Performance Analysis of Location-Based Data Consistency Algorithms in Mobile Ad Hoc Networks

Ing-Ray Chen, Jeffery W. Wilson
Department of Computer Science
Virginia Tech
{irchen, wilsonjw}@vt.edu

Frank Driscoll, Karen Rigopoulos
The MITRE Corporation
{fdriscoll, karenr}@mitre.org

ABSTRACT: Many applications of mobile *ad hoc* networks require real-time data consistency among the moving nodes within a geographical area of interest to function correctly, e.g., battlefield command and control applications. While it is operationally desirable to maintain data consistency among nodes within a large geographical area, the time required to propagate state changes to all mobile nodes in that geographical area limits its size. This paper investigates the notion of *location-based data consistency* in mobile *ad hoc* networks, and analyzes the tradeoff between data consistency and timeliness of data exchange among nodes within a location-based group in a geographical area of interest. By utilizing a Petri net performance model, we analyze performance characteristics of location-based data consistency maintenance algorithms and identify design conditions under which the system can best tradeoff consistency for timeliness (reflecting the time to propagate a state change) while satisfying the imposed data consistency requirement, when given a set of parameters characterizing the application in the underlying mobile *ad hoc* network environment.

1. Introduction

Many applications operate in mobile *ad hoc* networking environments with no fixed infrastructure connecting mobile nodes in the field. Many of these applications require that nodes have some degree of data consistency within a community of interest for them to function correctly. An example is a battlefield application in which nodes within a geographical area of interest must have consistent data structures that contain kinematic and other characteristic state data that describes friendly, enemy, and neutral aerospace objects to satisfy the command and control functionality requirements. Such an *ad hoc* environment is characterized by mobile nodes, multi-hop routing, planned and unplanned node disconnection, node failure, relatively low communication system throughput, and unreliable communication. Node mobility is a particular challenge because mobility translates into multi-hop network topology changes, which are reflected in frequent packet route changes and network partitions. The physical environment exacerbates the challenges caused by node mobility; by nature of their mobility, nodes can be expected to exploit cultural and natural features of the physical environment (e.g., taking shelter in buildings, maneuvering around hills) that will have a deleterious effect on communication.

The problem we are addressing in this paper is analyzing performance characteristics of algorithms for maintaining "location-based data consistency" among a group of nodes in a mobile *ad hoc* environment for applications that require *location-based data consistency*, e.g., the DoD Single Air Integrated Picture (SIAP) project that requires all mobile nodes to maintain a consistent view of tracked objects for combat missions [2]. It is well known that the problem of reaching an agreement ("consensus") among all nodes in asynchronous distributed systems in the presence of failures is deterministically non-solvable even if communication is reliable and at most one peer may crash [3]. A less constrained problem, known as the group membership consistency for maintaining a "single agreed view" of the group membership among all peers, was also shown to be non-solvable in asynchronous distributed systems where communication is reliable and at most one peer may crash [1]. The notion of *location-based data consistency* considered in the paper does not require a "single agreed view" to be maintained (for which there is no solution). Rather, it allows mobile nodes to join and leave location-based groups, allowing multiple data views to coexist in different location-based groups as long as nodes within the same group have the same view of data. This requirement, although less strong in data consistency, is very useful for battlefield applications where data carry geographical meanings, e.g., tracking friend or foe objects entering, traveling and leaving a geographical area by all units in the area.

The general problem of group communication in mobile *ad hoc* wireless networks to maintain consistent group membership and, as an extension, to maintain data consistency among members of a group is a relatively new research area. Killijian et al. [5] introduced the definition of proximity group communication in which group membership depends on location. They associated each proximity group with a static or mobile area of interest within which the group members should be located. They gave a sketch of using a partition anticipator executed on every node to detect suspicious partitioning events of a proximity group due to node and link failures in order to take preventive actions for consistency group membership maintenance. However, no concrete mechanisms were given by which the partition anticipator may be implemented. Roman et al. [4, 7] use the idea of *safe distance* for implementing consistent group membership wherein membership is based on the location information of mobile nodes. The basic idea is to group nodes logically connected within a safe distance ("close enough") in a geographic area and to perform membership-change operations atomically as nodes move in and out of the geographic area. Physical connections that are *susceptible* to disconnection are considered as announced disconnections so the system can perform membership change in one indivisible operation to ensure group membership consistency. In this way, a group expands and contracts atomically, preserving consistent group membership. However, their safe distance-based algorithm is based on the somewhat unrealistic assumption that disconnections are only caused by node movements, so the algorithm breaks when disconnections are caused by node failures.

