Using a Backup Coordinator to Compensate for Blocking of Atomic Transactions in MANETs

Joos-Hendrik Böse, Jürgen Broß

Using a Backup Coordinator to Compensate for Blocking of Atomic Transactions in MANETs

Joos-Hendrik Böse and Jürgen Broß

Freie Universität Berlin, Institute for Computer Science
Takustr. 9, 14195 Berlin, Germany
{joos.boese,juergen.bross}@fu-berlin.de

Abstract. We present a probabilistic model to evaluate the reliability of atomic commit for distributed transactions in mobile ad-hoc networks (MANETs). Our model covers arbitrary MANET scenarios as well as strict and semantic transaction models. We evaluate the approach to integrate a backup coordinator to reduce blocking risks. For an example MANET scenario we show that the considered blocking probability is very low.

Keywords: Atomicity, Mobile Ad-Hoc Networks, Transactions

1 Introduction

To provide for robustness and reliability of applications deployed to volatile environments such as MANETs, transaction processing is a key concept. Atomic transactions guarantee consistency of data and system states. Guaranteeing atomicity of a distributed transaction requires agreement among transaction participants on the outcome of the transaction. This is typically achieved by an atomic commit protocol (ACP). It is generally known that in the presence of node or communication failures such protocols cannot avoid blocking [10]. While in fixed networks such situations are rare due to low probabilities of site and communication failures, MANETs are a more challenging environment.

Generally, a blocking situation arises when participants can no longer terminate their transaction branch independently, but are forced to wait until they can learn about the global transaction decision. Compared to ACPs tailored to MANETs like [5, 6], the use of a backup coordinator (BC) [8] is a more lightweight strategy to compensate for blocking. However, it is unclear whether the use of a BC is generally beneficial in a MANET scenario, as it introduces an additional source of failure. In this article we provide an in-depth analysis of the BC scheme for MANETs. We present a calculation model to answer the question whether blocking is a relevant problem in a specific MANET and transaction scenario and thus may require use of a BC. Additionally, the model then allows to predict to which degree blocking is reduced.

2 System and Transaction Model

2.1 System and Failure Model

A MANET $\mathcal{A}$ is established between nodes located in a specific area. Due to potential node and communication failures, we do not assume that $\mathcal{A}$ is fully connected. For each node a chance exists to completely leave the area and thus to disconnect

from $\mathcal{A}$. We describe the probability for this event to happen until time t by the cumulative distribution function (cdf) $F_L(t)$. We assume that a multi-hop routing protocol, such as AODV or DSDV, is used. Although message delays in $\mathcal{A}$ depend on the hop count of communication paths, for sake of simplicity, we assume an average message delay δ_m for all messages.

Communication characteristics of $\mathcal{A}$ are captured by the cdf $F_c(t)$, describing the probability that a communication path breaks until time t. In the following we refer to this time as *path duration*. The according probability density function (pdf) is denoted by $f_c(t)$. Note that in this article we do not consider the case that a link may recover. In addition to communication failures, a node may suffer from exhausted energy resources or general technical failures. For these events we assume cdfs $F_E(t)$ and $F_T(t)$. Given the assumptions above, the proposed probabilistic failure model covers the following two types of failures:

Node Failures denote all events that cause a node to disconnect from $\mathcal{A}$. Hence, $F_N(t)$, the cdf of a node failure until time t, is given by the probability that a node leaves $\mathcal{A}$, exhibits exhausted energy resources or suffers from a technical failure until t. $F_N(t)$ is calculated by considering complementary probabilities: $F_N(t) = 1 - ((1 - F_L(t))(1 - F_E(t))(1 - F_T(t)))$

Communication Failures cause the break of a communication path, that was functional before. The failure of the link is induced by mobility or by node failures of relaying nodes. The cdf for a communication failure is denoted by $F_C(t)$. We show in Section 3 that $F_C(t)$ depends on the initial hop distance of communication partners.

