Process
CQI	Channel Quality Indicator
CSI	Channel State Information
CU	Continuous Updating
ECG	Electrocardiography
ED	Events-Driven
FS	Fixed Scheduling
FCFS	First Come First Served
FIFO	First-In-First-Out
H2H	Human-to-Human
HARQ	Hybrid Automatic Repeat Request
IaaS	Infrastructure as a Service
INT	Inter-Arrival Time
IoT	Internet of Things
ITS	Intelligent Transportation System
LAN	Local Area Network
LCFS	Last-Come-First-Served
LTE	Long Term Evolution
M2M	Machine-to-Machine communication
M2MD	Machine-to-Machine Communication Device
MA	Maximum Age
MCM	Machine Communication Model
MMPP	Markov Modulated Poisson Process
MTC	Machine-Type-Communications
NP	Number of transmitted Packets
OP	Optimal Peak Age
PA	Peak AoI
PDF	Probability Density Function
RB	Resources Blocks
SNR	Signal to Noise Ratio
SR	Scheduling Request
ST	Stabilising mode
TCP/IP	Transmission Control Protocol/Internet Protocol
TP	Transition Probability
ZW	Zero-Wait Policy

Chapter 1

Introduction

Research Question:

How can we deliver fresh information?

Research Gaps:

- ***The Lack of empirical work in the information freshness research scope.***
- ***Examine the optimally of Zero-Wait policy.***
- ***A policy to deliver fresh information in real-world scenarios.***
- ***A policy to deliver fresh information with a high throughput.***
- ***Investigate the accuracy of the queuing models.***

"If the only tool you have is a hammer, you tend to see every problem as a nail."; Professor Abraham Maslow, the creator of the Needs Hierarchy Theory.

The motivation for contributing to this thesis research scope is presented in the first section of this chapter. The second section introduces the research question this thesis is answering and the main knowledge gaps that exist in the literature.

1.1 Motivation

Acquiring accurate information in a timely manner has helped in shaping the world. For instance, one of the turning points in world history was the Greco-Persian Wars [1], which lasted for more than 40 years from 492 BC until 449 BC. In this war, communicating information in a timely manner changed the course of the war, and hence, the world. The decisive battle, that determined the outcome of the war, was the Battle of Marathon, September 490 BC. In this battle, according to the legend, an Athenian messenger was sent from Marathon to Athens, a distance of about 40 kms, and there he announced the Persian defeat before dying of exhaustion. At those days communicating the announcement took two days, which was sufficiently quick to change the outcome of the war.

In 16th century England, an innovative system to communicate information in a timely manner had played a role in improving the defences. The system was the beacons, which were used to warn of the approaching Spanish Armada [2] [1]. Several hills in England were named Beacon Hill after such beacons. Thus, the English were able to prepare for any invasion, using the information at the right time obtained from the beacons.

The last century had an extraordinary revolution in information transmission through telecommunications systems. Several significant contributions had made this revolution a reality. Each of the contributions opened the path to a new way of thinking and pushed the limits of what is possible. The revolution led us to invent new means of communication to move from handwritten letters to text, audio and video. As we are currently on the fringe of a new revolution in the Internet of Things (IoT), robotics and autonomous systems, it is critical to revisit some of the assumptions and the questions that we as researchers had been asking ourselves.

[1] Spanish Armada was a fleet of 130 ships that sailed from Corunna in late May 1588, with the purpose of escorting an army from Flanders to invade England.

One of the most prominent questions was asked over a century ago by A. K. Erlang[2], i.e., *'How long do we have to wait to be assigned a phone line?'*. To answer this question Erlang in his seminal paper [3] put in place the cornerstones of what is now called Queuing Theory. Queuing theory is defined as the primary methodological framework for analysing network delay [4]. It usually utilises Poisson and Markov Theories [5] to calculate the probability of waiting to be assigned a phone line for a given number of users and call duration (service time).

Although Erlang's work is now over a century old, it is still relevant to us nowadays, and the question he answered is critical for the new means of communication. In particular, to understand the limitations of the existing networks to serve the IoT [6]. The IoT is the network that carries the Machine-to-Machine communication (M2M) [7]. The M2M communication occurs due to functions of the machines rather than the people. Typically a device transmits a reading of a sensor to a database or a decision maker.

M2M communication takes place in several applications that range from healthcare monitoring [8], smart grids [9] and asset monitoring [10]. Excessive errors and delays in these applications can be life-threatening. For instance, consider the case of Intelligent Transportation System (ITS) which aims to enhance the safety of passengers. For ITS [11], excessive delays might cause severe accidents. Robotic surgeries [12] equipment are another example in which the requirements of these systems must be carefully determined.

For such applications, new fundamental questions have been raised, such as *'Is the waiting time the most efficient way to evaluate the steadiness of the information?'* and *'To what extent do we care about the communication throughput and networks delay?'*. These questions led researchers in the 90s to rethink the way we look at what we should measure [13]. The question was raised in the context of real-time database systems by scholars at Stanford University. They were proposing policies for a trading application that communicates with an external database such as the New York Stock Exchange.

One of the insightful questions raised was *'For Real-Time systems, do we care about the network performance (delay) or do we care more about delivering the information as fresh as possible to its destination?'*. To answer this question, it is necessary to understand the difference between high data throughput (or short delay time) and delivering fresh information in the context of data networks.

[2]A. K. Erlang (1878-1929), was a mathematician, statistician and engineer, who worked as the Chief Engineer of the Copenhagen Telephone Company.

Let us consider a time-varying process that we are interested in monitoring, say the location of an ambulance vehicle. The instantaneous location (status) is sent from the vehicle to an external database. If it is intended to propose a policy for communicating the status updates to the database, several approaches might be taken.

One approach to design the policy can be to focus our concern on having the shortest delay as possible. This can be achieved by regulating the time between the transmission of the updates to be sent with sufficient time between them. This would reduce the number of updates queuing to be delivered to the database. However, this approach would reduce the accuracy of our monitoring. The second approach can be to transmit as much updates as possible, which would give the monitor a continuous log of the process of interest. However, using this approach, would increase the number of updates in the system and hence increase the delay time. On the other hand, if the concern of the policy is to acquire the most accurate information regarding the monitored process, i.e., ambulance location. We will deliver fresh information, that would lead us to have a higher accuracy monitoring. The information freshness is affected by both the time of generating the information and the time to communicate it.

1.2 Research Question and Gaps

The novel concept, i.e., information freshness, led to this thesis research question, i.e., ***'How can we deliver fresh information?'***. The literature (reviewed chapter 2) proposed several policies to answer this question; nevertheless, there is still a long way to find a comprehensive answer. In the literature review, the author identified several gaps in current knowledge. For instance, it can be observed that most of the work done in the literature had been on abstract theoretical models, and very little work had been done empirically. Hence, in chapter 3, a method to evaluate the information freshness using experiments is proposed. That work aims to making it straightforward to estimate the freshness using experiments, thus motivate more experimental work on this research scope.

Several scholars identified a policy, as optimal for information freshness, i.e., Zero-Wait policy. Moreover, it can be observed that it became common knowledge that the Zero-Wait policy is optimal. Thus, that encouraged the author to investigate the preciseness of this claim. It was found that the Zero-Wait is only optimal in specific queuing models, which were identified in chapter 4. To validate this conclusion, the

author conducted experiments on several scenarios. The results supported the claims and showed the limitation of the Zero-Wait policy, as shown in chapter 4.

Although the limitations of the Zero-Wait findings were notable, they did not answer our research question. Consequently, the Adaptive Status Arrivals Policy (ASAP) was proposed, which can deliver the updates as fresh as possible in several scenarios. The ASAP was able to overcome the limitations of the Zero-Wait policy; moreover, it can deliver fresh information through a very challenging environment such as the internet. More detail regarding ASAP is presented in chapter 5.

In chapter 5, it was shown that to deliver fresh information, the number of updates we have about the process of interest have to be tamed. On the other hand, for several applications delivering the freshest information might not be the most critical Key Performance Indicator. Accruing a significant amount of relevant information might be more critical. Consequently, in chapter 6, the Clustered Acknowledgement Policy (CAP) is proposed, in which it (CAP) can provide much more information and maintain the freshness of the information. In particular, our experiments showed that it delivers three orders of magnitude increase in the number of updates compared with the theoretical optimal policies.

In the investigation of the policies that can deliver fresh information, it was noticed that the scholars are using the models that were defined by Erlang over a century ago [3]. Accordingly, the author was interested in knowing if these models are still the best representation of our queues. Hence, chapter 7 started with an evaluation of the assumptions made, and it was found that they might not be the most suitable for all the data networks. Consequently, the Machine Communication Model (MCM) was proposed, which takes into consideration much more parameters affecting the network's performance. The model was compared with literature and showed that it could evaluate the traffic with more accuracy.

This thesis is written in a way that each chapter builds up to the next one; nevertheless, each chapter can be independent of other chapters. Furthermore, after each chapter, the list of the related references are presented, and the complete list of references is provided at the end of this thesis.

Chapter References

[1] Green, P., 1996. The Greco-Persian Wars. Univ of California Press.

[2] Hill, D. and Sharp, S., 1997. An Anglo-Saxon Beacon System. Names, Places and People: An Onomastic Miscellany for John McNeal Dodgson, pp.157-165.

[3] Erlang, A.K., 1909. The theory of probabilities and telephone conversations. Nyt. Tidsskr. Mat. Ser. B, 20, pp.33-39.

[4] Bertsekas, D.P., Gallager, R.G. and Humblet, P., 1992. Data networks (Vol. 2). New Jersey: Prentice-Hall International.

[5] Gallager, R.G., 2012. Discrete stochastic processes (Vol. 321). Springer Science & Business Media.

[6] Atzori, L., Iera, A. and Morabito, G., 2010. The internet of things: A survey. Computer networks, 54(15), pp.2787-2805.

[7] Anton-Haro, C. and Dohler, M. eds., 2014. Machine-to-machine (M2M) communications: architecture, performance and applications. Elsevier.

[8] Islam, S.R., Kwak, D., Kabir, M.H., Hossain, M. and Kwak, K.S., 2015. The internet of things for health care: a comprehensive survey. IEEE Access, 3, pp.678-708.

[9] Gungor, V.C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C. and Hancke, G.P., 2011. Smart grid technologies: Communication technologies and standards. IEEE transactions on Industrial informatics, 7(4), pp.529-539.

[10] Garcia, C.B., 2001. Method for monitoring and trading stocks via the internet displaying bid/ask trade bars. U.S. Patent 6,272,474.

[11] Zhang, J., Wang, F.Y., Wang, K., Lin, W.H., Xu, X. and Chen, C., 2011. Data-driven intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems, 12(4), pp.1624-1639.

[12] Lanfranco, A.R., Castellanos, A.E., Desai, J.P. and Meyers, W.C., 2004. Robotic surgery: a current perspective. Annals of surgery, 239(1), p.14.

[13] Adelberg, B., Garcia-Molina, H. and Kao, B., 1995, June. Applying update streams in a soft real-time database system. In ACM SIGMOD Record (Vol. 24, No. 2, pp. 245-256). ACM.

Chapter 2

Literature Review

- **Research Gap:**
  To understand the background and the state of the art policies to answer the research question
- **Most relevant papers:**
  - D. Bertsekas and R. Gallager, Data networks. 1992, vol. 2. Prentice Hall, 1992.
  - Adelberg, B., Garcia-Molina, H. and Kao, B., 1995, June. Applying update streams in a soft real-time database system. In ACM SIGMOD Record (Vol. 24, No. 2, pp. 245-256). ACM.
  - S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?," in Proceedings - IEEE INFOCOM, 2012, pp. 2731–2735.
  - Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or Wait: How to Keep Your Data Fresh," IEEE Transactions on Information Theory, vol. 63, no. 11, pp. 7492–7508, Nov. 2017.

"Stand on the shoulders of giants"; Sir Issac Newton.

Throughout the last century, since Erlang's seminal work in the early 1900s [1], several major contributions were made. Most of the contributions were aiming to analyse the queuing system's behaviour and minimising the queue delays. Recently, the freshness of the updates had started to be considered in the data networks. In this literature review, the author categorised the work done in the literature as foundation work and state of the art. The foundation work presents a brief introduction of Queuing Theory, data networks delays and the early work on information freshness. The state of the art section presents the most recent work on the information freshness. The first part of the state of the art section presents the recently proposed metrics for quantifying the information freshness; the second part presents some of the policies proposed in the literature to optimise the information freshness. An overview of the literature review chapter is presented in Fig. 2.1.

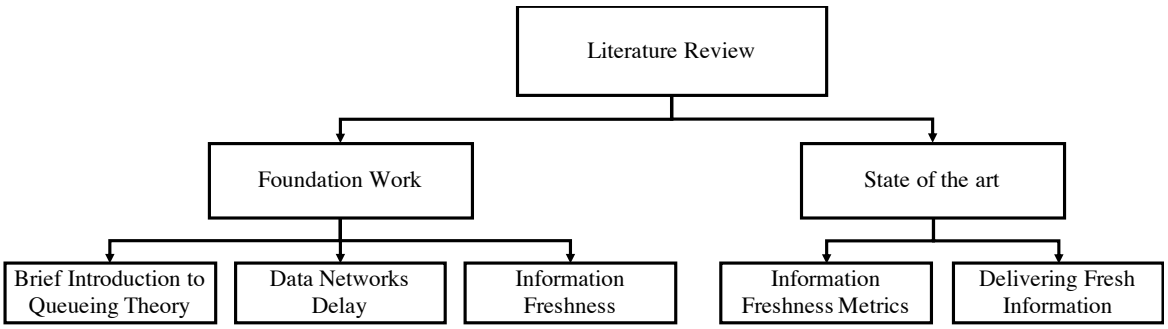

Fig. 2.1 Literature Review Main Sections.

2.1 Foundation Work

This subsection briefly presents the research area background and the main definitions used throughout the thesis. The foundation work is divided into three main parts, i.e., Queuing Theory Fundamentals, Data networks delays and Information freshness. The first part concisely presents the essential background for Queuing Theory. In the data networks delay part, the author presents the intersection between Queuing Theory and data networks, in particular in the analysis of the waiting time in the data networks. While in the Information freshness subsection, the focus was on queuing in real-time database systems.

2.1.1 Brief Introduction to Queuing Theory

Similar to most of the new sciences, the scholars started their investigation by classifying the major categories, i.e., in queuing theory, the queues, to better understand their behaviour. The classification started with defining the parameters that can be used to identify a queue. Consider an Automated Teller Machine (ATM) as an example, where the users join a queue to use it, as shown in Fig. 2.2. To identify this queue, one critical parameter is the rate of customers joining the queue. Hence, the first parameter is the inter-arrival time (X) or rate (λ). The inter-arrival time is defined as the time interval between two consecutive arrivals [3]. The inter-arrival time in the ATM example can be affected by the location of the machine, for instance, if it was located in a shopping centre, the number of customers using the machine would be affected by the centre opening hours. Thus, the inter-arrival time can be a random variable that follows a certain distribution such as an exponential distribution.

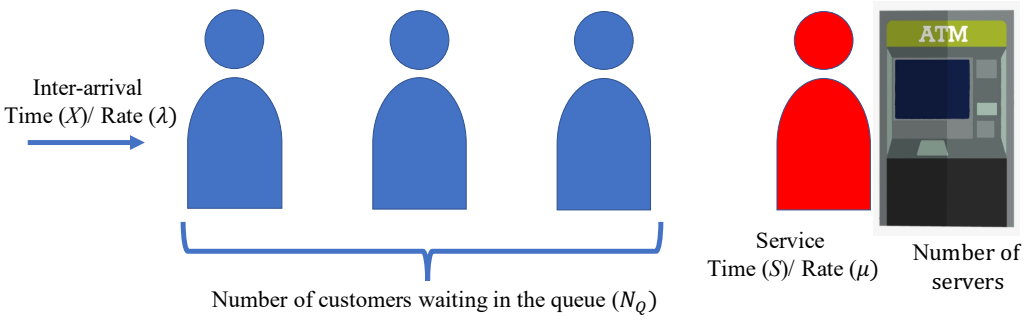

Fig. 2.2 A general queue showing the main parameters to identify a queue.

The number of customers waiting in the queue (N_Q) is critical, as if it is 'long', it might influence the satisfaction of the customers. The number of customers in the queue can also have physical limitations, such as the maximum space that the queue can occupy. However, it is not the only parameter that affects the customers; for instance, if the number of customers in the queue was 'big', but the queue is moving quickly (short waiting time), then the customers might still be satisfied with the service. It is worth mentioning that the waiting time is affected by inter-arrival time and the time spent by a customer using the machine.

The time spent by a customer using the machine, named the service time, is also an important parameter. Several parameters affect the service time; for instance, the time that the machine takes to process a transaction may also it can be affected by the response time of the users [2]. Hence, the service time is commonly modelled as a

random variable following a distribution such as a Poisson. To minimise the service time, several machines can be used. Hence, the number of machines or servers is a parameter to identify a queue.

As shown in the previous example, several parameters play a role in the queue behaviour. To identify a queue, Kendall's notation [1] is commonly used [4]. This notion consists of three main parameters, the inter-arrival time, the service time and the number of servers. Hence they are exposed as follows Arrival/Service/Number of servers. For example, if we consider a queue with periodic arrival and the service time is fixed, and a single server is referred to as D/D/1 queue. The D here refers to a deterministic or fixed value.

A commonly used queue is the M/M/1 queue, which refers to a queue where both the arrivals and the service time follows an exponential distribution. The exponential distribution is a probability distribution [2] of the time between events in a Poisson point process [3] [4]. The exponential distribution has a crucial property, i.e., Memory-less property. A random variable X that poses the Memory-less property is a non-negative non-deterministic and satisfy the following condition if, for every x $\geq$ 0 and t $\geq$ 0,

$$Pr\{X > t + x\} = Pr\{X > x\}\ Pr\{X > t\}. \tag{2.1}$$

For the M/M/1 queue, hence the inter-arrivals time and service time follows an exponential distribution, and both times are independent of each other. The exponential distribution probability density function (pdf) is,

$$Pr\{\widetilde{x} > x\} = e^{(-\lambda x)};\ \text{ for } x > 0. \tag{2.2}$$

The exponential distribution[5] for rate μ is plotted in Fig. 2.3

The M/M/1 queue is commonly used in literature, since it is a behaviour very close to 'real-life' systems, such as the arrival of calls to a central station or customers to a shopping centre. It can be argued that the queue in the ATM example can be modelled

[1] This notation was proposed by David George Kendall in the 1930s.

[2] A probability distribution is a mathematical function that provides the probabilities of occurrence of different possible outcomes in an experiment.

[3] Poisson point process is a renewal process in which the inter-arrival intervals have an exponential distribution function.

[4] A renewal process is an arrival process for which the sequence of inter-arrival times is a sequence of identical and independent random variables.

[5] For the exponential distribution the mean value equals $1/\mu$, where μ is the rate of arrivals or the number of occurrences in a time unit.

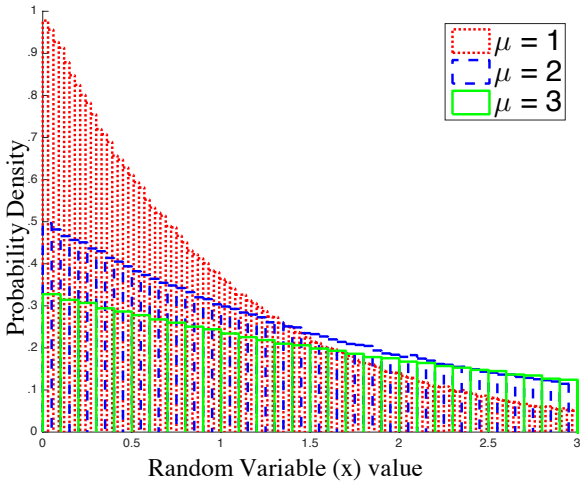

Fig. 2.3 Exponential distribution Probability Density Function for rate $\mu = 1$, 2 and 3.

as an M/M/1 queue. It is also commonly used because it is relatively easy derive a mathematical expression for [6].

In the next subsection, the focus would be on the data networks. In particular, one of the most critical metrics for systems design, i.e., delay. The delay differs according to the queue. However, it is possible to model it for a generic queue that can be used according to the queue parameters, as shown in the next subsection.

2.1.2 Data Networks Delay

To evaluate the performance of a data network, several metrics can be applied. For instance, one commonly used is the data throughput, i.e., the number of bits per unit of time; alternatively one can use the mean delay time required to deliver a piece of information or a status update from its origin to destination. Although both metrics can be used to evaluate the data network, the delay can be used in assessing several aspects of the devices communicating through the network, such as the buffer size.

To understand the delay, let us consider a temperature sensor that sends its readings to a heater to regulate its heating power. Without loss of generality, let us assume that the sensor was initiated at time 0, after a predefined period at time (t_1) it generates its first reading and starts to transmit it to the heater. The time instant in which the sensor reading or status update will be received at the heater is (r_1). The instantaneous number of updates at time (t) transmitted from the sensor is denoted to as ($A(t)$), and the instantaneous number of received updates ($D(t)$). The number of updates that were generated and have not been received (number of updates in the system)

is $N(t)$. As shown in Fig. 2.4 the delay affects the number of updates in the system ($N(t)$) and the number of updates received ($D(t)$); hence, the delay can be used in the design of the network devices buffer size. On the other hand, the delay is affected by the number of updates generated and received per unit of time.

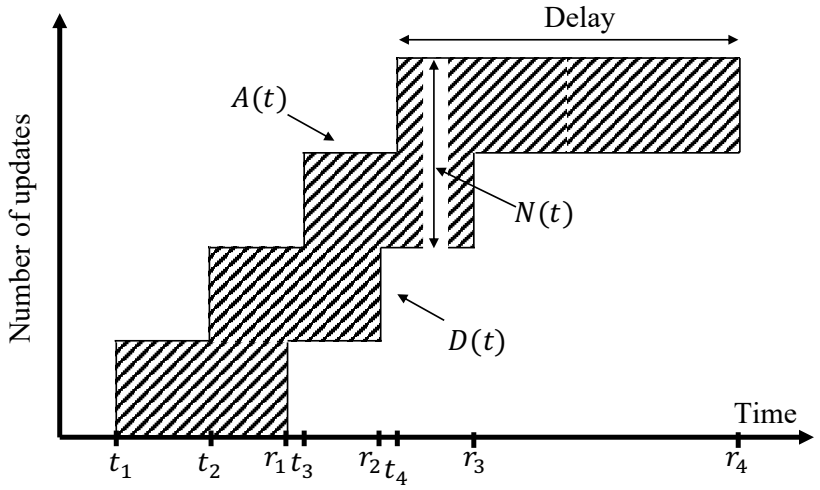

Fig. 2.4 The relation between the delay and the number of arrivals/departures. $A(t)$ refers to the number of arrival updates, $D(t)$ is the number of received updates and $N(t)$ represents the number of updates in the system.

To define the relationship between the delay and the number of updates, it is critical to have a precise definition of the delay [4]. The delay time of update i is calculated using the time of generating (t_i) and receiving it (r_i) as follows,

$$T_i = r_i - t_i. \tag{2.3}$$

The time average (mean) delay [6] for the period 0 up to t, is defined as,

$$T_t = \frac{\sum_{i=0}^{A(t)} T_i}{A(t)}. \tag{2.4}$$

To use the delay as a metric to evaluate the performance of a data network the steady-state time average delay is defined as,

$$T = \lim_{t \to \infty} T_t. \tag{2.5}$$

[6]Time average is the mean time spent in the system per update.

The relation between the number of updates and the delay was defined by John Little [7] [10] as follows,

$$N = \lambda T, \tag{2.6}$$

where λ is the rate of generating updates and $N(t)$ is the steady state number of updates in the system. Both metrics can be calculated as follows

$$\lambda_t = \frac{A(t)}{t}, \tag{2.7}$$

and the steady-state inter-arrival rate is

$$\lambda = \lim_{t \to \infty} \lambda_t. \tag{2.8}$$

Similarly, the number of updates in the system up to time t can be calculated by

$$N_t = \frac{1}{t} \int_0^t N(\tau) d\tau. \tag{2.9}$$

the steady state number of updates in the system is

$$N = \lim_{t \to \infty} N_t. \tag{2.10}$$

From (2.6), it can be concluded that the steady-state delay affects the number of updates in the system, which can be used in designing the network components. On the other hand, it is affected by the rate of generating the updates and the rate of receiving the updates (since scholars were aiming to minimise delay to make the network more efficient). Thus, they started to classify the components of the delay, which are [4]:

- *Transmission delay* which represents the period between the first and last bit of the communication.
- *Waiting delay* also known as the *Queuing delay*; is defined as the time between the generation of the information until it starts to be transmitted.
- *Propagation delay* represents the time between the generation of the last bit at the origin until it is received at its destination.

[7] John Little is an Institute Professor at the Massachusetts Institute of Technology.

- *Processing delay* stands for the period between generating the first bit of the update until it is assigned to the corresponding queue.

Some of these delays are constructed by the physical limitations of the network (e.g. the physical distance and the speed of light), such as the *Propagation delay*. On the other hand, other delays can be eliminated by the network policies and protocols, such as *Waiting delay*. Hence, the delay minimisation had been extensively investigated in the literature [5]. Erlang proposed the investigation in the context of telephone lines in 1908 [1]. In the research done following Erlang until the 1990s, the delay was the typical metric for evaluating the information staleness [4, 5]. However, with the developments in computing and telecommunications, the necessity for new metrics had been born.

2.1.3 Information Freshness as a Metric

In the 1990s, several developments occurred in the telecommunication systems. These developments had significantly changed the way information is communicated. One of the systems that required information delivery promptly was the stock exchange databases [7–9]. Excessive delays in these systems might have severe consequences. Accordingly the authors of [7], were aiming to answer the following question *'How can we maintain consistent data delivery for a database* [8] *that was located far from the source of the updates?'*. They started their investigation by characterising the updates as follows:

- *Complete versus partial updates*: complete updates contain every detail of the monitored process, even if it did not change since the previous update. The partial updates only include the aspect that had changed since the last update.

- *Periodic versus aperiodic updates*: periodic updates have a periodic interval between the updates. The aperiodic updates occur when the monitored process changes.

In their investigation [7], they proposed to evaluate the performance by comparing the time that elapsed since the generation of the update (which was named update Age Δ) to a predefined value named Maximum Age (MA). The importance of the update determined the value of the MA. Thus, they categorised the updates into two

[8] Database is defined as any collection of data or information, that is specially organised for rapid search and retrieval by a computer.

categories, i.e., *High and Low importance*. An example, of *High importance,* was the exchange rate of Dollar to Yen; while the *Low importance*, might be the stock price of a corner shop.

If the update Age was greater than the MA, it was considered as violating the staleness constraint. In their simulations, they used the ratio of the updates that violated the constraint to the total updates to evaluate the performance of the scheduling algorithm. In particular, the fraction of updates violating the stale constrain for *low importance* data was referred to as f_l and *high importance* as f_h. The average fraction of stale updates is calculated by,

$$\overline{f_l} = \frac{\int_0^\tau f_l(t)dt}{\tau} \text{ and } \overline{f_h} = \frac{\int_0^\tau f_h(t)dt}{\tau}, \tag{2.11}$$

where τ is the time of the simulation. The algorithms that were tested are *Do update First*, *Do Transition First*, *Split Updates* and *Apply Updates On Demand*. Further details of these algorithms can be found in [7].

Although their contribution was insightful, it was not complete. Hence, they did not model the updates Age, and they only observed it from the simulation as they evaluated the violating probability. Thus, recently several scholars acknowledged the importance of the Age metric and aimed to model it. Also, the Age metric was used to evaluate scheduling algorithms. In other words, they intended to generalise the Age metric to be used for all the real-time systems.

2.2 State-of-the-art Work

In this section, a brief introduction of the recent contributions made in the last ten years is provided. In particular, it is focused on the contribution made in the information freshness research scope.

2.2.1 Information Freshness Metrics Definitions

Recently, several metrics to evaluate information freshness was proposed. For instance, the Age metric which was briefly mentioned in the previous section. In the latest contributions, the Age and the other metrics were explicitly defined, and that inspired several researchers to revisit this research scope.

Age of Information

In 2012, Yates et al. [11] aimed to answer this insightful question *'How often should we transmit the updates?'*. Although the question had been previously visited, the approach they took was useful. As they started their investigation by defining the goal of real-time systems, i.e., to ensure that the updates (from the process of interest) are delivered to the monitor as timely as possible. Afterwards, they derived an expression for the Age of the general queuing system as follows; consider a sensor that measures the speed of the vehicle and transmits its reading to a central control unit. At time t the last status update the sensor transmitted was generated at time $u(t)$. The status update Age was defined as

$$\Delta(t) \triangleq t - u(t). \tag{2.12}$$

where $\triangleq$ refers to equal by definition.

In a monitoring system, at time $t = 0$, let us assume that no updates were previously sent and all the network queues empty. Considering update i was generated at time t_i, its Age then is

$$\Delta(t_i) = t_i - u(t_i) = 0. \tag{2.13}$$

Then the Age increases linearly, as shown in Fig. 2.5. To calculate the average Age for the updates that had been transmitted during the period $(0 \rightarrow \tau)$, the integral of the Age pattern can be used as follows,

$$\Delta_\tau = \frac{1}{\tau} \int_0^\tau \Delta(t) dt. \tag{2.14}$$

To calculate the integral, the sum of the trapezoids area can be used.

In the literature, they derived the expression time average Age for M/M/1, M/D/1 and D/M/1 queues[9] (details regarding these queues can be found in section 2.1.1). More details regarding these expressions are presented in chapter 3.

Although the Age metric had been defined, we can observe that the time average Age might be challenging to apply and minimise. Hence, another metric had been proposed, i.e., Peak Age [12] (which is presented in the next subsection). Moreover, their definition has not been empirically validated.

[9]They attempted to derive a closed-form expression for the M/D/1 queue Age, however, as they showed it could only be evaluated numerically.

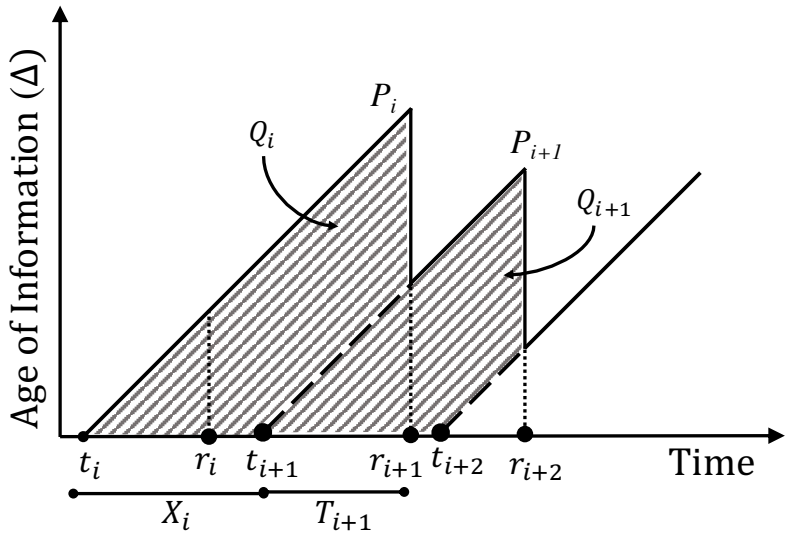

Fig. 2.5 Age of Information illustration. Shown that the ith update was generated at time t_i and received at time r_i. At t_i the Age equals to 0 and starts to increase linearly until it reaches its peak value P_i. The shaded trapezoids Q_i and Q_{i+1} are used to calculate the time average Age.

Peak Age of Information

The Peak Age also expresses the information freshness; however, it is much easier to optimise than the Age. The Peak Age was proposed in [12], as the Age achieved immediately before receiving an update, i.e., the maximum value of Age. Accordingly, it can be used in systems that require the information age to be lower than a certain threshold. We can observe the Peak Age value in Fig. 2.5 as the peak of the Age sawtooth pattern.

The Peak Age (PA) of the i^{th} update can be calculated as follows [13]

$$P_i \triangleq X_i + T_i, \tag{2.15}$$

where $\triangleq$ refers to equal by definition, X_i refers to the inter-arrival time, i.e., the time between generating two updates; and T_i is the delay time [10]. The delay time is the duration between generating the update and instant of finishing processing it. It is also worth mentioning that T_i is depend on the X_i, consequently, changing the mean value for the distribution of any of them would affect the other.

The average PA value P can be obtained by calculating the expected value as follows

[10]Delay time also known as the system time in queuing theory literature.

$$P = \mathbb{E}[X] + \mathbb{E}[T], \tag{2.16}$$

where, $\mathbb{E}$ is expectation operator.

As shown in (2.16), deriving a closed-form expression for the PA is more straightforward than the AoI, and consequently, it is simpler to utilise [12, 13]. Moreover, the PA represents the information freshness. Consequently, it was widely used in the literature as a metric for evaluating the information freshness.

In the literature, the theoretical derivation of both the average age of information and the peak age was proposed. However, until the work done in chapter 3, no method existed to empirically evaluate them. Hence, the first research gap in this thesis is ***'How can we evaluate the Average Age and Average Peak Age from experiments?'***.

2.2.2 Delivering Fresh Information Policies

In the literature, several policies had been proposed to deliver fresh information. The methods to deliver fresh information are commonly optimised to a specific application. For instance, a substantial work focused on delivering fresh information was in the context of energy harvesting sensors, such as sensors that charge from solar cells [14–17]. Also, a significant amount of contributions had been made on delivering fresh information over wireless channels such as [18–20]. This work is critical, since a considerable amount of data is being transmitted over wireless networks. Fresh information is also vital for content caching [11]; thus, several scholars contributed in this field [21–25].