In this paper, we utilize the concept of partitionable group membership for achieving *location-based data consistency* in *ad hoc* systems so that each member in a location-based group has knowledge of other members in its group and that such knowledge would be consistent across the entire group [4, 7]. However, instead of maintaining membership consistency *all the time* by using the concept of safe distance, we allow temporary inconsistency to exist during membership changes to tradeoff consistency for timeliness. The degree of inconsistency is bounded by the way we check and perform membership changes as a result of nodes leaving and joining location-based groups, the rate of which can be adjusted to satisfy the data consistency requirement of the application.

One key design issue for maintaining *location-based data consistency* within a geographical community of interest is to determine the “logical size” of each geographical area. The size directly impacts the time taken for achieving data consistency among members within a geographical area, thus reflecting the bound on data consistency. The optimal size is affected by environmental conditions characterized by mobility rate (for mobile units in and out of an area), node failure rate, membership-change detection rate, data change rate (e.g., for tracking objects entering, traversing and exiting the area), etc. A smaller area incurs a lower latency for message transfer because fewer hops are required but incurs a higher overhead for membership change operations because of a higher rate of members leaving and joining geographically smaller areas. Thus when mobile hosts have low mobility rates, it may dictate a smaller geographical area. On the other hand, a larger area incurs a higher latency for data change operations because of more hops in the larger geographical area but incurs a lower overhead for operations associated with consistent group membership maintenance due to node mobility/failure events. Thus there exists an optimal size, when given a set of parameters characterizing the mobile application in the underlying *ad hoc* network environment. In this paper, we aim to analyze performance characteristics of location-based data consistency maintenance algorithms in terms of the optimal size that can best tradeoff consistency for timeliness (reflecting the time to propagate a state change) while satisfying the imposed data consistency requirement.

2. System Model

We assume a mobile *ad hoc* network consisting of one or more peers. The network is heterogeneous, with peers in the system having greatly different capabilities. For a battlefield application, for example, one end of the capability spectrum is represented by large command and control nodes (mobile or fixed), such as an aircraft carrier or fixed surface-to-air missile site. At the other end of the spectrum we find human-portable devices or pilot-less vehicles with more modest command and control capabilities. Each peer has one or more communication devices and may have organic sensors whose data is shared with other peers in the distributed system. Additionally, each peer has one database in which sensor and other state data is stored.

2.1 Geographical Community of Interest

The notion of location-based data consistency considered in this work is based on the concept of a geographical community of interest. This concept allows us to move from a requirement for data consistency among *all* peers in the system (which is not achievable) to a requirement for data consistency only among peers who belong to the same geographical community of interest. We also note that in battlefield applications, as in any other problem-solving, team-oriented application, it is more important (in most cases) to have greater consistency with nearby peers than with ones farther away. In this paper we will use the terms “geographical community of interest” and “location-based group” (or just “group” for short) interchangeably.

The state of the mobile application is characterized by the values of state variables (e.g., track objects in the example battlefield application) maintained by peers in the system. The location-based data consistency requirement means that all the peers within the same geographical community of interest will have the same values for state variables.

The area of a geographical community of interest can be modeled many ways. We can divide the terrain into geometric shapes like squares or hexagons. Figure 1 is a coverage model showing three possible ring sizes for modeling a geographical community of interest based on hexagons, i.e., covering 1, 7 and 19 hexagons, respectively by ring 0, ring 1 and ring 2. The size of each geographical community of interest may vary depending on the operating conditions. For example, if the mobility rate is low for most mobile hosts, then the size can be small to optimize the performance.

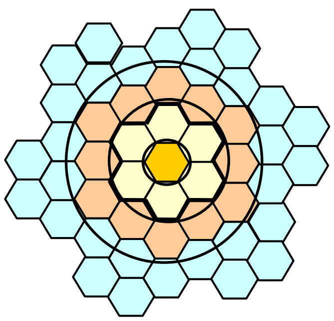

Figure 1: A Representation of a Geographical Area based on a Hexagonal Coverage Model.

