Modelling and Performance Evaluation of a Manual Logon System for Electronic Fee Collection

Steffen Tacke[1], Joerg Sommer[2], Hanna Zuendel[2], Christoph M. Gauger[2]

[1]DaimlerChrysler AG, Research and Technology, Vehicle IT and Services Research
HPC 050/G021 - REI/VA, 70197 Stuttgart, Germany
Email: steffen.tacke@daimlerchrysler.com

[2]University of Stuttgart, Institute of Communication Networks and Computer Engineering
Pfaffenwaldring 47, 70569 Stuttgart, Germany
Email: {sommer|zuendel|gauger}@ikr.uni-stuttgart.de

Abstract— In this paper we model a toll-station terminal logon process of an electronic fee collection system as used in the German toll collection system. Currently, approximately 734,000 vehicles participate in this electronic fee collection (EFC) system and up to nine million users are estimated to use a future Europe-wide solution. Thus, a clear view on traffic characteristics and performance of this large-scale, distributed telematics system are paramount for the overall architecture, parameterisation, and operation. Especially, reactions to downtime and recovery conditions must not result in an overload situation which could eventually lead to a total failure of the system. We describe the German toll collection system, detail the manual logon process, and, present a detailed system model. Then, we evaluate by simulation the process of delivering booking data records, which are stored in the terminals during a downtime of the central data centre, after system recovery. Finally, we compare different algorithms for sending data records during this phase regarding the time needed to deliver all queued records as well as regarding the load on the servers in the data centre.

Index Terms— telematic system, electronic fee collection, backoff algorithm, manual logon, terminal, queueing network

Fig. 1.   Manual logon process for EFC

I. Introduction

CURRENTLY, several different variants of fee collection system for vehicles are being deployed throughout various countries. Beyond the installation of toll booths alongside the charged roads, which is rather inflexible and requires extensive infrastructure investments, there are more sophisticated EFC approaches using e.g. dedicated short-range communication (DSRC) between in-car and roadside equipment or in-vehicle positi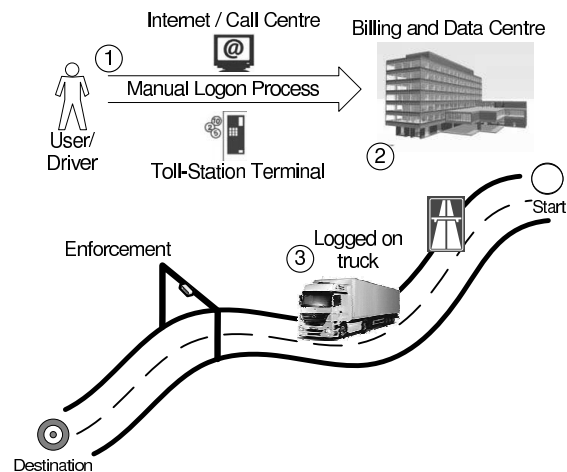oning and communication [3]. In 2005, an EFC system [6] which is based on already well established technologies like the global navigation satellite system (e.g. GPS) and the mobile communication system (e.g. GSM) was introduced in Germany. This system is the worldwide first implementation of a EFC system using the aforementioned technologies. Such systems are often referred to as a global navigation satellite system/cellular network (GNSS/CN) [4]. Currently, the basic system concept is that vehicles are equipped with a so called *on-board unit* (OBU) which tracks the vehicle's position via GPS. As soon as the truck enters a toll road the OBU starts the charging process until the vehicle leaves the toll road. After having accumulated a certain amount of fees the OBU sends all gathered data over the GSM network

to a backend billing data management. This backend system is then responsible for all further accounting and billing. Key advantages of this approach are that the collection and the billing of toll relevant data is done transparently for the driver and, moreover, that the system can be easily extended towards a bigger road network or a differentiated toll system. In [15], we have modelled and evaluated the traffic flow of this automatic GNSS/CN EFC system.

Although only trucks are charged on the German highway system, there are over 734,000 registered vehicles and over 490,000 trucks equipped with an on board unit by July 2005 [2]. With a possible future European EFC approach based on GNSS/CN the number of users in Europe might grow to over nine millions by 2012.