By $F(t)$ we denote the cdf for the general failure that either a communication or a node failure occurs until t.

2.2 Distributed Transaction Models

The basic model we consider is the flat ACID transaction model. We assume that ACID is guaranteed locally on every node and transactions are only aborted due to node or communication failures. Following the X/Open DTP model [11] a transaction consists of a set of operations that are issued by an *application*. All operations received by a *participant* constitute a local transaction branch of the global transaction. To avoid the need for initially choosing a *coordinator*, we assume that the application process and the transaction coordinator are co-located. In order to detect node and communication failures each execution of an operation is acknowledged by the participant. If an acknowledgment is not received during timeout Δ_{op}, the transaction is globally aborted.

We distinguish between *processing* and *decision phase* of a distributed transaction. The processing phase begins at time t_s, when the transaction is initiated and ends at time t_p, when the acknowledgment of the last operation of the global transaction is received by the coordinator.

A participant i receives the last operation of its transaction branch at some random time t_o. For each participant, the random variable t_o is distributed within the interval $[t_s, t_p]$ according to the cdf $O(t_o)$. Here, we assume $O(t_o)$ to be uniformly distributed. If no failure is detected during $[t_s, t_p]$, the decision phase is initiated by starting an ACP at time t_p. The strict and semantic transaction model are based on this general model.

Strict Transaction Model The strict transaction model requires that *strict atomicity* as for example defined in [2] is guaranteed. We assume that this is achieved by the use of the well-known two-phase commit protocol (2PC) [11]. Due to its popularity, we omit a detailed presentation of the protocol here.

Blocking Risk with Strict Atomicity In 2PC a participant enters its so called *window of uncertainty* if it answers a commit-request of the coordinator with a positive vote. During this period the participant cannot autonomously proceed until it learns the global decision (i.e. it is blocked).

The decision phase begins at time t_p (i.e. the ACP is started at t_p). In fact, we assume that all *commit-request* messages are sent at t_p. The coordinator awaits a timeout of Δ_{vo} for the participant's votes. Hence, the length of the critical window ΔU, when a participant is vulnerable to a coordinator's node failure, is at minimum $\Delta U_{min} = 2\delta_m$ and at maximum $\Delta U_{max} = 2\delta_m + \Delta_{vo}$. In case that no undetected participant failure has occurred, ΔU has length ΔU_{min}. Otherwise the coordinator has to await timeout Δ_{vo}, so that ΔU increases to ΔU_{max} .

In the following we discuss how a BC can be integrated into 2PC. We present the basic idea and the additional blocking risk that is introduced by the use of a BC. For a more detailed description we refer to [8].

Integration of a Backup Coordinator The *BC* is integrated into 2PC between *commit-request* and *commit* phase. If the main coordinator (*MC*) learns that all participants voted *yes*, it sends a *decided_ commit* message to the BC. The *BC* saves the decision to stable log and acknowledges with a *recorded_ commit* message. Then the *MC* initiates the commit phase of 2PC. If the *MC* learns that the transaction needs to be aborted, it directly informs the participants without contacting the *BC*. The *BC* possesses a veto right to abort the transaction until it writes a commit decision to its log. In case a participant is blocked, it contacts the *BC*. Depending on its local state, the *BC* either answers with a *commit* message or makes use of its veto right and aborts the transaction. If the *BC* has aborted the transaction it answers the *decided_ commit* request of the *MC* with an *aborted* message. In effect transaction commit requires both coordinators, while the decision to abort a transaction can be made autonomously by one of the coordinators.

The added redundancy of coordinators is assumed to lower the risk of blocking due to a node failure, but it also introduces a new blocking situation: By issuing a *decided_ commit* message, the *MC* hands over transaction control to the *BC*. Until a *recorded_ decision* or *aborted* message is received, the *MC* remains uncertain and cannot conclude the transaction. To reduce this risk, the *MC* may perform a *reachability test* at t_p by pinging the BC before issuing a *decided_ commit* message. If it is unsuccessful, the transaction is aborted.

