

A Queueing-theoretic Analysis of the Threshold-based Exhaustive Data-backup Scheduling Policy

Dieter Claeys^{1,2,a),b)}, Jan-Pieter Dorsman^{3,4,c)}, Apoorv Saxena^{2,d)}, Joris Walraevens^{2,e)} and Herwig Bruneel^{2,f)}

¹*Department of Industrial Systems Engineering and Product Design, Ghent University, Technologiepark 903, B-9052 Zwijnaarde, Belgium*

²*SMACS Research Group, Department TELIN, Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium*

³*Korteweg-de Vries Institute, University of Amsterdam, P.O. Box 94248, 1090 GE Amsterdam, The Netherlands*

⁴*Mathematical Institute, Leiden University, P.O. Box 9512, 2300 RA Leiden, The Netherlands*

^{a)}Corresponding author: Dieter.Claeys@ugent.be

^{b)}URL: <http://ea18.ugent.be/user/163>

^{c)}J.L.Dorsman@uva.nl

^{d)}Apoorv.Saxena@ugent.be

^{e)}Joris.Walraevens@ugent.be

^{f)}Herwig.Bruneel@ugent.be

Abstract. We analyse the threshold-based exhaustive data backup scheduling mechanism by means of a queueing-theoretic approach. Data packets that have not yet been backed up are modelled by customers waiting for service (back-up). We obtain the probability generating function of the system content (backlog size) at random slot boundaries in steady state.

INTRODUCTION

The last decade has witnessed a tremendous growth in data produced and stored. In a recent McKinsey report, it was stated that “by 2009, nearly all sectors in the US economy had at least an average of 200 terabytes of stored data (twice the size of US retailer Wal-Mart’s data warehouse in 1999) per company with more than 1,000 employees” and that “the increasing volume and detail of information captured by enterprises, the rise of multimedia, social media, and the Internet of Things will fuel exponential growth in data for the foreseeable future” [1]. This creates various challenges, among which extracting valuable data from large data sets, i.e., data mining, security (authenticity, integrity), making backups to protect against data loss due to hardware failure, accidental deletion, theft, et cetera.

In order to alleviate the backup assignment of the huge amounts of data of various users, research has been carried out to reduce the amount of data to backup, e.g. by compression techniques. We refer to Tan et al. [2] and Vrable et al. [3] and the references therein. However, even with these data reduction techniques, the amount of data to be backed up and the number of users remain large. Hence, there is a need for efficient data backup scheduling policies, tackling the trade-off between not backing up too frequently to maintain sufficient capacity while keeping the age of data and backlog size sufficiently small to minimize the amount of data at risk of loss.

We analyse a basic data backup scheduling mechanism by means of a queueing-theoretic approach. The backup scheduling mechanism considered here is a threshold-based exhaustive policy, that is, a data backup is initiated as soon as the backlog size reaches or exceeds some threshold, and an ongoing backup is finished when the backlog size becomes zero, that is when all data is backed up including new data that has been created during the backup process. The corresponding queueing model is described and its analysis is briefly sketched in the subsequent sections.

To the best of our knowledge, there is a lack of formal mathematical models for backup scheduling. We are only aware of the paper of van de Ven et al. [4], where a discrete-time queueing model for a distributed backup scheduling protocol for corporate networks with homogeneous users (i.e., same data production and connectivity patterns) is

studied. Particular attention was paid to the tradeoff between frequent backups and reducing the network peak load, taking into account connectivity patterns of users.

DATA BACKUP QUEUEING MODEL

The objective of developing a queueing model is to obtain a formula for the backlog size, that is the data that has not yet been backed up. The queueing model has the following features:

- Time is slotted, that is, a discrete-time queueing model is considered.
- Data packets that have not been backed up are modelled to await backup (service) in a queue.
- During a slot a number of packets can arrive at the queue, that is, new packets are created; the number of arrivals during some slot k is denoted by A_k and $\{A_k; k \geq 0\}$ is modelled as a sequence of independent and identically distributed (iid) random variables, with common probability mass function (pmf) $a(n)$, $n \geq 0$ and probability generating function (pgf) $A(z)$. The mean number of arrivals during a random slot is denoted by λ .
- When no backup is being carried out, a backup will be initiated at the first slot boundary where the backlog size, that is the system content, reaches or exceeds the threshold l .
- The capacity of the backup server equals c , meaning that maximum c packets can be backed up in a slot.
- The server continues backing up until the backlog size becomes 0 again. Hence, at backup completion, the backlog size equals 0.
- Arriving customers can immediately, that is in their slot of arrival, be taken into service if service is going on in that slot and capacity is available. This feature is in literature referred to as early arrivals.

In fact, the queueing model considered here is a discrete-time queueing model with batch service and N -policy (with $N = l$). In literature, the combination of batch service and N -policy has rarely been considered. To the best of our knowledge, only Arumuganathan and Jeyakumar [5], and Reddy et al. [6] incorporate both features, but in continuous time, which is not useful for our purpose.