In addition to the aforementioned automatic system a manual payment system was introduced as shown in figure 1 allowing the manual logon and the payment of fees without the OBU. It is an alternative access point to the EFC system for users who travel infrequently into the toll controlled area, who do not like to spend money for an on-board system or whose OBU is malfunctioning. With this system it is possible to specify and bill toll relevant data prior to the start of the trip.

The access to the manual system is realised by either using the internet or a call centre, or by using one of over 3600 toll-station terminals installed close to motorway access ramps all over Germany and the bordering countries (denoted with step 1 in figure 1. Similar to the automatic system, the actual toll processing and billing is done by a central backend system (step 2). With a valid manual logon to the EFC system the driver is allowed to drive on the billed roads (step 3). So far, approximately twenty percent of all registered users participate in the manual system.

Due to the large number of users and equipment installed in the field, and the severe financial consequences of system outages, scalability and stability is a major concern for such an EFC. Although the manual system usually only serves a fraction of the EFC users, it also serves as the backup solution for the unlikely case that the automatic system fails. In order to analyse performance and scalability of such a system we modelled it according to the system specifications and conducted simulation studies for key scenarios. Naturally, such studies can neither be examined experimentally in the running system nor do they lend themselves to lab experiments.

In this paper, the main focus of the evaluation is on the analysis of recovery situations just after system downtimes. These extreme situations are characterised by a burst communication which induces a significant load onto the backend data centre systems as all terminals try to send back-logged data. We will present what kind of traffic will result from this situation and weather this traffic load will cause any further downtime or delay due to an overload situation and resource restrictions in the dial-in system and backend server. We study the algorithms and parameters, which control the communication and system behaviour during backlog resolution after recovery, in depth.

The remainder of this paper is structured in four main parts. First the description of the overall manual logon and payment system architecture is given in section two. Then, in section three the detailed model of the system is described on which the further analysis is based with all relevant parameter settings. Section four comprises the simulation results of the performance evaluation and finally a conclusion and an outlook is given in section five.

II. THE TERMINAL-BASED LOGON SYSTEM

The architecture of the terminal-based EFC system with all relevant components is shown in figure 2. Users enter all necessary information into a toll-station terminal, further denoted as terminal. The toll record contains all relevant data required for the billing process: vehicle license number, starting point, route (via waypoints), ending point and intended departure time. Then, the following process is initiated: As soon as a complete billing data record (*BDR*) is present on the terminal the user gets a confirmation and can proceed with the payment. Additionally, the terminal tries to establish an ISDN dial-in connection to a remote access server (*RAS*) and sends the record via a system connection to the billing data management (*BDM*). Upon reception of the correct BDR, the BDM is responsible for sending an acknowledgement back to the original terminal via the same connection and within a certain time limit T_{ack}.

The failure-free case described so far characterises the so called *normal operational mode*. Apart from this, it is also possible that the terminal does not receive an acknowledgement in time. It will then fall back into an *partial autonomous mode* in which it sends all BDRs to the control centre (*CC*). If the terminal is neither able to deliver its data to the BDM nor to the CC it switches into an *autonomous mode*. This may happen during regular maintenance work, system failure or downtime of the ISDN network, the RAS, the CC and/or the BDM. In this mode the terminal gathers all BDRs locally in a database and continues to issue confirmation notices to all correct logon procedures done by a user. In the further analysis, we neglect the partial autonomous mode and concentrate
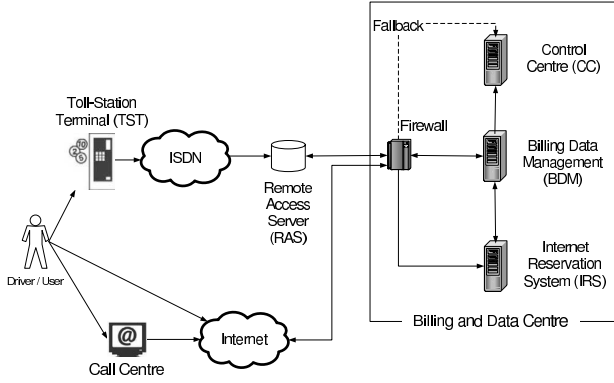
Fig. 2.　Terminal-based EFC system architecture

on the recovery process. After system recovery, terminals use a specific algorithm to deliver backlogged BDRs. In the autonomous mode, a terminal repeatedly tries to establish a connection to the BDM or CC in certain intervals, the so-called *AutoReconnectIntervals* (*ARI*). If the connection establishment succeeds the terminal sends gathered BDRs to the billing management (BDM or CC).