We distinguish between *early* and *late selection* of the *BC*. In *early selection*, the *BC* is chosen with transaction initiation, while with the *late selection* it is chosen right before t_p and participants learn about the *BC* with the commit-request messages of the coordinator. In either case, participants know about the *BC before* they become uncertain. A reachability test is only required in case of early *BC* selection.

Semantic Transaction Model In the semantic model a local transaction branch is committed as soon its last operation is successfully processed. This weaker atom-

icity notion called *semantic atomicity* [7] is maintained by means of compensating transactions, which semantically undo the effects of an already committed branch. An associated compensation transaction has to be executed in the case that a local commit conflicts with the global decision.

Hence, as long as a participant is uncertain about the global decision it has to maintain conditions that allow for compensation. A general problem here is that effects of the committed branch are visible to other transactions and compensation must consider these so called *dependent transactions*. E.g., the correctness criterion *soundness* introduced in [7] may require dependent transaction to be rejected, thus hindering the overall progress. Another negative effect is that uncertainty about the global decision possibly leads to non-optimal behavior. For instance, in trading applications, a participant that is uncertain about the global decision cannot use money or goods affected by the transaction in doubt, in other transactions.

Blocking Risk with Semantic Atomicity Within the semantic model, processing and decision phase are not strictly separable: While in strict atomicity ΔU is the same for all participants, with semantic atomicity ΔU is individual for each participant. This is due to the fact that a participant i commits its local transaction branch right after successfully executing its last operation at time t_o, moving into uncertainty afterward. We assume that participants know which operation is the last one. The ack of this operation is an implicit *yes* vote to the coordinator. The coordinator later derives the global decision, without requiring an additional vote from this participant. In this model we define that the processing phase ends at t'_p, which is the time the coordinator sends the last operation to the last participant denoted by PA_{last}. The coordinator derives the global decision at $t_u = t'_p + 2\delta_m + \Delta_{ex}$, where Δ_{ex} is a constant time required by PA_{last} for the execution of the last operation.

The size of the uncertainty window is generally wider with semantic atomicity. If no participant failure occurs, the individual uncertainty period begins at $t_{i,o}$ and ends with $t_u + \delta_m$. If a participant failure is detected at t_f, two cases have to be distinguished: In case $t_{i,o} \leq t_f$, the phase of uncertainty is described by the interval $[t_{i,o}, t_f + \delta_m]$. For $t_{i,o} > t_f$ the participant is never uncertain, as it receives the coordinator's decision before moving into uncertainty (abort during processing phase). In semantic atomicity a participant does not block as in strict atomicity. Analogous to blocking in strict atomicity, we denote this situation as *extended uncertainty*.

Note that semantic atomicity only allows for early selection of the *BC*, as no explicit commit-request messages are sent.

3 Failure Probabilities

While for the calculation model presented in Section 4 it is non-relevant how failure probabilities are derived, the main intention of this section is to prove that derivation of the probabilities assumed in the system model is generally possible for MANETs.

We demonstrate their derivation for an example MANET scenario. In this scenario we assume that 15 mobile nodes move within a city area of 500x500m according to the Random Waypoint (RWP) mobility model at 2.0–5.0mps, relaying messages for each other using AODV, while the transmission range of nodes is 100m. Nodes use an electronic currency to trade electronic goods such as mp3 files.

The exchange of goods and money has to happen in a transactional manner. Batteries of nodes are assumed to deliver 2h of service, while each node is expected to remain within the area for 30min before moving away. The mean time to failure due to a technical failure is 500h.

It has been observed that the path duration (described by $F_C(t)$) is mainly influenced by the hop distance of nodes at transaction start (called *initial hop distance* (ihd)). We distinguish between ihd> 2 (*case 2+ ihd*) and ≤ 2 (*case 1-2 ihd*), because resulting path durations mainly influences the probability of transaction abort [3].

