Modeling and Simulation of Secure Automatic Energy Meter Reading and Management Systems using Mobile Agents

Radwan Tahboub†, Dan Lazarescu†, Vasile Lazarescu†

Applied Electronics and Information Engineering Dept., POLITEHNICA University of Bucharest, Romania

Summary

The current energy saving technologies rely on conventional data logging systems, where the current methods in retrieving the energy data is not convenient, and the cost of the data logging systems is high. The Automatic energy Meter Reading (AMR) technology is network-based applications used by electric/energy companies to increase performance and reliability of their energy meter reading and management systems. A mobile agent is a program (code and data) that can autonomously migrate from host to host in a network of heterogeneous computer systems and fulfill a task specified by its owner. Intelligent and secure Automatic Meter Reading and Management using Mobile Agents can be of great importance for municipalities and energy distribution companies so as to minimize the number of traditional visits required by the distribution company, hence decreasing the number of employees used in performing this traditional time consuming and high cost work. In this work we will start by surveying the current technologies and techniques used in handling remote meter reading and management. New modeling and simulation results for different configurations and techniques of automatic energy meter reading and management systems using mobile agents will be presented and compared to traditional client server techniques.

Key words:

Automatic Meter Reading (AMR), Mobile Agents, Client Server, JADE, OPNET.

1. Introduction

Remote system control and management is becoming much easier to manage since new products are becoming networked and smarter than before. The Automatic Meter Reading (AMR) and Supervisory Control and Data Acquisition (SCADA) technologies are network-based applications used by electric utilities to increase performance and reliability of their power transmission and distribution systems. AMR-based networks are utilized for theft detection, outage management, customer energy management, load management, on/off services, and distributed automation, among others. SCADA is used for controlling and managing the grid of power distribution system[7]. Different types and technologies of communication links can be utilized as the communication medium in an AMR or automatic meter reading systems. This includes telephone lines, local area network technologies, Internet, and the power line carrier or PLC[7].

A mobile agent is a program (code and data) that can migrate from host to host in a network of heterogeneous computer systems and fulfill a task specified by its owner. It works autonomously and communicates with other agents and host systems. During the self-initiated migration, the agent carries its code, may be data, and some kind of execution state with it. On each host they visit, mobile agents need a special software that is named agency or host, which is responsible to execute/host agents, provides a safe execution environment, and offers several services for agents residing on this host. A mobile agent system is the set of all agencies of the same type together with the agents running on these agencies as part of an agent-based application [10,12,14].

Based on the traditional traveling sales man problem we will introduce the term Mobile Agent Automatic Meter Reader and or Manager (MAMR/M or simply MAMR). Simulations and implementation testing will be presented to show the applicability of such a new approach. Also there exist many security issues that need to be addressed to protect the proposed automatic meter reading and management systems. Such security issues consist of protecting the newly considered power metering systems and protecting the MAMR used in the management and information retrieval processes from these devices.

In this work, we present new approaches for Secure Automatic Meters Reading and Management systems using Mobile Agents' information retrieval systems. Namely the retrieval and management of Energy Meter Readings densely distributed in LAN / WAN configuration.

2. Traditional Meter Reading Systems

Meter Reading and Billing are among the most time consuming functions performed by municipalities and energy distribution companies. These functions have a major influence on the utilities cost, efficiency, productivity, structure and cash flow as well. Solutions based on recording readings manually, then entering it into a central billing system are time consuming, prone to errors and delays in delivering bills to customers with

Manuscript received January 5, 2007
Manuscript revised January 25, 2007

negative effect on cash flow. There are many methods involved in the meter reading process; this includes traditional manual methods up to fully automatic meter reading systems [13]:

- Traditional Systems:
  - Read by walk: Once by 2 months or by one year.
  - Read by customer. Where customers will be responsible for reading the meter readings.

It should be clear that such methods are very time consuming and does not satisfy the business requirements for the power company, in addition to the large number of errors incorporated in the reading process.

- Hand-Held units for meter reading and billing system: speed, accuracy, and cost effectiveness are the strongest features in this system.
- Prepaid AMR. Where each meter has a smart card reader and the meter will be on as long as the smart card has credit.
- AMR Systems with the following main components:
  - The meter interface unit (MIU) with an embedded server capability, communication or networking system, and power company offices system.
  - Different networking methods for data transmission are used; this includes telephone lines, Cable TV, LANs/WANs, fiber, wireless and power line.
  - Micro Web-Servers can be used in the design and implementation of power meter reading systems to facilitate the Automatic Meter Reading process [12].
  - There are many different forms of communication network technologies in both LAN and WAN scales that can be utilized as the communication medium in AMR systems.
  - Traditional and state of the art Wireless, Bluetooth and Power-Line Communication (PLC) techniques can be used to construct such LANs. For the WAN part, the most common technology options for providing broadband services for AMR includes: xDSL, Cable Modem, Fiber-to-the-Home, Wireless, Satellite, and PLC[13].