However, in order to avoid an overload situation the terminals should not send all BDRs at once. The algorithm currently applied is controlled via two parameters and only allows to send a certain number of BDRs (at least one) in FIFO order upon every successful connection establishment. Again, ARI is used for the interval between two connection attempts, while the number of BDRs, *n*, allowed to be sent during a connection is derived from the *AutoSendInterval* (*ASI*) according to the following equation:

$$n = \begin{cases} 1 & \text{ASI} < t_{\text{BDR}} \\ \frac{\text{ASI}}{t_{\text{BDR}}} & \text{ASI} \geq t_{\text{BDR}}. \end{cases} \quad (1)$$

Here, t_{BDR} denotes the time necessary to send one BDR message to the server system. New BDRs entering during the recovery from the autonomous mode are automatically stored locally in the terminal along with the already queued ones. After the last record was sent and acknowledged the terminal returns to the normal operational mode.

As mentioned before, BDRs in the manual system can also be sent via the internet or a call centre. All messages are sent to the internet reservation system (*IRS*) from where the BDRs are immediately transferred to the BDM. In this case it is a completely online process, i.e., there is no autonomous mode in which records are stored locally if the backend system cannot be reached. This mode of operation implies that if the logon system (besides the IRS) cannot be contacted, no messages can be transferred. Therefore, the traffic induced by the

internet or call centre BDRs in the following can be modelled as an additional load to the BDM during times in which the backend is operational.

III. Modeling the manual logon environment

We modelled the manual logon environment by an open queueing network as shown in figure 3, which can be further subdivided into two parts: *frontend* and *backend*. The frontend comprises the terminals and the RAS, while the backend represents the BDM in the data centre.

The *m* terminals operate independently and are modelled as unbounded waiting queues with FIFO queueing discipline and arrival rate λ_i. Thus, in the frontend part the total request arrival rate is $\lambda = \sum_{i=1}^{m} \lambda_i$.

The RAS has a capacity of *s* ports and accepts up to 24 connection establishments per second, which we considered in our simulation studies by blocking excess requests. Due to the limited number of RAS ports the number of active connections between the terminals and the data centre is also limited.

The holding time of RAS ports T_{Hport} depends on the communication and processing delays inside the data centre. Particularly, it depends on the operation time T_{op} as terminals wait for an acknowledgement from the BDM. In figure 3, this dependence is symbolised by the dashed line. In detail, T_{Hport} is calculated as

$$T_{\text{Hport}} = T_{\text{op}} + T_{\text{followup}} + \\ \sum T_{\text{connect}} + \sum T_{\text{trans}}, \quad (2)$$

where $\sum T_{\text{connect}}$ is the sum of all connection establishment delays between the RAS and the BDM. T_{followup} is the follow-up time for which the connection will remain open after reception of an acknowledgement, e.g., to exchange network management information. $\sum T_{\text{trans}}$ corresponds to the sum of all transmission times of one BDR message and all other communication delays inside the data centre.

The backend part of the architecture represents the BDM which was modelled as a *M/D/k multi-server-delay-system*. This model is used to calculate the mean waiting time of a *k* server-delay-system under Poisson arrivals of rate λ_{BDM}. The *k* identical servers are assumed to have a deterministic service time *h*, i.e., $\rho = \frac{\lambda_{\text{BDM}} h}{k} \leq 1$.