2.2 Location-based Group Membership and Data Consistency Algorithm

We assume that each mobile host has a unique host identifier and is equipped with location sensing devices such as a Global Positioning System (GPS) receiver, so it can determine its own location as well as reason about its location relative to the locations of its neighbors within radio range. For a geographical community of interest identified by a group identifier, if the cardinality of the membership set (containing members that are connected in the *ad hoc* environment) is not zero, the mobile host with the smallest host identifier will be elected as the leader. The leader broadcasts its presence within the community of interest periodically. Should the leader fail, the failure event will be detected and a re-election protocol will be followed to select a new leader. If two or more leaders announce their presence, the leader with the smallest host identifier wins and the rest will relinquish their roles.The group discovery protocol is location-based. When a mobile host moves out of a geographical community of interest, it voluntarily informs the group leader of its departure, who in turn will perform a membership change operation to exclude the host in the group. Conversely, when a mobile host enters a new geographical area of interest, it broadcasts a *hello* message containing its location information and host identifier to discover the new location-based group to join. When a host, say A, receives a *hello* message from host B, it informs the corresponding group leader which in turn will perform a membership change operation to include B in the group. If the leader receives multiple messages regarding B's new membership, it accepts the first and ignores the rest.

Each mobile host also periodically sends an update message to the leader regarding its location and identifier so the leader is aware of who are still within the community of interest. When the leader detects that a member mobile host has not sent its update message, it assumes that the member has been disconnected either due to mobility or failure and will remove the mobile host from the group membership. A mobile host can always send a hello message to request for membership reinstatement if it suspects that it has been removed from the group membership by the leader. This periodic maintenance event thus allows the leader to actively gather information regarding new and missing members to maintain consistent group membership.

Within a geographical community of interest, if there is a state change detected by any member in the group (e.g., a hostile object approaching), the member will send a message to the leader which in turn will forward the message to all members in the group. A multicast tree is built dynamically to permit the leader to reach all members more efficiently, reliably, and securely.

2.3. Traffic Model and Performance Metric

We assume that each mobile host has its own distinct mobility rate in and out of geographical communities (groups) of interest. For example, helicopters move faster than tanks which in turn move faster than human beings in general. We assume that the terrain is virtually partitioned into equal-area regions (e.g., hexagons) for ease of analysis and presentation as shown in Figure 1. Let the mobility rate of mobile host *i* be σ_i moving in and out these regions. Each mobile host also has its own distinct failure rate. Let the failure rate of mobile host *i* be ϕ_i. Following the group membership protocol described earlier, let T be the time period between which each mobile host sends its location information and identifier to the leader in an update message. The time required for the leader to perform a membership change operation depends on the size of the geographical area. Let $\mu_{mc}(n)$ be the rate at which a membership change operation is executed in a geographical community of interest with a ring size of n (see Figure 1 for illustration), including the time to rebuild a multicast tree by the leader. Similarly the time required for the leader to perform a state update operation also depends on the size of the geographical area. Let $\mu_u(n)$ be the rate at which the leader can propagate an update to members within a geographical community of interest of size n where n is the ring size of the geographical community of interest. Finally, as a larger geographical community of interest is likely to maintain a larger set of state variables (e.g., sensor data in the example battlefield application), let $\delta(n)$ be this sensor-update rate with n again the ring size of the geographical community of interest. This data update rate also depends on the objects to be tracked, for example, the data update rate to track a theater ballistic missile is generally different than that required to track an air-breathing missile or aircraft. Later in the paper we will show how these parameters can be parameterized (i.e., be given values) properly reflecting the design choice such as the size of a geographical area of interest.

Our performance metrics of interest would measure "timeliness" and "consistency" of state information distributed to members within the geographical community of interest. The timeliness metric is measured by the response time R required to achieve data consistency whenever there is a state change detected by any member within a geographical area of interest. On the other hand, the consistency metric is measured by the proportion of time the system is in a consistent state, which can be broken up into two measures. The first measure PT_m is the proportion of time the group membership is consistent, while the second measure PT_{md} is the proportion of time both membership and state data are consistent among the node members of a location-based group. Our goal is to satisfy the response time requirement while making the consistency measures as high as possible. When there is a constraint in the consistency requirement, the goal is to minimize the response time measure by identifying the best geographical community of interest area size while satisfying the imposed consistency requirement, when given a set of model parameters identified and parameterized characterizing the operational conditions of the mobile application in *ad hoc* networking environments.