ANALYSIS OF THE MODEL

Through a partial-generating function approach, we obtain the pgf $U(z)$ of the system content, that is the data that has not yet been backed up, at random slot boundaries in steady state. The pgf $U(z)$ is obtained as follows:

$$U(z) := \sum_{n=0}^{\infty} \lim_{k \rightarrow \infty} \mathbb{P}[U_k = n] z^n = \sum_{n=0}^{l-1} u_0(n) z^n + U_1(z) ,$$

with

- $u_i(n) := \lim_{k \rightarrow \infty} \mathbb{P}[U_k = n, \tau_k = i]$, $n \geq 0$, $i \in \{0, 1\}$,
- U_k : system content at slot boundary k ,
- $\tau_k = 0$ if no backup is going on in slot k ; $\tau_k = 1$ otherwise,
- $U_1(z) := \sum_{n=1}^{\infty} u_1(n) z^n$.

After some calculations, the following expression for $U_1(z)$ is obtained:

$$U_1(z) [z^c - A(z)] = z^c A(z) \sum_{j=0}^{l-1} u_0(j) z^j - z^c \sum_{j=0}^{l-1} z^j \sum_{n=0}^j u_0(n) a(j-n) - \sum_{n=1}^c v_1(n) z^n , \quad (1)$$

with

$$v_1(n) := \lim_{k \rightarrow \infty} \mathbb{P}[U_k + A_k = n, \tau_k = 1] .$$

Equation (1) contains $c+l$ unknown constants $u_0(0), \dots, u_0(l-1), v_1(1), \dots, v_1(c)$, which can be determined once a set of $c+l$ linearly independent equations in these unknowns are determined. First, $c-1$ linearly independent equations are found by means of the standard approach of relying on Rouché's theorem and the analytic property of pgfs [7]:

$$z_i^c A(z_i) \sum_{j=0}^{l-1} u_0(j) z_i^j - z_i^c \sum_{j=0}^{l-1} z_i^j \sum_{n=0}^j u_0(n) a(j-n) - \sum_{n=1}^c v_1(n) z_i^n = 0 , \quad 1 \leq i \leq c-1 ,$$

with z_i the $c - 1$ zeroes of $z^c - A(z)$ different from 1 inside the closed complex unit disk $\{z \in \mathbb{C} : |z| \leq 1\}$. The other equations can be found through probabilistic reasoning, that is linking the state in a slot with feasible states in the previous slot, leading to

$$u_0(n) = \sum_{j=0}^n u_0(j)a(n-j) , \quad 1 \leq n \leq l-1 ,$$

$$u_0(0) = u_0(0)a(0) + \sum_{j=1}^c v_1(j) ,$$

and by applying the normalization condition on $U(z)$:

$$(c - \lambda) \left[1 - \sum_{n=1}^{l-1} u_0(n) \right] = \sum_{j=0}^{l-1} u_0(j)[j + c + \lambda] - \sum_{j=0}^{l-1} (j + c) \sum_{n=0}^j u_0(n)a(j-n) - \sum_{n=1}^c v_1(n)n .$$

CONCLUSIONS AND FUTURE WORK

A queueing system is developed and analysed to model a threshold-based exhaustive backup policy. This paper is one of the first to address the data backup challenge from a queueing-theoretic point of view. To the best of our knowledge, only van de Ven et al. also consider this point of view [4]. Whereas we study a threshold-based policy, van de Ven et al. examine a probabilistic backup policy, that is at each time slot the user decides with some probability whether to initiate a backup, regardless of the backlog size.

There are several directions for future research. The assumption of iid sequence $\{A_k\}$ may be relaxed, because in practice it is likely that the arrival process exhibits some cyclic pattern, as described by van de Ven et al. [4]. Our approach will then have to be slightly adapted by using partial or vector generating functions instead of probability generating functions, in order to keep track of the position within the cycle. Also, other backup policies could be studied and compared one to another. We have started analysing the threshold-based gated backup policy through a similar analysis technique, but we have obtained a functional equation which we have not yet been able to solve. Another possibility is to study a model that combines a threshold policy with a timer policy to ensure that the age of the data does not become excessive. This would result in some kind of discrete-time batch-service queueing model with NT policy.

ACKNOWLEDGMENTS

Dieter Claeys is a Postdoctoral Fellow with the Research Foundation Flanders (FWO-Vlaanderen), Belgium. Part of his research has been funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. The research of the second author was performed when visiting Ghent University in the context of the IAP Bestcom project funded by the Belgian government, while being a Ph.D. student at Eindhoven University of Technology and Centrum Wiskunde & Informatica funded in the framework of the STAR-project ‘‘Multilayered queueing systems’’ by the Netherlands Organisation for Scientific Research (NWO).

REFERENCES

- [1] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Byers, McKinsey Global Institute, Tech. Rep., May 2011 .
- [2] Y. Tan, H. Jiang, E. Sha, Z. Yan, and D. Feng, *Journal of Signal Processing Systems* **72**, 209–228 (2013).
- [3] M. Vrabie, S. Savage, and G. Voelker, *ACM Transactions on Storage* **5**, 1–28 (2009).
- [4] P. van de Ven, B. Zhang, and A. Schorgendorfer, ‘‘Distributed backup scheduling: modeling and optimization,’’ in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2014), Toronto, April 27 - May 2 (2014)*, pp. 1644–1652.
- [5] R. Arumuganathan and S. Jeyakumar, *Applied Mathematical Modelling* **29**, 972–986 (2005).
- [6] G. K. Reddy, R. Nadarajan, and R. Arumuganathan, *Computers and Operations Research* **25(11)**, 957–967 (1998).
- [7] I. Adan, J. van Leeuwen, and E. Winands, *Operations Research Letters* **34**, 355–360 (2006).