In this thesis, the author is aiming to deliver fresh information for any communication system. Consequently, the focus of this section of the literature review is to present the methods proposed to deliver fresh information in a generic communication system. The policies to maximise information freshness can be categorised into four groups. The groups are (i) Queue discipline; (ii) Number of servers; (iii) Queue size; and (iv) Zero-Wait Policy, also known as just-in-time policy.

[11] Content caching is a mechanism to enhance the timing of data delivery in which data is delivered from the closest servers.

Queue Discipline

One approach to minimise the Age of Information in queuing systems is to control the queue discipline. The queuing systems discipline represents the order of serving the updates. The commonly used discipline in the literature, is the *First-come-first-Served* (FCFS), also known as *First-in-first-out* (FIFO), in which the system serves the update that joined the queue first as shown in Fig. 2.6 [29].

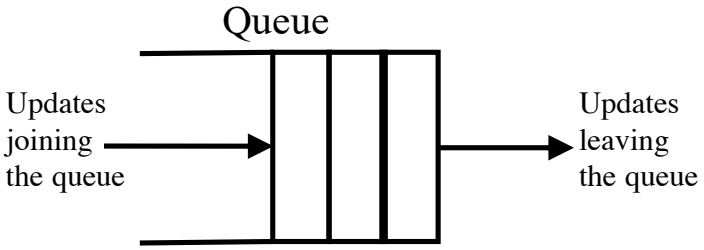

Fig. 2.6 First-Come-First-Served discipline, in which the update that is being served is the update that first joined the queue.

To minimise the Age of Information, the *Last-Come-First-Served* (LCFS) discipline was proposed [30, 31]. In LCFS, the updates that will be served are the updates that joined the queue the last, as shown in Fig. 2.7. The LCFS queues with and without the ability to preempt the packet currently in service were analysed. The LCFS queue with a preamble is a queue that drops the update being processed at the instance a new update joined the queue. It was shown that the LCFS with preamble Age outperforms the LCFS without preamble.

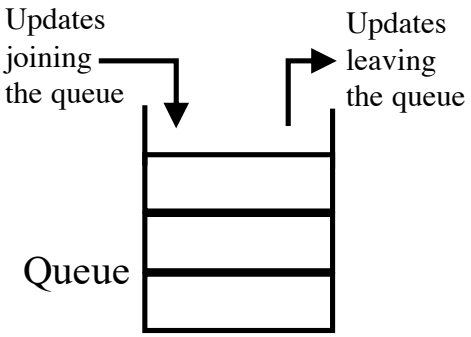

Fig. 2.7 Last-Come-First-Served queue discipline, in which the update that is being served is the update that last joined the queue.

The LCFS was also analysed for discrete-time queues [32]. A discrete-time queue represents a wireless communication channel, where the information can be sent through

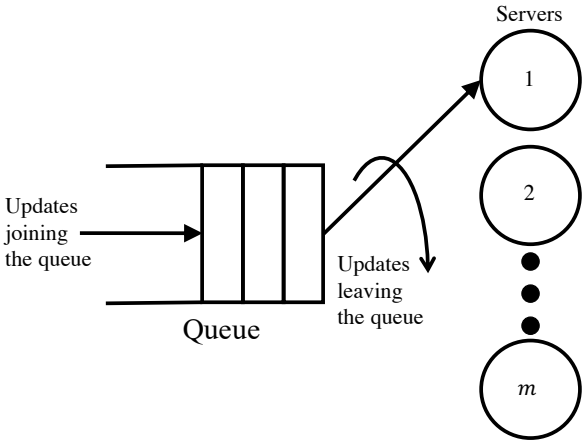

Fig. 2.8 A queue with m servers to maximise the information freshness [33].

the channel at the beginning of the time slot and it is assumed that it would finish the communication at the end of the slot.

Although this approach can achieve a low Age of the updates, it can be considered as not suitable for several applications. For instance, discontinuous of the information of the observed process might cause some issues in the performance of the system. For example, if the monitored process was the location of an ambulance vehicle, with the LCFS, the vehicle will appear to be stopping and leaps to the next location instead of having a consistent monitoring, each time an update taking longer than the mean service time. In several other examples, using the LCFS might be not the most efficient policy; thus, other approaches were proposed in the literature.

Number of Servers

The number of servers affects the queue service time [33–36]. Two approaches for interrupting multiple servers had been proposed in the literature, the first approach used in [33], is the approach typically used in queuing theory, in which the server is the destination, where the updates terminate. Hence, in this approach, as shown in Fig. 2.8, having several servers reduces the service time, consequently the delay time and maximise the information freshness.

In the second approach [34–36], the authors refers to the servers as a vehicle used to deliver the updates to a server, as shown in Fig. 2.9 [12]. This approach was proposed to represent a wireless base station, in which there where a certain number of channels to handle the data transmitted from the mobile phones to the base station.

[12]In queuing theory terms, the server here referees to the resources in a resources allocation problem.

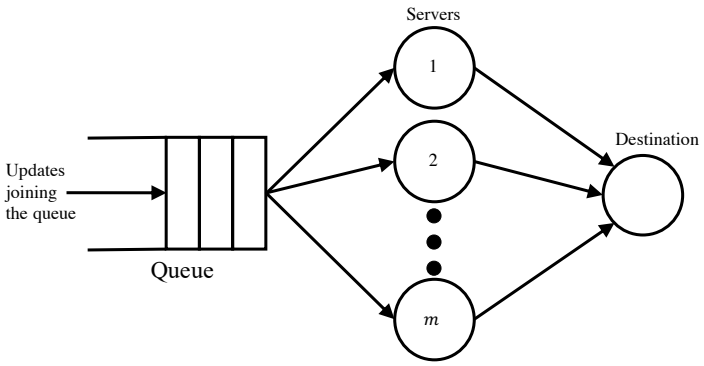

Fig. 2.9 A queue with m servers to deliver the updates to a destination [34–36].

Although this approach can reduce the Age value, it requires additional servers, hence extra communication channels for the case of wireless base stations and extra hardware for the case of databases. Thus, it can be considered as inadequate for several applications.

Queue Size

The queue size can also contribute to delivering fresh information [37]. Controlling the queue size delivers fresh information by minimising the mean service time, by eliminating the waiting updates, as shown in Fig. 2.10. In other words, this approach controls the maximum number of packets waiting in a queue. If we consider a queue size of one as shown in Fig. 2.10, if the queue has an update waiting and a new update was generated, the queue would drop an update. In the literature, there are two approaches to drop the update. The first approach would drop the newly generated update, and the second approach would drop the 'oldest' update, and the newly generated update will join the queue. Although, this approach can deliver fresh information it has some disadvantages as well, for example in this approach the sensor would consume some energy to generate an update and then it would drop it to maintain the freshness. Also, when dropping the update the information on the destination side would not be consistent.

Zero-Wait Policy

The main aim of the Zero-Wait policy is to eliminate the waiting time in the queue and, hence, optimise the information freshness. Eliminating the waiting time minimise the delay time (T) and hence, the Peak Age value; as shown in (2.16), (i.e., $P = \mathbb{E}[X]+\mathbb{E}[T]$). Eliminating the waiting time is achieved by having the destination of the updates sends

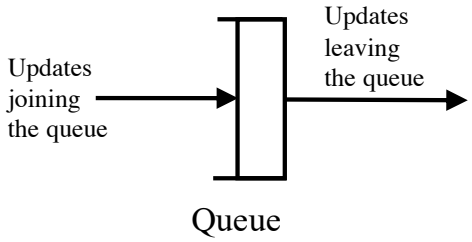

Fig. 2.10 A queue size of one.

an Acknowledgement (ACK) to the source of the update after processing it, as shown in Fig. 2.11. Hence, the number of updates waiting in the queue will be equal to zero [26–28].

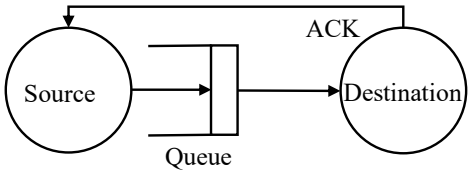

Fig. 2.11 Zero-Wait policy system diagram.

Although the Zero-Wait policy seems to be delivering fresh information, it was only considered to be operating on a simple queuing system. In chapter 4, the author examined the Zero-Wait policy empirically and highlighted the limitations of it.

Several other policies had been proposed to deliver fresh information in generic networks. For example, applying reinforcement learning algorithms to deliver fresh information such as in [38–41]. The Lazy updating policy was also proposed to deliver fresh information in generic queues. Lazy updating is a novel idea of adding an extra delay to optimise the Age performance of the queue [42]. The idea was proposed in the context of energy harvesting [13] nodes.

[13]Energy harvesting is the process of capturing the energy that otherwise would be lost s heat, light, sound, vibration or movement [43].

Chapter References

[1] Erlang, A.K., 1909. The theory of probabilities and telephone conversations. Nyt. Tidsskr. Mat. Ser. B, 20, pp.33-39.

[2] Keates, S., Langdon, P., Clarkson, P.J. and Robinson, P., 2002. User models and user physical capability. User Modeling and User-Adapted Interaction, 12(2-3), pp.139-169.

[3] Gallager, R.G., 2012. Discrete stochastic processes (Vol. 321). Springer Science & Business Media.

[4] Bertsekas, D.P., Gallager, R.G. and Humblet, P., 1992. Data networks (Vol. 2). New Jersey: Prentice-Hall International.

[5] Gross, D., 2008. Fundamentals of queueing theory. John Wiley & Sons.

[6] Feller, W., 2008. An introduction to probability theory and its applications (Vol. 2). John Wiley & Sons.

[7] Adelberg, B., Garcia-Molina, H. and Kao, B., 1995, June. Applying update streams in a soft real-time database system. In ACM SIGMOD Record (Vol. 24, No. 2, pp. 245-256). ACM.

[8] Kuo, T.W. and Mok, A.K., 1993, December. SSP: A semantics-based protocol for real-time data access. In 1993 Proceedings Real-Time Systems Symposium (pp. 76-86). IEEE.

[9] Song, X. and Liu, J.W.S., 1990, October. Performance of multiversion concurrency control algorithms in maintaining temporal consistency. In Proceedings., Fourteenth Annual International Computer Software and Applications Conference (pp. 132-139). IEEE.

[10] Little, J.D., 1961. A proof for the queuing formula. Operations Research, 9(3), pp.383-387.

[11] S. Kaul, R. Yates, and M. Gruteser, “Real-time status: How often should one update?,” in 2012 Proceedings IEEE INFOCOM, 2012, pp. 2731–2735.

[12] M. Costa, M. Codreanu, and A. Ephremides, “On the Age of Information in Status Update Systems With Packet Management,” IEEE Trans. Inf. Theory, vol. 62, no. 4, pp. 1897–1910, Apr. 2016.

[13] L. Huang and E. Modiano, “Optimising age-of-information in a multi-class queueing system,” in 2015 IEEE International Symposium on Information Theory (ISIT), 2015, pp. 1681–1685.

[14] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, 2015, pp. 3008-3012. doi: 10.1109/ISIT.2015.7283009

[15] A. Arafa, J. Yang, S. Ulukus and H. V. Poor, "Age-Minimal Online Policies for Energy Harvesting Sensors with Incremental Battery Recharges," 2018 Information Theory and Applications Workshop (ITA), San Diego, CA, 2018, pp. 1-10. doi: 10.1109/ITA.2018.8503180

[16] B. T. Bacinoglu, E. T. Ceran and E. Uysal-Biyikoglu, "Age of information under energy replenishment constraints," 2015 Information Theory and Applications Workshop (ITA), San Diego, CA, 2015, pp. 25-31. doi: 10.1109/ITA.2015.7308962

[17] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, 2015, pp. 3008-3012. doi: 10.1109/ISIT.2015.7283009

[18] A. Valehi and A. Razi, "Maximizing Energy Efficiency of Cognitive Wireless Sensor Networks With Constrained Age of Information," in IEEE Transactions on Cognitive Communications and Networking, vol. 3, no. 4, pp. 643-654, Dec. 2017. doi: 10.1109/TCCN.2017.2749232

[19] Ning Lu, Bo Ji, and Bin Li. 2018. Age-based Scheduling: Improving Data Freshness for Wireless Real-Time Traffic. In Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc ’18). ACM, New York, NY, USA, 191-200. DOI: https://doi.org/10.1145/3209582.3209602

[20] Bin Li, Ruogu Li, and Atilla Eryilmaz. 2013. Heavy-traffic-optimal scheduling with regular service guarantees in wireless networks. In Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '13). ACM, New York, NY, USA, 79-88. DOI=http://dx.doi.org/10.1145/2491288.2491304

[21] Corneo, L. and Gunningberg, P., 2018, July. Scheduling at the edge for assisting cloud real-time systems. In Proceedings of the 2018 Workshop on Theory and Practice for Integrated Cloud, Fog and Edge Computing Paradigms (pp. 9-14). ACM.

[22] Zhang, S., Li, J., Luo, H., Gao, J., Zhao, L. and Shen, X.S., 2018, October. Towards Fresh and Low-Latency Content Delivery in Vehicular Networks: An Edge Caching Aspect. In 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP) (pp. 1-6). IEEE.

[23] Kam, C., Kompella, S., Nguyen, G.D., Wieselthier, J.E. and Ephremides, A., 2017, June. Information freshness and popularity in mobile caching. In 2017 IEEE International Symposium on Information Theory (ISIT) (pp. 136-140). IEEE.

[24] Yates, R.D., Ciblat, P., Yener, A. and Wigger, M., 2017, June. Age-optimal constrained cache updating. In 2017 IEEE International Symposium on Information Theory (ISIT) (pp. 141-145). IEEE.

[25] Gao, W., Cao, G., Srivatsa, M. and Iyengar, A., 2012, June. Distributed maintenance of cache freshness in opportunistic mobile networks. In 2012 IEEE 32nd International Conference on Distributed Computing Systems (pp. 132-141). IEEE.

[26] Kuang, Q., Gong, J., Chen, X. and Ma, X., 2019. Age-of-Information for Computation-Intensive Messages in Mobile Edge Computing. arXiv preprint arXiv:1901.01854.

[27] Champati, J.P., Al-Zubaidy, H. and Gross, J., 2019. On the distribution of AoI for the GI/GI/1/1 and GI/GI/1/2* systems: Exact expressions and bounds. In IEEE INFOCOM.

[28] Wang, M., Chen, W. and Ephremides, A., 2019. Real-Time Reconstruction of Counting Process through Queues. arXiv preprint arXiv:1901.08197.

[29] Daigle, J.N., 2005. Queueing theory with applications to packet telecommunication. Springer Science & Business Media.

[30] S. K. Kaul, R. D. Yates and M. Gruteser, "Status updates through queues," 2012 46th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, 2012, pp. 1-6.

[31] Bedewy, A.M., Sun, Y. and Shroff, N.B., 2019. Minimising the AAge of information through queues. IEEE Transactions on Information Theory.

[32] Tripathi, V., Talak, R. and Modiano, E., 2019. Age of Information for Discrete Time Queues. arXiv preprint arXiv:1901.10463.

[33] C. Kam, S. Kompella and A. Ephremides, "Effect of message transmission diversity on status age," 2014 IEEE International Symposium on Information Theory, Honolulu, HI, 2014, pp. 2411-2415.

[34] Yates, R.D., 2018, June. Status updates through networks of parallel servers. In 2018 IEEE International Symposium on Information Theory (ISIT) (pp. 2281-2285). IEEE.

[35] Yates, R.D., 2018, April. Age of information in a network of preemptive servers. In IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 118-123). IEEE.

[36] Bedewy, A.M., Sun, Y. and Shroff, N.B., 2016, July. Optimising data freshness, throughput, and delay in multi-server information-update systems. In 2016 IEEE International Symposium on Information Theory (ISIT) (pp. 2569-2573). IEEE.

[37] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier and A. Ephremides, "On the Age of Information With Packet Deadlines," in IEEE Transactions on Information Theory, vol. 64, no. 9, pp. 6419-6428, Sept. 2018.

[38] Sert, E., Sönmez, C., Baghaee, S. and Uysal-Biyikoglu, E., 2018, May. Optimising AAge of information on real-life TCP/IP connections through reinforcement learning. In 2018 26th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

[39] Elgabli, A., Khan, H., Krouka, M. and Bennis, M., 2018. Reinforcement learning based scheduling algorithm for optimising AAge of information in ultra reliable low latency networks. arXiv preprint arXiv:1811.06776.

[40] Ceran, E.T., Gündüz, D. and György, A., 2018, September. A reinforcement learning approach to AAge of information in multi-user networks. In 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) (pp. 1967-1971). IEEE.

[41] Ceran, E.T., Gündüz, D. and György, A., 2019. Average Age of information with hybrid ARQ under a resource constraint. IEEE Transactions on Wireless Communications, 18(3), pp.1900-1913.

[42] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, 2015, pp. 3008-3012.

[43] Priya, S. and Inman, D.J. eds., 2009. Energy harvesting technologies (Vol. 21, p. 2). New York: Springer.

Chapter 3

Measuring the Average and Peak Age of Information in Real Networks

- **Research Gap:**
  The lack of empirical work on the information freshness.
- **Published paper:**
  - B. Barakat, H. Yassin, S. Keates, K. Arshad and I. J. Wassell, 'How to Measure the Average and Peak Age of Information in Real Networks?' Accepted in IEEE 25^{th} European Wireless conference (EW) 2019.
- **Most relevant papers:**
  - C. Kam, S. Kompella, and A. Ephremides, "Experimental evaluation of the ageof information via emulation," in Proceedings - IEEE Military Communications Conference MILCOM, 2015, vol. 2015-Decem, pp. 1070-1075.
  - C. Sönmez, S. Baghaee, A. Ergişi and E. Uysal-Biyikoglu, "Age-of-Information in Practice: Status Age Measured Over TCP/IP Connections Through WiFi, Ethernet and LTE," 2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Batumi, 2018, pp. 1-5.

"It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment, it's wrong", Richard P. Feynman (Nobel prize in Physics).

The Age of Information (AoI) was proposed in the literature to quantify the freshness of information. The majority of the work done in this area has theoretically evaluated AoI and its Peak (PA). In this chapter, a method for obtaining the value of AoI and PA from experiments is proposed. The author conducted an experiment emulating M/M/1, D/D/1 and M/D/1 queues and used the proposed method to evaluate AoI and PA. The values were compared to the expressions presented previously in the literature. The results show that the proposed method is accurate for the tested queues. A statistical test was conducted to confirm the reliability of this conclusion.

3.1 Introduction

Over the decades, continuous breakthroughs in communication technologies gave birth to a range of applications with different requirements. Many Internet of Things (IoT) applications are based on receiving updates about the status of a remote agent to help in decision making. Examples include wireless channel quality estimation [1], telehealth, environmental and industrial monitoring. For some of these applications, it is crucial that, at any point in time, the status that the decision maker has is up to date and represents as much as possible the current status at the source. There is a subtle difference between this requirement and the traditional low latency (delay) requirement because the latter is seen purely from the perspective of network performance, while the former is seen from the destination's perspective. In other words, low-latency is not equivalent to the freshest updates at the receiver; it very much depends on how frequently the updates are being generated at the source.

To address this requirement, a new metric called Age of Information (AoI) was introduced in [7] to measure and quantify the freshness of information from the receiver's perspective. It is defined as the time since the last update received was generated. The main difference between AoI and conventional network delay metrics is that AoI is observed from the receiver's perspective, while the delay is observed from the network's perspective. The Peak Age of Information (PA), introduced in [3], is another metric related to AoI and represents the worst case AoI. It is defined as the maximum time elapsed since the preceding piece of information was generated. The PA metric has a

simpler formulation and is a more mathematically tractable metric [3]. Consequently, modelling PA has gained attention in the literature such as in [3, 4]. Also, minimising the PA by optimising network functions was extensively investigated [3, 5].

However, the majority of work on AoI and PA has been theoretical and assumed simple queuing models to derive theoretical results about these metrics. In [6], an emulation-based validation of the theoretical models was presented. More recently, [11] presented experimental results that validated the non-monotonous nature of AoI as a function of link utilisation. However, in both papers, no clear explanation was provided as to how exactly the metrics were evaluated from the experiment. This chapter aims to bridge this gap and provide a clear and concise tutorial for experimental researchers that wish to evaluate these metrics on real networks. The contributions of this chapter can be summarised as follows:

- the author provide an intuitive formulation of how the AoI and PA metrics can be estimated from the recorded time-stamps in an experiment,
- the author validate these expressions by performing an experiment comprising $M/M/1$, $D/D/1$ and $M/D/1$ queues, and
- the author present a simple methodology for conducting experiments.

The rest of the chapter is organised as follows. In section 3.2, a definition for the new metrics and some related quantities. In section 3.3, we present the proposed method to estimate the delay, AoI, and PA. In section 3.4, we present a case study to validate our method and we compare the proposed estimates against their theoretical counterparts. We conclude the chapter in section 3.5.

3.2 Definitions and Previous Work

In this section, the main definition that is used throughout the thesis is presented.

3.2.1 Age of Information

Consider a destination and an information source that is generating updates at discrete times (possibly by sampling a process) and then instantaneously transmit them to the destination through a communication network. We denote by t_i the time at which the ith update was generated/transmitted at the source and by r_i the time at which

it was received at the destination. We define $X_i = t_i - t_{i-1}$ as the time between the generation of updates i and $i-1$, i.e., the updates inter-arrival time. We also define the delay time (system) $T_i = r_i - t_i$ as the time it took, from the generation of ith update, until its reception at the destination. The Age of Information (AoI) at time t, denoted $\Delta(t)$, is defined as the time elapsed since the last received update at the destination was generated at the source. Mathematically, it is given by $\Delta(t) = t - u(t)$, where $u(t)$ is the generation time of the last received update at time t. Fig. 3.1 illustrates an example of how the information age evolves with time as a sawtooth function [7].

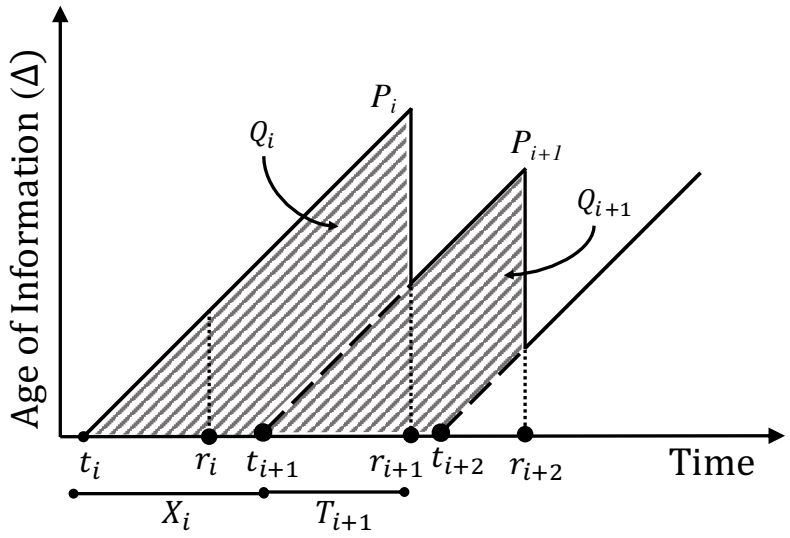

Fig. 3.1 Age of Information as a function of time. The updates inter-arrival time are referred to as X_i and the delay (system time) is T_i, i.e., the service time plus the queuing time. The PA of information i is represented by P_i. The time of generating update i is t_i and the time of receiving it is r_i.

3.2.2 Time Average Peak AoI

For time interval $(0, \mathcal{T})$, where $\mathcal{T}$ is assumed, for simplicity, to coincide with the receipt of the nth update, i.e., $\mathcal{T} = r_n$, where r_n is the time that the last piece of information, n, was received in the time interval. The average (mean) time delay in this period can be written as

$$T_{\mathcal{T}} = \frac{1}{n} \sum_{i=1}^{n} T_i. \tag{3.1}$$

The Peak Age of an update is its age at the time of receipt of the next update [3], i.e., the Peak Age of the ith update is $\Delta(r_{i+1}) = X_i + T_i$. Therefore, it is possible to define the time-average Peak AoI in the period $\mathcal{T}$ as follows

$$P_{\mathcal{T}} = \frac{1}{n-1}\sum_{i=1}^{n-1}\Delta(r_{i+1}). \tag{3.2}$$

In the literature, there is an interest in the stationary case when $\mathcal{T} \to +\infty$. It can be seen that the delay time converges to $\mathbb{E}[T]$ and the peak average AoI converges to

$$P = \lim_{\mathcal{T}\to+\infty} P_{\mathcal{T}} = \mathbb{E}[X+T]. \tag{3.3}$$

3.2.3 Time Average Age of Information

The time average AoI in the interval $(0, \mathcal{T})$, denoted $\bar{\Delta}_{\mathcal{T}}$, is the area under the sawtooth function normalised by the observation period $(\mathcal{T})$, and it is given by

$$\Delta_{\mathcal{T}} = \frac{1}{\mathcal{T}}\int_0^{\mathcal{T}} \Delta(t)\, dt. \tag{3.4}$$

By a geometric argument, the area under the curve can be re-written as the sum of the of the areas in Fig. 3.1. Starting from $t = 0$, these are the areas of the polygon Q_1, trapezoids Q_i for $2 \leq i \leq n$, and the triangle of length T_n between t_n and r_n. The first update i.e., $(i = 1)$ is shown in Fig. 3.1. Hence,

$$\Delta_{\mathcal{T}} = \frac{Q_1 + \sum_{i=2}^{n} Q_i + T_n^2/2}{\mathcal{T}} \tag{3.5}$$

The trapezoid area Q_i can be written as the area difference between the right isosceles triangles of legs $T_i + X_i$ and T_i, respectively. This is

$$Q_i = \frac{1}{2}(T_i + X_i)^2 - \frac{1}{2}T_i^2 \tag{3.6}$$

$$= T_i X_i + \frac{X_i^2}{2}. \tag{3.7}$$

For the steady state the average Age converges to

$$\Delta = \lim_{\mathcal{T}\to+\infty} \Delta_{\mathcal{T}} = \frac{\mathbb{E}[Q]}{\mathbb{E}[X]}. \tag{3.8}$$

$$\Delta = \lambda\Big(\mathbb{E}[XT] + \frac{\mathbb{E}[T^2]}{2}\Big), \tag{3.9}$$

where X is the random variable representing the inter-arrival time of updates at the source (generation) with its rate $\lambda = 1/X$, and T is the random variable representing the system time (delay) of an update. The expectations in the expressions depend on the network. To abstract the details of the underlying network, it is common to assume idealised queuing models such as the M/M/1, M/D/1, D/M/1 [3, 7]. Each model makes a different assumption about the update generation (X) and system time (T) which is composed of waiting time in the queue and service time in the network.

3.3 Estimation of the Metrics from Experiments

Consider a setup in which the source transmits an update at generation time (t_i) and that the receiver records its time of receipt (r_i). It is recommended that all calculations be performed after the end of the experiment as the time to calculate the metrics might affect the accuracy of the logged timings.

The delay that the ith update exhibits is calculated by

$$T_i = r_i - t_i. \tag{3.10}$$

The expected delay of (3.10) can be estimated using the sample median of the vector that contains all the delays of all the updates transmitted in the time interval $(0, \mathcal{T})$, i.e., the vector

$$T_{(1\rightarrow n)} = \Big[T_1, T_2, ..., T_n\Big], \tag{3.11}$$

where n is the total number of updates communicated. The sample median is employed instead of the sample average because in some cases, the network protocol might re-transmit some packets, as in TCP/IP and HARQ protocols, if they suffered from significant errors. The re-transmission will increase the delay time hence would significantly affect the mean value.

Similarly, the average PA can be estimated using the sample median of the vector containing the PA of all the status updates communicated in the experiment, i.e., the vector

$$P_{(1\rightarrow n)} = \Big[P_1, P_2, ..., P_n\Big]. \tag{3.12}$$

To use the initial definition $P_i = X_i + T_i$ requires first finding the inter-arrival time and the delays. In the following, we provide an easier formulation using only the times of generation and receipt as shown in Fig. 3.2. In the figure, it is clear that the peak

age of an update stretches from its time of generation until the time of receipt of the next update. Hence, P_i can be evaluated as

$$P_i = r_{i+1} - t_i. \tag{3.13}$$

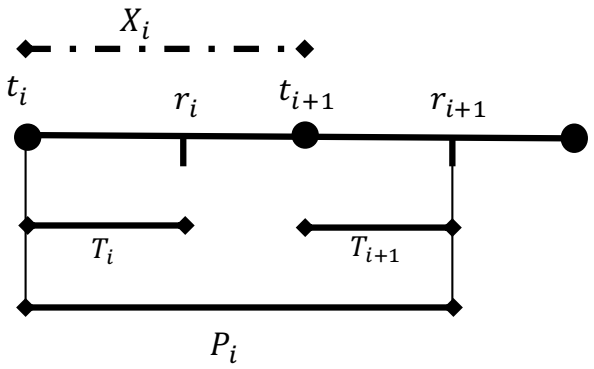

Fig. 3.2 Peak Age of Information measuring method illustration, where t_i is the time that information i was generated, r_i is the time in which information i was received by the server, where X_i represents the updates inter-arrival time. The Peak Age can be considered as the difference between the time of receiving the next update and the time of generating the update.

Finally, to estimate the average AoI the areas of Q_i should be calculated. The area of a trapezoid can be calculated from (3.6), as follows,

$$\begin{aligned} Q_i &= \frac{1}{2}(T_i + X_i)^2 - \frac{1}{2}T_i^2 \\ &\because X_i = t_{i+1} - t_i \ \text{ and } \ T_i = r_i - t_i \\ &\therefore Q_i = \frac{1}{2}\left(\Big((t_{i+1} - t_i) + (r_i - t_i)\Big)^2 - \Big(r_i - t_i\Big)^2\right). \end{aligned} \tag{3.14}$$

The average AoI can then be estimated as follows

$$\Delta = \frac{\sum_{i=2}^{n} Q_i}{(r_n - t_2)}. \tag{3.15}$$

Therefore, in this section, we can express the metrics of interest in terms of the observable time-stamps coming from an experiment.

3.4 Tested Case Studies

In this section, we validate our estimators by emulating $M/M/1$, $D/D/1$ and $M/D/1$ queues and comparing the measured to the theoretical results. We start by reviewing the theoretical results for the queues; then we give our validation setup before presenting and discussing the results.

3.4.1 M/M/1 Queue

The method of measurement was tested on an M/M/1 queue, where the inter-arrival time and the service time follow an exponential distribution with rates λ and μ, respectively. The expected delay of such model, denoted $\mathbb{E}[T]$, is given by [10]

$$\begin{aligned}\mathbb{E}[T] &= \mathbb{E}[W+S] \\ &= \frac{\lambda}{\mu(\mu-\lambda)} + \frac{1}{\mu}, \end{aligned} \tag{3.16}$$

where $\mathbb{E}[W] = \lambda/(\mu(\mu-\lambda))$ is the expected waiting time in the queue and $\mathbb{E}[S] = 1/\mu$ is the expected service time. From (3.3), (3.16), and the fact that $\mathbb{E}[X] = 1/\lambda$, the theoretical $M/M/1$ average Peak AoI is [3]

$$\begin{aligned}P^{M/M/1} &= \mathbb{E}[X+T] \\ &= \frac{1}{\lambda} + \frac{\lambda}{\mu(\mu-\lambda)} + \frac{1}{\mu} && (3.17) \\ &= \frac{1}{\mu}\left(1 + \frac{1}{\rho} + \frac{\rho}{1-\rho}\right), && (3.18)\end{aligned}$$

where $\rho = \lambda/\mu$ is the link utilisation.