3. AMR Data Collection using Mobile Agents

Mobile agents consists of a self-contained piece of software that can migrate and execute on different machines in a dynamic networked environment, and that senses and (re)acts autonomously and proactively in this environment to realize a set of goals or tasks as described in[13]. Mobile agents are commonly used in distributed information-retrieval applications.

In this section, we consider such a retrieval and remote management system, namely the management and retrieval of Energy Meter Readings densely distributed in LAN / WAN configuration. Based on the traditional traveling sales man problem we will introduce the term **Mobile Agent Automatic Meter Reader and Management (MAMR/M or simply MAMR).** A model for these MAMR system activities will be presented in this section. The simulation using 2 different tools will be shown and discussed to show the applicability of this new method. Then security issues related to our new approach and a new secure collection protocol will be presented.

3.1 The Mobile Agent AMR (MAMR) Problem:

Let us assume that a specific task should be executed in $n+1$ power / energy meters m_i with embedded servers where $0 \le i \le n$. Also the Power Company's main server is m_0 .

Starting and ending at m_0 , the MAMR can continue to visit the next planed meter to execute on it. When the reading and management tasks are executed and finished at a meter m_i MAMR visits the next one, and so on. When no more meters are to be visited or all the remaining meters to be visited are not reachable then the MAMR should return to m_0 by the most expedient route which may or may not be taking the direct path. Returning back requires l_{k0} travel time if the MAMR has finished execution, reading and managing at m_k . If the task is not executed then the MAMR will visit another meter with no need to return back to m_0 . This is one of the basic differences between MAMR and the Traveling Agent Problem presented in [12] where the MA returns back as soon as it finishes the first successful task.

Formally, the MAMR Agent Problem is defined as follows: There are $n+1$ energy meters m_i with embedded servers that are capable of hosting and running MAMR where $0 \le i \le n$ including the Power Company's main server, m_0 . A probability, $p_i = 1$, of being able to successfully visit meter m_i and complete the agent's task (reading or monitoring, or other specified tasks), also $p_i = 0$ means that MAMR can not visit m_i or meter m_i is not reachable. These probabilities are independent of each other. An execution time $x_i > 0$, is required for the agent to attempt the visit and communicate with the next power meter before moving to a meter m_{i+1} regardless of whether it is successful or not. And a time r_i where a MAMR reads and manages a meter m_i , including compression, encryption/decryption, hashing, and digitally signing read information. Travel times or latencies for the

MAMR to move between sites are given by $l_{ij} \geq 0$ for moving between meter m_i and meter m_j . When the MAMR task has been successfully completed at all possible meters, the MAMR must return to the main companies server from which it started (i.e., m_0). For meter m_0 , $p_0 = 1$. The Mobile Agent Automatic Reader Problem is to successfully complete the reading and managing tasks. One can also consider minimizing the expected time to complete this task.

This problem can be formulated as a Markov Decision Problem or discrete stochastic control problem [13] in which the state space consists of vectors indexed by meters with coordinate values indicating whether a meter has been visited already or not. Standard dynamic programming algorithms could be used on this formulation but since the state space is exponentially large in the number of meters, this formulation is not scalable. However, in certain cases, the state space can be simplified leading to efficient dynamic programming solutions [13].

A tour for the Mobile Agent Automatic Reader Problem consists of specifying the order in which to read the meters, namely a permutation $< i_1, i_2, ..., i_n >$ of 1 through n. Such a permutation is called a tour in keeping with the tradition for such problems [12].

The expected time to complete the tasks and read all meters provided that all meters are reachable, for a tour $< i_1, i_2, ..., i_n >$ representing an arbitrary permutation of the existing energy meters:

$$T_{MAMR} = \begin{cases} l_{0i_1} + x_{i_1} + r_{i_1} + \\ \sum_{k=2}^{n} \left\{ (\prod_{j=1}^{j=k} p_{i_j}) \right\} (l_{i_{(k-1)} i_k} + x_{i_k} + r_{i_k}) + \\ (\prod_{j=1}^{j=z} p_{i_j}) l_{i_z i_0} \end{cases} \quad (3.1)$$