The system parameters are fitted to the performance numbers provided by the specification, namely $\tilde{T}_{\text{op}}$ and the performance bound of a loaded running system. These numbers contain the mean waiting time $E[W]$ under a given BDM utilisation ρ. To estimate the values of the parameter *h* and *k*, we used the mean value analysis.
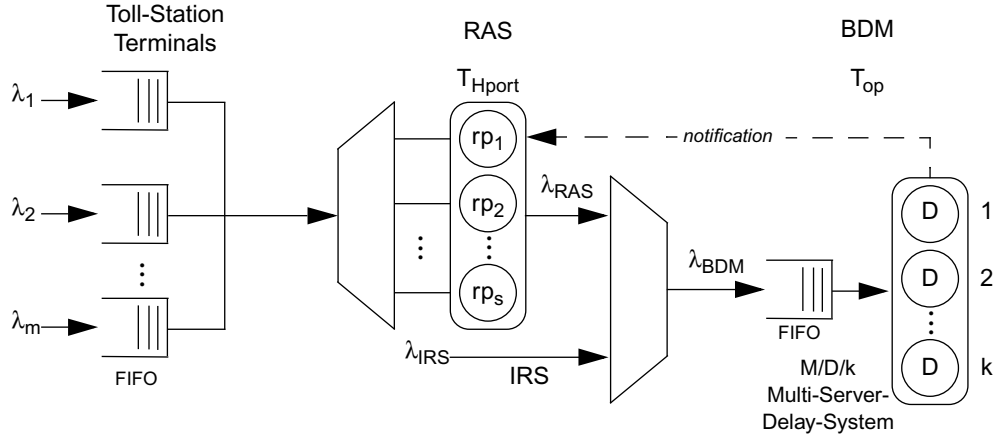
Fig. 3.   Open queueing network of the terminal-based EFC system architecture

We assumed a Markov arrival process to estimate the parameter *h* and *k*. This assumption aim only for the *normal operational mode*, not for the *autonomous mode*. We do not consider the use of an exact computation, because this might be complex and high precision is not required in the BDM environment.

The computation of the mean values in a $M/D/k$ multi-server-delay-system is not trivial. In the literature different approximations are available ([1], [5], [8], [11], [12], and [13]).

We applied the following approximation to derive the mean waiting time of the $M/D/k$ system $E[W_{M/D/k}]$ for the known mean waiting times of the $M/D/1$, $M/M/1$, and $M/M/k$ pure-delay-systems ([7], [9], and [10]):

$$\frac{E[W_{M/M/1}]}{E[W_{M/M/k}]} \approx \frac{E[W_{M/D/1}]}{E[W_{M/D/k}]}, \qquad (3)$$

$$E[W_{M/D/k}] \approx \frac{\frac{\rho}{2(1-\rho)}}{\frac{\rho}{(1-\rho)}} * E[W_{M/M/k}], \qquad (4)$$

$$E[W_{M/D/k}] \approx \frac{1}{2} * E[W_{M/M/k}], \qquad (5)$$

with

$$E[W_{M/M/k}] = \frac{\mu\rho^k}{(k-1)!(k\mu - \lambda_{\mathrm{BDM}})^2} \qquad (6)$$

$$* \left(\sum_{j=0}^{k-1} \frac{\rho^j}{j!} + \frac{\rho^k}{k!} \frac{k\mu}{k\mu - \lambda_{\mathrm{BDM}}} \right)^{-1},$$

and $\mu = \frac{1}{h}$.

In the case of ligth utilisation $\rho \leq 0,5$ the relative variations from the exact solution are high. But the absolute variations are small due to the fact that the values

are small $\leq 0,1721s$ [8]. With increasing utilisation our approximation becomes closer to the exact solution.

With this formula the system parameters *h* and *k* are fitted to the performance numbers provided by the specification with the least-square-method under consideration of the stationary condition. This estimate is approached by determining the values of the system parameters, so the sum of the squares of deviations between the given values $\tilde{T}_{\mathrm{op}}(\rho)$ and the fitted function $E[T_{\mathrm{op}}]$ is a minimum. Therefore:

$$\Delta(h,k) = min\left\{ \sum \left(\tilde{T}_{\mathrm{op}} - E[T_{\mathrm{op}}] \right)^2 \right\}, \qquad (7)$$

with the mean sojourn time

$$E[T_{\mathrm{op}}] = h + E[W_{\mathrm{M/D/k}}]. \qquad (8)$$

In our case we have more than one result for the system parameters. We choose the parameters to $h = 2s$ and $k = 50$.

Above-mentioned a detailed system model of the terminal-based EFC system architecture was presented. With this model a performance evaluation of the process of delivering booking data records after system recovery can be accomplished.

IV. PERFORMANCE EVALUATION

The model was implemented and simulated with the time discrete simulation environment ns-2 [14]. In the simulation each ISDN channel was modelled with a user data rate of 64 kbps and a delay of 50 ms. Over these channels, i.e. the 100 Mbps Ethernet connection between the RAS and the BDM, as well as between IRS and BDM, TCP is parametrised with a maximum segment size of 1500 bytes. The steady-state simulation

starts at the point in time right after the failed system is recovering.