Probability of Node Failures Node failures cause a disconnection of a node from the MANET. In our example MANET setting a node's battery capacity is expected to hold for 2h and a technical failure is expected after 500h. Hence, $F_E(t)$ describing the remaining battery capacity of a random node is uniformly distributed in $[1s, 7200s]$. The probability for technical failures $F_T(t)$ is assumed to be exponentially distributed with $\lambda_T = 1/18 * 10^5$. The distribution of *sojourn times* $F_L(t)$ depends on the individual node behavior. For simplicity we assume an exponential distribution, here with parameter $\lambda_L = 1/1800$, as the expected sojourn time is 0.5h. The derived probability for a node failure is low for small periods, e.g. $F_N(10s) = 0.0069$. Communication failures are more likely in the scenario, as shown in the following.

Probability of Communication Failures The cdf $F_C(t)$ is primarily influenced by *node density*, *radio range*, *node mobility* and *routing scheme* of the given MANET scenario. $F_C(t)$ additionally depends on the hop count of the path at initiation time. Due to the complexity in modeling the numerous dependent events, most scholars propose a statistical analysis of this problem. E.g. in [9, 1] it is shown that for the common mobility models RWP, Manhattan and Freeway mobility the path duration in case 2+ ihd can be approximated by exponential distributions. In contrast to the work cited above, we explicitly address case 1-2 ihd. To derive $F_c(t)$ for both cases in our example MANET setting we did a simulation study using the ns2 network simulator and a statistical analysis of its results.

The results of [9, 1] towards path durations for case 2+ ihd could be confirmed by fitting an exponential distribution with $\lambda = 0.051$ to the measured data. However, for case 1-2 ihd path durations are better fit by a log-normal distribution, here with $\mu = 3.53$ and $\sigma = 0.67$. Thus, the important observation is that short after link initiation, for case 1-2 ihd a communication failure is unlikely, whereas exponentially distributed path durations always show the same failure rate. As average message delay δ_m we measured $180ms$ in the example scenario. We refer to our technical report [3] for a detailed description of the simulation study.

4 Calculation Model and Results

In this section, we derive a calculation model that allows to estimate blocking probabilities that are caused by a node failure of the coordinator. The formulae we present have to be interpreted from a participant's perspective, i.e. they describe the probability for an individual participant to block. The individual participant is denoted by PA, the set of the other $n-1$ participants is denoted by PA_{other}.

4.1 Preliminary Considerations

In many cases we calculate the probability that an event happens during an interval $[t_1, t_2]$. In the following we use the notation $F(t_1..t_2)$ for this probability (with F being a probability distribution function). We generally neglect events that occur in intervals $[t_1, t_2]$ of size $\leq \delta_m$.

An important factor is the probability that a transaction enters the decision phase. The decision phase is not entered, if the transaction is aborted before, because the coordinator detects a participant's failure. A participant's failure is only detected if it occurs in $[t_s, t_o]$, since then the coordinator would observe a missing ack. The probability $P_{o>f}(t_p)$ denotes that a participant's failure is detected within interval $[t_s, t_p]$ and is given by $P_{o>f}(t_p) = \int_0^{t_p} \int_0^{t_o} o(t_o) f(t_f)\, dt_f dt_o$ (with $t_s = 0$). The subscript $_{o>f}$ indicates that this probability only considers failures that happen before acknowledgment of the last operation. Analogous the probability that a failure is not detected, is calculated by $P_{o<f}(t_p) = \int_0^{t_p} \int_{t_o}^{t_p} o(t_o) f(t_f)\, dt_f dt_o$.

4.2 Strict Atomicity

In the following we derive the blocking probability with and without use of a BC. For lack of space, the calculations of blocking risk with a single coordinator are described only briefly. A more detailed description is presented in [4, 3].