This formula can be explained as: The first meter, m_{i_1} , on the tour is always read first (this is not necessarily the first meter m_1) and requires traveling time l_{0i_1} to be reached. Upon arrival, time $x_{i_1} + r_{i_1}$ must be spent there for reading and monitoring and executing. With probability p_{i_1} =1 the meter m_{i_1} is reachable and task is successfully completed in which case the agent can move to the next meter m_{i_2} in the tour, however, with probability p_{i_2} either 0 or 1 to reach it. The expected value of the contribution involving moving from meter m_{i_1} to meter m_{i_2} and succeeding there is ($p_{i_1} p_{i_2}$) ($l_{i_1 i_2} + x_{i_2} + r_{i_2}$), and so on. Finally, the last term $(\prod_{j=1}^{j=z} p_{i_j})$ arises when all reachable meters in the tour with the last reachable meter is m_{i_z} and the MAMR must return to the originating server m_{i_0} . If all meters where read successfully then $z = n$. Figure 1 shows a possible tour done by a single MAMR (dotted lines).

Now, recalling that MAMR can be defined as a set of v code blocks with size ($\sum_{s=0}^{v} B_{bc_s}$ bytes of code), data and status information with size $B_{data_i + state_i}$, then the MAMR increases in size while moving from one meter to another specially the reading information (data). Hence the payload size of the sending MAMR from m_{i_j} to m_{i_k} can be considered as:

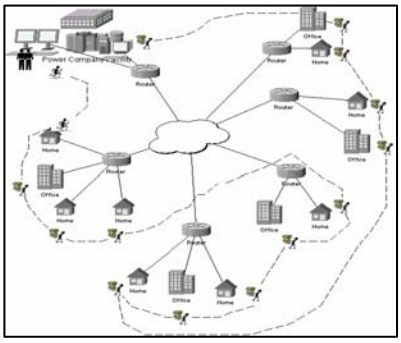

Fig 1. Tour done by single MAMR (Virtual Tour)

$$B_{MAMR}\big|_{m_{i_j} \to m_{i_k}} = \begin{cases} \sum_{s=0}^{v} B_{bc_s} + \\ B_{data_{i_j} + state_{i_j}} + \\ (1-\sigma) B_{rep_{i_k}} \end{cases} \quad (3.2)$$

Where $B_{rep_{ik}}$ is the size of the optional reply from m_{i_k} to m_{i_j} , σ denotes the selectivity of the agent, that is, how much the B_{rep} is reduced by remote processing. Accordingly, the time required to transfer this agent m_{i_j} to m_{i_k} :

$$l_{i_j i_k}\big|_{m_{i_j} \to m_{i_k}} = \begin{cases} 2\delta(m_{i_j}, m_{i_k}) + \\ \dfrac{\sum_{s=0}^{v} B_{bc_s} + B_{data_{i_j} + state_{i_j}} + (1-\sigma) B_{rep_{i_k}}}{\tau(m_{i_j}, m_{i_k})} \end{cases} \quad (3.3)$$

Where $\delta(m_{i_j}, m_{i_k})$ is the average network delay including DNS and other networking and routing latencies. The throughput between power meter m_{i_j} and the next power meter m_{i_k} is $\tau(m_{i_j}, m_{i_k})$. The MAMR can query a network status from a directory server to find latencies times and estimated throughputs, $\delta(m_{i_j}, m_{i_k})$ and $\tau(m_{i_j}, m_{i_k})$. Inserting equation 5.4 into 5.2 yields:

$$T_{MAMR} = \begin{cases} 2\delta(m_{i_0}, m_{i_1}) + \dfrac{\sum_{s=0}^{v} B_{bc_s} + (1-\sigma)B_{rep_{i_0}}}{\tau(m_{i_0}, m_{i_1})} + x_{i_1} + r_{i_1} + \\ \sum_{k=2}^{n}\left\{(\prod_{j=1}^{j=k} p_{i_j})\right\}\cdot\left\{\begin{cases} 2\delta(m_{i_j}, m_{i_k}) + \\ \dfrac{(\sum_{s=0}^{v} B_{bc_s} + B_{data_{i_j}+state_{i_k}} + (1-\sigma)B_{rep_{i_k}})}{\tau(m_{i_j}, m_{i_k})} + \\ x_{i_k} + r_{i_k} \end{cases}\right\} + \\ (\prod_{j=1}^{j=z} p_{i_j})(2\delta(m_{i_z}, m_{i_0}) + \dfrac{\sum_{s=0}^{v} B_{bc_s} + B_{data_{i_z}+state_{i_z}} + (1-\sigma)B_{rep_{i_0}}}{\tau(m_{i_0}, m_{i_1})} + x_{i_0}) \end{cases} \quad (3.4)$$