In this paper only the failure and recovery of the BDM is considered, since other communication failures of e.g. the RAS will lead to similar situations. As we were mostly interested in longer downtimes, most terminals has reached the autonomous mode during the failure phase of the BDM. By periodically attempting to setup connections, they probe the system until they detect its recovery. The time after which a terminal identifies system recovery corresponds to the forward recurrence time of the constant ARI and is thus uniformly distributed between zero and ARI.

All following results are based on simulations with $m = 3600$ terminals and $s = 1920$ RAS ports. The downtime T_{off} of the BDM was selected to be 30 minutes. The limit T_{ack} for server timeout is 35s, the follow-up time T_{followup} until the terminal closes the ISDN connection is 45s.

In the analysed scenario the terminals are divided into 4 different classes with regard to their utilisation as table I shows. This is motivated by the fact that terminal usage is quite inhomogenous across Germany and the neighbouring regions. Terminals and IRS are triggered individually by their arrival distributions both creating traffic at the BDM. The arrival rate of messages (BDRs) at the terminals as well as from the IRS is modelled as a Poisson process.

A. The booking traffic profile

The traffic characteristics of connection requests to the RAS as well as of BDR transmissions to the BDM provide important information for system analysis and dimensioning. The BDR rate offered to the BDM describes the utilisation of the system in the backend. In the simulations a downtime of 30 minutes was assumed, which leads to the respective configuration of the terminals shown in table I. This means that at the beginning of the recovery phase approximately 13.536 BDRs have been stored during the downtime of the BDM.

Class	1	2	3	4
Percentage of terminals	5	40	45	10
$1/\lambda_i$ [s]	180	300	1000	3600
average BDR count after 30 min downtime	10	6	1.8	0.5

TABLE I

CONFIGURATION OF THE TERMINALS

Figures 4 and 5 show the utilisation of the BDM with different values for the reconnect interval parameter ARI and for different values of ASI. The default configuration of the terminals is $ARI = 150$ and $ASI = 0$.

In case where $ASI = 0$ (see figure 4) it can be seen that with the start of the recovery process the number of BDRs processed at the BDM varies largely and stabilises further on a rather large value between 20 and 24 $\frac{\text{BDR}}{s}$ before dropping to approximately 5 $\frac{\text{BDR}}{s}$. The larger amplitude for $ARI = 150$ indicates that the utilisation of the BDM or the RAS is not optimal at this point, therefore smaller values of ARI where choosen. With $ARI = 60$ the large variation in amplitude values diminishes and the average utilisation of the BDM increases significantly. Also the point in time where the BDR rate decreases vastly changes from approximately 2000s to 1000s. This behaviour can be seen with all decreasing steps of ARI values ($ARI = 150, 100, 80, 60$) though there is no significant increase in the utilisation of the BDM and decrease of recovery time beyond $ARI = 100$. This is mainly due to the fact that the connection establishments at the RAS are limited to $24 \frac{\text{connection}}{s}$. This means that with a lower ARI the usage of the RAS and the overall BDR rate, that can be processed at the BDM without an overload situation, has increased.

Though it has to be considered that a further decrease of the ARI and hence an increase in RAS load will lead to an unacceptable number of ISDN connection attempts.

In case where $ASI = 5$ (see figure 5), the behaviour differs as the RAS utilisation does not dominate the BDM utilisation anymore. The throughput of BDRs is much higher because the time T_{followup} is only applied once for each connection and therefore the BDM utilisation is much higher as every terminal can send up to 10 BDRs per connection.

Considering a limited processing power of the BDM through the configuration of the $M/D/k$ system a further increase of ASI would lead into an overload situation at the BDM and hence into longer waiting times for each individual BDR in the $M/D/k$ FIFO-queue.

B. Influence of the logon parameters

The duration T_{recovery}, after which a terminal returns to the normal mode, significantly depends on the choice of the terminal parameters ARI and ASI (n respectively, see equation 1).