The average AoI of an $M/M/1$ queue from (3.9) and (3.16), can be written as follows,

$$\begin{aligned}\Delta^{M/M/1} &= \lambda\left(\mathbb{E}[XT] + \frac{\mathbb{E}[T^2]}{2}\right) && (3.19) \\ &= \lambda\left(\mathbb{E}[XS + XW] + \frac{\mathbb{E}[T^2]}{2}\right). && (3.20)\end{aligned}$$

The service time does not affect the inter-arrival time or $\mathbb{E}[X]$ is independent of the $\mathbb{E}[S]$. Then,

$$\mathbb{E}[XS] = \frac{1}{\lambda} \times \frac{1}{\mu}. \tag{3.21}$$

On the other hand, the waiting time depends on the inter-arrival time. Hence, a short inter-arrival would increase the number of updates in the queue (consequently increase in the waiting time). Therefore, $\mathbb{E}[XW]$ is calculated using

$$\mathbb{E}[XW] = \frac{\rho}{\mu^2(1-\rho)}. \tag{3.22}$$

From (3.21) and (3.22),

$$\mathbb{E}[XT] = \mathbb{E}[XS + XW] \tag{3.23}$$

$$= \frac{1}{\lambda\mu} + \frac{\rho}{\mu^2(1-\rho)} \tag{3.24}$$

$$= \frac{1}{\mu}\left(\frac{1}{\lambda} + \frac{\rho}{\mu(1-\rho)}\right). \tag{3.25}$$

In the M/M/1 queue, the inter-arrival time an exponential distribution, hence,

$$\mathbb{E}[T^2] = \frac{2}{\lambda^2}. \tag{3.26}$$

Hence, the average age is,

$$\Delta^{M/M/1} = \lambda\left(\frac{1}{\mu}\big(\frac{1}{\lambda} + \frac{\rho}{\mu(1-\rho)}\big) + \frac{1}{\lambda^2}\right) \tag{3.27}$$

$$= \lambda\left(\frac{1}{\mu}\big(\frac{1}{\lambda} + \frac{\rho}{\mu(1-\rho)} + \frac{1}{\rho\lambda}\big)\right). \tag{3.28}$$

Finally, the closed expression for M/M/1 queue average Age is then,

$$\Delta^{M/M/1} = \frac{1}{\mu}\left(1 + \frac{1}{\rho} + \frac{\rho^2}{1-\rho}\right). \tag{3.29}$$

3.4.2 D/D/1 Queue

For a $D/D/1$ queue, the inter-arrival time and the service time are a fixed number, or following a deterministic distribution. For a stable queue, where the inter-arrival rate is less than the service rate, i.e., $\lambda < \mu$, or the time between two inter-arrivals time is

longer than the service time. Hence, the waiting time equals to zero, accordingly, the delay time equals to the service time, as follows,

$$T^{D/D/1} = \mathbb{E}[W + S] \tag{3.30}$$
$$= \mathbb{E}[S] = \mathbb{E}[\frac{1}{\mu}] \tag{3.31}$$
$$= \frac{1}{\mu}. \tag{3.32}$$

The Peak Age for the $D/D/1$ queue can be obtained from (3.3) and (3.32), as follows,

$$P^{D/D/1} = \mathbb{E}[X + T] = \mathbb{E}[X + S] \tag{3.33}$$
$$= \mathbb{E}[\frac{1}{\lambda}] + \mathbb{E}[\frac{1}{\mu}], \tag{3.34}$$
$$= \frac{1}{\lambda} + \frac{1}{\mu}. \tag{3.35}$$

The average AoI of an $D/D/1$ queue is given using (3.9) and (3.32), as follows

$$\Delta^{D/D/1} = \lambda\Big(\mathbb{E}[XT] + \frac{\mathbb{E}[T^2]}{2}\Big) \tag{3.36}$$
$$= \lambda\Big(\mathbb{E}[XS] + \frac{\mathbb{E}[S^2]}{2}\Big)$$
$$= \lambda\Big(\mathbb{E}[\frac{1}{\lambda} \times \frac{1}{\mu}] + \frac{\mathbb{E}[(\frac{1}{\mu})^2}{2}\Big)$$
$$= \lambda\Big(\mathbb{E}[\frac{1}{\lambda} \times \frac{1}{\mu}] + \frac{1}{2\mu^2}\Big)$$
$$= \lambda\Big(\frac{1}{\lambda} \times \frac{1}{\mu} + \frac{1}{2\mu^2}\Big)$$
$$= \frac{1}{\mu} + \frac{\lambda}{2\mu^2}$$
$$= \frac{1}{\mu} + \frac{\rho}{2\mu}. \tag{3.37}$$

Finally, Average Age of Information closed form expression for $D/D/1$ queue is,

$$\Delta^{D/D/1} = \frac{1}{\mu}\Big(1 + \frac{\rho}{2}\Big). \tag{3.38}$$

3.4.3 M/D/1 Queue

For an M/D/1 queue the Delay time can be calculated using the Pollaczek–Khinchine formula [10]. The expression for the delay time of general queue is

$$T = \overline{S} + \frac{\lambda \times \overline{S^2}}{2\Big(1-\rho\Big)}, \tag{3.39}$$

where, $\overline{S} = \mathbb{E}[S]$ is the expectation of the service time, and $\overline{S^2} = \mathbb{E}[S^2]$ is the second moment [1].

For an M/D/1 queue, the inter-arrival follows an exponential distribution and hence, $\overline{S} = \mathbb{E}[S] = 1/\mu$ and $\overline{S^2} = 1/\mu^2$. From (3.39), the delay time can be derived as follows

$$T = \frac{1}{\mu} + \frac{\lambda}{2}\frac{\Big(1/\mu^2\Big)}{(1-\rho)}, \tag{3.40}$$

or

$$T = \frac{1}{\mu} + \frac{\rho}{2\mu(1-\rho)} \tag{3.41}$$

The PA fro M/D/1 queue is derived using the PA general queue (G/G/1) formulation [3], i.e.,

$$P^{G/G/1} = \mathbb{E}\Big[X + T\Big]. \tag{3.42}$$

By substituting the value of the M/D/1 queue delay time from (3.41), The PA is,

$$P^{M/D/1} = \frac{1}{\lambda} + \frac{1}{\mu} + \frac{\rho}{2\mu(1-\rho)}; \tag{3.43}$$

by taking $\frac{1}{\mu}$ as a common factor and hence $\rho = \lambda/\mu$, the PA of an M/D/1 queue is equal to,

$$P^{M/D/1} = \frac{1}{\mu} \times \left(\frac{\mu}{\lambda} + 1 + \frac{\rho}{2(1-\rho)}\right). \tag{3.44}$$

[1] It is worth mentioning, that the second moment differs from the variance. The service time second moment is $\mathbb{E}[S^2]$, while the variance, $\mathbb{E}\{S - \mathbb{E}[S]^2\}$.

or,

$$P^{M/D/1} = \frac{1}{\mu} \times \left(1 + \frac{1}{\rho} + \frac{\rho}{2(1-\rho)}\right). \tag{3.45}$$

The average age closed form expression is not derived until the time of writing the thesis. However, as shown in the literature, it can be obtained numerically [7]. Hence, in this chapter and the following chapter we will use the Peak Age for M/D/1 queues.

3.4.4 Experimental Setup

To validate the proposed method a simple network employing $M/M/1$, $D/D/1$ and $M/D/1$ queues were emulated locally. A Client-Server model was used as shown in Fig. 3.3. The updates were sent using TCP/IP from the client to the server. The client transmitted the instantaneous time-stamps of when the updates were generated, i.e., t_i. Upon sending an update, the client sleeps for a random duration, which is obtained from an exponential distribution with mean (inter-arrival rate) (λ). The rate λ was varied between 1 and 8 updates per second. Fig. 3.4 presents the flow-chart of the client's behaviour.

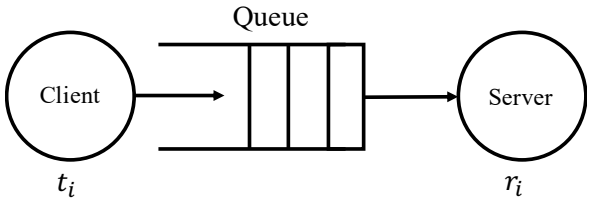

Fig. 3.3 Network System model showing the Client, where the time-stamp of generating updates i, i.e., (t_i). The Server saves the time of receiving the update (r_i).

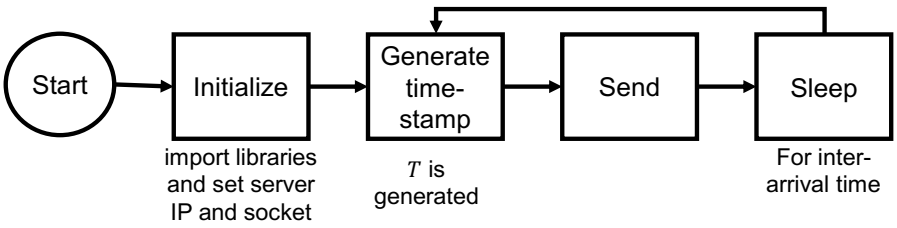

Fig. 3.4 Client's flow chart.

At the server side, the time of receipt r_i is recorded for each packet i along with the transmit time embedded in the update, t_i. The service time, is exponentially distributed

for the $M/M/1$ queue and deterministic for the $D/D/1$ and $M/D/1$ queues. The mean service rate for all the queues is $\mu = 10$ updates per second. To emulate the service time the server will sleep for the service time period (upon receipt of the update), and then wake up and terminate the session with the client and record the time as the time of receipt r_i. After receiving a predefined number of updates, (we used one thousand updates per client in the experiments, hence, increasing the number of updates more than that would have a negligible impact on the results), the experiment terminated and the estimations presented in the previous section were performed.

The experiments were done using Apple MacOS with a 2.2 GHz Intel Core i7 processor and 16 GB 1600 MHz DDR3 Memory. To make sure the clients and server remained tightly synchronised during the experiment, a single clock for all measurements was used. The updates were communicated using a *Python 3.6 socket* module [8]. To obtain the time-stamps, a *Python time* module [9] was used and the readings were obtained by using the object *time.time()*. All the code and the description of the code used is presented in Appendix A.

3.4.5 Results and Discussions

In this section, the performance of the Delay time, Peak Age and average Age is presented. The performance is illustrated by showing the median value for the transmitted updates and the percentage error.

3.4.6 Delay Time Performance

The delay performance, for $M/M/1$, $D/D/1$ and $M/D/1$ queues, are presented in Fig. 3.5, Fig. 3.6 and Fig. 3.7 and compared with the theoretical, given by (3.16), (3.32) and (3.41) respectively. It can be seen that the experimental and theoretical results are in good agreement. In particular, the mean percentage error does not exceed 6%. To better illustrate the accuracy, Table 3.1, presents the percentage error for the different queues. The percentage error is calculated as follows,

$$E = \frac{|\text{Theoretical Value - Median Empirical Value}|}{\text{Theoretical Value}} \times 100\%, \tag{3.46}$$

where $|.|$, represent the absolute value.

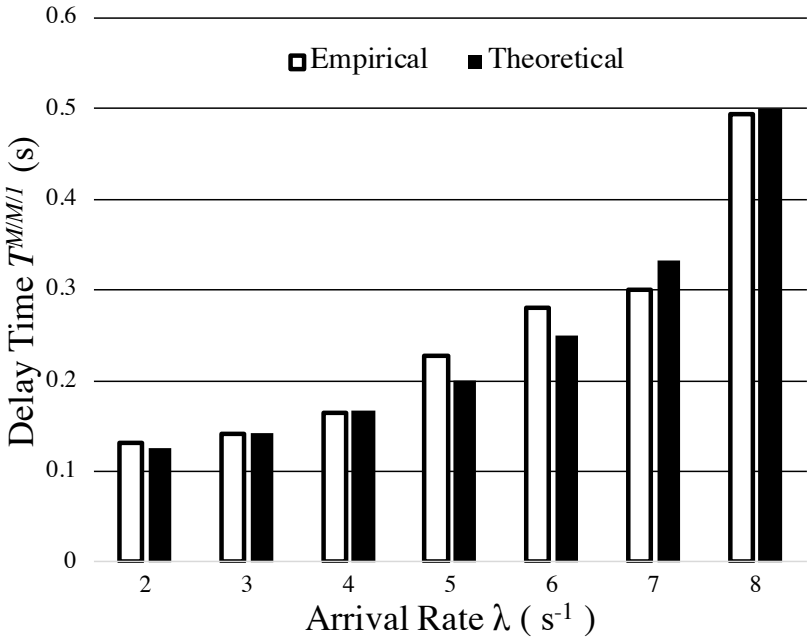

Fig. 3.5 Delay versus Arrival rate for M/M/1 queue calculated theoretically from (3.16) and measured in the experiment using the median of (3.11).

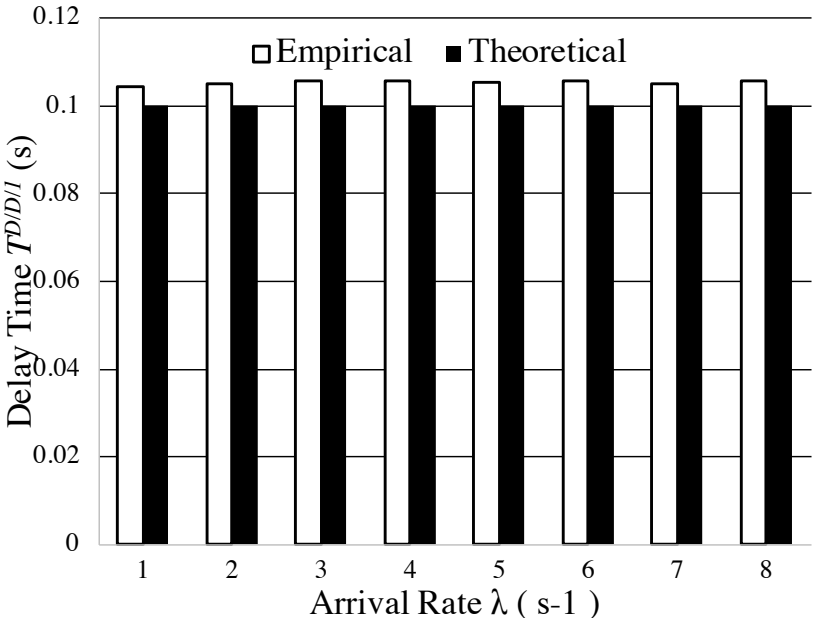

Fig. 3.6 Delay time versus Arrival rate for D/D/1 queue calculated theoretically from (3.32) and measured in the experiment using the median of (3.11).

3.4.7 Peak Age Performance

Next, we move to consider the PA estimator which was compared with its theoretical counterpart given by (3.18) for the $M/M/1$ queue, (3.35) for the $D/D/1$ queue and

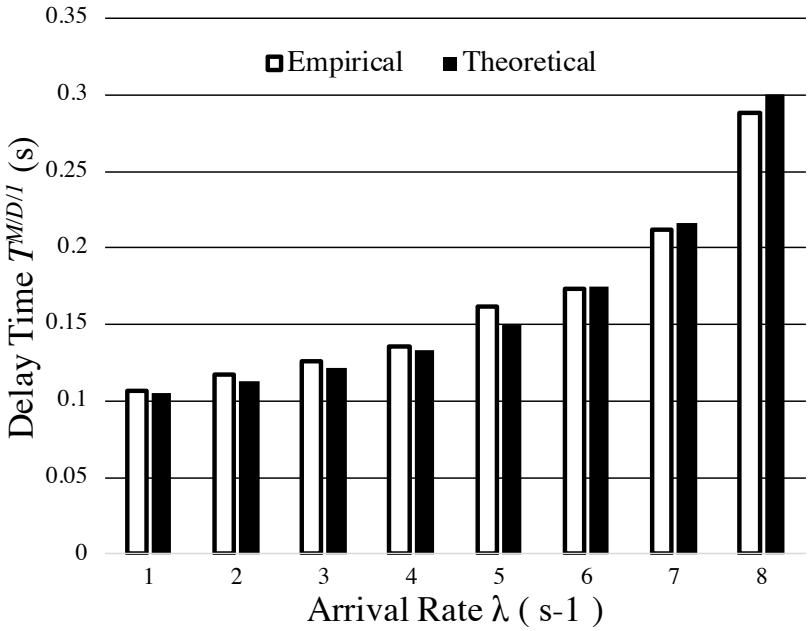

Fig. 3.7 Delay time versus Arrival rate for M/D/1 queue calculated theoretically from (3.41) and measured in the experiment using the median of (3.11).

Table 3.1 Percent errors for $M/M/1$, $D/D/1$ and $M/D/1$ queues, validating the proposed method for evaluating the Delay time.

Inter-arrival rate	Tested Queues		
λ	$M/M/1$	$D/D/1$	$M/D/1$
1	2.71	4.28	0.97
2	5.61	4.92	4.44
3	1.03	5.67	3.90
4	1.55	5.65	1.76
5	13.64	5.50	7.91
6	12.11	5.74	0.72
7	10.06	5.07	2.12
8	1.22	5.70	3.91
Mean	5.99	5.32	3.22

(3.45) for the $M/D/1$ queue. Fig. 3.8, Fig. 3.9 and Fig. 3.10; show the Peak Age performance for the $M/M/1$, $D/D/1$ and $M/D/1$ queues respectively.

The results in Fig. 3.8 show that the estimated PA is in very good agreement with the theoretical values. The percentage error of the PA is presented in Table 3.2, shows that the mean error does not exceed 5%. Thus, we can argue that the experimental model proposed can obtain the PA and delay time for the proposed system model accurately (hence, the error does not exceed 5%).

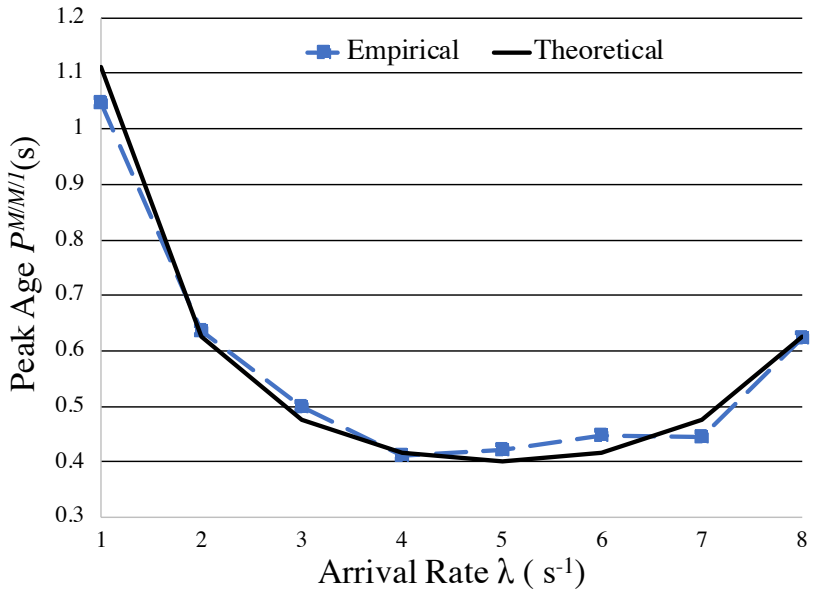

Fig. 3.8 Peak Age versus Arrival rate for $M/M/1$ queue calculated theoretically from (3.18) [3] and obtained experimentally using (3.12).

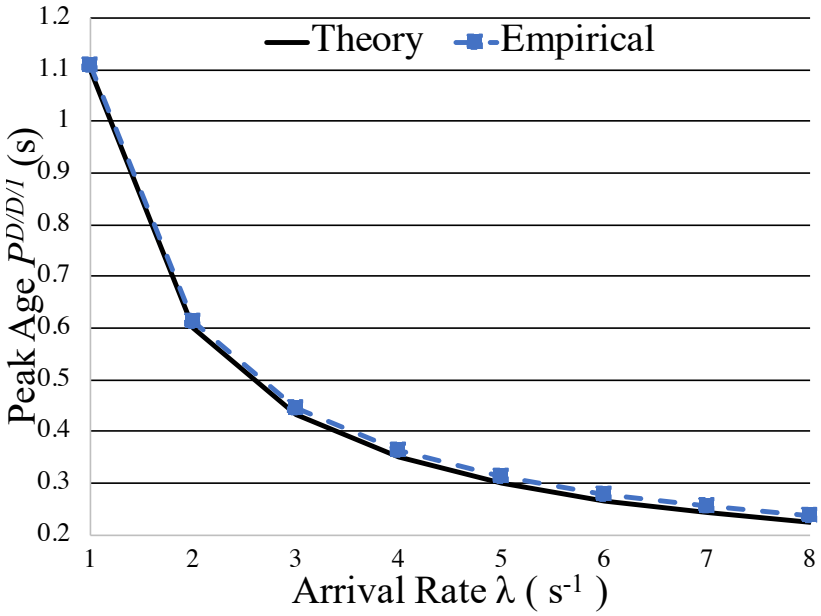

Fig. 3.9 Peak Age versus Arrival rate for D/D/1 queue calculated theoretically from (3.35) and obtained experimentally using (3.12)

.

3.4.8 Average Age of Information Performance

The estimated and theoretical average AoI are shown in Fig. 3.11 for $M/M/1$ queue and Fig. 3.12 for $D/D/1$ queue as a function of the arrival rate. The theoretical values

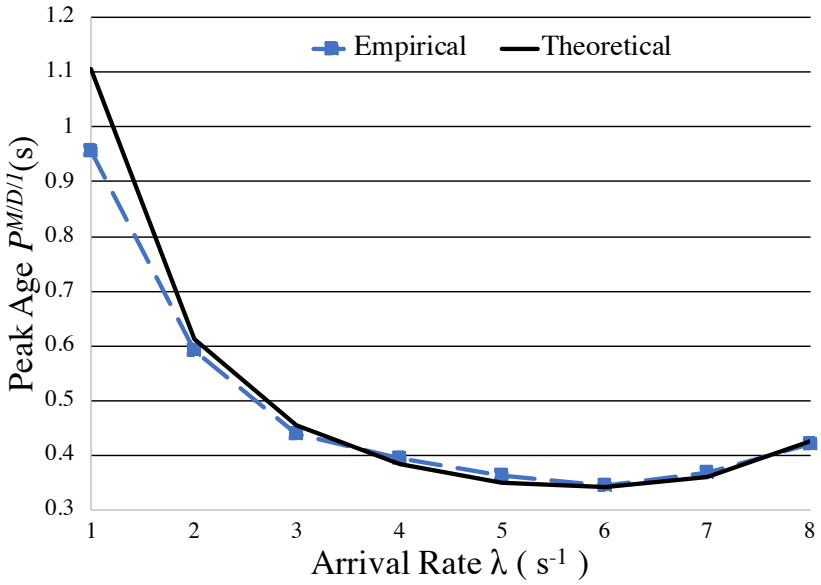

Fig. 3.10 Peak Age versus Arrival rate for M/D/1 queue calculated theoretically from (3.45) and obtained experimentally using (3.12)

.

Table 3.2 Percent errors for $M/M/1$, $D/D/1$ and $M/D/1$ queues, validating the proposed method for evaluating the Peak Age Performance.

Inter-arrival rate	Tested Queues		
λ	$M/M/1$	$D/D/1$	$M/D/1$
1	5.78	0.82	13.55
2	1.89	1.93	3.39
3	5.04	2.62	3.42
4	0.95	3.23	2.65
5	5.58	3.67	3.61
6	7.34	4.13	1.12
7	6.57	4.78	2.05
8	0.16	5.05	1.11
Mean	4.16	3.28	3.86

are obtained from (3.29) for $M/M/1$ queue and (3.38) for $D/D/1$ queue [2]. The results show that the estimated and theoretical results are in agreement. It also observed that the mean percentage error here does not exceed 6% as shown in Table 3.3.

[2]It is worth mentioning that the theoretical Age of Information for $M/D/1$ queue does not have a closed form expression as shown in [7].

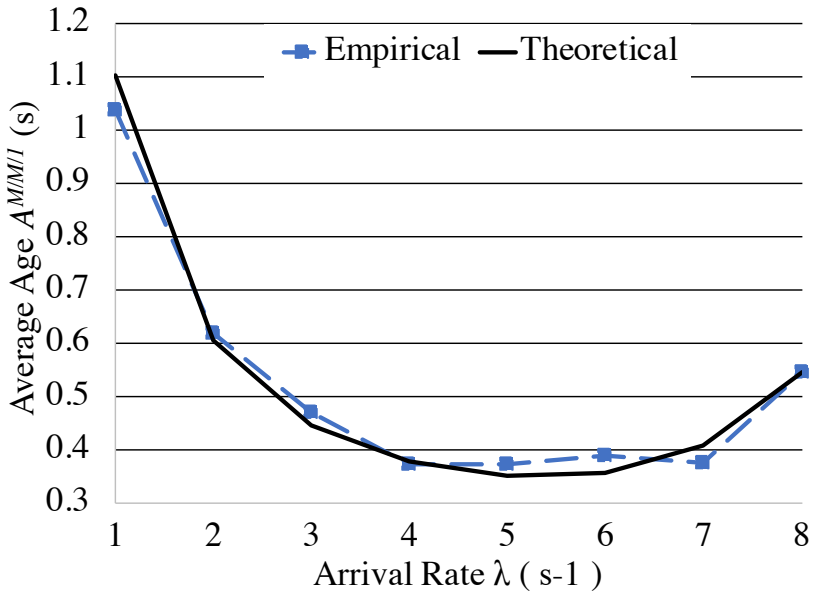

Fig. 3.11 Average Age versus Arrival rate for M/M/1 queue calculated theoretically from (3.11) [7] and obtained experimentally using (3.15).

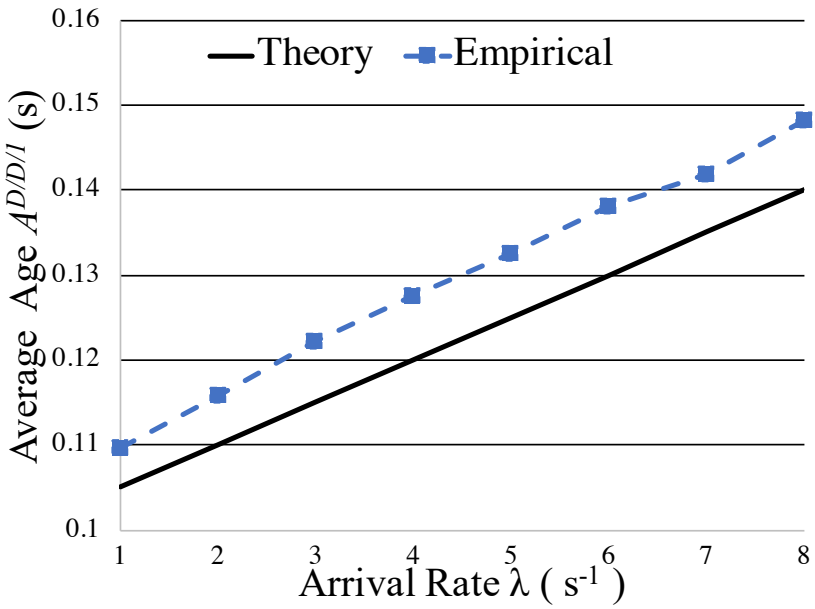

Fig. 3.12 Average Age versus Arrival rate for D/D/1 queue calculated theoretically from (3.38) [7] and obtained experimentally using (3.15).

3.4.9 Statistical Test

To assess the accuracy of the proposed experiments, two statistical tests were conducted. The first test, i.e., *Chi-square test*, was conducted to investigate the relationship between the theoretical results and the experiments. The *Chi-square test* calculates

Table 3.3 Percent errors for $M/M/1$, $D/D/1$ and $M/D/1$ queues, validating the proposed method for evaluating the Average Age Performance.

Inter-arrival rate	Tested Queues	
λ	$M/M/1$	$D/D/1$
1	5.84	4.47
2	1.95	5.26
3	5.38	6.20
4	1.05	6.21
5	6.37	6.01
6	8.59	6.28
7	7.70	5.09
8	0.18	5.89
Mean	4.63	5.68

the probability (p-values) that there is a relationship between the two sets. Initially the null hypothesis was that both the experiments and the theoretical values has a relationship, the alternative hypothesis is that there is no relationship between the two data sets. To evaluate the accuracy of the null hypothesis the *Chi* value (χ^2) is calculated as follows,

$$\chi^2 = \sum \frac{(O-E)^2}{E}, \tag{3.47}$$

where, O is the observed value from the experiments and E is the expected value from the theory. After calculating the *Chi* value, the probability that the null hypothesis is correct is calculated, using the χ distribution [17].

Table 3.4 presents the *Chi-square test* p-values for the delay time, PA, and AoI measurements. As shown in Table 3.4, the difference in the means can be described as not significant. Thus, it can be concluded that statistically their is a significant relationship between the experiments results and the theoretical derivation.

Table 3.4 *Chi-square test p* values for $M/M/1$, $D/D/1$ and $M/D/1$ queues, validating the proposed method for evaluating the Delay, Peak Age and Average Age.

Queue	*P-value*		
	Delay (T)	Peak Age (P)	Age (A)
M/M/1	1	0.999999999	0.999999999
D/D/1	1	1	1
M/D/1	1	0.999999987	-

To test if there a significant statistical difference between the experiments and the theory a *student t-test* was conducted [18]. A *student t-test* or *t-test*, is a statistical

hypothesis test to determine if the means (sum of the values divided by their the number of values) of two sets of data are significantly different from each other. The null hypothesis here is that there are no significant deference between both sets (experiment and theoretical values) and the alternative hypothesis is that there is a significant difference between them. The t value in this test is calculated as follows,

$$t = \frac{\overline{X} - \mu}{\hat{\sigma}/\sqrt{n}}, \tag{3.48}$$

where $\overline{X}$ refers to sample mean value, μ is the population mean. The standard deviation is refereed to as $\hat{\sigma}$ and n is the number of samples.

Table 3.5, presents the p-values (probability) that the null hypothesis is correct, of the Peak Age for the tested queues.

Table 3.5 *t-test* p values for $M/M/1$, $D/D/1$ and $M/D/1$ queues, validating the proposed method for evaluating the Delay, Peak Age and Average Age.

Queue	*P-value*
	Peak Age (P)
M/M/1	0.99
D/D/1	0.94
M/D/1	0.87

From Tables 3.4 and 3.5, it can be concluded that their is no significant difference between the experiments and the theoretical derivation. Also, that the number of readings is sufficient to precisely calculate the results. Hence, this method was used to evaluate the PA throughout this thesis.

3.5 Chapter Conclusions and Forthcoming Work

The AoI is a novel metric suggested to measure the freshness of information. A considerable amount of work has been done to evaluate and optimise AoI and PA theoretically. This chapter, aims to motivate more experimental work on the AoI by making it straightforward to estimate the metrics from experiments. The proposed method was validated on emulated $M/M/1$, $D/D/1$ and $M/D/1$ queues. It was shown that the proposed method could achieve estimates that are very close to the theoretical counterparts. The obvious next step is to test the accuracy of some of the policies proposed in the literature to minimise the AoI and PA ; which will be done in the next

chapter. In particular, the author aimed to investigate the validity of the argument that Zero-Wait policy is optimal for delivering fresh information with a high throughput.

3.6 Appendix A, Python programming language

Python is one of the most used programming languages [13]. It is an interpreted [3] high level programming language [4]. Python can be interrupted on most of the major operating systems such as Windows, Mac OS, Linux and Unix [15]. Python has several useful modules. In this thesis, Python was used in most of the technical chapters. It was very useful in implementing the proposed policies. In particular, the author used the Python *Socket module* and the *Time module in this chapter.*

Socket Module

To transmit information through a computer network Python has several useful tools, such as the *Socket module*. The Python *Socket module* rely on the socket functionally supplied by the operating system. In this thesis, the *Socket module* had been used to communicate the information to the network. In particular, it has been used to transmit updates from the client to the server. In the experiments were the client and the server were local, the client and the server were executed on the same computer.

Time Module

The time stamps sent by the *Socket module*, were generated by the Python *Time module* [16]. The *Time module*, is useful in dealing with time related functions. In particular, *time.time()* function, which returns the time since the epoch as a floating point number. The epoch was inherited from the Unix time and it is 00:00:00 Thursday, 1^{st} January 1970. Both the *Socket* and *Time* modules, were used in the client and the server. To demonstrate how the modules were used a sample code for both the client and the server is presented.

[3] Interpret programming language is a type of programming languages in which the translator would convert the instructions to a machine languages line by line. In other words, the code would be executed one instruction at a time [14].

[4] High level programming languages is a category of programming languages that is easy for us to understand, unlike the Low level programming languages that are closer to machine language than to natural language. [14]

The following present a sample client python code,

```
#           ***  The Client  ***
import socket           # Importing the Socket module.
import time             # Importing the Time module.
host = ‘’               # Server IP address.
port =                  # Server port number.
No=0
inter-arrival_time =1   # inter-arrival time in seconds.
def Main():
     for j in range(1,1000):
     # A loop to transmit 1000 updates.
         s = socket.socket()
         # Define the socket as an object s.
         s.connect((host, port))
         #  Connect to the server using the provided
         # IP address and port number.
         crnt-tim=time.time()
         # Assign the current time to the variable.
         message = str(No)+str(crnt-tim)
         # prepare the message to transmit.
         s.send(message.encode(‘ascii’))
         # Transmit the message to the server.
         time.sleep(inter-arrival_time)
         # Sleep for the inter-arrival time interval.
         NO+=1
         # Increase the message counter.
         s.close()
         # Terminate the connection.

Main()

        # ***     End of the Client code.       ***
```

The following code is for a generic server,

```
#           ***   The SERVER   ***
import socket               # Importing the Socket module.
import time                 # Importing the Time module.
host = ''
# IP address to receive the information from.
port = 6000                 # Port number of the communication.
s = socket.socket()
filename = 'Results.csv'
# Name of the file which will contain the results.
service_time = 1            #Service time in seconds

def Main():
    s.bind((host,port))
    #Define the IP and port address.
    while True:
     s.listen()
     #Listen to any incoming connections.
     c, addr = s.accept()
     # Accept the Client connection.
     T = c.recv(1024).decode('ascii')
     # Receive the message from the Client.
     time.sleep(service_time)
     # Sleep for the duration of the service time.
     R = float(time.time())
     # Record the time of receiving the message.
     Saver(T[0],T[1:],R)
     #Save the message and the time of receiving it.
     c.close()           #Terminate the connection.

def Saver(No,T,R):          #A function to save the message.
 appendFile = open(filename,'a')
 from datetime import datetime
 appendFile.write( str(No) + ',' +str(T) ...
 + ',' + str(R) + ',' + str(datetime.now()))
```

```
 appendFile.write('\n')
 appendFile.close()

if __name__ == '__main__':
 Saver('No','T','R')
 Main()

        # ***     End of the Server code.       ***
```

Chapter References

[1] B. Barakat and K. Arshad, “An adaptive hybrid scheduling algorithm for LTE-Advanced,” in 2015 22nd International Conference on Telecommunications (ICT), 2015, pp. 91-95.

[2] M. Costa, M. Codreanu, and A. Ephremides, “On the Age of Information in Status Update Systems With Packet Management,” IEEE Trans. Inf. Theory, vol. 62, no. 4, pp. 1897-1910, Apr. 2016.

[3] L. Huang and E. Modiano, “Optimizing age-of-information in a multi-class queueing system,” in 2015 IEEE International Symposium on Information Theory (ISIT), 2015, pp. 1681-1685.

[4] E. Najm and R. Nasser, "Age of information: The gamma awakening," 2016 IEEE International Symposium on Information Theory (ISIT), Barcelona, 2016, pp. 2574-2578.

[5] A. Kosta, N. Pappas, and V. Angelakis, “Age of Information: A New Concept, Metric, and Tool,” Found. Trends Netw., vol. 12, no. 3, pp. 162-259, 2017.