3. Performance Model

In this Section, we develop a Stochastic Petri net (SPN) performance model to describe the behavior of a mobile application operating under the location-based data consistency algorithm described earlier in Section 2. Later in Section 4 we will utilize this performance model to calculate the timeliness and data consistency metrics to analyze the tradeoff between data consistency and timeliness, given a set of parameter values characterizing a given mobile *ad hoc* environment.

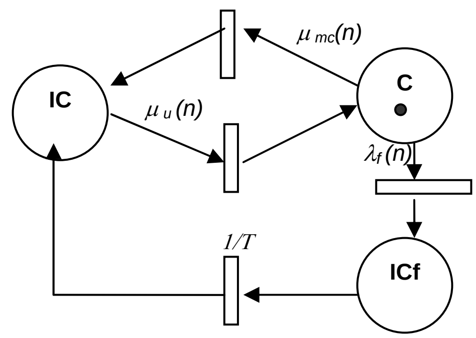

Figure 2: Petri Net Model for Location-Based Data Consistency Algorithm.

Figure 2 shows an SPN model for describing the behavior of the system operating under the location-based membership and data consistency protocol within a geographical area of interest of size n. The SPN model can be viewed as a continuous-time finite state machine which reacts to system events that occur in the system. There are 3 places in the Petri net model, with "C" standing for the state in which the system is consistent in membership, "IC" standing the state in which the system is inconsistent in membership due to nodes moving in and out of the geographical area of interest, and "ICf" standing for the state in which the system is inconsistent in membership due to unannounced node failures or disconnections. Initially the system is in a consistent state, represented by having a token deposited in place "C". We use the place at which the token resides to represent the current state of the system as time progresses, so the initial state is "C" as the token is initially placed there.

Whenever there is a membership change due to arrivals and departures of mobile nodes in and out of the geographical area of interest of size n with rate $\mu_{mc}(n)$, the system migrates from state "C" and state "IC". Our algorithm requires mobile hosts to inform the leader of the membership changes when they move in and out of the location-based group of size n, the discovery rate of which is $\mu_{mc}(n)$. After a membership change detection event occurs, the leader then sends a membership update operation to all members in the location-based group, the rate of which is the same as that for the state-update operation, i.e., $\mu_u(n)$. These behaviors are captured by the two transitions in the upper part of the SPN model with rates $\mu_{mc}(n)$ and $\mu_u(n)$, respectively. Note that the token flows from state "C" to state "IC" and then to state "C" again, reflecting that a membership update event is taken sequentially following a membership change detection event.

Whenever there is a membership change due to unannounced failure or disconnection of mobile nodes with rate $\lambda_f(n)$, the system migrates from state "C" to state "ICf". This event is modeled by a lower right transition in the SPN model with rate $\lambda_f(n)$. Periodically, the leader will collect and analyze beacon messages sent from mobile members of the location-based group and detect if any member needs to be removed from the membership because of unannounced failure or disconnection events. Consequently, any unannounced failure or disconnection will be detected by the system after a period of time T has elapsed. This detection event is modeled by the lower left transition with a deterministic time period T. Afterward the token flows to place "IC" in which the system performs a membership change operation with rate $\mu_u(n)$ again to bring the group membership consistent. The last event is modeled by having the token flow from place "IC" to place "C" through a transition with rate $\mu_u(n)$ to inform all members of the membership change.

Note that " data change" events are not modeled in the SPN model as one can image emanating from each state, whenever there is a state change due to sensor detection with rate $\delta(n)$, the system will migrate to another state in which the system will propagate the data update from the mobile user detecting the data change (through sensors presumably) to the leader and then from the leader to all mobile nodes in the location-based group using the multicast tree maintained by the leader with the rate of data propagation being $\mu_u(n)$. To avoid clutter, we do not explicitly model this behavior in the SPN model and instead will consider it through probabilistic arguments when we later derive expressions for computing the consistency and timeliness performance metrics.