The basic idea to derive the blocking risk with a single coordinator (denoted $P_u(t_p)$) is to calculate the probability that an uncertainty window either of size ΔU_{max} or ΔU_{min} is entered (the probabilities are called $P_{Umax}(t_p)$ and $P_{Umin}(t_p)$ respectively) and that the coordinator experiences a node failure during this period. For a transaction with n participants $P_{Umax}(t_p)$ is given by $P_{Umax}(t_p) = (1 - P_{o>f}(t_p))^{n-1} - (1 - F(t_p))^{n-1}$, while $P_{Umin}(t_p) = (1 - F(t_p))^{n-1}$. We denote the probability of a coordinator's node failure within interval $[t_p, t_p + \Delta U_{min}]$ by $C_{Umin}(t_p)$. It is given by $F_n(t_p..t_p + \Delta U_{min})$. $C_{Umax}(t_p)$ is defined analogously. The probability that PA does not encounter any failure until $t_p + \Delta U_{min}$, given by $1 - F(t_p + \Delta U_{min})$, is denoted by $PA_{Umin}(t_p)$. Analogously we define $PA_{Umax}(t_p)$. Blocking occurs, if the coordinator suffers from a node failure during ΔU.

Backup Coordinator with Early Selection Here, the BC is chosen at t_s as a node that lies within 1-2 hop distance from all participants and the coordinator. A reachability test is executed as described in Subsection 2.2. We distinguish between standard blocking situations due to a coordinator failure and the additional situation where the MC is alive, but blocked. Four cases may lead to a standard blocking situation:

(i) At least one unrecognized failure of a node in PA_{other} occurred until t_p. Then PA is blocked if MC and BC are unreachable at recovery time (the last point in time the global decision can arrive: $t_p + \Delta U_{max}$). As MC must have survived until t_p, only a node failure within $[t_p, t_p + \Delta U_{max}]$ needs to be considered. The link between BC and PA may have been broken within $[t_s, t_p + \Delta U_{max}]$. Then $BC1(t_p)$ describes the probability that PA blocks:

$$BC1(t_p) = P_{Umax}(t_p) * C_{Umax}(t_p) * F(t_p + \Delta U_{max}) \quad (1)$$

(ii) No node in PA_{other} fails, but BC encounters a communication failure with MC until t_p. In this case the reachability test fails and MC decides on abort at $t_p + \Delta U_{max}$. Participant PA is blocked if MC fails in $[t_p, t_p + \Delta U_{max}]$ and additionally, BC fails within $[t_p, t_p + \Delta U_{max}]$ or its link to PA brakes in $[t_s, t_p + \Delta U_{max}]$. This probability is given by:

$$BC2(t_p) = (1 - F(t_p))^{n-1} * F_c(t_p) * C_{Umax}(t_p) \\ * [F_n(t_p..t_p + \Delta U_{max}) + F_c(t_p + \Delta U_{max})] \quad (2)$$

(iii) No node in PA_{other} fails, but BC encounters a node failure until t_p. Again, the reachability test fails and the transaction is aborted. A node failure of MC leads to blocking of PA, as the BC has already failed and is not reachable:

$$BC3(t_p) = (1 - F(t_p))^{n-1} * F_n(t_p) * C_{Umax}(t_p) \quad (3)$$

(iv) Neither a node in PA_{other} nor the BC fails until t_p. A coordinator's node failure in $[t_p, t_p + 2\delta_m]$ leads to a blocking situation in the case that also BC has suffered from a failure in this interval given by $BC4(t_p)$:

$$BC4(t_p) = (1 - F(t_p))^{n-1} * F_n(t_p..t_p + 2\delta_m) * F(t_p..t_p + 2\delta_m) \quad (4)$$