[6] C. Kam, S. Kompella, and A. Ephremides, “Experimental evaluation of the age of information via emulation,” in Proceedings - IEEE Military Communications Conference MILCOM, 2015, vol. 2015-Decem, pp. 1070-1075.

[7] S. Kaul, R. Yates, and M. Gruteser, “Real-time status: How often should one update?,” in Proceedings - IEEE INFOCOM, 2012, pp. 2731-2735.

[8] “18.1. socket - Low-level networking interface — Python 3.3.7 documentation.” [Online]. Available: https://docs.python.org/3.3/library/socket.html. [Accessed: 06-Mar-2018].

[9] "16.3. time - Time access and conversions - Python 3.6.4 documentation." [Online]. Available: https://docs.python.org/3/library/time.html. [Accessed: 07-Mar-2018].

[10] D. Bertsekas and R. Gallager, Data Networks. Prentice Hall, 1992.

[11] C. Sönmez, S. Baghaee, A. Ergişi and E. Uysal-Biyikoglu, "Age-of-Information in Practice: Status Age Measured Over TCP/IP Connections Through WiFi, Ethernet and LTE," 2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Batumi, 2018, pp. 1-5.

[12] B. Barakat, S. Keates, I. J. Wassell, and K. Arshad, "Is the Zero-Wait Policy Always Optimum for Information Freshness (Peak Age) or Throughput?," in IEEE Communications Letters, doi:10.1109/LCOMM.2019.2907935, (in press).

[13] Lutz, M., 2001. Programming python. " O'Reilly Media, Inc.".

[14] Bennett, J.M., Prinz, D.G. and Woods, M.L., 1952, September. Interpretative sub-routines. In Proceedings of the 1952 ACM national meeting (Toronto) (pp. 81-87). ACM.

[15] Peters, T., 2010. The zen of python. In Pro Python (pp. 301-302). Apress.

[16] Python Software Foundation 2019, Python 3.37 documentation accessed 06-Mar-2018, https://docs.python.org/3/library/time.html.

[17] Brightman, H.J., 1998. Data Analysis in Plain English: With Microsoft Excel. International Thomson Publishing.

[18] Marshall, E. and Boggis, E., 2016. The statistics tutor's quick guide to commonly used statistical tests. University of Sheffield. Available online: http://www. statstutor. ac. uk/resources/uploaded/tutorsquickguidetostatistics. pdf.

Chapter 4

Examining the Optimality of the Zero-Wait Policy

- **Research Gap:**

  Is the Zero-Wait Policy Always Optimum for Information Freshness (Peak Age) or Throughput?

- **Published paper:**

  - B. Barakat, S. Keates, I. Wassell and K. Arshad, "Is the Zero-Wait Policy Always Optimum for Information Freshness (Peak Age) or Throughput?", in IEEE Communications Letters.

- **Most relevant papers:**

  - Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or Wait: How to Keep Your Data Fresh", IEEE Trans. Inf. Theory, vol. 63, no. 11, pp. 7492–7508, Nov. 2017.
  - B. Barakat, H. Yassine, S. Keates, K. Arshad and I. J. Wassell, "How to Measure the Average and Peak Age of Information in Real Networks?", in IEEE 25^{th} European Wireless (EW) 2019.

"The problems that exist in the world today cannot be solved by the level of thinking that created them"; Albert Einstein.

The Zero-Wait (ZW) policy is widely held to achieve maximum information 'freshness', i.e., to achieve minimum Peak Age (PA) and maximum throughput, for real-time Internet-of-Things applications. In the previous chapter, a method for evaluating the PA was proposed and validated. In this chapter, it was shown through a series of experiments that the ZW policy is not necessarily the optimum policy for freshness nor throughput in all real-world scenarios. Firstly, the effect of delay on the ZW policy was shown on a LAN. Afterwards, the server was located on the Internet, and it was shown that the ZW policy incurred a two-fold PA and throughput performance degradation compared with continuously sending status updates.

4.1 Introduction

One of the main challenges for designing Internet of Things (IoT) applications is how to deliver information in a sufficiently timely manner. Outdated information may interfere with the operation of critical applications, potentially endangering lives, such as in telehealth or autonomous cars. Recently, a new metric was proposed to measure and quantify the information 'freshness', i.e., Age of Information (AoI) [1]. More recently, the Peak Age (PA) metric, which is defined as the mean maximum AoI of a piece of information, was proposed as a more tangible (utilisable) metric [2, 3]. Both metrics look at the information freshness from the destination's perspective and hence are reactive measures. They do not determine a proactive policy for the sending of the information to ensure maximum freshness. This raises the following question: *"How can we ensure that we are delivering the 'freshest' possible information from a time-varying process?"*.

Several policies have been proposed in the literature to achieve minimum PA, i.e., maximum information 'freshness', such as [1]-[5]. One widely used policy is the Zero-Wait (ZW) policy, which minimises the PA by eliminating the waiting (queuing) time, as explained in section 4.2. The ZW policy is often referred to as a logical mechanism for minimising the PA and maximising the number of communicated status updates per time unit, i.e., maximising the throughput [5]. Recently it was shown that the ZW policy is unable to achieve optimum PA performance when the status update service time changes continuously between long and short duration [5]. However,

it can be argued that such scenario only reflects one special case and might not be the best characterisation of the majority of real-time IoT applications. Nevertheless, the ZW policy is still considered as an optimum throughput policy [5]. Furthermore, most of the published work to minimise PA is purely theoretical without providing proof-of-concept. Additionally, no previous work investigated the performance of ZW in a cloud scenario (in which the conventional theoretical models fail to evaluate the PA).

In this chapter, the outcomes of an experimental study are presented to evaluate the PA and inter-arrival time (INT) of the ZW policy. Two scenarios are proposed, tested and evaluated. The first scenario (S1) has both the source (client) and the destination (server) of the status updates located in the same Local Area Network (LAN). In the second scenario (S2), the server is located in the cloud, i.e., hosted by a remote service provider. Firstly, the effect of delay on the ZW policy was investigated by deriving an expression for the ZW PA performance. Afterwards, the threshold delay value in which the ZW fails to deliver the freshest information is presented in section 4.3. The PA and INT performance of the ZW policy are compared with those achieved by the Continuous Updating (CU) policy. The experimental system model is presented in section 4.4.

The PA and INT performance of the ZW policy was significantly different in the two given scenarios, as shown in section 4.5. In S1, the ZW PA performance clearly outperformed the CU policy, as predicted. However, the ZW PA performance in S2 is much worse than in S1. In particular, the CU policy outperformed the ZW in terms of PA and INT by a factor of two in S2. The results of statistical tests are also presented to validate the results. Section 4.3 presents the cases in which the ZW policy fails to achieve optimum PA performance. Finally, conclusions and proposals for future work are presented in section 4.6.

4.2 Peak Age and Zero-Wait Policy

The PA metric is defined as the mean of the maximum elapsed time since the latest status update received, was generated [2]. Let X_n be the period between generating two consecutive status updates, i.e., INT. T_n is the status update delay time (T), i.e., the time between the generation and completion of processing update n, i.e.,

$T_n = \mathbb{E}[1/\mu_n + W_n]$. Hence, P_n is given as [3]

$$P_n = \mathbb{E}[X_n + T_n] = \mathbb{E}\left[\frac{1}{\lambda_n} + \frac{1}{\mu_n} + W_n\right], \forall n, \tag{4.1}$$

where $\mathbb{E}[.]$ is the expectation operator, $1/\mu_n$ is the status service time and W_n is the waiting time. In other words, the PA is equal to the inter-arrival duration plus the delay time.

It can be argued that ZW can minimise PA and X_n by achieving a zero waiting time i.e., $W_n = 0, \forall n$ [3]. In the ZW policy, the clients (e.g., sensors) may generate status updates only when the server is idle (the ZW system model is shown in Fig. 4.1). Hence, the clients must wait for the server to send an Acknowledgement (ACK) for each status update [5]. Hence, X_n depends on the delay time of both the status update and the ACK (i.e., T_n^{ACK}). Fig. 4.2 illustrates the PA and INT of the ZW policy. The ZW model proposed in [3, 6] was based on the assumption that the client would receive the ACK and generate a new status update instantaneously. However, this assumption does not represent many IoT applications accurately and holds true only in some special cases, such as point to point communication.

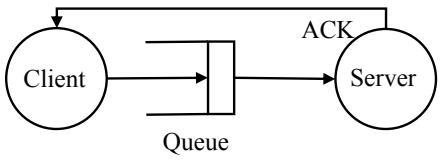

Fig. 4.1 Zero-Wait policy network as in [5], where the client sends the updates through the queue to the server, and the server sends an Acknowledgement (ACK) to the client.

4.3 When is the Zero-Wait Not Optimal?

To understand the limitations of ZW policy, it is important to explicitly identify the scenarios in which the ZW is not optimal. Hence, it is useful to have a closed form expression for the ZW PA, as (4.1) represents the general case of PA. For the ZW, the waiting time is equal to zero, i.e., $W_n = 0, \forall n$. On the other hand, the inter-arrival times depend on the service time for the update and the corresponding ACK delay time (T^{ACK}). In particular, the inter-arrival time for ZW

$$X_n^{ZW} = S_n + T_n^{ACK}, \tag{4.2}$$

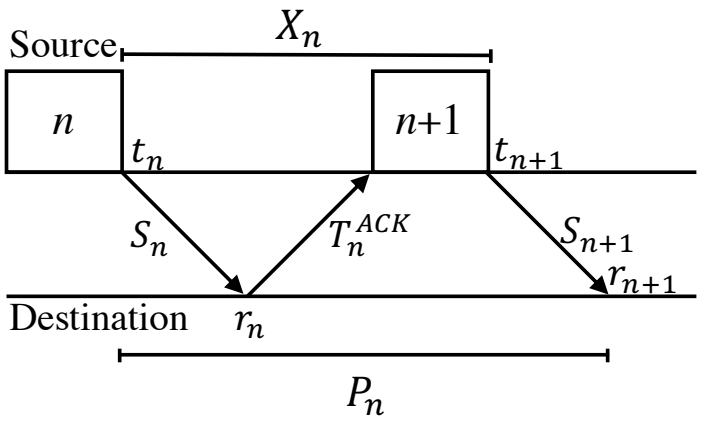

Fig. 4.2 An illustration of Peak Age for the Zero-Wait policy, where t_n is the time at which update n was generated, r_n is the time update n was received, T_n^{ACK} is n ACK delay time, X represents the inter-arrival time and P_n is the Peak Age of update n.

and therefore the Peak Age of Zero-Wait P^{ZW} can be derived as follows

$$P_n = X_n + T_n,$$

by substituting X^{ZW} from (4.2),

$$\begin{aligned} P^{ZW} &= S_n + T_n^{ACK} + T_n = S_n + T_n^{ACK} + S_n, \\ P_n^{ZW} &= 2S_n + T_n^{ACK} = \frac{2}{\mu_n} + T_n^{ACK}. \end{aligned} \tag{4.3}$$

Hence, using (4.1) and (4.3) the ZW would achieve a longer PA than a general status updating policy (P) if,

$$P < P^{ZW} \text{ iff } \mathbb{E}\Big[w + \frac{1}{\mu} + \frac{1}{\lambda}\Big] < \mathbb{E}\Big[\frac{2}{\mu} + T^{ACK}\Big] \tag{4.4}$$

$$\text{iff } \mathbb{E}\Big[w + \frac{1}{\lambda} - \frac{1}{\mu}\Big] < \mathbb{E}\Big[T^{ACK}\Big]. \tag{4.5}$$

Consider an M/M/1 queue where the client updates inter-arrival time follows an exponential inter-arrival time; its waiting time is $w = \frac{\lambda}{\mu(\mu-\lambda)}$. If the waiting time is substituted in (4.5), the condition can be written as,

$$P^{M/M/1} < P^{ZW} \text{ iff } \mathbb{E}\Big[\frac{\rho}{\mu - \lambda} + \frac{1}{\lambda} - \frac{1}{\mu}\Big] < \mathbb{E}\Big[T^{ACK}\Big]. \tag{4.6}$$

For an M/D/1 queue, the $w = \frac{\rho}{2\mu(1-\rho)}$ and the threshold is,

$$P^{M/D/1} < P^{ZW} \text{ iff } \mathbb{E}\Big[\frac{\rho}{2\mu(1-\rho)} + \frac{1}{\lambda} - \frac{1}{\mu}\Big] < \mathbb{E}\Big[T^{ACK}\Big]. \tag{4.7}$$

where ρ, is the server utilisation λ/μ .

From (4.6) and (4.7), the threshold ACK delay time τ (in which the ZW would be not optimal) for an M/M/1 and M/D/1 queue with $\mu = 1$ is shown in Fig. 4.3. In other words, if the ACK delay time exceeds the τ value the ZW would fail to deliver the information as fresh as a general updating policy.

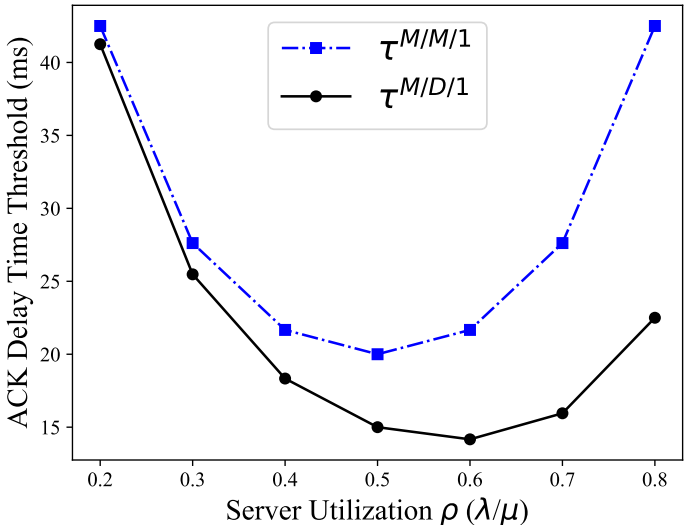

Fig. 4.3 The threshold ACK delay time (τ) in which updating using the PA of M/M/1 and M/D/1 queues with $\mu = 100$ is shorter than ZW.

Moreover, the ZW would fail to deliver the freshest information if the updates were sent through the internet. On the other hand, in several IoT applications the sensors transmit status updates to a server located in the cloud and which is provided by a service provider. Indeed, several IoT service providers rely on Infrastructure as a Service (IaaS) to support the sensors. As an example, Amazon provides Amazon Web Services (AWS), Google offers the Google Cloud Platform and Microsoft provides Microsoft Azure Platform. Thus, for such IoT networks, it is of paramount importance to consider the effect of ACK delay time on the PA performance. However, the condition proposed in (4.5) is only applicable to a closed queue, and it is not applicable to the cloud scenario. Hence, to understand the limitation of the ZW in the cloud it must be evaluated experimentally.

4.4 Experiment Setup

In this chapter, the performance of the ZW policy in terms of PA and throughput is obtained using a Server-Client network topology (introduced in the previous chapter [8]) for the following two scenarios: S1 and S2. In S1, both server and client were placed on the same LAN while in S2 a virtual server was used that was located on an IaaS provider and the client was located at the University of Greenwich, Medway campus.

The status updates were sent from the client to the server. Each update contained the instantaneous time-stamp representing the time of the generation of the update (t). Subsequently, the client only generated a new update and time-stamp after receiving an ACK from the server. A flowchart demonstrating the client function is presented in Fig. 4.4. In S1, a delay was added in the server to investigate the effect of the service time on the PA and throughput performance. The delay followed an exponential distribution with mean range from 0.1 up to 1 updates per second. After the delay period, the server sent an ACK back to the client. In S2, as soon as the server received an update, it sent an ACK back to the client with no added delay to investigate the effect of delays arising from using a cloud-based server rather than the previous LAN scenario. Subsequently, in both scenarios, the server recorded the corresponding instantaneous time-stamp (r) for each update received and logged both the time of the generation and the receipt of the update.

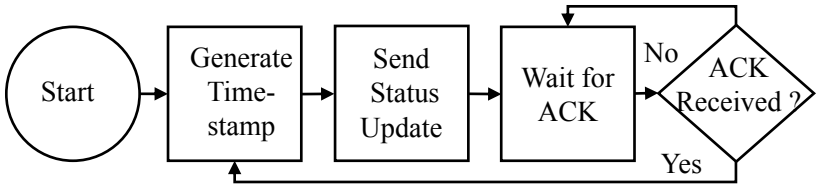

Fig. 4.4 Zero-Wait client flowchart. The client generates a time-stamp, sends it, then waits for an Acknowledgement (ACK) and then it will repeat this procedure.

The PA and INT performance of the ZW policy was compared with the CU policy. The CU policy is a very simple policy in which the client generates a time-stamp and sends it to the server. Instantly after that, it would generate the next time-stamp. Therefore, the inter-arrival time depends on how quickly the client can generate and transmit an update (or on the client's processor clock frequency). In the experiments conducted, in S1 $\lambda \approx \mu$.

The PA and delay time of the n^{th} status update can be calculated as follows:

$$P_n = r_{n+1} - t_n \text{ and } T_n = r_n - t_n. \tag{4.8}$$

The PA vector is defined as $\boldsymbol{\delta} = (P_1, P_2, ..., P_{N-1})$, where N is the total number of status updates sent. Similarly, the inter-arrival time vector is $\boldsymbol{x} = (X_1, X_2, ..., X_{N-1})$. The experimental PA (P) and the value of INT, as calculated in experiments, denoted by X, is equal to the median value of the $\boldsymbol{x}$ vector, can be calculated as

$$P = \tilde{\boldsymbol{\delta}} \quad \text{and} \quad \boldsymbol{X} = \tilde{\boldsymbol{x}} \tag{4.9}$$

where $\tilde{\delta}$ is the median value of the vector δ, and $\tilde{x}$ is the median value of the vector X.

4.5 Zero-Wait Peak Age and Throughput Performance

Initially, the effect of the ACK delay on the ZW PA performance was investigated. Fig. 4.5 presents the PA of ZW and M/M/1 with $\mu = 100$. As proposed in (4.6), the PA of ZW exceed the PA of the M/M/1 queue when the ACK delay approaches the τ value. Also, it is observed that the results validate the expression derived in (4.3).

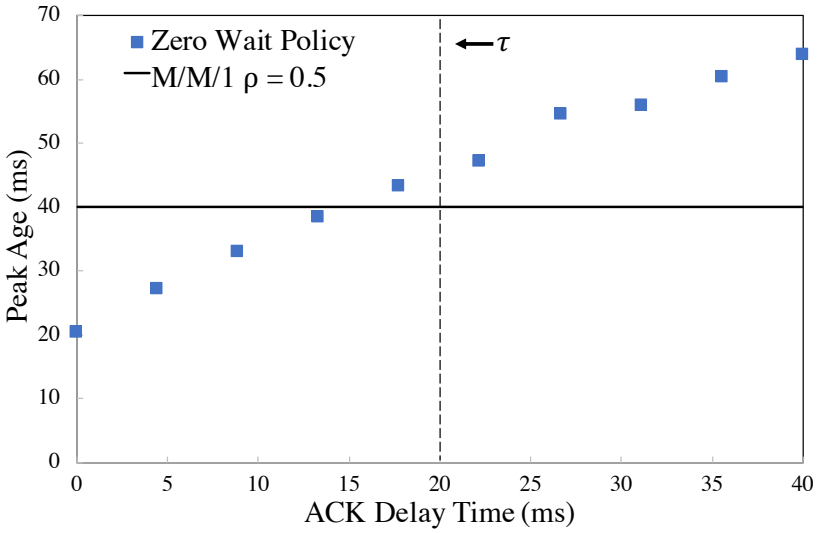

Fig. 4.5 ZW and M/M/1 queue, PA performance (P) versus ACK delay time.

The ZW policy shows significantly different performance for the two given scenarios. Fig. 4.6 and Fig. 4.7 show P and X respectively for 1000 updates in S1 plotted against the mean service time. In S1, the Zero-Wait policy outperformed the CU policy in terms of both P and X, as shown in Fig. 4.6 and Fig. 4.7. In particular, the CU PA is 60 times longer than the ZW policy for all the service times that were tested. This considerable difference was due to the exponentially increasing waiting time of the CU policy. On the other hand, the ZW policy achieved a low waiting time. The median INT X, as a function of mean service time, was similar for both policies as shown in Fig. 4.7.

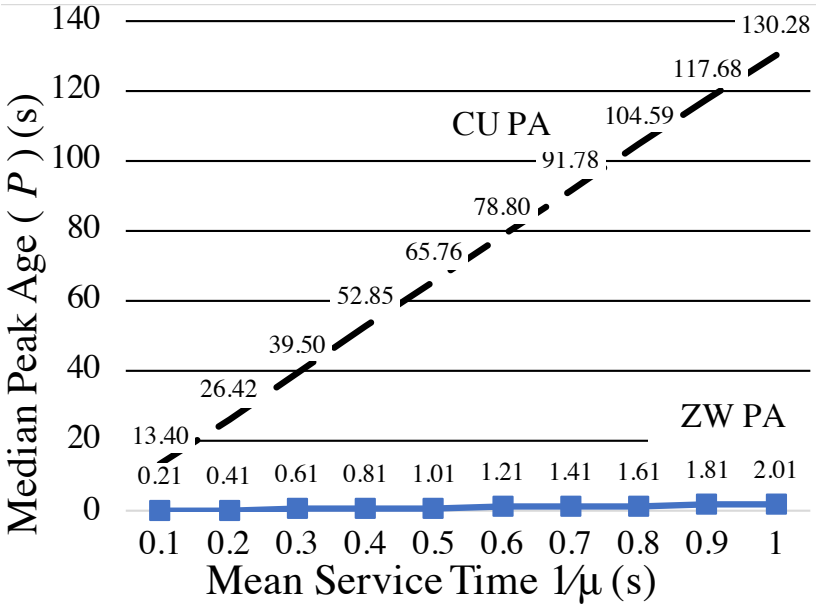

Fig. 4.6 ZW and CU, PA performance (P) versus service time for S1. The PA value for both policies increase with the service time, however, its value for CU is notably higher.

The results shown in Fig. 4.6 and 4.7 are plausible and in agreement with the literature [2, 3]. In particular, the ZW policy achieved low PA and INT times. Thus, in S1, it can be assumed that the ACK service time is short enough to be neglected in comparison with the waiting and service time.

In S2 (when the server is in the cloud), the ACK delay time has a noticeable impact on the PA and INT times. In particular, the argument that the ZW policy is optimum in terms of PA and throughput does not hold true in S2, as shown in Fig. 4.8. Here both the P and X for the ZW policy are approximately twice that of the CU policy. Consequently, in this scenario, it can be argued that CU would outperform ZW in terms of both PA and throughput.

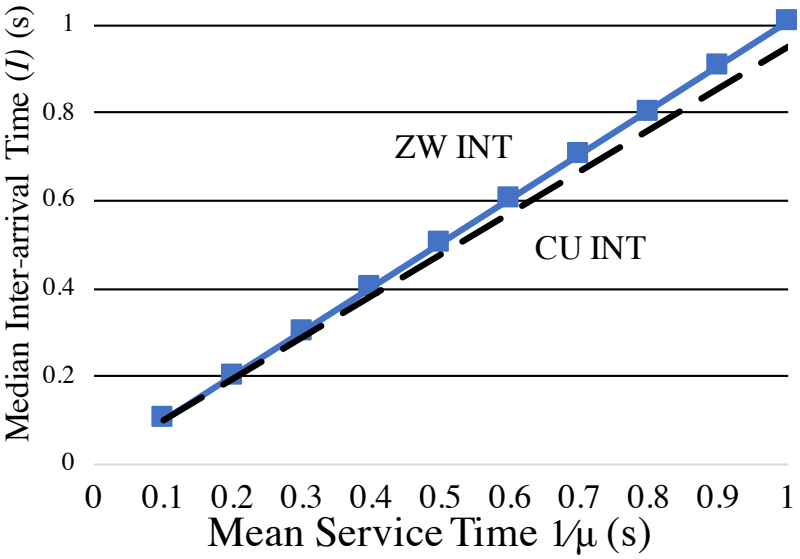

Fig. 4.7 ZW and CU, INT performance (X) versus service time for S1. The inter-arrival time performance of both policies is approximately equal.

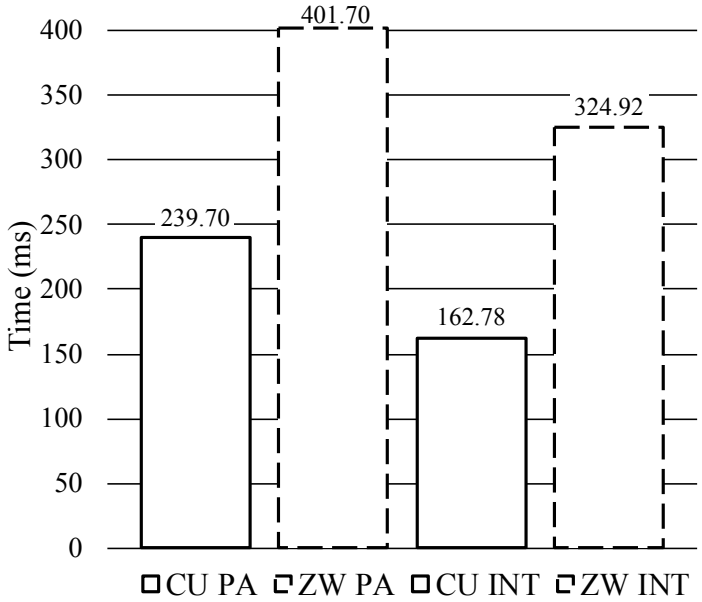

Fig. 4.8 ZW and CU, performance when the server is located in the cloud (S2). The CU policy outperforms the ZW policy for both the PA and the INT.

To validate the experimental results, a statistical analysis (presented in Table 4.1) was performed. An analysis was conducted on the delay time to make sure that the performance was not unduly affected by the fluctuations in the Internet load, which would affect the data propagation time. The analysis results are shown in Table 4.1. It is observed that the difference in the T between the two policies is negligible (less than 0.1 ms). It can also be observed from the measures of the variance of T for both policies that the number of updates communicated was sufficient to mitigate for any Internet load fluctuations. Consequently, it is noted that the fluctuations in the T did not have a major effect on the performance of the policies. Also, the analysis was performed on the PA and INT to show the significant difference in the performance.

Table 4.1 Statistical analysis of the second scenario results for delay time T, Peak Age δ and Inter-arrival time x

Parameter	T		PA (δ)		INT (x)	
Percentile	CU	ZW	CU	ZW	CU	ZW
10% (ms)	76.57	76.64	237.70	398.30	160.80	321.50
25% (ms)	76.62	76.69	238.30	399.80	161.30	322.90
50% (Median) (ms)	76.85	76.75	239.70	401.70	162.80	324.90
75% (ms)	77.23	76.93	241.40	403.30	164.60	326.40
90% (ms)	77.49	77.23	242.50	405.10	165.40	328.30
Mean (ms)	77.02	76.86	240.20	408.60	163.20	331.70
SD	0.002	0.000	0.004	0.127	0.004	0.127

The median PA value, i.e., P, differs significantly between the ZW and CU policies. As presented in Fig. 4.8, the P for the CU and ZW policies are approximately 240 (ms) and 402 (ms) respectively. Consequently, it can be claimed that the CU outperforms ZW in terms of P performance. Furthermore, the median INT duration i.e., X, of the CU is approximately half that of the ZW policy. Thus, it can be seen clearly that CU outperforms ZW by a factor of two for this scenario.

To investigate the difference in the performance of both policies, a Student t-test was performed on the PA and INT results i.e., δ and x, for both policies. As shown in Table 4.2, the variance of the policies differed notably, hence the t-test with unequal variances was used [7]. It was assumed that the hypothesis was that CU outperforms the ZW policy (in both PA and INT) is $\mathcal{H}_1$ hypothesis and the null hypothesis, $\mathcal{H}_0$ is that there is no difference between the policies. The t value and the p value was calculated as shown in chapter 3. Table 4.2 shows the null hypothesis $\mathcal{H}_0$ can be rejected at the 1% level. Consequently, it can be argued that the ZW policy resulted

Table 4.2 T Test: Two-Sample Assuming Unequal Variances

Parameter	PA (δ)		INT (x)	
Policy	CU	ZW	CU	ZW
Mean	0.24	0.41	0.16	0.33
Variance	0.000	0.016	0.000	0.016
t Stat	-41.86		-41.91	
P(T<=t) one-tail	<1%		<1%	
t Critical one-tail	1.646		1.646	

in statistically significantly higher (hence, the high value of the t) PA and INT times. Hence it is asserted that the ZW policy PA and INT performance in the cloud-based server scenario does not outperform the CU policy. Indeed, the CU policy outperforms it significantly and thus, the ZW policy is not optimum in this scenario.

4.6 Chapter Conclusions and Forthcoming Work

In the previous chapter, the author showed that we can use experiments to evaluate the value of AoI and PA. Using the experiments, the author used the experiments to examine the optimality of the Zero-Wait policy. In this chapter, it has been shown that the Zero-Wait policy is not always the optimum policy for either PA or throughput. The results presented contradict the current paradigm that Zero-Wait is always the optimum throughput policy.

The next step is to propose a policy that can deliver fresh information at several real-world scenarios. The ultimate goal is to provide guidance on which policy is the most likely to be optimum for a range of known scenarios. In the case of time-critical applications, such as telehealth applications, such choices could carry significant consequences.

Chapter References

[1] S. Kaul, R. Yates and M. Gruteser, "Real-time status: How often should one update?," 2012 Proceedings IEEE INFOCOM, Orlando, FL, 2012, pp. 2731-2735.

[2] M. Costa, M. Codreanu and A. Ephremides, "On the Age of Information in Status Update Systems With Packet Management," in IEEE Transactions on Information Theory, vol. 62, no. 4, April 2016 , pp. 1897-1910.

[3] L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, 2015, pp. 1681-1685.

[4] S. K. Kaul, R. D. Yates and M. Gruteser, "Status updates through queues," 2012 46th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, 2012, pp. 1-6.

[5] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal and N. B. Shroff, "Update or Wait: How to Keep Your Data Fresh," in IEEE Transactions on Information Theory, vol. 63, no. 11, Nov. 2017, pp. 7492-7508.

[6] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, 2015, pp. 3008-3012.

[7] S. S. Sawilowsky, 'The Probable Difference Between Two Means When σ 1 2 $\neq$ σ 2 2,' vol. 1, no. 2, 2002, pp. 461-472.

[8] B. Barakat, H. Yassine, S. Keates, K. Arshad and I. J. Wassell, "How to Measure the Average and Peak Age of Information in Real Networks?" To appear in IEEE 25^{th} European Wireless (EW) 2019.

Chapter 5

Adaptive Status Arrivals Policy (ASAP) to Minimize Peak Age

- **Research Gap:**
  How can we deliver information as fresh as possible?
- **Published paper:**
  - B. Barakat, S. Keates, I. Wassell and K. Arshad, “Adaptive Status Arrivals Policy (ASAP) Delivering Fresh Information (Minimise Peak Age) in Real World Scenarios,” In International Conference on Human-Computer Interaction.
- **Most relevant papers:**
  - Shreedhar, T., Kaul, S.K. and Yates, R.D., 2018, October. ACP: Age Control Protocol for Minimizing Age of Information over the Internet. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (pp. 699-701). ACM.
  - R. D. Yates, "Age of information in a network of preemptive servers," IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, 2018, pp. 118-123.

"The measure of intelligence is the ability to change." , Albert Einstein.

Real-time systems make their decisions based on information communicated from sensors. Consequently, delivering information in a timely manner is critical to such systems. In the previous chapter it was shown the the Zero-Wait policy is not always optimal, especially when the updates delivered to a cloud server. In this chapter, a policy for delivering fresh information (or minimising the Peak Age of the information) is proposed. The proposed policy, i.e., the Adaptive Status Arrivals Policy (ASAP), adaptively controls the timing between updates to enhance the Peak Age (PA) performance of real-time systems. Firstly, an optimal value for the inter-arrival rate is derived. Afterwards, we implemented the policy in three scenarios and measured the ASAP PA performance. The experiments showed that ASAP is able to approach the theoretical optimal PA performance. Moreover, it can deliver fresh information in scenarios where the server is located in the cloud.

5.1 Introduction

Recently, several applications relying on real-time communications have been investigated such as autonomous cars, tactile internet and telehealth. Delivering the information in a timely manner is critical for these applications. For instance, in a robotic surgery, excessive delays might be life-threatening. Hence, several researchers proposed polices for delivering information with as low latency as possible. However, most of the work done is based on assumptions that might be oversimplifying. These assumptions and the whole latency current paradigm should be questioned.

Consider a vegetable market as an analogy. An insightful question would be what do we mostly care about: (i) the speed of the vegetable carrier, (ii) the time the vegetables took to arrive at the market or (iii) the freshness of the vegetables? Usually, one only care about the freshness of the vegetables. It is obvious that freshness is affected by both the speed of transporting the vegetables and the time it took for them to arrive at the market. Moreover, the freshness is also affected by when the vegetables were grown. This analogy can represent several systems where the information has a window of time in which it is useful and after that window, the information loses its usefulness, for example, an adaptive control system or systems with online machine learning algorithms. In queuing theory/data networks terms, the speed of transmitting information in a network is called the data throughput, the time taken to communicate

a piece of information is the delay or latency and the time to grow the vegetables is the time to generate the package/information. However, until recently no metric has focused on information freshness.

To evaluate the freshness of an update, let us assume that we have a stopwatch that starts counting as soon as an update was generated. The stopwatch stops immediately after receiving the next update at the destination. The time elapsed until the stopwatch is stopped is called Age of the Information (AoI) [1]. The final value the stopwatch shows is the Peak Age of the Information (PA) [2]. The PA is defined as the maximum value of AoI [3]. It can be noticed that the PA consists of the inter-arrival time and the delay time. Unlike the conventional queuing theory delay metric, PA evaluates the freshness from the destination's perspective. In other words, PA reflects the information's freshness, not the time it took to communicate it.