The system evolves over three states, namely, "C", "IC" and "ICf", as time progresses. Thus, there exists a steady-state probability that the system can be found in one of the three states. Let P_C, P_{IC}, P_{ICf} be the steady state probabilities of states "C", "IC", and "ICf" respectively, which can be obtained by evaluating the SPN model constructed after model parameters are parameterized (i.e., given specific values) characterizing environment- and application-specific operating conditions. Then we can calculate consistency metrics, i.e., PT_m and PT_{md}, as follows:

$$PT_m = P_C \qquad (1)$$

$$PT_{md} = \mu_u(n)\, P_C \,/\, (\mu_u(n) + \delta(n)) \qquad (2)$$

Equation (1) above gives the proportion of time the system is consistent in membership, which is exactly the same as the equilibrium probability that the system is found in state "C". Equation (2) gives the proportion of time the system is consistent in both data and membership, which is equal to the equilibrium probability that the system is found in state "C" multiplied with the probability that the system is consistent in data, given that the system is consistent in membership. This can be reasoned by considering splitting state "C" into two states "C1" and "C2" such that "C1" is a state that is consistent in both membership and data while state "C2" is a state that is consistent in membership only because a data update propagation operation is still taking place. If one draws a two-state model with "C1" and "C2" such that the rate from "C1" to "C2" is $\delta(n)$ for the data-change transition (due to sensing) while the rate from "C2" to "C1" is $\mu_u(n)$ for the data-update transition (for propagating updated data to members), then one will see that the probability that the system is consistent in both membership and data, i.e., in state "C1", given that it is in state "C", is equal to $\mu_u(n) / (\mu_u(n) + \delta(n))$.

The timeliness metric can be calculated by the average of the response times obtained in various states weighted by their respective state probabilities, i.e.,

$$R = (P_C + P_{ICf})/\mu_u(n) + P_{IC}\,(1/\mu_u(n) + 1/\mu_u(n)) \qquad (3)$$

Here the first term accounts for the response time when the system is in either state "C" or state "ICf", which incurs an average update propagation time of $1/\mu_u(n)$, while the second term accounts for the response time when the system is in state "IC" which incurs a waiting time of $1/\mu_u(n)$ to account for the extra time required to process the membership change operation before taking another $1/\mu_u(n)$ time to process the data propagation operation by the system (leader). Here we note that while the system is in state "ICf", the leader will only propagate data to members inconsistently since in state "ICf" the system is in a state in which the leader is not aware of the fact that the group membership is inconsistent. Contrarily, the system is fully aware of its membership inconsistency in state "IC", in which case the leader is in the process of performing a membership change operation, so a data propagation operation newly arriving must wait for the membership operation to execute to completion before being processed by the leader, thus incurring a waiting time to the response time.

We also note that when T is small, the probability of the system found in state "ICf" will be small since the moment the system is in state "ICf" it will transit to state "IC" quickly in which a membership change operation will be executed to maintain membership consistency.

Thus a small T improves membership and data consistency while compromising the response time performance metric, and vice versa, and there exists a tradeoff between the consistency metrics (as given by Equations (1) ad (2)) and the timeliness metric (as given by Equation (3)).

4. Analysis

4.1 Parameterization

Consider a mobile *ad hoc* network modeled by a hexagonal network coverage model as illustrated in Figure 1 with the center region in ring *0*. Also consider a location-based group with a geographical area of interest of size n covering ring 0 through ring $n-1$. For a mobile node, say, node i, in the area, let λ^n_i be the "outward" mobility rate of mobile node i to go out of ring n into ring $n+1$ and μ^n_i be the "inward" mobility rate of the mobile node to go out of ring n into ring $n-1$.

The specific values of λ^n_i and μ^n_i for mobile node i depend on the semantics of the mobile applications and the mobility model of the mobile node. As an example, consider the node follows a random walk mobility model. It can be shown that [6] when a mobile node is in ring n, the probabilities of the mobile node with random walk moving outward to ring $n+1$, moving inward to ring $n-1$, and staying within ring n, upon a movement out of a hexagon region, denoted by P_{omove}, P_{imove} and P_{smove}, respectively, are given by:

$$
\begin{aligned}
P_{omove} &= \begin{cases} 1 & if\ n=0 \\ \dfrac{2n+1}{6n} & otherwise \end{cases} \\
P_{imove} &= \begin{cases} 0 & if\ n=0 \\ \dfrac{2n-1}{6n} & otherwise \end{cases} \\
P_{smove} &= \begin{cases} 0 & if\ n=0 \\ \dfrac{2n}{6n}=\dfrac{1}{3} & otherwise \end{cases}
\end{aligned}
\tag{4}
$$