The MC is blocked, if the *recorded_ decision* message of the BC is awaited, but not received. A presumption for this to happen, is that no failure of any node has occurred until t_p. We distinguish two cases that actually lead to blocking of the MC. In the first case A, the BC encounters a node failure after sending the ack of the reachability test and before sending its *recorded_ decision* message ($[t_p + \delta_m, t_p + 3\delta_m]$). Then, the MC gets blocked and the BC is not reachable for PA. In case B, the BC suffers from a communication failure with MC in $[t_p + 2\delta_m, t_p + 4\delta_m]$. If additionally the communication link between BC and PA breaks until $t_p + \Delta U_{max}$, MC is blocked and PA cannot reach BC. Since A and B are not independent, probability has to be computed as $P(A) + P(B) - P(A) * P(B)$:

$$BC5(t_p) = (1 - F(t_p))^{n-1} * (1 - F_n(t_p + \Delta U_{max})) \\ * \Big[F_n(t_p + \delta_m..t_p + 3\delta_m) + F_c(t_p + 2\delta_m..t_p + 4\delta_m) \\ * F_c(t_p + \Delta U_{max}) - F_n(t_p + \delta_m..t_p + 3\delta_m) \\ * F_c(t_p + 2\delta_m..t_p + 4\delta_m) * F_c(t_p + \Delta U_{max}) \Big] \quad (5)$$

The probability of blocking with early selection of a BC is then given by:

$$P_{u,BCe}(t_p) = PA_{Umax}(t_p) * \Big[BC1(t_p) + BC2(t_p) + BC3(t_p) \\ + BC5(t_p) \Big] + PA_{Umin}(t_p) * BC4(t_p) \quad (6)$$

Backup Coordinator with Late Selection Here, the BC is chosen at t_p, thus a working communication path is now required to last for ΔU only. The increased probability that BC and PA can communicate has the greatest influence on the end result. The computation of the blocking probability is analogous to 4.2. Cases (ii) and (iii) from above are not considered, as the BC is guaranteed to be available at t_p. In the other formulae, intervals have to be adjusted, as no reachability test

is executed. Line (1) of the following formula considers case (i) from above, line (2) considers case (iv) and line (3–4) describe the situation when MC is blocked:

$$\begin{aligned} P_{u,BCl}\ (t_p) = & PA_{Umax} * P_{Umax}(t_p) * C_{Umax}(t_p) * F(\Delta U_{max}) \\ & + PA_{Umin} * P_{Umin}(t_p) * F_n(t_p..\Delta U_{min}) * F(\Delta U_{min}) \\ & + PA_{Umax} * \Big[F_n(2\delta_m) + F_c(2\delta_m) * F_c(4\delta_m) \\ & - F_n(2\delta_m) * F_c(2\delta_m) * F_c(4\delta_m) \Big] \end{aligned} \tag{7}$$

Figure 1(a) compares the blocking probabilities as computed by $P_u(t_p)$, $P_{u,BCe}(t_p)$ and $P_{u,BCl}(t_p)$ for our example scenario with $n = 3$ and $\Delta_{vo} = 1s$. We assume only transactions with 1–2 ihd (all $F_c(t)$ are log-normal distributed), as the abort rate for 2+ ihd is unacceptable [3]. The important observation is that the risk of blocking caused by a coordinator's node failure is generally low in our scenario ($P_u(t_p)$<0.0003). The scheme with early BC selection ($P_{u,BCe}(t_p)$) shows to be even more susceptible for blocking than a single coordinator. Only for very short transactions with $t_p < 9s$ a reduction of the blocking risk is observed. However, this reduction is explained by an increased abort rate caused by unsuccessful reachability tests. For $t_p > 9s$ the risk of the additional blocking situations overcompensates the benefit of the increased coordinators availability. In case late integration of the BC is used, the blocking risk is significantly reduced for $5s < t_p < 60s$ compared to a single coordinator.

4.3 Semantic Atomicity

We examine the probability of extended uncertainty from the perspective of PA that is not PA_{last}. The probability of PA to suffer from extended uncertainty due to a node failure of the (single) coordinator is called $P'_u(t'_p)$. For lack of space we omit a detailed description of how $P'_u(t'_p)$ is derived and refer to [4, 3].