Several policies have been proposed to minimise PA in the literature and can be classified into two main categories [1]. The first category controls the buffer size, which regulates the maximum number of updates waiting in the buffer, e.g., in [3]. The second category attempts to minimise the updates' waiting time and hence reduce the PA. For instance, the Zero-Wait (ZW) policy minimises the waiting time by only permitting the updates' source to communicate a new update after it has ensured that the destination is idle, by waiting for an Acknowledgement (ACK) from the destination after every transmitted update [4]. The state-of-the-art policies reduce PA, however, their performance does not always approach the minimum PA, as shown in section 5.2.

In this chapter, a policy that is able to reach a near-optimal PA performance is proposed. The proposed policy, i.e., Adaptive Status Arrivals Policy (ASAP), adapts the status updating inter-arrivals rate according to the service time to reach the optimal PA performance. The ASAP was tested in three scenarios, as detailed in section 5.2. The first scenario is a single queue with single service rate (μ). The second scenario also has a single queue, however, the mean service time can take four values (this emulates a system where the load on the server changes between four service duration or a wireless channel with adaptive coding and modulation [6, 7]). In the third scenario, the updates were transmitted through the internet to a destination located in the cloud. The ASAP continuously changes the inter-arrivals rate (λ) to reach the optimal value, which is derived in section 5.3. The PA performance of the ASAP presented in section 5.4 shows that it can deliver the information with a freshness that is close to the theoretical optimal freshness. This chapter is concluded in section 5.5.

5.2 Problem Statement and System Model

In this section, the problem this chapter is solving is presented. In the first part, the Peak Age metric is presented and then the tested scenarios are presented in the second part.

5.2.1 Peak Age Metric

The PA metric was proposed as a policy for calculating the peaks of a typical AoI sawtooth profile [1]. The AoI is defined as the time elapsed from the generation of the last successfully received message [1]. In Fig. 5.1, an illustration of AoI evolution is presented. When an update is generated, the AoI is zero and as time passes, the AoI grows linearly until the receipt of the next update. The highest value of AoI, which forms the peak of the AoI pattern, is PA, which represents the highest value of AoI.

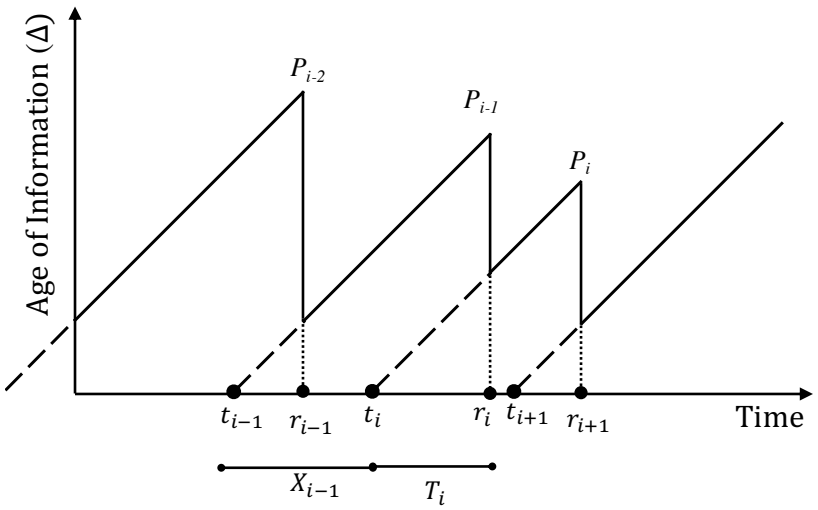

Fig. 5.1 Age of Information illustration, the AoI for update (i) starts when an update is generated (t_i) and keeps counting until the server receives the next update (r_{i+1}). The maximum value of AoI is called Peak Age (P_i), which is equal to inter-arrival time (X) plus the system time (T).

As shown in Fig. 5.1, the PA for a general queue with a single server (G/G/1) can be obtained as follows [3],

$$P = \mathbb{E}[X] + \mathbb{E}[T] = \mathbb{E}[X] + \mathbb{E}[S] + \mathbb{E}[W]. \tag{5.1}$$

where $\mathbb{E}[.]$ is the expectation operator, X is the inter-arrival time, i.e., $1/\lambda$ and T is the delay time, i.e., the the service time (S) plus the queuing time (W).

The service time and the waiting time affect the delay time, and hence, the PA. The service time is either deterministic or follows a random distribution, depending on the time the server takes to process the information. The waiting time depending on the service time, inter-arrival time, queue service discipline, number of servers and the maximum queue length.

5.2.2 Tested Scenarios

In this chapter, the proposed policy is tested on three main scenarios. The first scenario is a single First Come First Serve (FCFS) (also known as First in First Out (FIFO)) queue and the service time follows a single distribution. The second scenario is also a single FCFS queue, however, the mean service time follows several values. The final scenario is that the server is located in the cloud.

Single queue and mean service time scenario

In this scenario, the proposed policy was tested on an M/D/1 queue, i.e., a queue where inter-arrival time distribution followed exponential distributions and the service time follows a deterministic one. This scenario emulates a system where the sensors transmit the information to a decision maker that is located in the same chip, such as a prosthetic limp that has a finger tip sensors and a micro-controller controls the power of the grip. Also, it is usually used to model a communication system where the distance between the source of the information and the destination is short such as Internet of things Machine to Machine communication (M2M) [12, 13].

For an M/M/1 queue, the delay time is [5]:

$$T^{M/M/1} = \frac{1}{\mu} + \frac{\lambda}{\mu(\mu - \lambda)}. \tag{5.2}$$

From (5.1) and (5.2), the PA is:

$$P^{M/M/1} = E[X] + \frac{1}{\mu} + \frac{\lambda}{\mu(\mu - \lambda)} = \frac{1}{\lambda} + \frac{1}{\mu} + \frac{\lambda}{\mu(\mu - \lambda)}. \tag{5.3}$$

Finally, the PA for an M/M/1 queue is equal to [2]

$$\begin{aligned} P^{M/M/1} &= \frac{1}{\mu}\left(\frac{\mu}{\lambda} + 1 + \frac{\lambda}{\mu - \lambda}\right) \\ &= \frac{1}{\mu}\left(1 + \frac{1}{\rho} + \frac{\rho}{1-\rho}\right), \end{aligned} \tag{5.4}$$

where ρ refers to the server utilisation, i.e., $\rho = \lambda/\mu$.

For an M/D/1 queue, the delay time ($T^{M/D/1}$) is calculated using the Pollaczek-Khinchine (P-K) formula [5] as follows,

$$\begin{aligned} T^{M/D/1} &= S + W, \\ &\because S = \frac{1}{\mu} \text{ and } W = \frac{\rho}{2\mu(1-\rho)}, \\ &\therefore T = \frac{1}{\mu} + \frac{\rho}{2\mu(1-\rho)}. \end{aligned} \tag{5.5}$$

From (5.1) and (5.5), the PA of the M/D/1 queue ($P^{M/D/1}$) is,

$$P^{M/D/1} = \frac{1}{\mu}\left(1 + \frac{1}{\rho} + \frac{\rho}{2(1-\rho)}\right). \tag{5.6}$$

Single queue with several mean service time values

In this scenario, the proposed policy was tested on a server with a mean service rate that can take several values. This scenario emulates a server that has several probable data processing speeds e.g. a data communication channel with Adaptive Modulation and Coding (AMC) such as Long Term Evolution (LTE) which has 15 Channel Quality Indicator (CQI) ranges [6, 7]. In this scenario, we have tested the ASAP for a server has four mean service rates as shown in Fig. 5.2.

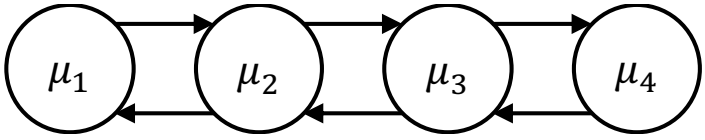

Fig. 5.2 Second Tested Scenario. In this scenario, the server mean service rate can take one of four mean values.

Status updating through the Internet

Several major cloud services offer Infrastructure as a service for Internet of Things (IoT) applications, e.g., Amazon and Google. However, delivering information that is as fresh as possible to the cloud might be challenging [8, 9]. Hence, the models proposed in the literature only consider abstract queues. In this scenario, the source located at University of Greenwich, Medway Campus, will be transmitting information through the internet to a server located in a cloud service as shown in Fig. 5.3. The inter-arrival time follows an exponential distribution with mean $1/\lambda$, where λ is the inter-arrival rate. The server service time mean is deterministic and is equal to $1/\mu$, where μ is the service rate.

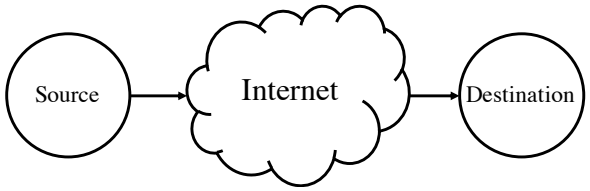

Fig. 5.3 Third tested scenario. In this scenario, the server is located in a cloud services provider.

Delivering fresh information in the third scenario, is more challenging than the previous two scenarios. On the other hand, it might be critical for the next generations of communication networks, since several new applications and systems rely on the remote control of other machines that might be very distant from the controller. For example, the control of the control of a fleet of self-driving vehicles (such as *Cambridge Minicar* [10]). Another example is the control of Base Stations antenna tilting in a cooperative self organising network [11].

5.3 Adaptive Status Arrivals Policy (ASAP)

In the proposed policy, the client adapts its status inter-arrivals rate λ to reach the optimal PA value, as shown in Fig. 5.4. The optimal inter-arrival rate (λ^{opt}) relies on the μ value, and hence it can be calculated using [5],

$$\lambda^{\text{opt}} = \rho^{\text{opt}} \times \mu, \tag{5.7}$$

where ρ^{opt} is the optimal server utilisation value. Now ρ^{opt} depends on the scenario, hence in the next section, ρ^{opt} is derived for the tested scenarios.

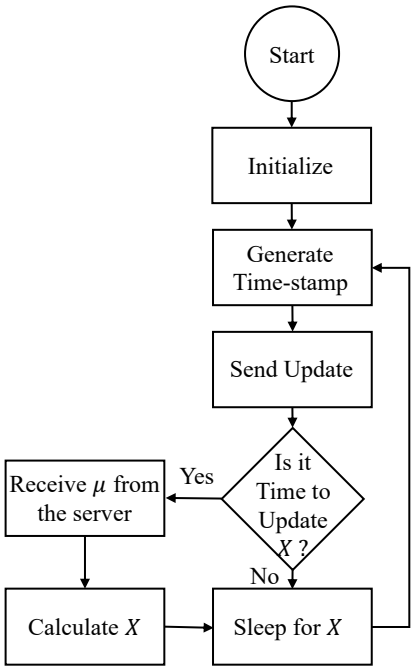

Fig. 5.4 ASAP client flow chart. The Client generates a time-stamp, sends the update and sleeps for the duration of the inter-arrival time (X). When its time to update the (X) it would receive the server service time μ and adapt its X accordingly.

5.3.1 Optimal Server Utilisation for Minimising the Peak Age

An optimisation problem with its objective to minimise the PA with respect to ρ was formulated, as follows:

$$\begin{aligned} P^{\text{opt}} \triangleq \min_{\rho} \quad & P(\rho) \\ \text{subject to:} \quad & \rho < 1 \\ & \lambda \leq \lambda^{\max}. \end{aligned} \tag{5.8}$$

The P^{opt} refers to the optimal PA value, the first constraint is to ensure that the queue is stable, since if ρ approach 1, the delay time T value would equal to ∞ [5]. The second constraint ensures that λ does not exceed the λ^{max}, which is determined by the device capabilities (for instance the sensor clock cycle).

The Optimal Peak Age (OP) for an M/M/1 queue ($OP^{M/M/1}$) can be obtained by solving

$$\frac{d}{d\rho} P^{M/M/1}(\rho) = 0, \tag{5.9}$$

for ρ. Form (5.4), the derivative would equal to

$$\frac{d}{d\rho}\left[\frac{1}{\mu}(1+\frac{1}{\rho}+\frac{\rho}{1-\rho})\right]=0 \tag{5.10}$$

and so,

$$\frac{d}{d\rho}P^{M/M/1}(\rho)=\frac{2\rho-1}{(\rho-1)^2\rho^2}=0. \tag{5.11}$$

From (5.11), the optimal server utilisation for M/M/1 queue is equal to

$$\rho^{opt}=\frac{1}{2}. \tag{5.12}$$

For an M/D/1 queue, the ρ^{opt} can be obtained as follows,

$$\frac{d}{d\rho}P^{M/D/1}(\rho)=0. \tag{5.13}$$

From (5.6) and (5.13), ρ^{opt} can be derived by,

$$\frac{d}{d\rho}\left[\frac{1}{\lambda}+\frac{1}{2\mu}\left(\frac{2-\rho^2}{1-\rho}\right)\right]=0 \tag{5.14}$$

$$\frac{d}{d\rho}P^{M/D/1}(\rho)=\frac{1}{2\rho-2}-\frac{2(\rho-2)}{(2\rho-2)^2}-\frac{1}{\rho^2}=0. \tag{5.15}$$

Solving (5.15) for ρ, the optimal server utilisation for the M/D/1 queue is,

$$\rho^{opt}\approx 0.5858. \tag{5.16}$$

The PA value for M/M/1 and M/D/1 queues are plotted in Fig. 5.5. It can observed that optimal ρ derived in (5.16) achieves the minimum PA value.

The derived optimal values for ρ are only applicable to a single queue. However, the PA in the third scenario (where the server is located on the cloud) the internet load would affect the inter-arrival time, and hence it must be considered. The inter-arrival time in this scenario (X), as observed from the server, is

$$X(n)=X^*+\epsilon(n), \tag{5.17}$$

where X^* refers to the inter-arrival time and $\epsilon(n)$ is a random value representing the time it takes the information (n) to be transmitted through the internet.

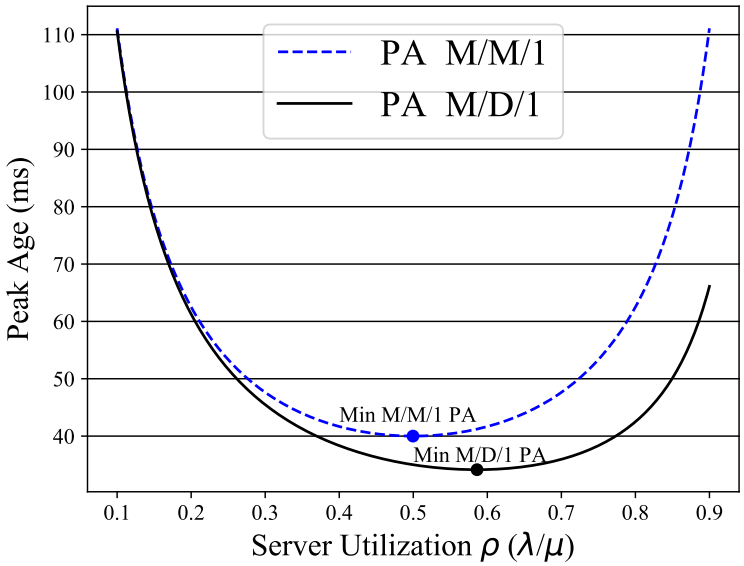

Fig. 5.5 Peak Age versus server utilisation, showing minimum value for M/M/1 and M/D/1 queue with $\mu = 100$. The optimal Server Utilisation value for the M/M/1 queue is equal to 0.5 and for M/D/1 queues is approximately equal to 0.5858.

From (5.17), the optimal inter-arrival rate in the third scenario is

$$\lambda(n)^* = \frac{1}{X^*} = \frac{1}{X(n) - \epsilon(n)}. \tag{5.18}$$

Hence,

$$\rho(n)^* = \frac{\lambda(n)^*}{\mu} = \frac{1}{(X(n) - \epsilon(n)) \times \mu}. \tag{5.19}$$

5.3.2 Experimental System Model

To evaluate the PA performance, we implemented an experimental system, consisting of a Client and Server as shown in Fig. 5.6. The Client sends status updates to the server; it would then sleep for the duration of the inter-arrival time, as shown in Fig. 5.7. In the experiment, each update consists of the instantaneous time-stamp (t_n). The server records time it received the updates (r_n). The updates were sent using TCP/IP protocol.

The PA in the experiment was calculated by using the logged time-stamps. As shown in Fig. 5.8, the PA of update n can be calculated as follows,

$$P_n = r_{n+1} - t_n. \tag{5.20}$$

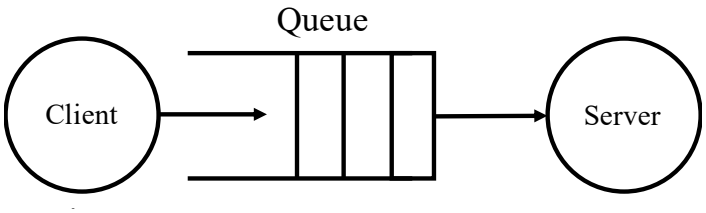

Fig. 5.6 Client-Server network model. The updates are generated in the Client and sent to the Server. The time of generating the update is t_n and the time of receiving the update is r_n.

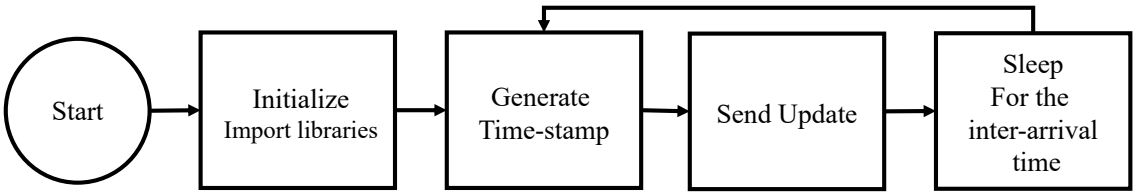

Fig. 5.7 Client flow chart. The Client initially import the socket libraries, then generate the instant time-stamp using the the Time module. After sending the update to the Server it sleeps for the inter-arrival duration.

The experimental PA was obtained by taken the median value of the PA of all the updates sent

$$P = \tilde{P}_{(1,2,\ldots,N)}, \tag{5.21}$$

where $\tilde{P}$ refers to the median value and N refers to the total number of transmitted updates.

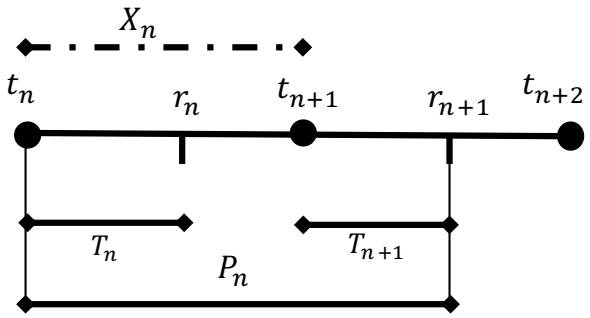

Fig. 5.8 Peak Age illustration. For the experiment the Peak Age value (P) is equal to the inter-arrival time (X) plus the update system time (T). Consequently, in the experiment, the Peak Age value of update n can be obtained by subtracting the time of generating the update t_n from the time of receiving the next update (r_{n+1}).

5.4 ASAP Peak Age Performance

The ASAP PA performance is presented in the three scenarios. In Fig. (5.9) the time series PA performance for an M/D/1 for the first scenario is shown. It can be observed that the ASAP PA performance varies with time as it keeps changing its inter-arrival time, i.e., λ, according to the server mean service time, μ. In our experiments, the server sends its service rate after receiving 100 updates and the sent value is the median service rate ($\tilde{\mu}$). Fig. 5.10 presents the mean PA value of the updates shown in Fig. 5.9. It is observed that the PA approaches the theoretical optimal PA value.

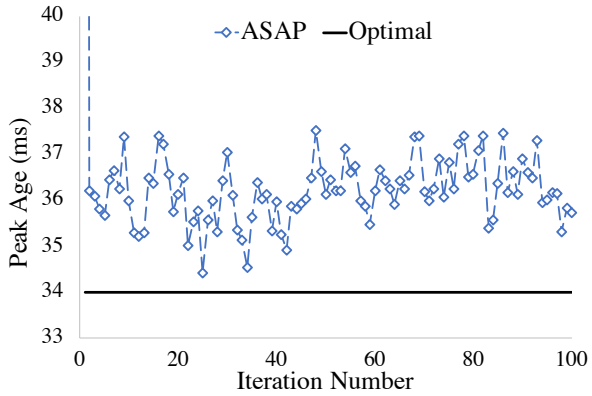

Fig. 5.9 ASAP Peak Age time-series performance. Each point in the ASAP represents the median value of 100 values. The presented values for the Optimal, represent the theoretical value for PA at the used service time. The Peak Age performance of ASAP policy continually changes, hence the service time is random.

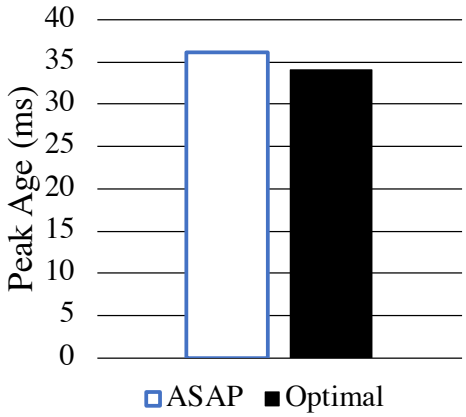

Fig. 5.10 ASAP Mean Peak Age performance. The values presented are the mean value of the results presented in Fig. 5.9.

The ASAP PA performance for the second scenario is presented in Fig. 5.11, where the service rate can take four possible mean values, the ASAP managed to deliver the

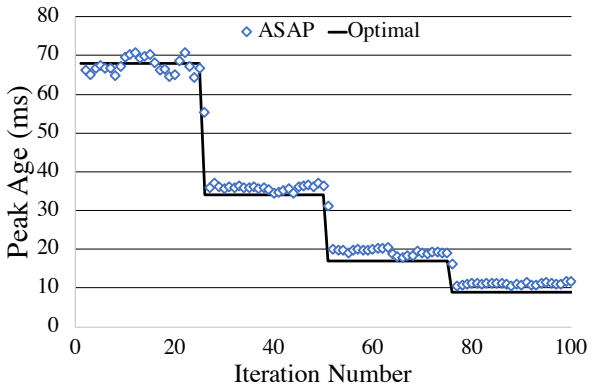

Fig. 5.11 ASAP Peak Age performance in the second scenario. Each point in the ASAP represents the median value of 1000 values. The presented values for the Optimal, represent the theoretical value for PA at the used service time. The Peak Age performance of ASAP policy continually changes, hence the service time is random. ASAP can outperform the Optimal value if the majority of the updates service time is less than the mean service time. Consequently, in the experiment the ASAP can outperform the Optimal value.

updates with almost optimal freshness. It can be noted that the achieved performance is very close to the theoretical optimal performance. It is worth mentioning that the achieved instantaneous PA performance might outperform the theoretical optimal PA, as the optimum represent the mean (average) performance.

In the third scenario, the ASAP managed to handle the internet load fluctuations as shown in Fig. 5.12. It is worth mentioning that the presented optimal performance in Fig. 5.12 represents the optimal PA for a single queue with the service time equal to the service time of the server plus the approximated value internet delay, hence the internet delays are random and hard to predict.

Changing the inter-arrival rate makes ASAP a dynamic policy that is able to change its sampling rate to best fit the server. This feature can be critical in real-world applications where the server might have several background processes running on it. Using ASAP, instead of the client impairing an extra load on the server, it can reduce its transmissions but maintain a near optimal freshness performance.

5.5 Chapter Conclusions and Forthcoming Work

In chapter3, a method for evaluating the Peak Age from experiments was presented; then in chapter 4, the experiments showed that he Zero-Wait policy is not always optimal. In this chapter, the ASAP policy was proposed for minimising the Peak Age of Information. The policy regulates the inter-arrival time of status updates to deliver

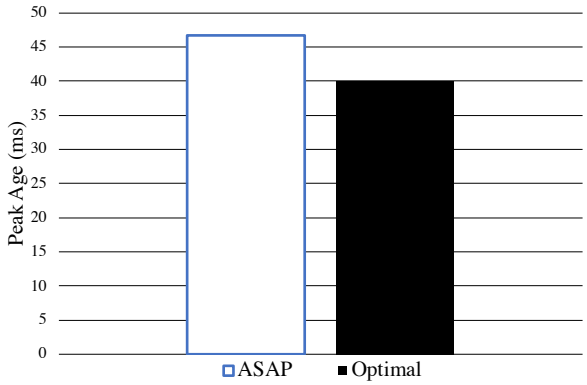

Fig. 5.12 ASAP Peak Age performance in the third scenario. Each bar in the ASAP represents the median value of 1000 values. The presented values for the Optimal, represent the theoretical value for PA at the used service time for a single queue.

fresh information. The performance was measured by conducting experiments on three scenarios. The ASAP Peak Age performance in the tested scenarios approaches the optimal value. Moreover, it can adapt to the server load and the varying load of the internet.

It was shown that to minimise the Peak Age value, the inter-arrival time was controlled. However, for several applications the inter-arrival time (which represents the throughput) is more critical to the operation of the application. Hence, in the next chapter, a policy that would deliver fresh information with high throughput is proposed.

Chapter References

[1] S. Kaul, R. Yates, M. Gruteser, "Real-time status: How often should one update?", Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM), pp. 2731-2735, Mar. 2012.

[2] L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, 2015, pp. 1681-1685.

[3] M. Costa, M. Codreanu and A. Ephremides, "On the Age of Information in Status Update Systems With Packet Management," in IEEE Transactions on Information Theory, vol. 62, no. 4, pp. 1897-1910, April 2016.doi: 10.1109/TIT.2016.2533395

[4] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal and N. B. Shroff, "Update or Wait: How to Keep Your Data Fresh," in IEEE Transactions on Information Theory, vol. 63, no. 11, pp. 7492-7508, Nov. 2017.

[5] D. Bertsekas and R. Gallager, Data Networks. Englewood Cliffs, NJ: Prentice-Hall, 1987

[6] B. Barakat and K. Arshad, "An adaptive hybrid scheduling algorithm for LTE-Advanced," 2015 22nd International Conference on Telecommunications (ICT), Sydney, NSW, 2015, pp. 91-95.

[7] S. O. Aramide, B. Barakat, Y. Wang, S. Keates and K. Arshad, "Generalized proportional fair (GPF) scheduler for LTE-A," 2017 9th Computer Science and Electronic Engineering (CEEC), Colchester, 2017, pp. 128-132.

[8] Shreedhar, T., Kaul, S.K. and Yates, R.D., 2018, October. ACP: Age Control Protocol for Minimizing Age of Information over the Internet. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (pp. 699-701). ACM.

[9] Yates, R.D., 2018, April. Age of information in a network of preemptive servers. In IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 118-123). IEEE.

[10] Prorok, A., Hyldmar, N., & He, Y. A Fleet of Miniature Cars for Experiments in Cooperative Driving. IEEE International Conference on Robotics and Automation.

[11] M. Sharsheer, B. Barakat and K. Arshad, "Coverage and capacity self-optimisation in LTE-Advanced using active antenna systems," 2016 IEEE Wireless Communications and Networking Conference, Doha, 2016, pp. 1-5.

[12] B. Barakat, S. Keates, K. Arshad and I. J. Wassell, "Deriving Machine to Machine (M2M) Traffic Model from Communication Model," 2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT), Amman, 2018, pp. 1-5.

[13] B. Barakat and K. Arshad, "Energy efficient scheduling in LTE-advanced for Machine Type Communication," 2015 International Conference and Workshop on Computing and Communication (IEMCON), Vancouver, BC, 2015, pp. 1-5.

Chapter 6

Clustered Acknowledgement Policy (CAP) for Fresh and Fast Status Updating

- **Research Gap:**
  How can we deliver information as fresh as possible without compromising the inter-arrival rate?

- **Published paper:**

  – B. Barakat, S. Keates, I. Wassell and K. Arshad, 'Three Orders of Magnitude Increase in Fresh Status Updates Throughput using Clustered Acknowledgement Policy (CAP)' poster presented at IEEE International Conference on Computer Communications INFOCOMM 2019.

- **Most relevant papers:**

  – Li, B., Li, R. and Eryilmaz, A., 2015. Throughput-optimal scheduling design with regular service guarantees in wireless networks. IEEE/ACM Transactions on Networking (ToN), 23(5), pp.1542-1552.

  – Kadota, I., Sinha, A. and Modiano, E., 2018, April. Optimizing age of information in wireless networks with throughput constraints. In IEEE INFOCOM 2018-IEEE Conference on Computer Communications (pp. 1844-1852). IEEE.

"I think it's important to reason from first principles rather than by analogy. The normal way we conduct our lives is we reason by analogy. [With analogy] we are doing this because it's like something else that was done, or it is like what other people are doing. [With first principles] you boil things down to the most fundamental truths... and then reason up from there"; Elon Musk, founder of SpaceX, Tesla, Neuralink, The Boring Company and OpenAI.

Recently, several policies have been proposed to deliver fresh information. However, the current understanding is that to deliver fresh information, the status updates throughput must be tamed. In particular, as shown in the previous chapter the ASAP can deliver fresh information by controlling the inter-arrival time. In this chapter, the author presents the Clustered Acknowledgement Policy (CAP) that can deliver fresh information while having a 'high' throughput. Experiments using two service rate distributions showed that the CAP delivers the information with a rate more than 5000 times than theoretically optimal policies.

6.1 Introduction

Designing a system for sensing a rapidly varying process is challenging. However, for several systems, delivering the sensor readings in a timely manner is critical; for example, the readings of medical devices or the location of a smart drone. Hence, it is common to take the timeliness of the information into consideration when designing such systems.

One possible approach for designing time-critical systems is to minimise the amount of data transmitted from the sensors. This can be achieved by compression of the data; however, this approach adds latency and also consumes computing power. Another possible approach is to reduce the frequency of the sensor readings or the status update throughput, which will worsen the time resolution of the readings.

Several metrics have been proposed to evaluate the timeliness of the information [1, 4–7]. The average Peak of the AoI (PA) was proposed as a metric for quantifying the freshness of the information. PA is defined as the time elapsed from the generation of a status update until the time of receiving the next update [8]. In other words, PA evaluates the information freshness from the destination perspective. The perspective from which the metric is observed is the main difference between the PA and other metrics, such as waiting time.

To deliver information that as fresh as possible requires increasing the time between the generation of the information (or the throughput) and hence its resolution [2, 3]. In most of the queues evaluated, it was shown that there is an optimal value for PA, such that even if the time between generating the information is shortened, the freshness of the delivered information decreases. In other words, after the optimal PA value, the faster the system generates updates, the lower is the freshness [1, 9, 10].

In this chapter, a policy that can deliver fresh information without compromising the frequency of generating readings (or updating throughput) is proposed. The Clustered Acknowledgement Policy (CAP) delivers the information in two modes; the first mode delivers a 'high' status throughput, while the second mode ensures the stability of the system.

The CAP PA and inter-arrival time (X) performances were compared to those for the Zero-Wait policy (ZW) and those for the theoretical optimum for M/M/1 and M/D/1 queues. It was shown that the CAP inter-arrival time is much shorter. In particular, for deterministic service rate with $\mu = 1$, the CAP inter-arrival rate is more than 5000 times higher than ZW and about 10000 times higher than optimal PA M/D/1 queue. For service rate following exponential distribution with $\mu = 1$, the CAP inter-arrival rate is more than 4000 times higher than ZW and 12000 times higher than optimal M/M/1 queue.

This chapter is organised as follows: section 6.2 states the problem this chapter solves. Section 6.3, describes the CAP operation and section 6.4 presents CAP performance. The chapter is concluded in section 6.5.

6.2 Problem Statement

The problem this chapter is addressing is *"How can we deliver fresh information and maintain a 'high' updating throughput without compromising the system stability?"*. The term throughput represents the frequency of transmitting a status update, that contains a reading of a time-varying process; in other words, the throughput represents the inter-arrival rate. We are aiming to have the inter-arrival time as short as possible. This requirement is beneficial for monitoring a fast varying process. For instance, in Fig. 6.1, for the sketched duration of the inter-arrival time X, some information between the two samples n and $n + 1$ will be irrecoverable, because the process of interest is rapidly varying. Hence, it is necessary to minimise the inter-arrival time. We are assuming that the readings will be sent to the destination as soon as they have

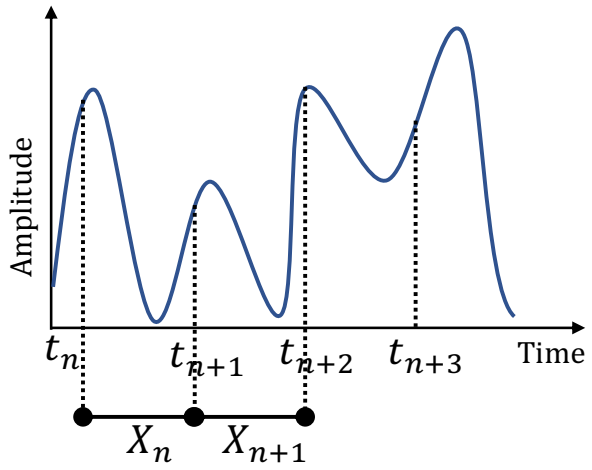

Fig. 6.1 Time varying signal, showing the times of generating updates sampling the amplitude at the epochs n and $n+1$. The time interval between two readings n and $n+1$ is X_n. In this chapter, X_n is used as a representation of the reading throughput. It is observed that some variation in the process of interest will be lost, because of the long duration of the inter-arrival time in this example.

been read. Hence, the time between reading is presented as inter-arrival time (X), and the rate is represented by λ.