Figure 6 depicts the impact of the ARI and ASI parameter setting on the recovery time T_{recovery}. It can be seen that the default $ARI = 150$ and $ASI = 0$ lead to a rather long recovery time of approximately $4900s$ until all backlogged BDRs are processed. Making the
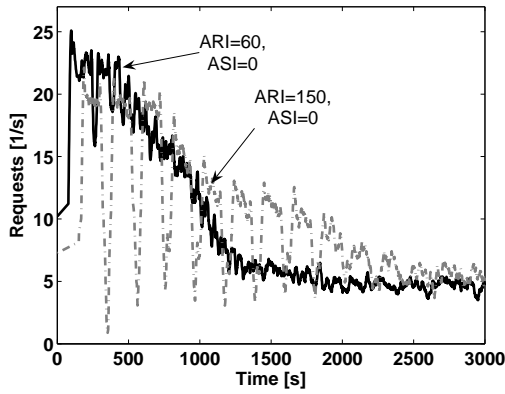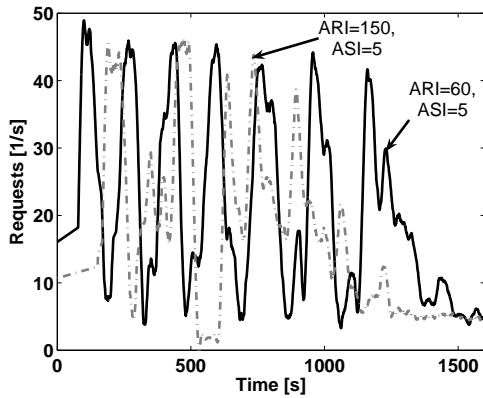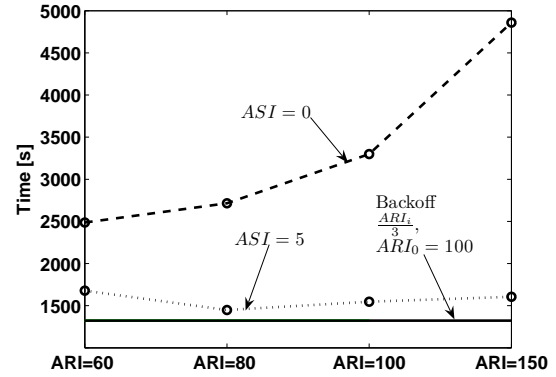
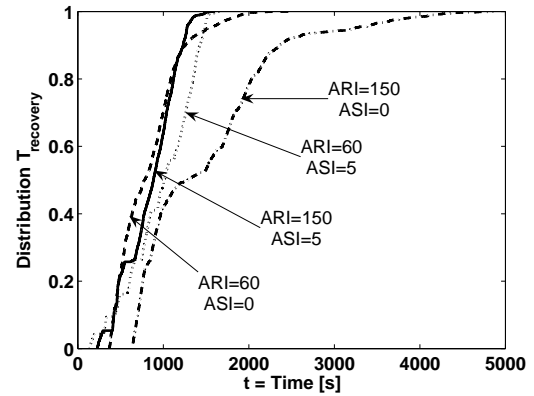Fig. 4.   BDM utilisation for different ARI values

Fig. 6.   Comparison of recovery times T_{recovery}

Fig. 5.   BDM utilisation for different ARI values

Fig. 7.   Comparison of CDF of recovery times T_{recovery} (n=10)

algorithm more aggressive by reducing the ARI value obviously reduces the recovery time to a value of $2500s$. However, as mentioned before, reducing the ARI value alone increases the number of connection setup attempts of each terminal and thus increases the load on the RAS server. Also, as the follow-up time T_{followup} before a connection can be terminated is comparably high, improvements to the recovery time T_{recovery} by shorter ARI values are limited.

Thus, alternatively or in addition, improvements can be expected by increasing the number of BDRs sent during one connection, i.e., the follow-up time is amortised by sending n BDRs. Figure 6 demonstrates this effect of transmitting more BDRs during a connection. With a value of $ASI = 5$, the system can recover much faster and reaches the normal operational mode earlier than with $ASI = 0$. The influence of the ARI parameter is neglegtible in this scenario and does not decrease the recovery time considerably.

Comparing the different cumulative distribution functions (CDF) in figure 7, regarding the variation of ARI and ASI values, it can be seen that the largest gain in overall performance yields from decreasing the ARI value. Whereas decreasing the ASI value leads to the best recovery time T_{recovery} for the terminals.

Summarising, a careful optimisation of ARI and ASI values as well as of system dimensioning is needed to obtain an overall efficient and stable system.