Backup Coordinator with Early Selection Extended uncertainty caused in $[t_s, t'_p]$ can be healed by the BC if it is reachable for PA at $t_u + \delta_m$. For this event the probability is given by:

$$BC'1(t'_p) = P'_u(t'_p) * F(t_u + \delta_m) \tag{8}$$

In $[t'_p, t_u]$, the reachability test has to be taken into account. The MC checks reachability of BC at t'_p. An answer is awaited until $t'_p + 2\delta_m + \Delta_{ex}$. Thus, at t_u MC knows the global decision as well if the reachability test was successful. To derive the desired probability for interval $[t'_p, t_u]$, multiple situations for both outcomes of the reachability test must be considered. Like in Subsection 4.2 we distinguish between standard extended uncertainty and the situation where the MC is reachable but blocked. We first consider the standard cases:

(i) If the reachability test fails, because the BC suffers from a node failure during $[t_s, t'_p + \delta_m]$, PA remains uncertain if the MC suffers from a node failure in interval $[t'_p, t_u + \delta_m]$:

$$BC'2(t'_p) = (1 - P_{o>f}(t'_p))^{n-1} * F_n(t'_p + \delta_m) * F_n(t'_p..t_u + \delta_m) \tag{9}$$

(ii) In case the reachability test fails due to a communication failure in $[t_s, t'_p + 2\delta_m]$, PA remains uncertain if MC suffers from a node failure in interval $[t'_p, t_u + \delta_m]$ and the BC is not reachable due to a node failure of BC in $[t'_p, t_u + \delta_m]$ or a communication failure in $[t_s, t_u + 2\delta_m]$:

$$BC'3(t'_p) = (1 - P_{o>f}(t'_p)^{n-1} * F_c(t'_p + 2\delta_m) * F_n(t'_p..t_u + \delta_m) \\ * \Big[F_n(t'_p..t_u + \delta_m) + F_c(t_u + 2\delta_m)\Big] \quad (10)$$

(iii) If no failure occurs until t'_p, PA remains uncertain, if MC suffers from a node failure in $[t'_p, t_u]$ and the BC is not reachable because of a node failure in $[t'_p, t_u]$ or a communication failure with PA in $[t_s, t_u + 2\delta_m]$:

$$BC'4(t'_p) = (1 - P_{o>f}(t'_p))^{n-1} * F_n(t'_p..t_u) * \Big[F_n(t'_p..t_u) + F_c(t_u + \delta_m)\Big] \quad (11)$$

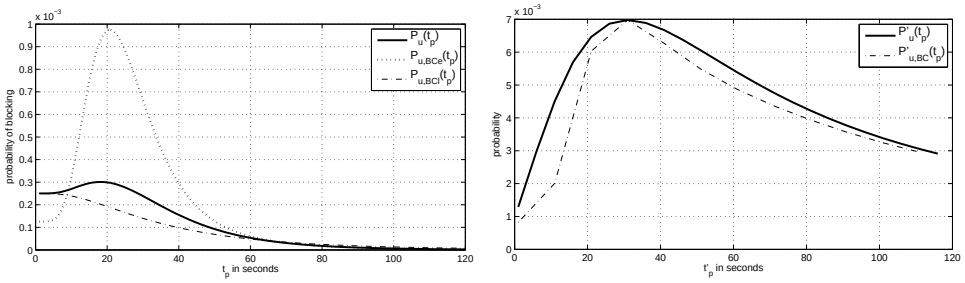
The situation, when PA remains uncertain although it can reach the MC, but the MC is blocked is derived by similar considerations that led to Formula (5) with different interval bounds. For lack of space we omit a detailed discussion of the derived Formula $BC'5(t'_p)$ here.

$$BC'5(t'_p) = (1 - P_{o>f}(t'_p))^{n-1} * [F_n(t'_p + \delta_m..t_u) + F_c(t'_p + 2\delta_m..t_u) \\ *F_c(t_u) - F_n(t'_p + \delta_m..t_u) * F_c(t'_p + 2\delta_m..t_u) * F_c(t_u)] \quad (12)$$

The probability for PA to remain uncertain if a BC is used is now given by:

$$P'_{u,BC}(t'_p) = BC'1(t'_p) + (1 - F(t_u)) * [BC'2(t'_p) + BC'3(t'_p) + BC'4(t'_p) + BC'5(t'_p)] \quad (13)$$

Figure 1(b) plots the reduction of blocking risk achieved by a BC in the semantic model. For the example scenario of this work, a BC only slightly reduces blocking risks. Like in Subsection 4.2, a log-normal cdf for all $F_C(t'_p)$ is assumed.