The PA value represents information freshness. The PA represents the Peak value of the Age pattern, as shown in Fig. 6.2. The PA for the ith update, i.e., P_i, is the maximum value the Age of Information attains over this interval; which is the Age value in the instance when the next update was received. The PA value can be calculated by taking the expectation value $\mathbb{E}[.]$ of inter-arrival time (X) and the delay time (T), as follows [8]:

$$P = \mathbb{E}[X + T]. \tag{6.1}$$

The ith update inter-arrival time (X_i) is the time between the generation of updates i and $i+1$, can be calculated by

$$X_i = t_{i+1} - t_i, \tag{6.2}$$

where t_i is the time of generating update i.

The inter-arrival time/rate affects the PA and the delay time. In particular, increasing the inter-arrival time X would increase the PA directly, but it also decreases the delay time. The delay time T is defined as the time elapsed from generating an update until receiving an update at the destination. In other words, the delay time is the waiting time W plus the service time S, the waiting time represents the time

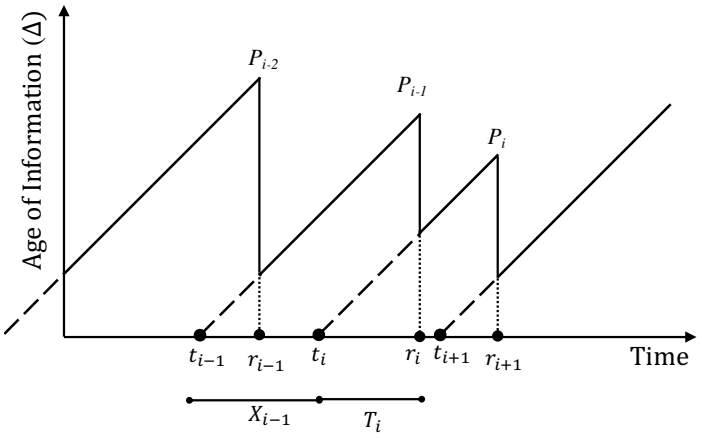

Fig. 6.2 Peak Age of Information illustration. t_i represent the time of generating the reading as shown in Fig. 6.1, r_i is the time of receiving the ith update.

an update spend in a queue while the service time is the time spent transmitting the update. For update i the delay time is

$$T_i = W_i + S_i. \tag{6.3}$$

The inter-arrival time (X) affects the delay time (T) by affecting the waiting time (W). For instance, if X is 'short', the number of updates generated in a given time interval will be 'large'; hence, the number of updates that waiting in the queue will be 'large'. On the other hand, if X is 'long' the number of updates waiting in the queue would be 'small', but the PA will be 'long' as shown in (6.3) and (6.1). Hence, as shown in the literature, the optimum X depends on the queue [8]. For example, if we consider the M/M/1 queue the delay time is [11]

$$\begin{aligned}\mathbb{E}[T^{M/M/1}] &= \mathbb{E}[W + S] \\ &= \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu},\end{aligned} \tag{6.4}$$

where μ represents the service rate. The PA for the M/M/1 queue can be calculated by [12]

$$\begin{aligned}P^{M/M/1} &= \mathbb{E}[X + T] \\ &= \frac{1}{\mu}\left(1 + \frac{1}{\rho} + \frac{\rho}{1-\rho}\right),\end{aligned} \tag{6.5}$$

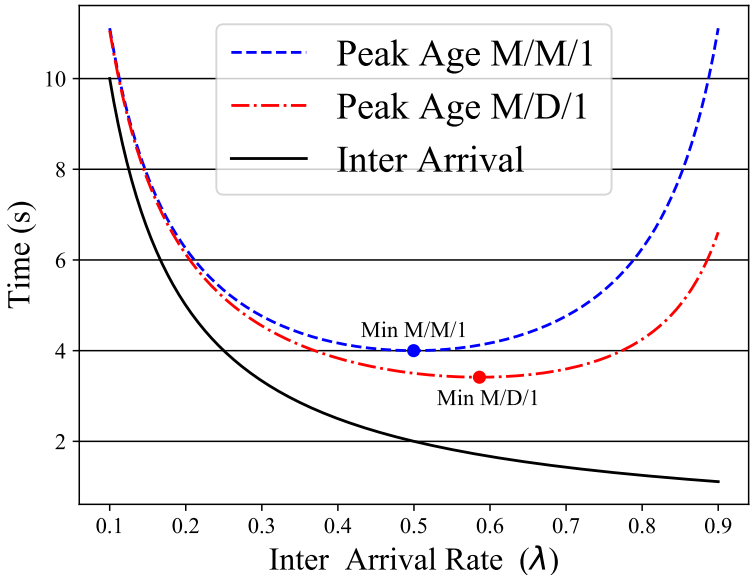

Fig. 6.3 Peak Age and Inter-arrival Time for M/M/1 and M/D/1 queues with $\mu = 1$

where $\rho = \lambda/\mu$ is the link utilisation.

For an M/D/1 queue, the delay time can be calculated using the Pollaczek–Khinchine formula [11, 10]

$$\mathbb{E}[T^{M/D/1}] = \frac{1}{\mu} + \frac{\rho}{2\mu(1-\rho)}. \tag{6.6}$$

Also, the PA is

$$\mathbb{E}[P^{M/D/1}] = \frac{1}{\mu}\left(1 + \frac{1}{\rho} + \frac{\rho}{1-\rho}\right). \tag{6.7}$$

Fig. 6.3 presents the PA and the Inter-Arrivals time for the M/M/1 and M/D/1 queues. As shown in Fig. 6.3, the optimal PA for the M/M/1 queue is at $\lambda = 0.5$, which is when $\rho = 0.5$ [10]. Hence, decreasing the inter-arrival time (increasing the inter-arrival rate) after the optimal point will increase the PA value. So using current approaches it is not possible to deliver fresh information using high inter-arrival rate. To summarise, one can either get fresh information or a high status updating throughput.

One possible solution for this problem is the Zero-Wait policy (ZW) which aims to deliver the updates as fresh as possible by eliminating the waiting time. In the ZW,

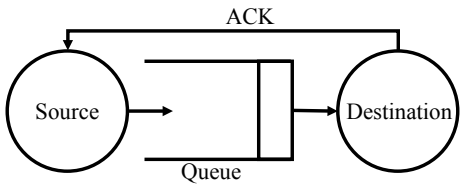

Fig. 6.4 Zero-Wait policy, in which the destination (server) sends an ACK after processing an update [13].

the destination of the updates send an Acknowledgement (ACK) [1] to the source of the updates as soon as it finishes serving an update as shown in Fig. 6.4 [13].

Although the ZW policy can achieve a notable decrease in the PA if compared with the conventional queues, it also affects the value of the inter-arrival time (since the source cannot generate an update until it receives an ACK for the previous update [13]).

6.3 Clustered ACK Policy (CAP)

CAP aims to deliver fresh information with 'high' throughput. However, as shown in the previous section, there is a trade-off between the PA and inter-arrival rate. The CAP manages to deliver fresh and high throughput updates by operating in two modes. The first mode, i.e., Continuous Updating (CU), aims to maximise the status updating throughput [13]. The CU mode transmits the updates as high throughput as possible. After transmitting N updates, the transmitter would operate in the second mode, i.e., Stabilising mode (ST). The ST mode forces the transmitter to wait until it receives an Acknowledgement (ACK) from the destination. Fig. 6.5 shows the flow chart of the CAP updating method. The CU mode ensures that the information is as high throughput as possible. In this mode, the inter-arrival time value will be the minimum time required to generate an update; hence, it is shorter than the update service times (S). In other words, the inter-arrival rate is much higher than the service rate ($\lambda >> \mu$), as shown in Fig. 6.7. Hence, the updates waiting time will increase with time; consequently, the number of updates waiting (N_Q) in the queue will increase as well. Nevertheless, the queue would remain stable [2]; hence, the number of updates waiting in the queue would be controlled by the ST modes. In particular, the worst

[1] The ACK in this chapter represents the receiving of the update and does not reflect the absences of errors in the updates.

[2] A queue is unstable if the number of updates reach ∞.

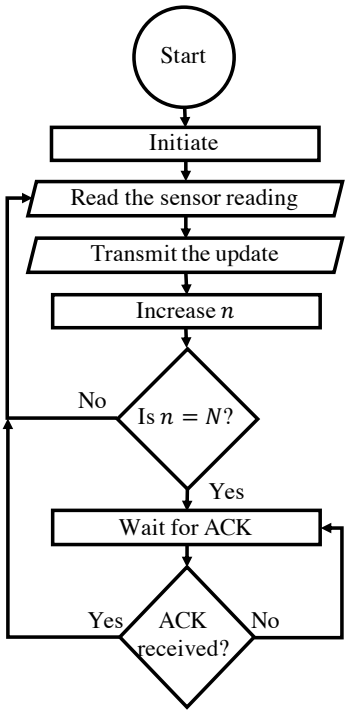

Fig. 6.5 The Clustered Acknowledgement Policy status updating flow chart. The source continuously generates and transmits the updates until the N^{th} update (where N can take an value depending on the requirements of the application), then it has to wait for an Acknowledgement (ACK) to be received.

case would be when the sum of inter-arrival time for all updates up to N is less than the service time of the first update (S_1), in this case, the number of updates in the queue will be N as follows:

$$N_Q = N, \quad \text{iff} \sum_{n=1}^{n=N} X_n < S_1. \tag{6.8}$$

Therefore, the number of updates waiting in the queue cannot exceed N, as shown in Fig. 6.6. Hence, the queue would remain stable; despite having ($\lambda >> \mu$).

To better understand the CAP waiting time, let us consider the updates timeline presented in Fig. 6.7. At the moment of turning on the source of the updates, say, a sensor, let us assume that the number of updates in the queue equals to zero. The first update waiting time will be zero, while for the second update, it had to wait until the first update finished serving, then it will start to be served. For the third update, it has to wait in the queue for the two previous updates to be served and then it will be served. Hence, the waiting time for the second update will be longer than the second update. To generalise, we can assume that the waiting time increases with each

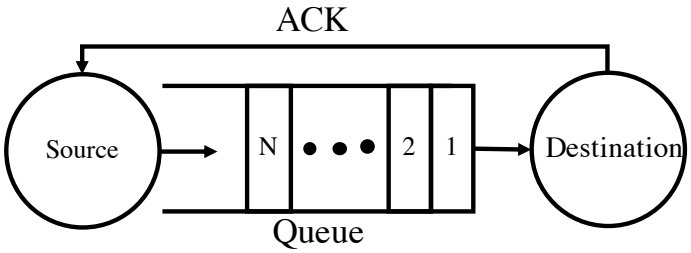

Fig. 6.6 The Clustered Acknowledgement Policy (CAP) diagram, shown the maximum number of updates waiting in the queue to be served, i.e., N.

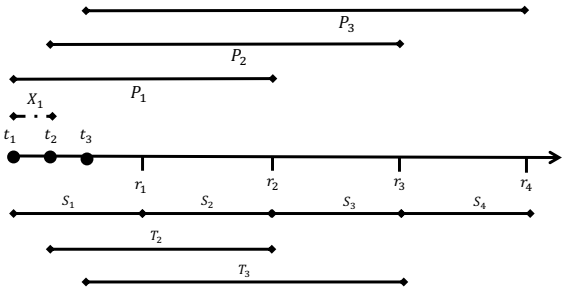

Fig. 6.7 Peak Age illustration of the Clustered Acknowledgement Policy Continuous Updating mode. Showing the 'short' duration of the inter-arrival time (X), which cause the update to wait for a 'long' duration to be served. The serving time is refereed to as S and the delay time (waiting and serving time) is T.

update. We are assuming that the waiting time increases linearly with each update; a validation for this assumption is shown in section 6.4.

We derived a closed form expression for the CAP PA by deriving an expression for PA for each update that had been sent in the period from 0 up to $\mathcal{T}$, and then by taking the median value [3] for them. Since we have two operating modes, we need to derive an expression for each of them. The PA as defined in (6.1) is the duration between the generation of an update until the time of receiving the next update; hence, we can use this definition to derive an expression for the CU mode. From Fig. 6.7, we derive the PA for nth update with the CU mode as

$$P_n^{CU} = n\mathbb{E}[S] + (\mathbb{E}[S] - (n-1)\mathbb{E}[X]). \tag{6.9}$$

To calculate the mean PA for updates 1 to N, we use the linearity assumption as follows;

[3]The median value was used as a representation of the average, hence, in experiemnts it is possible to have a considerable error in the tranmission of the data, which would causes a long delay in the measurements and conseqently in the mean value.

$$P^{CU} = \frac{P_N^{CU}}{2}, \tag{6.10}$$

in other words, the mean peak age value for the CU mode for update from 0 to N, is equal to half of the peak age value of update N.

If we implement the CU mode on its own for a 'long' duration the number of the updates waiting in the queue (Q) would reach ∞,

$$\lim_{t\to\infty} Q = \infty. \tag{6.11}$$

Hence, the queue with the CU is not stable.

The Stabilizing mode (ST) operates similarly to the Zero-Wait policy, as it depends on ACKs to be sent from the destination to the sources. The PA depends on both the service time of the updates and ACK service time. Fig. 6.8 illustrates the PA of the ST mode, which can be calculated by

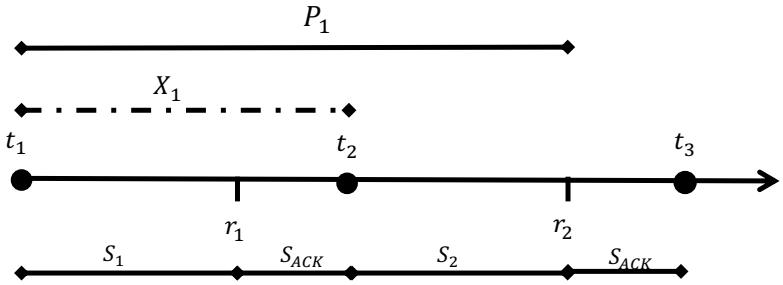

Fig. 6.8 Peak Age illustration of Clustered ACK Policy Stabilising mode. The update is generated at time t and received at time r; then the destination transmits the ACK, hence, the ACK service time is refereed to as S_{ACK}.

$$P^{ST} = 2\mathbb{E}[S] + \mathbb{E}[S_{ACK}], \tag{6.12}$$

where S_{ACK} represent the time to communicate the ACK.

We can derive an expression for the CAP PA (P^{CAP}) using (6.9) and (6.12) as follows,

$$P^{CAP} = \alpha P^{CU} + \beta P^{ST}, \tag{6.13}$$

where α represents the ratio of updates sent with the CU mode and β represents the ratio of updates sent with the ST mode.

$$\alpha = \frac{N}{N+1},$$
$$\beta = \frac{1}{N+1}. \tag{6.14}$$

6.4 Clustered Acknowledgement Policy (CAP) Performance

Here we present the throughput and freshness performance of the CAP. The performance is obtained by evaluating the PA and Inter-arrival time for updates sent from the source, i.e., a client implemented with a Python Socket object to the destination, i.e., a server. The source of the updates emulates a sensor transmitting its readings, and the destination emulates a decision maker. The mean service time is $S = 1/\mu = 1$ second. Two service time distributions were tested, i.e., deterministic and exponential.

6.4.1 Validation Results

Before evaluating the CAP PA performance, the assumption that the PA increases linearly in the CU regime is shown empirically by examining its time series performance. Fig. 6.9, shows the CAP PA performance for a deterministic service time with 50 updates per ACK ($N = 50$). Similarly, Fig. 6.10 presents the time series for an exponential service time, again for $N = 50$.

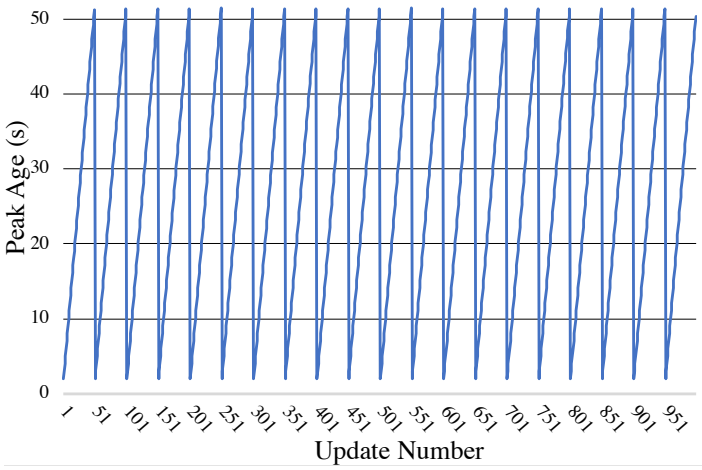

Fig. 6.9 CAP Peak Age time series for a deterministic service time and $N = 50$

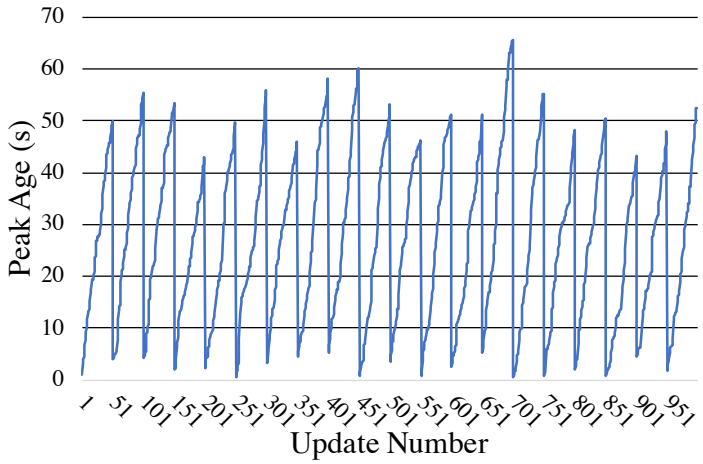

Fig. 6.10 CAP Peak Age time series for a exponential service time and $N = 50$

It can be observed that for the first update the PA initially starts with its minimum value and then increases linearly until the Nth update. At update $N + 1$ the PA restarts from the minimum value and follows the same pattern. For the exponential service time, the PA also increases; however, it has some variability because of the random update time. Although the pattern is not exactly linear, on average, it is reasonable to claim that the linearity assumption is valid.

Next, we validate the mathematical model proposed in (6.13). The number of updates per ACK used in our experiments is $N = 10, 25, 50, 75, 100, 125, 150, 175, 200, 225, 250$. Fig. 6.11 presents the CAP PA with N varies from 10 upto 250 updates per ACK, for a deterministic service time, evaluated by both experimentally and theoretically. We observe a close match between the results, and we also observe that the PA for the CAP increases with N.

6.4.2 CAP Peak Age and Throughput Performance

Peak Age performance

As shown in Fig 6.11, the CAP PA performance changes with the number of updates per ACK. In the following results, we present the CAP performance for $N = 10$; hence, it achieves the lowest (best) PA performance. CAP PA performance was compared to the optimal performance of M/M/1[4], and M/D/1[5] queues and also with the ZW policy.

[4]M/M/1 queue represents a queue where the inter-arrival time and service time follows an exponential distribution [11].

[5]M/D/1 queue represents a queue where the inter-arrival time follows an exponential distribution, and service time is a deterministic value [11].

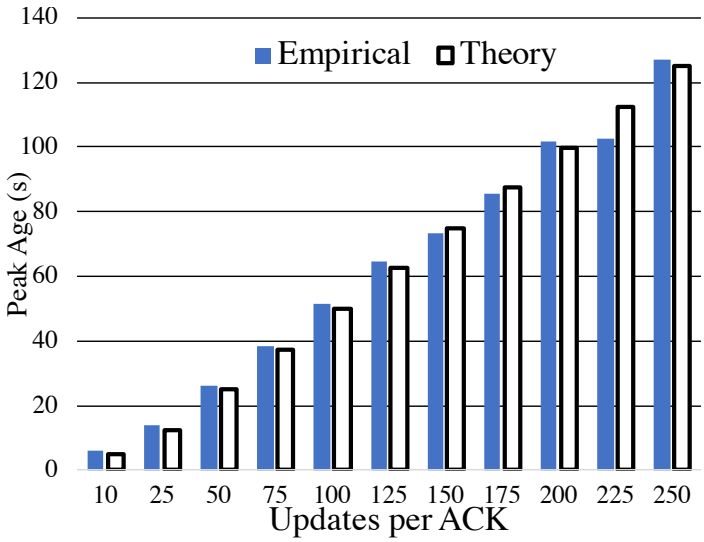

Fig. 6.11 Peak Age of CAP for number of updates per ACK ranges from 10 up to 250

Fig. 6.12, shows the PA performance with the service time following an exponential distribution. We can observe that the ZW PA outperforms the M/M/1 and CAP. In particular, the ZW PA is approximately one-third that of the PA achieved by CAP.

The PA for the deterministic service time is shown in Fig. 6.13. Similar to that observed for the exponential service time, the ZW has the shortest PA while the CAP is the longest. Hence, we can conclude that the ZW outperforms the CAP regarding PA performance.

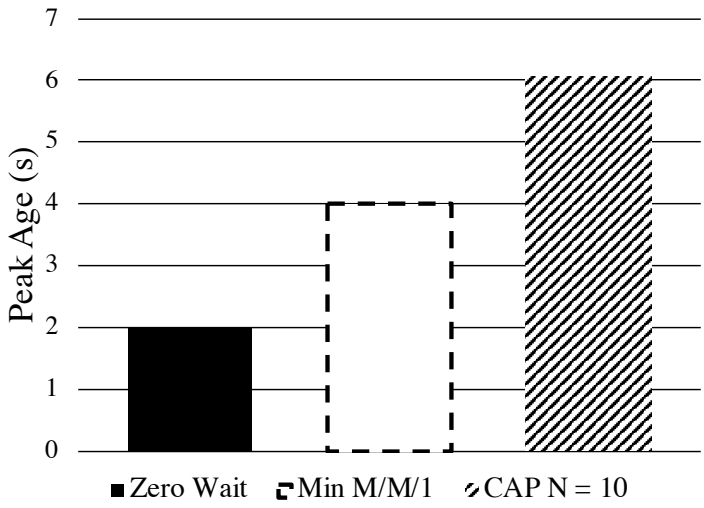

Fig. 6.12 Peak Age of CAP $N = 10$, Zero-Wait and Min M/M/1 for the service time that follows an exponential distribution.

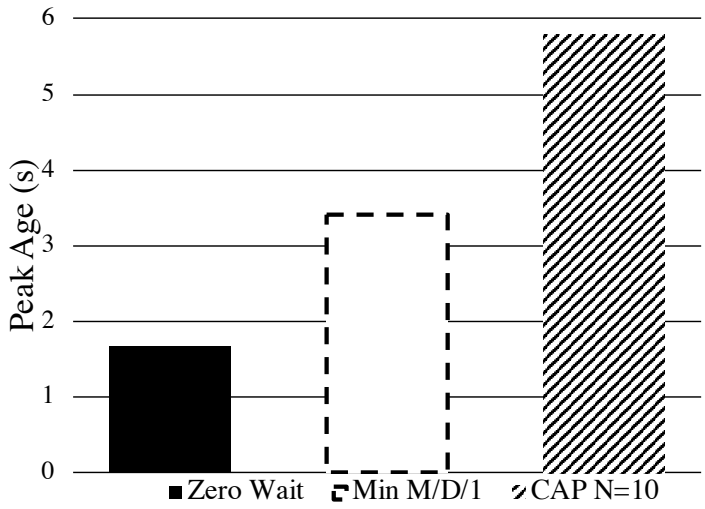

Fig. 6.13 Peak Age of CAP $N = 10$, Zero-Wait and Min M/D/1 for a deterministic service time.

CAP Updating Throughput Performance

Here, we are evaluating the time between the updates (X) or the rate of updating (λ). For the deterministic service time, the performance is presented in Table 6.1. We can observe that the CAP inter-arrival rate is considerably smaller than that of ZW and optimum M/D/1. For the exponential service time, Table 6.2 presents the X and λ results.

Table 6.1 Inter-arrival Time X and rate λ of a deterministic service time for CAP, ZW, optimal M/D/1 queue

Policy	X(s)	λ
CAP	0.00017	5882.61
ZW	1.00566	0.99
Optimal M/D/1	1.71	0.59

We can observe that the CAP PA performance is longer than the ZW; however, it outperforms the ZW in the updating throughput by several order of magnitude. Hence, for applications that require high throughput, at the cost of degraded the freshness, CAP can be sufficient for them.

We can conclude that the CAP can archive a much higher inter-arrival rate than ZW policy. However, its PA is longer than the ZW. Hence, CAP is suitable for applications that require as much information about the process of interest as possible. For instance,

Table 6.2 Inter-arrival Time X and rate λ of a exponential service time for CAP, ZW, optimal M/M/1 queue

Policy	X(s)	λ
CAP	0.00016	6250.82
ZW	0.703349	1.42
Optimal M/M/1	2	0.5

applications that will use the information to recognise a pattern, for example, in a telehealth application, where electrocardiogram readings are being sent to an emergency team to make treatment decisions. In such a scenario, if the information is not very fresh, say 2 seconds late (such as the results presented in Fig. 6.12), it would not be a critical issue, since in this example it is more important to have sufficient readings to diagnose a heart attack.

6.5 Chapter Conclusions and Forthcoming Work

In the previous chapter, it was shown that in order to deliver fresh information the throughput must be controlled. This chapter presents the Clustered Acknowledgement Policy, a policy that delivers fresh information without severely compromising its updating throughput. Clustered Acknowledgement Policy was tested in the presence of deterministic and exponential service times. The experiments showed that the Clustered Acknowledgement Policy inter-arrival rate performance substantial outperforms the Zero-Wait policy and the theoretical optimal queues.

In the previous chapters, the author had used some of the most commonly used queues. These queuing modes were used as an abstraction of the real life queues. However, very little work had been done to evaluate these models. In the next chapter, the author proposed to model the internet-of-things communication. The model was used to evaluate the traffic generated by the machine-to-machine communication.

Chapter References

[1] S. Kaul, R. Yates and M. Gruteser, "Real-time status: How often should one update?," 2012 Proceedings IEEE INFOCOM, Orlando, FL, 2012, pp. 2731-2735.

[2] Kadota, I., Sinha, A. and Modiano, E., 2018, April. Optimizing age of information in wireless networks with throughput constraints. In IEEE INFOCOM 2018-IEEE Conference on Computer Communications (pp. 1844-1852). IEEE.

[3] Li, B., Li, R. and Eryilmaz, A., 2015. Throughput-optimal scheduling design with regular service guarantees in wireless networks. IEEE/ACM Transactions on Networking (ToN), 23(5), pp.1542-1552.

[4] E. Najm, R. Nasser and E. Telatar, "Content Based Status Updates," 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, 2018, pp. 2266-2270.

[5] R. Devassy, G. Durisi, G. C. Ferrante, O. Simeone and E. Uysal-Biyikoglu, "Delay and Peak-Age Violation Probability in Short-Packet Transmissions," 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, 2018, pp. 2471-2475.

[6] R. Devassy, G. Durisi, G. C. Ferrante, O. Simeone and E. Uysal, "Reliable Transmission of Short Packets Through Queues and Noisy Channels Under Latency and Peak-Age Violation Guarantees," in IEEE Journal on Selected Areas in Communications, vol. 37, no. 4, pp. 721-734, April 2019.

[7] J. Zhong, R. D. Yates and E. Soljanin, "Two Freshness Metrics for Local Cache Refresh," 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, 2018, pp. 1924-1928.

[8] M. Costa, M. Codreanu and A. Ephremides, "On the Age of Information in Status Update Systems With Packet Management," in IEEE Transactions on Information Theory, vol. 62, no. 4, pp. 1897-1910, April 2016.

[9] B. Barakat, H. Yassine, S. Keates, K. Arshad and I. J. Wassell, "How to Measure the Average and Peak Age of Information in Real Networks?" in IEEE 25^{th} European Wireless (EW) 2019.

[10] B. Barakat, S. Keates, I. Wassell and K. Arshad, "Adaptive Status Arrivals Policy (ASAP) Delivering Fresh Information (Minimise Peak Age) in Real World Scenarios," In International Conference on Human-Computer Interaction.

[11] D. Bertsekas and R. Gallager, Data Networks. Englewood Cliffs, NJ: Prentice-Hall, 1987

[12] L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, 2015, pp. 1681-1685.

[13] Barakat, B., Keates, S., Wassell, I. and Arshad, K., 2019. Is the Zero-Wait Policy Always Optimum for Information Freshness (Peak Age) or Throughput. IEEE Communications Letters.

Chapter 7

Machine Communication Model (MCM)

- **Research Gap:**
  Are the Queuing Models used in the literature accurate?
- **Published paper:**
  - B. Barakat, S. Keates, K. Arshad and I. J. Wassell, "Deriving Machine to Machine (M2M) Traffic Model from Communication Model," 2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT), Amman, 2018, pp. 1-5.
  - Barakat, B., Keates, S., Wassell, I. J., and Arshad, K., "Modelling IoT devices communication employing representative operation modes to reveal traffic generation characteristics." International Journal of Parallel, Emergent and Distributed Systems (2019), pp.1-13.
- **Most relevant papers:**
  - O. Al-Khatib, W. Hardjawana, and B. Vucetic, "Traffic modeling for Machine-to-Machine (M2M) last mile wireless access networks," 2014 IEEE Glob. Commun. Conf. GLOBECOM 2014, pp. 1199–1204, 2014.
  - V. Paxson and S. Floyd, "Wide Area Trdfie: The Failure of Poisson Modeling," IEEE/ACM Trans. Netw., vol. 3, no. 3, pp. 226–244, 1995.

"The formulation of the problem is often more essential than its solution, which may be merely a matter of mathematical or experimental skill"; Albert Einstein

Several traffic models for the Internet of Things (IoT) have been proposed in the literature. However, they can be considered as heuristic models since they only reflect the stochastic characteristic of the generated traffic. In this chapter, we propose a model to represent the communication of IoT devices. The model was used to obtain the traffic generated by the devices. Therefore, the proposed model is able to capture a wider understanding of device behaviour than existing, state-of-the-art traffic models. The proposed model illustrates the behaviour of Machine-to-Machine uplink communication in a network with multiple-access limited information capacity shared channels. In this chapter, we analysed the number of transmitted packets using the traffic model extracted from our proposed communication model and compared with the state-of-the-art traffic models using simulations. The simulation results show that the proposed model has significantly higher accuracy in estimating the number of transmitted packets compared with the current models in the literature.

7.1 Introduction

The amount of data carried through wireless networks has increased by more than 100 fold in the past decade [1]. Several market research studies have predicted that the amount of data will continue to grow exponentially [2]. Furthermore, the number of connected devices is also expected to grow exponentially. The increase in the number of connected devices is occurring due to the variety of new applications coming on to the market, such as smart homes and wearable devices. Handling this extraordinary increase in the amount of communication data and number of connected devices is the driving force for researchers around the world investigating the next generation of wireless communication, i.e., the fifth generation (5G).

For the previous two generations of wireless communications, the typical challenges were energy efficiency [3], data throughput [4, 20], coverage [5] and end-to-end latency. For 5G, these issues are still considerably challenging; however, serving the expected number of connected devices might be overwhelming. The Internet of Things (IoT) is one of the leading forces in increasing the number of connected devices. The IoT can be defined as the network connecting billions of Machine-to-Machine communication (M2M) devices. M2M, also known as Machine-Type-Communications (MTC), is defined

as the communication between machines or from machine to the network with little or no human intervention [6]. IoT is expected to play a crucial role in several sectors, including smart grids [7], environmental monitoring, surveillance, healthcare [8], and intelligent transport systems [9]. Several market studies have predicted that there will be more than 50 billion M2M devices in operation by 2020 [10]. Providing a ubiquitous service for this extraordinary number of connected devices and the consequent volume of data generated by those devices is the biggest challenge for network operators [14, 18].

To design a network that can serve a large number of IoT devices, it is critical to have a comprehensive understanding of IoT communication and the traffic generated by its devices. It is known that the characteristics and the traffic patterns of M2M differ significantly from the conventional Human-to-Human (H2H) communication (mobile phone calls and computer video calls)[11, 13, 33]. For instance, commonly M2M applications generate short bursts of periodic data, and the cellular network is not well adapted for such short messages [14–17].

In this chapter, a model for IoT communication is proposed. The model is used to represent the traffic generated by IoT devices extending the work done in [37]. To better understand the communication model, let us consider a conference as an analogy. If it is intended to model the noise that will be produced by the audience, we can model it as a random process (an analytical approach is shown in section 7.2.1), or alternatively, use a sensor to record the noise level at several conferences and then generalise the measured noise level (an empirical approach that will be presented in section 7.2.2). However, it would be much more comprehensive to investigate when the noise is generated, which is usulay the breaks period in the conference, hence, perceiving the conference program can be used to estimate the noise level. The conference program here is analogous to the communication model.

Consequently, the traffic extracted from the IoT communication model considers several related factors (as shown in Fig.7.1). The first factor is the channel information capacity. The channel information capacity plays a significant role in the time required to transmit data. Most traffic models available in the literature do not consider the information capacity as they are mainly based on the Erlang model [19] (such as [6]). The Erlang model was proposed for telephone networks (i.e., circuit switched networks) and are arguably not valid for M2M traffic. Hence, in the circuit switched networks, at the moment a communication starts, a communication channel is designated to carry the data through them, accordingly the data throughput is constant. While, in packet

switched networks (currently used) the data is not always a voice communication and the data would be fragmented and communicated through the network, hence, the data throughput varies with time.