C. The Backoff Algorithm

In this section we examine a dynamic algorithm of adjusting the ARI and ASI values for delivering booking data records after system recovery to further decrease the recovery time T_{recovery} and to keep the utilisation of the BDM at a balanced level of high utilisation. So far, the terminal parameters ARI and n (respectively ASI, see equation 1) were configured as fixed parameters and had the same values for all terminals. This approach is

deterministic and suboptimal for sending data records during the recovery process. The aim is to minimise the duration T_{recovery} and to optimize the overall BDM utilisation.

After a terminal switches into the autonomous mode an initial $ARI_0 = 150s$ ($ARI_0 = 100s$, $ARI_0 = 50s$ respectively) and $n_0 = 1$ will be choosen.

The ARI and n increase if the terminal establishes a connection to the RAS and delivers backlogged BDRs successfully by the following equations:

$$ARI_{i+1} = ARI_0 + \frac{ARI_i}{a}, \tag{9}$$

and

$$n_{i+1} = \begin{cases} n_i * 2 & \text{if } 2 * n_i \leq q_l, \\ q_l & \text{else.} \end{cases} \tag{10}$$

If the terminal cannot establish a connection to the BDM or cannot deliver backlogged BDRs ARI and n will be increased according the following equations:

$$ARI_{i+1} = \begin{cases} ARI_i - \frac{ARI_i}{a} & \text{if } ARI_i - \frac{ARI_i}{a} \geq ARI_0, \\ ARI_0 & \text{else.} \end{cases} \tag{11}$$

and

$$n_{i+1} = \begin{cases} \frac{n_i}{2} & \text{if } n_i \geq n_0 \\ n_o & \text{else.} \end{cases} \tag{12}$$

The parameter a in the adjustment of the ARI value determines the aggressiveness of the approach regarding the change of ARI values. As can be seen in figure 8 an initial value of $a = 2$ and $ARI_0 = 150s$ resulted only in a slight improvement regarding the already achieved results in the previous chapter. However with setting $a = 3$ and $ARI_0 = 100s$ the behaviour of the alternative backoff algorithm showed major improvements. A further variation of $a = 4$ or alternatively a variation of $ARI_0 = 50$ yielded into an increase of the revocery time T_{recovery}, so the setting of $a = 3$ and $ARI_0 = 100$ was choosen to be optimal for the backoff algorithm. With this parameter setting an equilibrium was reached between the aggressiveness of the change of values and the decrease of recovery time T_{recovery}. The direct comparison of all relevant revocery times T_{recovery} can be seen in figure 6.

Regarding the utilisation of the BDM, the backoff algorithm showed a better usage of the available ressources which can be seen in figure 9. Since the capacity of the BDM was modelled as a system with limited ressources by the aforementioned $M/D/k$ system the backoff algorithm overcomes the initial limitation induced by the behaviour of the RAS. Furthermore the algorithm showed a better overall utilisation of the BDM compared to the best case with fixed $ARI = 80$ and $ASI = 5$. The peak utilisation of the BDM of the backoff algorithm is
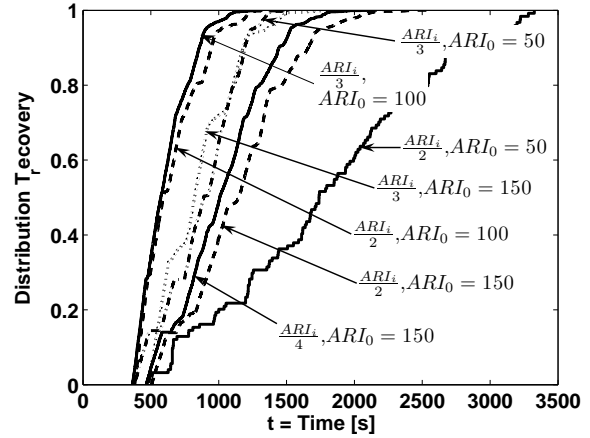