Let σ_i represent the user mobility rate of mobile node i moving across hexagonal areas. Again let λ^n_i be the outward mobility rate of mobile node i to go out of ring n into ring $n+1$ and μ^n_i be the inward mobility rate of the mobile node to go out of ring n into ring $n-1$. Then,

$$
\begin{aligned}
\lambda_i^n &= \begin{cases} \sigma_i & if\ n=0 \\ \dfrac{2n+1}{6n}\sigma_i & otherwise \end{cases} \\
\mu_i^n &= \begin{cases} 0 & if\ n=0 \\ \dfrac{2n-1}{6n}\sigma_i & otherwise \end{cases}
\end{aligned}
\tag{5}
$$

Now consider that the mobile *ad hoc* network is populated with mobile nodes with an average density of M_i users per hexagonal area located at ring i. For the uniform density case, all M_i's are the same, say, equal to M_0. The more reasonable case is that there are more mobile nodes close to the center of the geographical area (since they are interested in the area and are members of the location-based group) and less nodes as we move further away from the center of the geographical area. This inhomogeneous density distribution can be modeled by a population function with an exponential decay behavior. Let M_0 be the density of the center hexagon in a geographical area of interest, then M_i is given by:

$$M_i = \frac{M_0}{b^i} \tag{6}$$

Here b is the population decay parameter whose magnitude represents how fast the population density decays as we move away from the center of attention in the geographical area, with the special case b=1 being the uniform density case. Since a geographical area of interest of size n contains $3n^2 - 3n + 1$ hexagons, so there are $(3(n+1)^2 - 3(n+1) + 1) - (3n^2 - 3n + 1) = 6n$ hexagons in ring n, with $n>0$. For example, ring 0 contains 1, ring 1 contains 6 and ring 2 contains 12 hexagons, and so on. Since only nodes in ring n moving inward to ring $n-1$ and nodes in ring $n-1$ moving outward to ring n will trigger a location-based membership change operation, the overall rate at which all the mobile nodes will trigger a membership change for a geographical area of interest of size n (consisting of ring 0 to ring $n-1$) due to mobility, defined as $\mu_{,mc}(n)$, is given by:

$$\mu_{mc}(n) = 6(n-1)M_{n-1}\lambda^{n-1} + 6nM_n\mu^n \tag{7}$$

Here the first term accounts for the rate at which mobile nodes move out of the geographical area of interest of size n, and the second term accounts for the rate at which mobile nodes move into the area, both triggering a membership change operation. Note that we have dropped the subscript i from λ^n_i and μ^n_i to refer to the fact we have considered all mobile nodes in Equation (6).

Since a geographical area of interest of size n on average will contain $M_0+6M_1+12M_2+ \ldots + 6(n-1)M_{n-1}$ mobile nodes, the rate at which mobile users within a geographical area of interest of size n fail or disconnect unannounced, $\lambda_f(n)$, is given by:

$$\lambda_f(n) = \phi(M_0 + \sum_{j=1}^{n-1} 6jM_j) \tag{8}$$

The time for a leader to propagate a state-update operation or a membership change operation to all members in the geographical area depends on if the propagation method is broadcast-based or multicast-based. Suppose that we adopt multicast-based for the sake of security. Then the propagation time depends on the number of members in the group and the way the leader builds the multicast tree to reach all members. Assume a perfect balance tree. Then on average it takes $log_2(M_0+\sum_{j=1}^{n-1} 6jM_j)$ hops to reach all members and the communication time per hop is τ depending on the underlying communication technology deployed in the *ad hoc* network. Consequently, the rate at which the leader performs a state-update operation to all members in the location-based group of size n, $\mu_u(n)$, is given by:

$$\mu_u(n) = \frac{1}{\tau \log_2[M_0 + \sum_{j=1}^{n-1} 6jM_j]} \tag{9}$$

Equations (5), (6), (7), (8) and (9) thus parameterize model parameters $\mu_{mc}(n)$, $\lambda_f(n)$ and $\mu_u(n)$ once we are given values of basic parameters M_0 and b (density of mobile nodes), σ (mobility rate per node), ϕ (failure rate per mobile node) and τ (communication delay per hop) characterizing the network and application operating conditions.