(a) Strict transaction model. (b) Semantic transaction model.

Fig. 1. Probability of blocking caused by a node failures of the coordinator.

5 Summary and Conclusion

In this article we presented an in-depth analysis of the use of a BC in MANETs. We developed a probabilistic model to predict the expected reduction of blocking risk in case a BC is used. Using the proposed model, we demonstrated the benefit of BC integration in transaction processing for an example MANET scenario. It showed that the time of BC selection is critical. In the strict model only late selection of the BC leads to a reduction of blocking risk, while early selection even increases

blocking probability. In the semantic model the risk for blocking is generally higher, as uncertainty periods of participants are larger. Here, a BC reduces the blocking risk only slightly for the example scenario.

We argue that for strict transactions early selection of a BC is generally not feasible, while for semantic transactions such a general statement cannot be made. The provided calculation allows to identify critical scenarios, where the blocking risk reaches considerably high values and the use of a recovery strategy such as a BC is indicated. However, we argue that for most standard MANET scenarios the discussed blocking risk is low and negligible for most applications.

In addition, our probabilistic model allows for easy integration of individual failure probabilities of BCs, i.e. using especially reliable BCs can be considered for scenarios where e.g. a base station can be used as BC.

References

1. F. Bai, N. Sadagopan, and A. Helmy. Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks, 2003.
2. Philip A. Bernstein, Vassco Hadzilacos, and Nathan Goodman. *Concurrency control and recovery in database systems.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1987.
3. Joos-Hendrik Böse. Abort and blocking risk of atomic transactions in mobile adhoc networks. Technical Report B-08-07, Freie Universität Berlin, ftp://ftp.inf.fu-berlin.de/pub/reports/tr-b-08-07.pdf, June 2008.
4. Joos-Hendrik Böse and Jürgen Broß. Predicting the blocking risk of atomic transactions in manets induced by coordinator failures. In *IADIS International Conference Wireless Applications and Computing.* (Accepted for publication), 2008.
5. Joos-Hendrik Böse, Stefan Böttcher, Le Gruenwald, Sebastian Obermeier, Heinz Schweppe, and Thorsten Steenweg. An integrated commit protocol for mobile network databases. In *IDEAS '05: Proceedings of the 9th International Database Engineering & Application Symposium (IDEAS'05)*, pages 244–250, Washington, DC, USA, 2005. IEEE Computer Society.
6. L. Gruenwald and S. Banik. A power-aware technique to manage real-time database transactions in mobile ad-hoc networks. September 2001.
7. Henry F. Korth, Eliezer Levy, and Abraham Silberschatz. A formal approach to recovery by compensating transactions. In Dennis McLeod, Ron Sacks-Davis, and Hans-Jörg Schek, editors, *16th International Conference on Very Large Data Bases*, pages 95–106. Morgan Kaufmann, 1990.
8. P.K. Reddy and M. Kitsuregawa. Reducing the Blocking in Two-Phase Commit Protocol Employing Backup Sites. *Proceedings of the 3rd IFCIS International Conference on Cooperative Information Systems*, pages 406–416, 1998.
9. N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy. Paths: analysis of path duration statistics and their impact on reactive manet routing protocols, 2003.
10. D. Skeen and M. Stonebraker. A formal model of crash recovery in a distributed system. *IEEE Trans. Softw. Eng.*, 9(3):219–228, 1983.
11. X/Open. Distributed transaction processing: Reference model, version 3. http://www.opengroup.org/bookstore/catalog/g504.htm, February 1996.