The second factor not accounted for in the existing M2M traffic models is the blocking incidence in which the user requires access to the shared channels, but the channels are fully occupied [21–23]. Additionally, the multiple-access technique is missing in the existing M2M traffic models [21–23]. For a shared channel, there are two main multiple-access techniques [26]: (i) Centralised Scheduled Access in which a centralised device determines what part of the channel is allocated to each user, and (ii) Distributed Access in which each user locally decides the channel to access.

Modelling the communication can be insightful to better understand the behaviour devices in networks. For instance, it can help the researchers to model the traffic generated by the devices. Another example application can be the modelling of the energy consumption of the devices. One application that the authors believe that the contribution made in this chapter can be very insightful; is the modelling for real-time systems. In particular, the work done on the Age of Information, in which several researchers assumed that the traffic is generated according to Poisson distribution [24, 25].

This chapter is organised as follows. Section 2 briefly present the state of the art traffic models. Section 3, presents the proposed Machine Communication Model; section 4 shows the simulation results in which we present the number of transmitted packets in a predefined time period. This chapter is concluded in section 5.

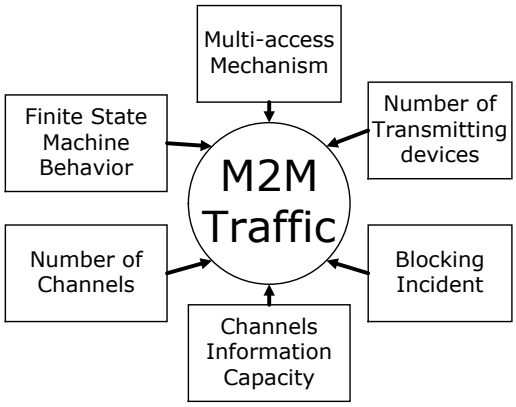

Fig. 7.1 Factors affecting M2M communication traffic.

7.2 Traffic Models proposed in literature

In the literature, two main approaches have been taken to model the traffic generated by the M2M devices (M2MDs). The first approach was to propose a stochastic model to evaluate the traffic (analytical approach) and the second approach was to measure the traffic generated by the M2MDs (an empirical approach) as shown in Fig. 7.2.

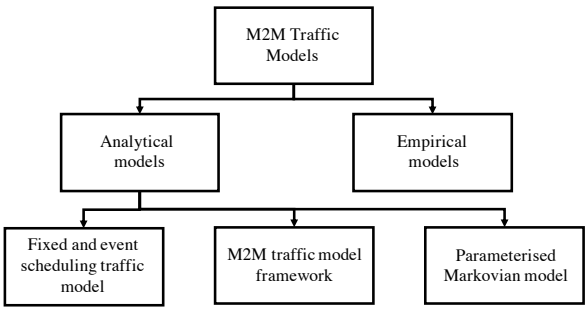

Fig. 7.2 Traffic models proposed in literature.

7.2.1 Analytical Approach

Fixed Scheduling and Event-driven M2MDS Traffic Model

The authors in [21, 23, 29] proposed splitting the M2MDs' traffic modelling into two distinct models according to the transmission periodicity. The first model considers the traffic generated by the periodic updates referred to as Fixed Scheduling (FS) nodes, e.g., sending a sensor measurement. The traffic generated by an FS node was assumed to follow a deterministic process. The second modelling problem was focusing on the non-periodic data traffic referred to as Events-Driven (ED) nodes, e.g., the report of an emergency alarm. The traffic packets generated by the ED notes are modelled as a Poisson Process with rate λ_D (number of packets sent in an explicitly defined time). Table 7.1 summarises the modelling classification:

Table 7.1 Machine-to-Machine communication devices classification proposed in [23].

M2MD node group	Traffic transmission periodicity	Transmission statistical distribution
Fixed Scheduling	Periodic	Deterministic
Events-Driven	Non-periodic	Poisson

Although the authors of [23] remarked on the inaccuracy of conventional traffic models, they made some inaccurate simplifying assumptions in their modelling. The first assumption made was to assume that the M2MDs can be either FS nodes or ED nodes. This assumption makes the model only applicable to specific devices. These devices can do only a particular job (such as periodically report the temperature, but where it cannot report an event such as when the temperature is higher than a set threshold), while most of the M2MDs at the moment in the market can be of both types. Assuming that the Fixed Scheduling nodes are synchronised is another one of the inaccurate assumptions. Hence, the authors in [30] investigated the synchronisation of machine-generated traffic such as router state update messages (a message that reports the current link state). It was demonstrated (analytically and empirically) that behaviour transition from asynchronous to synchronous is practically abrupt even if it was affected by an external influence (such as turning the devices *On* simultaneously). The synchronisation in the case of M2MDs would be an even more significant challenge. Hence most of the M2MDs will be connected to the network through a wireless connection; the propagation delays and multi-path will play a vital role in preventing synchronisation.

M2M Traffic Model Framework [21]

The authors in [21] made a remarkable contribution in demonstrating the differences between human to human communication (H2H) and M2M traffic. They proposed an M2M traffic model similar to the Engset Traffic model (also known as the *On-Off* model [27]). The only difference between the two models was that in the model proposed they assumed a Semi-Markov chain while in the Engset model, it is a Markov chain. The principal difference between a Markov chain and a Semi-Markov chain is the time between successful states transitions. In particular, in the Semi-Markov process, the states transition times are random variables [31].

The M2M traffic model proposed in [21] is shown in Fig. 7.3. It assumes that the transmission of data occurs in one the following instances: (1) Periodic Update data referred to as PU; (2) Event-Driven data referred to as ED; or, (3) Payload Exchange which refers to the data traffic following the PU and ED traffic. A Timer or an Event drive the transition from the OFF state to ON state. On the other hand, the transition between the ON state and OFF state occurs when data transmission finishes.

They also proposed a model for the Sensor-Based Alarm and Event Detection device shown in Fig. 7.4. In this model, they used the sub-states of the ON state in

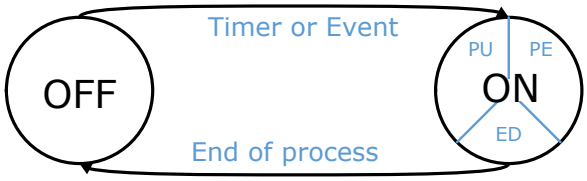

Fig. 7.3 Machine-to-machine communication (M2M) traffic model proposed in [21]. PU refers to Periodic Update, ED refers to Event Driven and PE refers to Payload Exchange.

Fig. 7.3 as main states. However, they did not use the PE exchange sub-state as they assumed that PU and ED are implicitly included in the PE state.

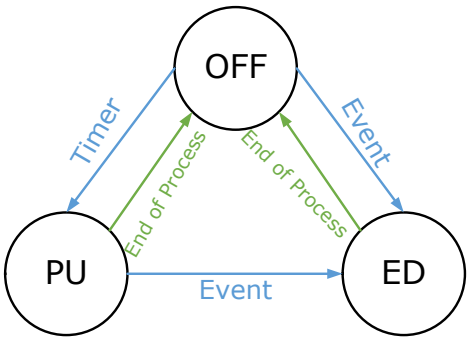

Fig. 7.4 Sensor based alarm and event detection model used in [21]. PU refers to Periodic Update, ED refers to Event Driven.

The inter-departure times between the states and the size of the packets are assumed to be identical and independent random variables. However, in practical cases, this does not reflect the situation of M2MD traffic unless it is an exceptional case in which the device transmits a very short burst of data traffic. Additionally, the researchers did not take into consideration the channel characteristics and the number of devices.

Coupled Markov Modulated Poisson Process Model [22]

The authors in [22] proposed a traffic model for M2MDs relying on a Markov Modulated Poisson Process (shown in Fig. 7.5). However, they used a Coupled Markov Modulated Poisson Process (CMMPP) to illustrate the M2MDs' synchronisation effect. The CMMPP refers to multiple Markov chains that influence each other's transition probabilities $P_n[t]$. The transition probability is defined as the probability of changing from one state into the next state in a defined unit of time. The CMMPP was initially proposed in the context of pattern recognition. They assumed that the arrival is a Poisson process. The arrival rate in the proposed model depends on the current state of the MMPP, e.g., λ_1 represents the rate of arrival of the first state.

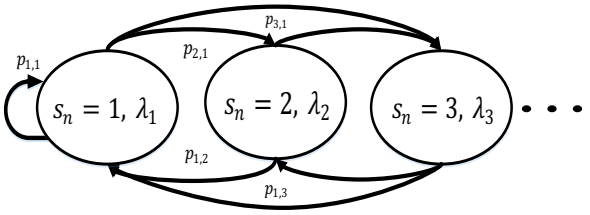

Fig. 7.5 Markov Modulated Poisson Process model used in [22]. s_n represents the number of M2MD transmitting data and λ represent the state arrival rate.

The model proposed was compared with those models proposed by 3rd Generation Partnership Project (3GPP)[1]. That was developed to model the aggregated traffic of several M2MDs. The focus of the comparison was to evaluate the complexity of computing and simulating the traffic. The simulation results showed that the CMMPP model would require a slightly higher simulation duration, but it can provide a better representation of the M2MD traffic than the 3GPP model.

Although the model proposed added a new aspect to the simulation (i.e., the effect of the M2MDs synchronisation) as compared it with the conventional traffic models, it still inherited various assumptions employed in the conventional models. In particular, they rely on the Markov Modulated model. As a result, they assumed that the arrival rates are still being considered as a Poisson Process. The Poisson Process arrivals assumption is very commonly used in the literature because of its simplicity. However, it is not the best representation of M2MD traffic. The principal reason for that is that typically M2MDs generate traffic periodically. Therefore, each periodically generated packet relies on the timing of the previous packet, which contradicts with the memory-less property of Poisson Processes [27].

Parameterised Markov Model [14]

The authors of [14] proposed a traffic model based on a Markov Process. Their main contribution was to evaluate the Blocking Probability [2] in a network that services both M2MD and H2H communication. The traffic model they used was similar to the model represented in Fig. 7.5 [22]. The parameters used in the evaluation of the traffic model blocking probability was adapted from field trials in literature. Fig. 7.6 represents the

[1] 3rd Generation Partnership Project is a standards organisation which develops protocols for mobile telephony

[2] Blocking Probability is the probability that a device would not be able to transmit data because of a lack of available channels.

approach they used to obtain the results by combining parameters from simulations and lab measurements.

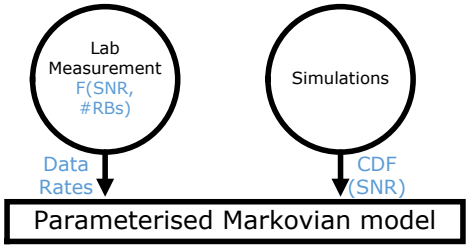

Fig. 7.6 The parameters used in the traffic model in [14]. The model proposed used the data rate (i.e., the data throughput achievable in terms of the number of bits that can be communicated using a communication channel) from a lab measurement. The lab measurement relies on the Signal to Noise Ratio (SNR) and the number of Resource Blocks (RB) to measure the Data Rate. The simulations were used to obtain SNR statistical properties (in particular, the Cumulative Distribution Function (CDF)).

Although the authors in [14] tried to bridge the gap between the analytical and empirical models, their analytical model still needs further enhancement. The analytical model used can be described as theoretical and does not reflect all the M2MD characteristics. The next subsection presents a brief introduction to the empirical models introduced in the literature.

7.2.2 Empirical Model

Empirical models rely on experiments and tests to evaluate a certain model. Typically, the models proposed using this methodology start by running the experiment, and afterwards, they try to fit the collected data into a certain statistical distribution. The seminal paper by Willinger et al. [32] used this approach to prove the deficiency of modelling computer network communication traffic as a Poisson Process. At the time of publication of that paper, the Poisson Process was the most commonly used approach, and it was highly accepted [28, 32, 34]. They also proposed the *Self-Similar Traffic* model for a *Local Area Networks* (LANs). The self-similar process refers to a type of Stochastic Process that seems to have the same behaviour when viewed at different scales [32].

Recently, the authors of [12, 33] used an empirical approach and measured the M2M traffic in a cellular network. They concluded that M2MD traffic would have a significant impact on the connectivity of smartphones. In particular, the M2MD would compete with the smartphones on the available channels, and therefore, the blocking

probability would increase. Although the empirical models illustrate the behaviour of several communication networks, they also have their shortcomings. Especially that they are a reactive approach to solve already existing problems. The empirical approach can only evaluate the considered scenario and is not be able to give a generalised model. This approach can only model aggregated traffic throughout the network. Therefore, modelling the source traffic (per device) is not possible.

7.3 PROPOSED M2M COMMUNICATION MODEL (MCM)

7.3.1 Overview

Most of the research done in the literature is focused on modelling the traffic generated by IoT devices, in this chapter we are modelling the communication that generates the traffic. In our investigation, we started with understanding the M2MDs, which are typically low computational complexity finite state machine that mainly consist of sensor(s), a microprocessor/controller and a communication unit. The M2MD's main function is to monitor the environment and send a report to a centralised node so that the data can be analysed along with data collected from other similar nodes. Fig. 7.7 illustrates a generic M2MD data communication flow chart. The M2MD initially, at start-up, monitors the environment (e.g., senses the motion in a room). After a predefined period, the M2MD sends a periodic update (i.e., Round Robin state update) to the base station or a centralised node. In the occurrence of a triggered interrupt (an event occurs, e.g., a movement detected), the M2MD also transmits exceptional, i.e., non-periodic data to report it.

The proposed M2MD Communication Model (MCM) is shown in Fig. 7.8. MCM is a discrete stochastic process that consists of four states: Sleep (s), Round Robin (r), Interrupt (i) and Buffer (b). At any time, the M2MD is considered to be in one of these four states and would change to another state with a certain probability referred to as the Transition Probability (TP). The TPs shown in Fig. 7.8 represent the Starting State and Finishing State. For example, for a TP $P_{s,b}$ the Starting State would be s and the Finishing State would be b.

The Sleep state represents the starting state of the finite state machine in which the M2MD is not transmitting any data. The Round Robin state represents the epoch in which the M2MD is transmitting routine periodic updates data, e.g., a periodic

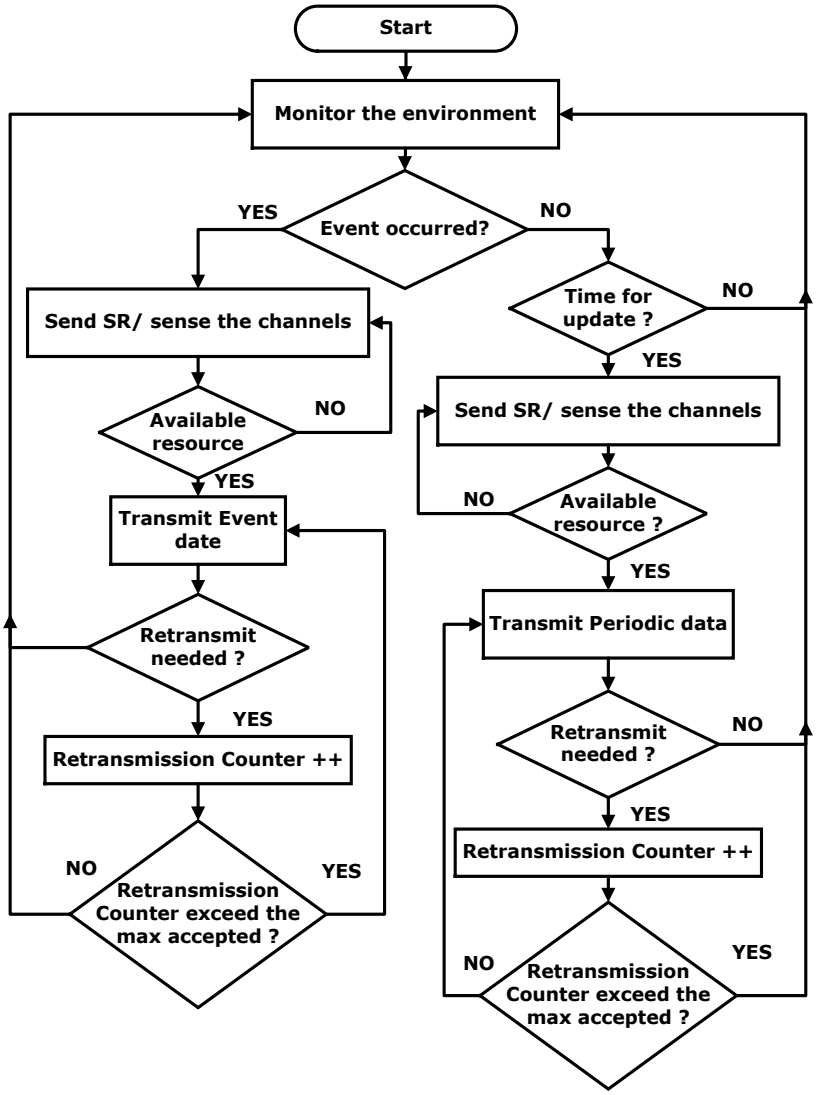

Fig. 7.7 Generic M2MD data communications flow chart. The flow chart shows the two types of data generated by an M2MD, i.e., periodic updates and non periodic data communication.

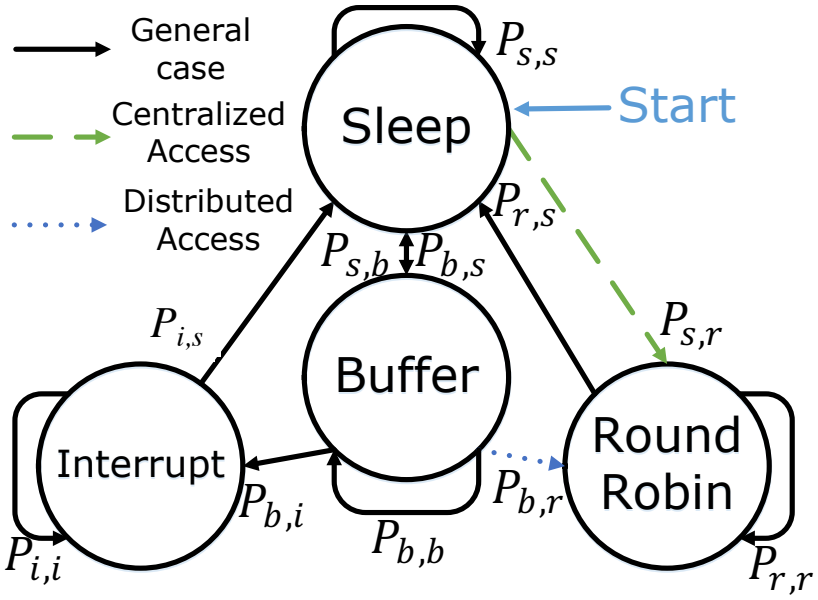

Fig. 7.8 Proposed M2MD's Communication Model, i.e., MCM, showing the four states that represents the IoT devices communication. Also shown are the probabilities of changing from one state to another.

report of room temperature. During the Buffer state, the M2MD has data to be sent, but it is still waiting to access the shared channels to transmit it. Additionally, in the case of fully occupied channels, the M2MD buffers the data packets until it can access a channel. The Interrupt state represents a non-periodic update event occurring in the M2MD in which it sends data representing the event, e.g., a burglar alarm is activated.

In MCM, the data traffic is transmitted during two distinct states, i.e., Interrupt and Round Robin. It differentiates between the two states for the following reasons:

- Typically, the data that has to be sent in the Round Robin updates are short data bursts, while in the Interrupt state data packet size is comparatively large. For instance, motion detectors would periodically send comparatively short data bursts (e.g., data sent containing the device identifier and, say, the battery state information). On the other hand, in the case of an exceptional event (e.g., a moving object had been detected), the M2MD would send a longer data burst that contains information of the event (e.g., a picture or the coordinates of the moving object);

- In the Round Robin state the communication is synchronised while communication in the Interrupt state is asynchronised;

- Consequently, the communication that occurs in the two states (Round Robin and Interrupt) would differ in their channel access approach, which relies on the network access technique.

7.3.2 MCM Transitions

The MCM is modelled as a discrete stochastic process in which at each time unit a state transition occurs. The transition can be to any possible state (including the starting state itself). The TPs determine which state is the one most likely to be moved to in the next time slot. The summation of the TPs going out of any state must equal to unity, as follows:

$$\begin{aligned} &P_{s,s} + P_{s,r} + P_{s,b} = 1 \\ &P_{b,b} + P_{b,i} + P_{b,s} + P_{b,r} = 1 \\ &P_{r,r} + P_{r,s} = 1 \\ &P_{i,i} + P_{i,s} = 1. \end{aligned} \tag{7.1}$$

The self-transition probabilities, i.e., staying in the same state, rely on several factors. In particular, $P_{s,s}$, which represents the probability of remaining in the Sleep state, depends on the frequency of both the periodic updates and the event occurring. The availability of channel resources directly affects the value of $P_{b,b}$. In particular, the value of $P_{b,b}$ is equal to the channel's instantaneous Blocking Probability. The length of the M2MD data packet and the channel quality, e.g., Signal to Noise Ratio (SNR), determines the value of both $P_{r,r}$ and $P_{i,i}$. Currently, let us only consider the SNR to be affecting the information data rate. Hence, the maximum achievable information rate by the kth M2MD in the jth channel ($R_{k,j}$) can be obtained by the Shannon capacity formula:

$$R_{k,j} = BW log_2(1 + SNR_{k,j}), \tag{7.2}$$

where BW is the channel bandwidth, and SNR refers to signal to noise ratio. The probability to remain in the round robin $P_{r,r}$ and the interrupt states $P_{s,s}$ can be

calculated by

$$\begin{aligned} P_{r,r \text{ or } i,i} &= \frac{1}{\gamma(t)_{r,r \text{ or } i,i}} \\ \gamma(t)_{r,r \text{ or } i,i} &= \left\lceil \frac{DR_{r \text{ or } i}}{R_{k,j}(t)} \right\rceil \end{aligned} \tag{7.3}$$

where γ represents the number of time units the data needs to be transmitted, $\lceil . \rceil$ refers to the ceiling function, t refers to the instantaneous time, and DR is the state data requirements.

In a network with shared channels, there are two main multi-access techniques (as classified in [26]). The first technique is Centralised Scheduling, in which the M2MD must send a Scheduling Request (SR) to a centralised device such as a Base Station (BS) to access the channel. The Base Station controls the M2MDs' channel's multiple-access scheduling [6]. The second technique is Distributed Scheduling where each M2MD makes a local decision whether it should access any particular channel based on channel sensing techniques, such as in [35].

In a Centralised Scheduling network, a central device such as a BS schedules the M2MD shared channel access. Consequently, the M2MD is required to send a Scheduling Request (SR) before starting to transmit data. After the BS receives the SR, it schedules a specified Resource (such as a time and bandwidth pair) for the M2MD. Thus, when an interrupt occurs (i.e., asynchronous data transmission is required) the M2MD needs to store the data in its buffer (i.e., the Buffer state). The time duration the data packets spend in the buffer represents the time of sending the SR to the BS, and for a resource to be scheduled. On the other hand, in a Round Robin update, data packets are transmitted in a predefined epoch (i.e., at an explicitly defined time). Accordingly, the M2MD sends the SR to the BS in advance, and M2MD periodic updates do not require data buffering.

However, in a Distributed Scheduling network all the data transmission (i.e., the data transmission owing both the Interrupt and Round Robin states) has to be buffered until the M2MD senses the channel and determines an unoccupied channel then transmits the data. Table 7.2 illustrates both data communication types (i.e., Round Robin state data and Interrupt state data for both multi-access approaches (i.e., Centralised and Distributed Scheduling).

Table 7.2 Data procedures for both types of Network Access, i.e., Centralised and Distributed Scheduling.

Data generating state	Data transmission procedure
Interrupt	Initially, the M2MD is in the Sleep state. When the data is ready to be transmitted the M2MD stores it in the Buffer. The M2MD remains in the buffer state until it either detects an unoccupied channel (for Distributed Scheduling) or it has been allocated a channel (for Centralised Scheduling).
Round Robin	In Centralised Scheduling the M2MD changes from the Sleep State (i.e., initial state) to the Round Robin State. In Distributed Scheduling, the M2MD changes from the initial state to the buffer state and stays there until it senses an unoccupied channel.

The Round Robin updates occur in a predefined epoch, so in a Centralised Scheduling network, $P_{s,r}$ follows a Deterministic distribution with a rate of λ. It is worth mentioning that the assumption that $P_{s,r}$ is deterministic is only acceptable if the SR was sent in sufficient time for the centralised device to allocate a channel resource to the M2MD. On the other hand, the interrupts occur randomly and hence $P_{s,b}$ can be modelled as a Discrete Poisson distribution with a mean μ. The probability of the M2MD discarding the packets it has previously prepared to transmit is represented bt $P_{b,s}$. This incident occurs when the packets have been blocked for a period of time; therefore, the information represented in the packet is not relevant anymore.

The TPs in the MCM model can thus be represented as a Transition Matrix (δ):

$$\delta = \begin{bmatrix} P_{s,s} & P_{s,r} & P_{s,b} & P_{s,i} \\ P_{r,s} & P_{r,r} & 0 & 0 \\ P_{b,s} & P_{b,r} & P_{b,b} & P_{b,i} \\ P_{i,s} & 0 & 0 & P_{i,i} \end{bmatrix}, \tag{7.4}$$

where the probabilities in each row have the same Starting State and the probabilities in each column share the same Finishing State.

The steady-state probabilities of the Sleep, Round Robin, Buffer and Interrupt states are referred to as P_s, P_r, P_b and P_i respectively. Accordingly, the steady-state probabilities can be expressed as a Stationary Vector (Q)

$$Q = [P_s\ P_r\ P_b\ P_i], \tag{7.5}$$

where,

$$P_s + P_r + P_b + P_i = 1. \tag{7.6}$$

The steady-state probabilities for the M2MD for the MCM can be obtained using the Balance equation:

$$\delta \times Q = Q \text{ or } Q(\delta - I) = 0 \tag{7.7}$$

where (I) is the identity matrix. Accordingly, from (7.2) and (7.4), expression (7.5) can be represented as:

$$P_s(P_{s,s} - 1) + P_r(P_{r,s}) + P_i(P_{i,s}) = 0 \tag{7.8}$$

$$P_s(P_{s,r}) + P_r(P_{r,r} - 1) = 0 \tag{7.9}$$

$$P_s(P_{s,b}) + P_b(P_{b,b} - 1) = 0 \tag{7.10}$$

$$P_s(P_{s,i}) + P_b(P_{r,i}) + P_i(P_{i,i} - 1) = 0. \tag{7.11}$$

Finally, by solving (7.5),(7.8),(7.9),(7.10) and (7.11), the values of Q and, hence, steady-state probabilities can be obtained.

The M2MD only transmits data in two states, i.e., Round Robin and Interrupt. Therefore, the number of transmitted packets (NP) can be derived from the MCM by using the probability of a device transmitting data (P_T) and the number of devices in the area of interest (n):

$$NP = P_T \times n \text{ where } P_T = P_r + P_i. \tag{7.12}$$

7.4 Evaluating the Number of Transmitted Packets

For simulating the M2MDs a discrete event simulator [36] was used to evaluate the network behaviour. In [26], it was shown that the Distributed Scheduling approach could outperform the Centralised Scheduling approach where there is delayed Channel State Information (CSI). In a high user density network (such as a network handling many M2MDs), the probability of delaying the CSI is high, therefore, in this report, let us study the packet transmission in a Distributed Scheduling network. The channel access probability is assumed to be equiprobable access across the M2MDs, i.e., all M2MDs are considered to have the same priority. For the simulations, five M2MDs (i.e., $n = 5$) sharing three channels was considered. The parameters and the associated values used to obtain the numerical and simulation results are given in Table 7.3. The parameters were chosen to be representative of a simple network, however, the model can also represent the traffic in other networks. The number of packets transmitted by the M2MDs with respect to the time units is shown in Fig. 7.9. As shown in the figure, the MCM can model the simulated M2MD traffic more accurately. In particular, in the case where $\gamma_{r,r}$ and $\gamma_{i,i}$ are equal to unity and three respectively (i.e., SNR 1), the MCM is able to predict the number of transmitted packets with significantly higher accuracy than the Poisson model (MMPP). For instance, in SNR1 the number of packets achieved by simulation is 3×10^4 for the 5×10^4 time unit, and using MCM is 3.041×10^4, that is less than 1.4% error. However, using the MMPP model, which does not adapt with respect to the SNR, the predicted number is 2.5×10^4, which is about 16.7% error.

7.5 Conclusions

In the literature, several traffic models for M2M communications traffic have been proposed. Those models are able to represent M2M traffic for a specific set of scenarios, but, they do not cope well with a different set of scenarios. In this chapter, a model for IoT communications was proposed by looking more closely at M2MD behaviour. The communication model was used to extract the traffic generated by the M2MDs. In the proposed method, the data traffic does not only rely on the statistical characteristics of the M2MD traffic. The extracted traffic has several other factors affecting it, such as the channel information capacity and multi-access technique used. The traffic model commonly used in the literature was simulated using a discrete event simulator and

Table 7.3 Numerical Parameters and Values.

Parameter	Value
Simulation duration	10×10^4 Time Units
Number of M2MD n / Channels	5/3
SNR 1 $\gamma_{r,r/i,i}$	1 for (r,r) / 10 for (i,i)
SNR 2 $\gamma_{r,r/i,i}$	3 for (r,r) / 30 for (i,i)
Round Robin update distribution in MCM	Deterministic with mean of 10
Interrupts Distribution in MCM	Poisson with mean 50
Data Requirements DR_r/DR_i	150 / 1500 Kbit
P_T for the Poisson model	Exponential distribution with mean 10
$P_{b,s}$	0

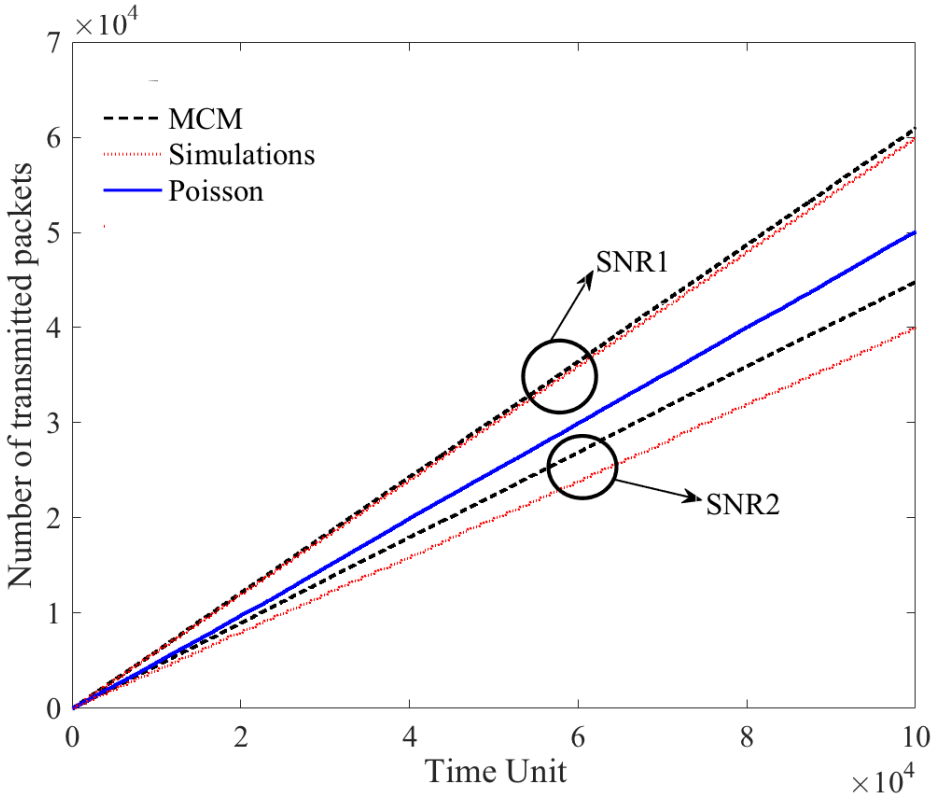

Fig. 7.9 Number of successfully transmitted packets with respect to the time unit.

compared with the analytical results obtained by extracting the generated traffic out of the proposed communication model. The results showed a significant improvement in predicting the number of packets with respect to time by using the proposed model.

Chapter References

[1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What Will 5G Be?," IEEE J. Sel. Areas Commun., vol. 32, no. 6, pp. 1065–1082, 2014.

[2] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update , 2010 – 2015," Growth Lakel., vol. 2011, no. 4, pp. 2010–2015, 2011.

[3] B. Barakat and K. Arshad, "Energy efficient carrier aggregation for LTE-Advanced," 2015 IEEE 8th GCC Conference & Exhibition, Muscat, 2015, pp. 1-5.

[4] S. O. Aramide, B. Barakat, Y. Wang, S. Keates and K. Arshad, "Generalized proportional fair (GPF) scheduler for LTE-A," 2017 9th Computer Science and Electronic Engineering (CEEC), Colchester, 2017, pp. 128-132.

[5] M. Sharsheer, B. Barakat and K. Arshad, "Coverage and capacity self-optimisation in LTE-Advanced using active antenna systems," 2016 IEEE Wireless Communications and Networking Conference, Doha, 2016, pp. 1-5.

[6] B. Barakat and K. Arshad, "Energy efficient scheduling in LTE-advanced for Machine Type Communication," in 2015 International Conference and Workshop on Computing and Communication, IEMCON 2015, 2015.

[7] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, "Toward Intelligent Machine-to-Machine Communications in Smart Grid" IEEE Commun. Mag., vol. 49, no. 4, pp. 60–65, 2011.

[8] N. Bui and M. Zorzi, "Health care applications: a solution based on the internet of things," Int. Symp. Appl. Sci. Biomed. Commun. Technol., pp. 0–4, 2011.