4.2 Numerical Data

Here we present numerical data obtained from evaluating the Petri net model developed using SPNP [8] based on Equation (1), (2) and (3) to show design tradeoffs between the timeliness (R) and consistency metrics (PT_m and PT_{md}) obtained, as a result of applying our location-based data consistency algorithm in mobile *ad hoc* networking environments. The set of parameters characterizing the mobile application in the underlying *ad hoc* network environment is given by τ=1, M_0=2, b=4, $\delta(n)$=0.01, σ=0.05, ϕ=0.001, T=5. These parameters are normalized with respect to τ=1 (hop-by-hop delay) for ease of presenatation, e.g., ϕ=0.001 means that the failure rate on average is once per 1000τ, and T=5 means that the periodic check is about once every 5τ period. We will analyze the effects of some of these parameters in the paper by changing their values to observe their impacts on R, PT_m and PT_{md} obtained. Figures 3 and 4 show the consistency metrics PT_m and PT_{md} verse the size of geographical area of a location-based group. Unlike the timeliness metric which monotonically increases with n, we observe that there exists an optimal n, say, n_{opt}, at which the consistency metric is maximized. For example, when b=1 or 2, n_{opt}=2, when b=4, n_{opt}=4 and when b=8 or 16, n_{opt},=3 in both Figures 3 and 4.

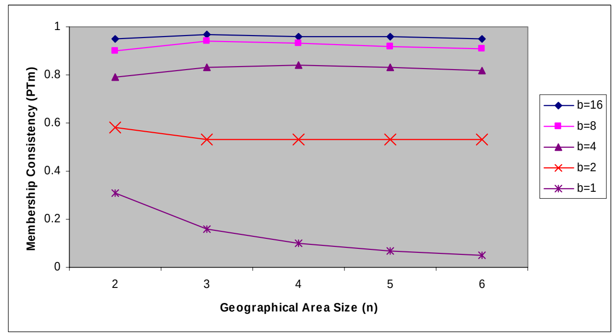

Figure 3: Membership Consistency (PT_m) vs. Geographical Area Size (n).

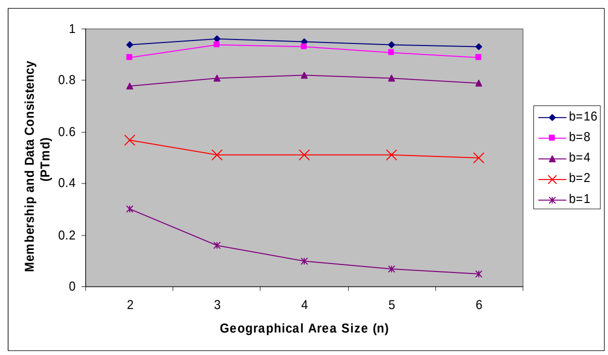

Figure 4: Membership and Data Consistency (PT_{md}) vs. Geographical Area Size (n).

The reason that an optimal area size exists for maximizing membership and data consistency is that membership inconsistency is attributed to the system being in state "IC" due to mobility events for nodes in and out of the group, and also in state "ICf" due to failure events for member nodes. The rate of node failure events is directly proportional to the number of member nodes in the location-based group. Thus, as n increases, more failure events are likely to occur as there are more member nodes in the group. On the other hand, the rate at which mobility events occur due to nodes moving into and out of the geographical area is not necessarily proportional to n. For the inhomogeneous population model defined by Equation (5), the rate of membership changes induced by user mobility actually decreases as n increases when b>2, because there are fewer nodes residing at the outer hexagons (due to exponential population decay) as we move away from the center hexagon of the location-based group, so most nodes in the group are likely to be contained within the area when n is large. These two effects counterbalance each other, thus resulting in an optimal area size that maximizes the membership and data consistency metrics.

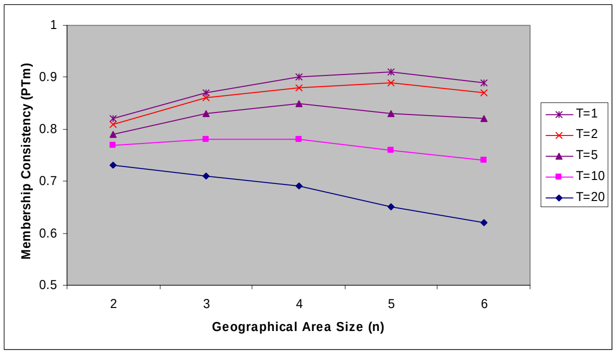

Figure 5: Effect of *T* on Membership Consistency (PT_m) and Optimal n_{opt}.