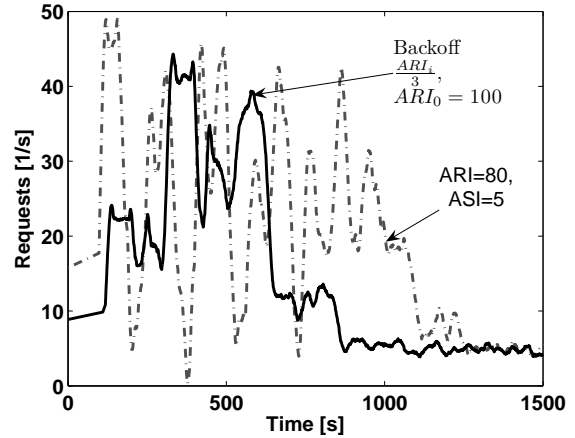Fig. 8. CDF of recovery time T_{recovery} with backoff algorithm

Fig. 9. Comparison of BDM utilisation backoff/ARI=80,ASI=5

approximately 10% lower and the average load during the recovery with backoff algorithm is $24.2\frac{\text{BDR}}{s}$ compared to $19.1\frac{\text{BDR}}{s}$.

V. CONCLUSION AND OUTLOOK

In this paper, we modelled the toll-station terminal logon process in an electronic fee collection system as used in the German toll collection system. We described the German toll collection system and presented, for the first time, a detailed system model.

In a performance evaluation we quantified the impact of the backlog resolution algorithm and parameters used to deliver data records, stored in the terminals during a downtime of the central data centre. We showed the positive impact of reducing the auto reconnect interval, i.e., making the algorithm more aggressive. Also, we

illustrated the trade-off of increasing the number of data records delivered during one connection attempt with respect to the speed of backlog resolution and request load on the servers in the data centre.

As was shown, with the backoff alogrithm further improvements regarding the recovery time were achieved by applying a dynamic approach. A modification of the backoff algorithm with random choice of the parameters ARI and ASI in a certain intervall could be researched.

Acknowledgement

References

[1] O. J. Boxma, J. W. Cohen, and H. Huffels, "Approximations of the Mean Waiting Time in an M/G/s Queueing System," *Operations Research*, vol. 27, no. 6, pp. 1115–1127, 1979.

[2] Bundesamt für Güterverkehr, "Marktbeobachtung Güterverkehr - Sonderbereicht über die Auswirkungen der Streckenbezogenen LKW-Maut," Published by Bundesamt für Güterverkehr (BAG), September 2005, in German language.

[3] I. Catling, "Innovative techniques for electronic road pricing," in *Electronic Tolling and Congestion Charging (Ref. No. 1999/092), IEE Seminar*, 1999, pp. 8/1–8/4.

[4] CEN TC278/ISO 17575, "Road Transport and Traffic Telematics (RTTT), Application requirements for Electronic Fee Collection (EFC) Systems based on GNSS/CN," CE/ISO-Norm, 2004.

[5] C. D. Crommelin, "Delay probability formulae," *The Post office electrical engineers' journal*, vol. 26, pp. 266–274, 1933.

[6] DaimlerChrysler Services, "Systemdescription ETC Germany," Published by DaimlerChrysler Services / Mobility Management, July 2003, in German language.

[7] G. Fiche and G. Hébuterne, *Communicating Systems & Networks: Traffic & Performance*. Kogan Page Science, 2004.

[8] P. J. Kuehn, *Tables on Delay Systems*. University of Stuttgart, Institute of Switching and Data Technics, 1976.

[9] ——, "[personal communication]," December 2005.

[10] L. Kleinrock, *Queueing Systems - Volume 1: Theory*. Wiley-Interscience, 1975.

[11] S. A. Nozaki and S. M. Ross, "Approximations in finite capacity multiserver queues with poisson arrivals," *Journal of Applied Probability*, vol. 15, no. 4, pp. 826–834, 1978.

[12] L. P. Seelen, H. C. Tijms, and M. H. van Hoorn, *Tables for Multi-Server Queues*. Elsevier Science Publishers B.V., 1985.

[13] H. Tijms, "New and old results for the M/D/c queue," *AEU - International Journal of Electronics and Communications*, vol. 60, no. 2, pp. 125–130, February 2006.

[14] University of California, "The network simulator – ns-2," http://www.isi.edu/nsnam/ns/, Berkely, 2005.

[15] J. Zeidler, S. Tacke, and L. Burgstahler, "Characterization of the traffic source behaviour of a GNSS/CN electronic fee collection system," *Proceedings of the 19th International Teletraffic Congress-ITC19, Beijing, China*, vol. 6a, pp. 697–706, August 2005.