[9] A. Zanella, L. Vangelista, N. Bui, A. Castellani, and M. Zorzi, "Internet of Things for Smart Cities," IEEE Internet Things J., vol. 1, no. 1, p. 22, 2014.

[10] C. Perera, C. H. I. H. Liu, S. Jayawardena, and M. Chen, "A Survey on Internet of Things From Industrial Market Perspective," IEEE Access, vol. 2, pp. 1660–1679, 2014.

[11] M. T. Islam, A. E. M. Taha, and S. Akl, "A survey of access management techniques in machine type communications," IEEE Commun. Mag., vol. 52, no. 4, pp. 74–81, 2014.

[12] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "Large-scale measurement and characterization of cellular machine-to-machine traffic," IEEE/ACM Trans. Netw., vol. 21, no. 6, pp. 1960–1973, 2013.

[13] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the random access channel of LTE and LTE-A suitable for M2M communications? A survey of alternatives," IEEE Commun. Surv. Tutorials, vol. 16, no. 1, pp. 4–16, 2014.

[14] C. Ide, B. Dusza, M. Putzke, C. Muller, and C. Wietfeld, "Influence of M2M communication on the physical resource utilization of LTE," in Wireless Telecommunications Symposium, 2012, pp. 1–6.

[15] M. Shirvanimoghaddam, Y. Li, M. Dohler, B. Vucetic, and S. Feng, "Probabilistic Rateless Multiple Access for Machine-to-Machine Communication," IEEE Trans. Wirel. Commun., vol. 14, no. 12, pp. 6815–6826, 2015.

[16] A. G. Gotsis, A. S. Lioumpas, and A. Alexiou, "M2M scheduling over LTE: Challenges and new perspectives," IEEE Veh. Technol. Mag., vol. 7, no. 3, pp. 34–39, 2012.

[17] E. Soltanmohammadi, K. Ghavami, and M. Naraghi-Pour, "A Survey of Traffic Issues in Machine-to-Machine Communications over LTE," IEEE Internet Things J., vol. 4662, no. c, pp. 1–1, 2016.

[18] T. Petsch, S. N. Khan Marwat, Y. Zakit, and C. Gorg, "Influence of future M2M communication on the LTE system," Proc. 2013 6th Jt. IFIP Wirel. Mob. Netw. Conf. WMNC 2013, 2013.

[19] A. K. Erlang, "The theory of probabilities and telephone conversations," Nyt Tidsskr. Mat. B, vol. 20, no. 33–39, p. 16, 1909.

[20] B. Barakat and K. Arshad, "An Adaptive Hybrid Scheduling Algorithm for LTE-Advanced," 2015, no. Ict, pp. 91–95.

[21] N. Nikaein, M. Laner, K. Zhou, P. Svoboda, D. Drajic, M. Popovic, and S. Krco, "Simple traffic modeling framework for machine type communication," 10th IEEE Int. Symp. Wirel. Commun. Syst. 2013, ISWCS 2013, pp. 783–787, 2013.

[22] M. Laner, P. Svoboda, N. Nikaein, and M. Rupp, "Traffic models for machine type communications," 10th IEEE Int. Symp. Wirel. Commun. Syst. 2013, ISWCS 2013, vol. 9, no. i, pp. 651–655, 2013.

[23] O. Al-Khatib, W. Hardjawana, and B. Vucetic, "Traffic modeling for Machine-to-Machine (M2M) last mile wireless access networks," 2014 IEEE Glob. Commun. Conf. GLOBECOM 2014, pp. 1199–1204, 2014.

[24] B. Barakat, H. Yassine, S. Keates, K. Arshad and I. J. Wassell, 'How to Measure the Average and Peak Age of Information in Real Networks?' To appear in IEEE 25^{th} European Wireless (EW) 2019.

[25] B. Barakat, S. Keates, K. Arshad and I. J. Wassell, 'Adaptive Status Arrivals Policy (ASAP) Delivering Fresh Information (Minimise Peak Age) in Real World Scenarios,' In International Conference on Human-Computer Interaction.

[26] M. Johnston and E. Modiano, "A New Look at Wireless Scheduling with Delayed Information," IEEE Int. Symp. Inf. Theory - Proc., pp. 1407–1411, 2015.

[27] D. P. Bertsekas, R. G. Gallager, and P. Humblet, "Data networks", vol. 2. Prentice Hall, 1992.

[28] V. Paxson and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modeling," IEEE/ACM Trans. Netw., vol. 3, no. 3, pp. 226–244, 1995.

[29] O. Al-Khatib, S. Member, W. Hardjawana, and B. Vucetic, "Traffic modeling and optimization in public and private wireless access networks for smart grids," IEEE Trans. Smart Grid, vol. 5, no. 4, pp. 1949–1960, 2014.

[30] S. Floyd and V. Jacobson, "The Synchronization of Periodic Routing Messages," IEEE/ACM Trans. Netw., vol. 2, no. 2, pp. 122–136, 1994.

[31] D. Gross, J. Shortle, F. Thompson, and C. Harris, "Fundamentals of Queueing Theory". 2008.

[32] K. Park and W. Willinger, "Self-Similar Network Traffic and Performance Evaluation", vol. 53. 1989.

[33] Shafiq, M.Z., Ji, L., Liu, A.X., Pang, J. and Wang, J., 2012. "A first look at cellular machine-to-machine traffic: large scale measurement and characterization." ACM SIGMETRICS performance evaluation review, 40(1), pp.65-76.

[34] R. Jain and S. Routhier, "Packet trains–measurements and a new model for computer network traffic," IEEE J. Sel. areas, vol. 4, no. 6, 1986.

[35] A. Aijaz and A. H. Aghvami, "Cognitive machine-to-machine communications for internet-of-things: A protocol stack perspective," IEEE Internet Things J., vol. 2, no. 2, pp. 103–112, 2015.

[36] Mathworks, "SimEvents®: User's Guide," MATLAB Manual, 2011.

[37] B. Barakat, S. Keates, K. Arshad and I. J. Wassell, "Deriving Machine to Machine (M2M) Traffic Model from Communication Model," 2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT), Amman, 2018, pp. 1-5.

Chapter 8

Thesis Conclusions and Future Work

"An expert is a person who has made all the mistakes that can be made in a very narrow field"; Niels Bohr (Nobel Prize Physicist).

The thesis research question was *'How to deliver fresh information for real-world applications?'*; this question inspired the author to contribute to the current knowledge. Initially, the author identified a gap in the literature, i.e., the little empirical work in the research scope. Hence, in chapter 3, the author aims to motivate more experimental work on the information freshness research area by making it straightforward to estimate the metrics from experiments. The proposed method was validated on an emulated $M/M/1$, $D/D/1$ and $M/D/1$ queues. It was shown that the proposed method could achieve estimates that are very close to the theoretical derivation.

A reliable empirical method to evaluate the Peak Age inspired the author to investigate another gap in the literature. This gap is the optimality of the Zero-Wait policy, which was considered as an optimal policy for delivering fresh information. In chapter 4, it has been shown that the Zero-Wait policy is not always the optimum policy for either Peak Age or throughput. The results presented contradict the current paradigm that Zero-Wait policy is always the optimum throughput policy.

Hence, chapter 4 showed that the Zero-Wait policy is not always optimal to deliver fresh information a new gap in the literature is formed, i.e., *How to deliver fresh information in real-world scenarios?*. In chapter 5, a policy for minimising the Peak Age of Information was proposed; i.e., Adaptive Status Arrivals Policy. The Adaptive Status Arrivals Policy regulates the inter-arrival time of status updates to deliver fresh information. The performance was measured by conducting experiments on three scenarios. The Adaptive Status Arrivals Policy Peak Age performance in the tested scenarios approaches the optimal value. Moreover, it can adapt to the server load and the varying load on the internet.

In chapter 5, it was shown that to deliver fresh information; the throughput must be controlled. This chapter presents the Clustered Acknowledgement Policy, a policy that provides fresh information without severely compromising its updating throughput. Clustered Acknowledgement Policy was tested in the presence of deterministic and exponential service times. The experiments showed that the Clustered Acknowledgement Policy inter-arrival rate performance substantial outperforms the Zero-Wait policy and the theoretical optimal queues.

In chapters 3, 4 and 5, the author had used some of the most commonly used queues. These queuing modes were used as an abstraction of real life queues. In the literature, several traffic models for machine-to-machine communication traffic have been proposed.

Those models can represent machine-to-machine communication traffic for a specific set of scenarios, but, they do not cope well with a different set of scenarios. In chapter 6, the author proposed to model Internet-of-things communication. The model was used to evaluate the traffic generated by machine-to-machine communication. In the proposed method, the data traffic does not only rely on the statistical characteristics of the device's traffic. The extracted traffic has several other factors affecting it, such as the channel information capacity and multi-access technique used. The traffic commonly used in the literature was simulated using a discrete event simulator and compared with the analytical results obtained by extracting the traffic out of the proposed communication model. The results showed a significant improvement in predicting the number of packets with respect to time by using the proposed model.

The contributions made in this thesis have opened the way to several other contributions to be made. In chapter 3, the author can extend the work to measure the Peak Age of several other queues types. Hence, the author can extend the work to measure the Peak Age of several other queues. For chapter 5, the author will extend the work to use a reinforcement learning algorithm to optimise the policy performance. Using a learning algorithm might be useful to deliver fresh information for an entirely random service time distribution.

In chapter 6, the author is currently working on applying the Clustered Acknowledgement Policy to a pattern recognition algorithm. In particular, it is aimed at detecting any irregularities in Electrocardiography (ECG) readings. The final aim is to design a system that can detect a pattern of heart diseases such as a heart attack by a central server. Building such a system requires a sufficient number of readings, which the Clustered Acknowledgement Policy can deliver, and an accurate detection algorithm. To have this algorithm to work with an acceptable level of accuracy, several readings must be collected to train the algorithm.

The work on the Machine Communication Model, presented in chapter 7, can be rigorously validated using large scale experiments and test-bed. Extending the work using the experiments can give the model extra authenticity. Also, it can be tested on several other scenarios. Afterwards, the model can be used to optimise the network resources (the allocated bandwidth and power of transmission) to design a highly reliable network.

Chapter 9

Author Publications

Table 9.1 The following publication were published by the author in a peer-reviewed conferences and journals and they are part of this thesis contributions.

Published	B. Barakat, S. Keates, K. Arshad and I. J. Wassell, 'How to Measure the Average and Peak Age of Information in Real Networks?' To appear in IEEE 25^{th} European Wireless (EW) 2019.
Published	Barakat, B., Keates, S., Wassell, I.J. and Arshad, K., 2019. Modelling IoT devices communication employing representative operation modes to reveal traffic generation characteristics. International Journal of Parallel, Emergent and Distributed Systems, pp.1-13. (Link)
Invited	B. Barakat, S. Keates, K. Arshad and I. J. Wassell, 'Adaptive Status Arrivals Policy (ASAP) Delivering Fresh Information (Minimise Peak Age) in Real World Scenarios,' Lecture Notes in Computer Science. (Link)
Published	B. Barakat, S. Keates, K. Arshad and I. J. Wassell, 'Is the Zero-Wait policy always optimum for information freshness (Age) or throughput?' IEEE Communication Letters (Link)
Published	B. Barakat, S. Keates, K. Arshad and I. J. Wassell, 'Deriving Machine to Machine (M2M) Traffic Model from Communication Model,' 2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT), Amman, 2018, pp. 1-5. (Link)

Table 9.2 The following publication were published by the author in a peer-reviewed conferences and journals and not part of this thesis main contributions.

Published	S. O. Aramide, B. Barakat, Y. Wang, S. Keates and K. Arshad, 'Generalized proportional fair (GPF) scheduler for LTE-A,' 2017 9th Computer Science and Electronic Engineering (CEEC), Colchester, 2017, pp. 128-132. (Link)
Published	M. Sharsheer, B. Barakat and K. Arshad, 'Coverage and capacity Self-Optimisation in LTE-Advanced using Active Antenna Systems,' 2016 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Doha, 2016, pp. 79-83. (Link)
Published	B. Barakat and K. Arshad, 'Energy efficient scheduling in LTE-advanced for Machine Type Communication,' 2015 International Conference and Workshop on Computing and Communication (IEMCON), Vancouver, BC, 2015, pp. 1-5. (Link)
Published	B. Barakat and K. Arshad, 'An adaptive hybrid scheduling algorithm for LTE-Advanced,' 2015 22nd International Conference on Telecommunications (ICT), Sydney, NSW, 2015, pp. 91-95. (Link)
Published	B. Barakat and K. Arshad, 'Energy efficient carrier aggregation for LTE-Advanced,' 2015 IEEE 8th GCC Conference & Exhibition, Muscat, 2015, pp. 1-5.(Link)

Chapter 10

Thesis References

1. Green, P., 1996. The Greco-Persian Wars. University of California Press.

2. Hill, D. and Sharp, S., 1997. An Anglo-Saxon Beacon System. Names, Places and People: An Onomastic Miscellany for John McNeal Dodgson, pp.157-165.

3. Erlang, A.K., 1909. The theory of probabilities and telephone conversations. Nyt. Tidsskr. Mat. Ser. B, 20, pp.33-39.

4. Kuo, T.W. and Mok, A.K., 1993, December. SSP: A semantics-based protocol for real-time data access. In 1993 Proceedings Real-Time Systems Symposium (pp. 76-86). IEEE.

5. Song, X. and Liu, J.W.S., 1990, October. Performance of multiversion concurrency control algorithms in maintaining temporal consistency. In Proceedings., Fourteenth Annual International Computer Software and Applications Conference (pp. 132-139). IEEE.

6. Bertsekas, D.P., Gallager, R.G. and Humblet, P., 1992. Data networks (Vol. 2). New Jersey: Prentice-Hall International.

7. Gallager, R.G., 2012. Discrete stochastic processes (Vol. 3). Springer Science & Business Media.

8. Atzori, L., Iera, A. and Morabito, G., 2010. The internet of things: A survey. Computer networks, 54(15), pp.2787-2805.

9. Anton-Haro, C. and Dohler, M. eds., 2014. Machine-to-machine (M2M) communications: architecture, performance and applications. Elsevier.

10. Islam, S.R., Kwak, D., Kabir, M.H., Hossain, M. and Kwak, K.S., 2015. The internet of things for health care: a comprehensive survey. IEEE Access, 3, pp.678-708.

11. Gungor, V.C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C. and Hancke, G.P., 2011. Smart grid technologies: Communication technologies and standards. IEEE transactions on Industrial informatics, 7(4), pp.529-539.

12. Garcia, C.B., 2001. Method for monitoring and trading stocks via the internet displaying bid/ask trade bars. U.S. Patent 6,272,474.

13. Zhang, J., Wang, F.Y., Wang, K., Lin, W.H., Xu, X. and Chen, C., 2011. Data-driven intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems, 12(4), pp.1624-1639.

14. Lanfranco, A.R., Castellanos, A.E., Desai, J.P. and Meyers, W.C., 2004. Robotic surgery: a current perspective. Annals of surgery, 239(1), p.14.

15. Adelberg, B., Garcia-Molina, H. and Kao, B., 1995, June. Applying update streams in a soft real-time database system. In ACM SIGMOD Record (Vol. 24, No. 2, pp. 245-256). ACM.

16. Little, J.D., 1961. A proof for the queuing formula. Operations Research, 9(3), pp.383-387.

17. S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?," in 2012 Proceedings IEEE INFOCOM, 2012, pp. 2731–2735.

18. M. Costa, M. Codreanu, and A. Ephremides, "On the Age of Information in Status Update Systems With Packet Management," IEEE Trans. Inf. Theory, vol. 62, no. 4, pp. 1897–1910, Apr. 2016.

19. L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," in 2015 IEEE International Symposium on Information Theory (ISIT), 2015, pp. 1681–1685.

20. Feller, W., 2008. An introduction to probability theory and its applications (Vol. 2). John Wiley & Sons.

21. Keates, S., Langdon, P., Clarkson, P.J. and Robinson, P., 2002. User models and user physical capability. User Modeling and User-Adapted Interaction, 12(2-3), pp.139-169.

22. R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, 2015, pp. 3008-3012. doi: 10.1109/ISIT.2015.7283009

23. A. Arafa, J. Yang, S. Ulukus and H. V. Poor, "Age-Minimal Online Policies for Energy Harvesting Sensors with Incremental Battery Recharges," 2018 Information Theory and Applications Workshop (ITA), San Diego, CA, 2018, pp. 1-10. doi: 10.1109/ITA.2018.8503180

24. B. T. Bacinoglu, E. T. Ceran and E. Uysal-Biyikoglu, "Age of information under energy replenishment constraints," 2015 Information Theory and Applications Workshop (ITA), San Diego, CA, 2015, pp. 25-31. doi: 10.1109/ITA.2015.7308962

25. A. Valehi and A. Razi, "Maximizing Energy Efficiency of Cognitive Wireless Sensor Networks With Constrained Age of Information," in IEEE Transactions on Cognitive Communications and Networking, vol. 3, no. 4, pp. 643-654, Dec. 2017. doi: 10.1109/TCCN.2017.2749232

26. Ning Lu, Bo Ji, and Bin Li. 2018. Age-based Scheduling: Improving Data Freshness for Wireless Real-Time Traffic. In Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '18). ACM, New York, NY, USA, 191-200.

27. Bin Li, Ruogu Li, and Atilla Eryilmaz. 2013. Heavy-traffic-optimal scheduling with regular service guarantees in wireless networks. In Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '13). ACM, New York, NY, USA, 79-88.

28. Corneo, L. and Gunningberg, P., 2018, July. Scheduling at the edge for assisting cloud real-time systems. In Proceedings of the 2018 Workshop on Theory and Practice for Integrated Cloud, Fog and Edge Computing Paradigms (pp. 9-14). ACM.

29. Zhang, S., Li, J., Luo, H., Gao, J., Zhao, L. and Shen, X.S., 2018, October. Towards Fresh and Low-Latency Content Delivery in Vehicular Networks: An

Edge Caching Aspect. In 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP) (pp. 1-6). IEEE.

30. Kam, C., Kompella, S., Nguyen, G.D., Wieselthier, J.E. and Ephremides, A., 2017, June. Information freshness and popularity in mobile caching. In 2017 IEEE International Symposium on Information Theory (ISIT) (pp. 136-140). IEEE.

31. Yates, R.D., Ciblat, P., Yener, A. and Wigger, M., 2017, June. Age-optimal constrained cache updating. In 2017 IEEE International Symposium on Information Theory (ISIT) (pp. 141-145). IEEE.

32. Gao, W., Cao, G., Srivatsa, M. and Iyengar, A., 2012, June. Distributed maintenance of cache freshness in opportunistic mobile networks. In 2012 IEEE 32nd International Conference on Distributed Computing Systems (pp. 132-141). IEEE.

33. Priya, S. and Inman, D.J. eds., 2009. Energy harvesting technologies (Vol. 21, p. 2). New York: Springer.

34. Daigle, J.N., 2005. Queueing theory with applications to packet telecommunication. Springer Science & Business Media.

35. S. K. Kaul, R. D. Yates and M. Gruteser, "Status updates through queues," 2012 46th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, 2012, pp. 1-6.

36. Bedewy, A.M., Sun, Y. and Shroff, N.B., 2019. Minimizing the age of information through queues. IEEE Transactions on Information Theory.

37. Tripathi, V., Talak, R. and Modiano, E., 2019. Age of Information for Discrete Time Queues. arXiv preprint arXiv:1901.10463.

38. C. Kam, S. Kompella and A. Ephremides, "Effect of message transmission diversity on status age," 2014 IEEE International Symposium on Information Theory, Honolulu, HI, 2014, pp. 2411-2415.

39. Yates, R.D., 2018, June. Status updates through networks of parallel servers. In 2018 IEEE International Symposium on Information Theory (ISIT) (pp. 2281-2285). IEEE.

40. Yates, R.D., 2018, April. Age of information in a network of preemptive servers. In IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 118-123). IEEE.

41. Bedewy, A.M., Sun, Y. and Shroff, N.B., 2016, July. Optimizing data freshness, throughput, and delay in multi-server information-update systems. In 2016 IEEE International Symposium on Information Theory (ISIT) (pp. 2569-2573). IEEE.

42. C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier and A. Ephremides, "On the Age of Information With Packet Deadlines," in IEEE Transactions on Information Theory, vol. 64, no. 9, pp. 6419-6428, Sept. 2018.

43. Kuang, Q., Gong, J., Chen, X. and Ma, X., 2019. Age-of-Information for Computation-Intensive Messages in Mobile Edge Computing. arXiv preprint arXiv:1901.01854.

44. Champati, J.P., Al-Zubaidy, H. and Gross, J., 2019. On the distribution of AoI for the GI/GI/1/1 and GI/GI/1/2* systems: Exact expressions and bounds. In IEEE INFOCOM.

45. Wang, M., Chen, W. and Ephremides, A., 2019. Real-Time Reconstruction of Counting Process through Queues. arXiv preprint arXiv:1901.08197.

46. Sert, E., Sönmez, C., Baghaee, S. and Uysal-Biyikoglu, E., 2018, May. Optimizing age of information on real-life TCP/IP connections through reinforcement learning. In 2018 26th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

47. Elgabli, A., Khan, H., Krouka, M. and Bennis, M., 2018. Reinforcement learning based scheduling algorithm for optimizing age of information in ultra reliable low latency networks. arXiv preprint arXiv:1811.06776.

48. Ceran, E.T., Gündüz, D. and György, A., 2018, September. A reinforcement learning approach to age of information in multi-user networks. In 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) (pp. 1967-1971). IEEE.

49. Ceran, E.T., Gündüz, D. and György, A., 2019. Average age of information with hybrid ARQ under a resource constraint. IEEE Transactions on Wireless Communications, 18(3), pp.1900-1913.

50. B. Barakat and K. Arshad, “An adaptive hybrid scheduling algorithm for LTE-Advanced,” in 2015 22nd International Conference on Telecommunications (ICT), 2015, pp. 91-95.

51. E. Najm and R. Nasser, "Age of information: The gamma awakening," 2016 IEEE International Symposium on Information Theory (ISIT), Barcelona, 2016, pp. 2574-2578.

52. A. Kosta, N. Pappas, and V. Angelakis, “Age of Information: A New Concept, Metric, and Tool,” Found. Trends Netw., vol. 12, no. 3, pp. 162-259, 2017.

53. C. Kam, S. Kompella, and A. Ephremides, “Experimental evaluation of the age of information via emulation,” in Proceedings - IEEE Military Communications Conference MILCOM, 2015, vol. 2015-Decem, pp. 1070-1075.

54. “18.1. socket - Low-level networking interface — Python 3.3.7 documentation.” [Online]. Available: https://docs.python.org/3.3/library/socket.html. [Accessed: 06-Mar-2018].

55. “16.3. time - Time access and conversions - Python 3.6.4 documentation.” [Online]. Available: https://docs.python.org/3/library/time.html. [Accessed: 07-Mar-2018].

56. C. Sönmez, S. Baghaee, A. Ergişi and E. Uysal-Biyikoglu, “Age-of-Information in Practice: Status Age Measured Over TCP/IP Connections Through WiFi, Ethernet and LTE," 2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Batumi, 2018, pp. 1-5.

57. B. Barakat, S. Keates, I. J. Wassell, and K. Arshad, “Is the Zero-Wait Policy Always Optimum for Information Freshness (Peak Age) or Throughput?," in IEEE Communications Letters, doi:10.1109/LCOMM.2019.2907935, (in press).

58. Lutz, M., 2001. Programming python. " O’Reilly Media, Inc.".

59. Bennett, J.M., Prinz, D.G. and Woods, M.L., 1952, September. Interpretative sub-routines. In Proceedings of the 1952 ACM national meeting (Toronto) (pp. 81-87). ACM.

60. Peters, T., 2010. The zen of python. In Pro Python (pp. 301-302). Apress.

61. Python Software Foundation 2019, Python 3.37 documentation accessed 06-Mar-2018, https://docs.python.org/3/library/time.html.

62. Brightman, H.J., 1998. Data Analysis in Plain English: With Microsoft Excel. International Thomson Publishing.

63. Marshall, E. and Boggis, E., 2016. The statistics tutor's quick guide to commonly used statistical tests. University of Sheffield.

64. Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal and N. B. Shroff, "Update or Wait: How to Keep Your Data Fresh," in IEEE Transactions on Information Theory, vol. 63, no. 11, Nov. 2017, pp. 7492-7508.

65. S. S. Sawilowsky, 'The Probable Difference Between Two Means When σ 1 2 $\neq$ σ 2 2,' vol. 1, no. 2, 2002, pp. 461-472.

66. S. O. Aramide, B. Barakat, Y. Wang, S. Keates and K. Arshad, "Generalized proportional fair (GPF) scheduler for LTE-A," 2017 9th Computer Science and Electronic Engineering (CEEC), Colchester, 2017, pp. 128-132.

67. Shreedhar, T., Kaul, S.K. and Yates, R.D., 2018, October. ACP: Age Control Protocol for Minimizing Age of Information over the Internet. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (pp. 699-701). ACM.

68. Yates, R.D., 2018, April. Age of information in a network of preemptive servers. In IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 118-123). IEEE.

69. Prorok, A., Hyldmar, N., & He, Y. A Fleet of Miniature Cars for Experiments in Cooperative Driving. IEEE International Conference on Robotics and Automation.

70. M. Sharsheer, B. Barakat and K. Arshad, "Coverage and capacity self-optimisation in LTE-Advanced using active antenna systems," 2016 IEEE Wireless Communications and Networking Conference, Doha, 2016, pp. 1-5.

71. B. Barakat, S. Keates, K. Arshad and I. J. Wassell, "Deriving Machine to Machine (M2M) Traffic Model from Communication Model," 2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT), Amman, 2018, pp. 1-5.

72. B. Barakat and K. Arshad, "Energy efficient scheduling in LTE-advanced for Machine Type Communication," 2015 International Conference and Workshop on Computing and Communication (IEMCON), Vancouver, BC, 2015, pp. 1-5.

73. Kadota, I., Sinha, A. and Modiano, E., 2018, April. Optimizing age of information in wireless networks with throughput constraints. In IEEE INFOCOM 2018-IEEE Conference on Computer Communications (pp. 1844-1852). IEEE.

74. Li, B., Li, R. and Eryilmaz, A., 2015. Throughput-optimal scheduling design with regular service guarantees in wireless networks. IEEE/ACM Transactions on Networking (ToN), 23(5), pp.1542-1552.

75. E. Najm, R. Nasser and E. Telatar, "Content Based Status Updates," 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, 2018, pp. 2266-2270.

76. R. Devassy, G. Durisi, G. C. Ferrante, O. Simeone and E. Uysal-Biyikoglu, "Delay and Peak-Age Violation Probability in Short-Packet Transmissions," 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, 2018, pp. 2471-2475.

77. R. Devassy, G. Durisi, G. C. Ferrante, O. Simeone and E. Uysal, "Reliable Transmission of Short Packets Through Queues and Noisy Channels Under Latency and Peak-Age Violation Guarantees," in IEEE Journal on Selected Areas in Communications, vol. 37, no. 4, pp. 721-734, April 2019.

78. J. Zhong, R. D. Yates and E. Soljanin, "Two Freshness Metrics for Local Cache Refresh," 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, 2018, pp. 1924-1928.

79. B. Barakat, H. Yassine, S. Keates, K. Arshad and I. J. Wassell, "How to Measure the Average and Peak Age of Information in Real Networks?" in IEEE 25^{th} European Wireless (EW) 2019.

80. B. Barakat, S. Keates, I. Wassell and K. Arshad, "Adaptive Status Arrivals Policy (ASAP) Delivering Fresh Information (Minimise Peak Age) in Real World Scenarios," In International Conference on Human-Computer Interaction.

81. J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What Will 5G Be?," IEEE J. Sel. Areas Commun., vol. 32, no. 6, pp. 1065–1082, 2014.

82. T. Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update , 2010 – 2015," Growth Lakel., vol. 2011, no. 4, pp. 2010–2015, 2011.

83. B. Barakat and K. Arshad, "Energy efficient carrier aggregation for LTE-Advanced," 2015 IEEE 8th GCC Conference & Exhibition, Muscat, 2015, pp. 1-5.

84. Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, "Toward Intelligent Machine-to-Machine Communications in Smart Grid" IEEE Commun. Mag., vol. 49, no. 4, pp. 60–65, 2011.

85. N. Bui and M. Zorzi, "Health care applications: a solution based on the internet of things," Int. Symp. Appl. Sci. Biomed. Commun. Technol., pp. 0–4, 2011.

86. A. Zanella, L. Vangelista, N. Bui, A. Castellani, and M. Zorzi, "Internet of Things for Smart Cities," IEEE Internet Things J., vol. 1, no. 1, p. 22, 2014.

87. C. Perera, C. H. I. H. Liu, S. Jayawardena, and M. Chen, "A Survey on Internet of Things From Industrial Market Perspective," IEEE Access, vol. 2, pp. 1660–1679, 2014.

88. M. T. Islam, A. E. M. Taha, and S. Akl, "A survey of access management techniques in machine type communications," IEEE Commun. Mag., vol. 52, no. 4, pp. 74–81, 2014.

89. M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "Large-scale measurement and characterization of cellular machine-to-machine traffic," IEEE/ACM Trans. Netw., vol. 21, no. 6, pp. 1960–1973, 2013.

90. A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the random access channel of LTE and LTE-A suitable for M2M communications? A survey of alternatives," IEEE Commun. Surv. Tutorials, vol. 16, no. 1, pp. 4–16, 2014.

91. C. Ide, B. Dusza, M. Putzke, C. Muller, and C. Wietfeld, "Influence of M2M communication on the physical resource utilization of LTE," in Wireless Telecommunications Symposium, 2012, pp. 1–6.

92. M. Shirvanimoghaddam, Y. Li, M. Dohler, B. Vucetic, and S. Feng, "Probabilistic Rateless Multiple Access for Machine-to-Machine Communication," IEEE Trans. Wirel. Commun., vol. 14, no. 12, pp. 6815–6826, 2015.

93. A. G. Gotsis, A. S. Lioumpas, and A. Alexiou, "M2M scheduling over LTE: Challenges and new perspectives," IEEE Veh. Technol. Mag., vol. 7, no. 3, pp. 34–39, 2012.

94. E. Soltanmohammadi, K. Ghavami, and M. Naraghi-Pour, "A Survey of Traffic Issues in Machine-to-Machine Communications over LTE," IEEE Internet Things J., vol. 4662, no. c, pp. 1–1, 2016.

95. T. Petsch, S. N. Khan Marwat, Y. Zakit, and C. Gorg, "Influence of future M2M communication on the LTE system," Proc. 2013 6th Jt. IFIP Wirel. Mob. Netw. Conf. WMNC 2013, 2013.

96. N. Nikaein, M. Laner, K. Zhou, P. Svoboda, D. Drajic, M. Popovic, and S. Krco, "Simple traffic modeling framework for machine type communication," 10th IEEE Int. Symp. Wirel. Commun. Syst. 2013, ISWCS 2013, pp. 783–787, 2013.

97. M. Laner, P. Svoboda, N. Nikaein, and M. Rupp, "Traffic models for machine type communications," 10th IEEE Int. Symp. Wirel. Commun. Syst. 2013, ISWCS 2013, vol. 9, no. i, pp. 651–655, 2013.

98. O. Al-Khatib, W. Hardjawana, and B. Vucetic, "Traffic modeling for Machine-to-Machine (M2M) last mile wireless access networks," 2014 IEEE Glob. Commun. Conf. GLOBECOM 2014, pp. 1199–1204, 2014.

99. M. Johnston and E. Modiano, "A New Look at Wireless Scheduling with Delayed Information," IEEE Int. Symp. Inf. Theory - Proc., pp. 1407–1411, 2015.

100. V. Paxson and S. Floyd, "Wide Area Trdfie: The Failure of Poisson Modeling," IEEE/ACM Trans. Netw., vol. 3, no. 3, pp. 226–244, 1995.

101. O. Al-Khatib, S. Member, W. Hardjawana, and B. Vucetic, "Traffic modeling and optimization in public and private wireless access networks for smart grids," IEEE Trans. Smart Grid, vol. 5, no. 4, pp. 1949–1960, 2014.

102. S. Floyd and V. Jacobson, "The Synchronization of Periodic Routing Messages," IEEE/ACM Trans. Netw., vol. 2, no. 2, pp. 122–136, 1994.

103. D. Gross, J. Shortle, F. Thompson, and C. Harris, “Fundamentals of Queueing Theory”. 2008.

104. K. Park and W. Willinger, “Self-Similar Network Traffic and Performance Evaluation”, vol. 53. 1989.

105. Shafiq, M.Z., Ji, L., Liu, A.X., Pang, J. and Wang, J., 2012. “A first look at cellular machine-to-machine traffic: large scale measurement and characterization.” ACM SIGMETRICS performance evaluation review, 40(1), pp.65-76.

106. T. S. Rappaport, “Wireless communications: principles and practice”. Prentice Hall PTR, 2002.

107. R. Jain and S. Routhier, “Packet trains–measurements and a new model for computer network traffic,” IEEE J. Sel. areas, vol. 4, no. 6, 1986.

108. A. Aijaz and A. H. Aghvami, “Cognitive machine-to-machine communications for internet-of-things: A protocol stack perspective,” IEEE Internet Things J., vol. 2, no. 2, pp. 103–112, 2015.

109. Mathworks, “SimEvents®: User’s Guide,” MATLAB Manual. pp. 1–458, 2011.

110. Barakat, B., Keates, S., Wassell, I. J., and Arshad, K., “Modelling IoT devices communication employing representative operation modes to reveal traffic generation characteristics.” International Journal of Parallel, Emergent and Distributed Systems (2019), pp.1-13.

"I am not a product of my circumstances. I am a product of my decisions." Stephen R. Covey