At a large group size (i.e., a large *n*) we can expect that practically there will be little mobility-induced membership changes since all mobile nodes would be reasonably contained within the area most of the time if *b*>2. Most membership change operations incurred in this case would be due to node failures whose rate increases as *n* increases. Figure 5 shows the effect of *T* on the optimal size n_{opt} for membership consistency. (The graph for the effect of *T* on the optimal size n_{opt} for both membership and data consistency exhibits the same trend and is not shown to avoid clutter.) We see that for the same operational condition (*b*=4 is chosen as the example), as *T* decreases n_{opt} increases, e.g., n_{opt} goes from 1 to 4 as *T* goes from 10τ to τ, because with a smaller *T*, membership changes due to node failures can be performed more rapidly, thus favoring a larger area for which membership changes are mostly due to node failures.

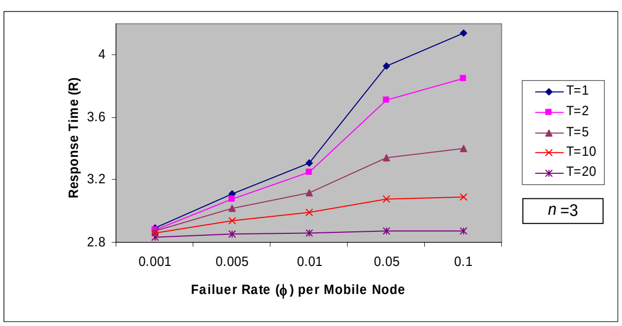

Figure 6: Pronounced Adverse Effect of *T* on *R* at High Failure Rate (ϕ).

With the above analysis, we know that we could achieve reasonably high membership consistency with a large geographical area size *n*, and a small *T*. Unfortunately, both a large *n* and a small *T* adversely degrade the response time metric. A more frequent periodic detection activity (i.e., a smaller *T*) degrades the response time metric more because more time will be spent by the leader to do membership maintenance induced by node failures, thus causing any concurrent state-change operation to be delayed. Isolating out *n*=3 as a case study, Figure 6 shows that the adverse effect of *T* on *R* is especially pronounced when the failure rate (ϕ) is high at which the leader must perform failure-induced membership change operations very frequently in order to maintain membership consistency, thus causing a high delay in the response time per state-change operation. Whether we should select a short *T* and a large area size *n* to yield high consistency at the expense of timeliness, or vice versa, depends on the application's QoS requirements in consistency and timeliness.

References

[1] T.D. Chandra, V. Hadzilacos, S. Toueg and B. Charron-Bost, "On the impossibility of group membership," *15th ACM Symposium on Principles of Distributed Computing*, New York, USA, 1996, pp. 322-330.

[2] J. Dahmann and J.W. Wilson, "MDA and HLA: Applying standards to development, integration and test of the single air picture integrated architecture behavior model," Fall Simulation Interoperability Workshop (SIW), Orlando, Florida, 2003.

[3] M.J. Fisher, N.A. Lynch and M.S. Paterson, "Impossibility of distributed consensus with one faulty process,'' *Journal of the ACM*, Vol. 32, April 1985, pp. 427-469.

[4] Q. Huang, C. Julien and G.-C. Roman, "Relying on safe distance to achieve strong partitionable group membership in ad hoc networks," *IEEE Transactions on Mobile Computing*, Vol. 3, No. 3, 2004.

[5] M.-O. Killijian, R. Cunningham, R. Meier, L. Mazare and V. Cahill, "Towards group communication for mobile participants,'' *ACM 1st International Workshop on Principle of Mobile Computing*, Newport, Rhode Island, USA.

[6] Y.B. Lin, L.F. Chang and A. Noerpel, "Modeling Hierarchical Microcell and Macrocell PCS Architecture", *ICC*, June 1995.

[7] G.-C. Roman, A. Huang and A. Hazemi, "Consistent group membership in ad hoc networks,'' *IEEE International Conference on Software Engineering*, 2001, pp. 381-388.

[8] Stochastic Petri Net Package (SPNP) User's Manual, Version 6, Department of Electrical and Computer Engineering, Duke University, 1999.