If not all of the MAMR code blocks is to be transferred each time MAMR moves from one energy meter to another and if only a subset of the code blocks will be requested from the source energy meter sending the MAMR then, the source energy meter sends a list of code blocks necessary to perform the specific MAMR operation. Using this list, the destination agent system can request only those code blocks not yet cached. Only one code block request is needed. Even though this method is more efficient, the source power meter system has to know the code blocks the next energy meter may need to run the MAMR. Accordingly, the payload size of the transferred MAMR in this case can be specified as:

$$B'_{MAMR}\big|_{m_{i_j} \to m_{i_k}} = \begin{cases} (1 - \prod_{s=0}^{v} P_s) B_{cr} + \\ \sum_{s=0}^{v} P_s B_{bc_s} + \\ B_{data_{i_j}+state_{i_j}} + B_{list_{i_j}} + \\ (1-\sigma) B_{rep_{i_k}} \end{cases} \quad (3.5)$$

And the corresponding new reading and managing time:

$$T'_{MAMR} = \begin{cases} 2\delta(m_{i_0}, m_{i_1}) + \dfrac{\sum_{s=0}^{v} P_s(B_{bc_s} + B_{cr}) + B_{list_{i_1}} + (1-\sigma)B_{rep_{i_1}}}{\tau(m_{i_0}, m_{i_1})} + x_{i_1} + r_{i_1} + \\ \sum_{k=2}^{n}\left\{(\prod_{j=1}^{j=k} p_{i_j})\right\}\cdot\left\{\begin{cases} 2\delta(m_{i_j}, m_{i_k}) + 2(\prod_{s=0}^{v}(1-P_s)\delta(m_{i_j}, m_{i_k})) + \\ \dfrac{(\sum_{s=0}^{v} P_s(B_{bc_s} + B_{cr}) + B_{list_{i_k}} + B_{data_{i_j}+state_{i_j}} + (1-\sigma)B_{rep_{i_k}})}{\tau(m_{i_j}, m_{i_k})} + \\ x_{i_k} + r_{i_k} \end{cases}\right\} + \\ (\prod_{j=1}^{j=z} p_{i_j})(2\delta(m_{i_z}, m_{i_0}) + \\ \dfrac{\sum_{s=0}^{v} P_s(B_{bc_s} + B_{cr}) + B_{list_{i_0}} + B_{data_{i_z}+state_{i_z}} + (1-\sigma)B_{rep_{i_0}}}{\tau(m_{i_z}, m_{i_0})} + x_{i_0}) \end{cases} \quad (3.6)$$

Where B_{cr}, is the size of the requesting list and P_s is the probability that a code block B_{cb_s} is requested to be transferred as requested in the transfer list.

It should be noted that the best optimum tour is the one that produces the minimum cost in terms of time and bandwidth usage. This cost can be compromised with the time and bandwidth used in the traditional client server AMR shown in Figure 2. The client server time can be computed as:

$$T_{cs} = Max(T_{cs}\big|_{m_{i_k} \to m_{i_0}}), \qquad k = 1, 2, ..., n \quad (3.7)$$

$$T_{cs}\big|_{m_{i_k} \to m_{i_0}} = p_{i_k}(l_{i_k i_0} + l_{i_0 i_k} + x_{i_k} + r_{i_k}) \quad (3.8)$$

$$T_{cs} = Max(p_{i_k}(l_{i_k i_0} + l_{i_0 i_k} + x_{i_k} + r_{i_k})), \qquad k = 1, 2, ..., n \quad (3.9)$$

So if the energy company calls an energy meter to send its data and state then the size of the payload of each of these meters reply will be as:

$$B_{cs}\big|_{m_{i_k} \to m_{i_0}} = B_{req_{i_0}} + B_{data_{i_k}+state_{i_k}} + B_{rep_{i_k}} \quad (3.10)$$

$$l_{i_0 i_k} = \lambda * (\delta(m_{i_0}, m_{i_k}) + \frac{(B_{reqi_0})}{\tau(m_{i_0}, m_{i_k})}) \quad (3.11)$$

$$l_{i_k i_0} = \lambda * (2 * \delta(m_{i_k}, m_{i_0}) + \frac{(B_{reqi_0} + B_{data_{i_k}+state_{i_k}} + B_{rep_{i_k}})}{\tau(m_{i_k}, m_{i_0})}) \quad (3.12)$$

And

$$T_{cs} = Max(p_{i_k} * \begin{cases} \lambda * (\delta(m_{i_0}, m_{i_k}) + \dfrac{(B_{reqi_0})}{\tau(m_{i_0}, m_{i_k})}) + \\ \lambda * (2 * \delta(m_{i_k}, m_{i_0})) + \\ \lambda * (\dfrac{(B_{reqi_0} + B_{data_{i_k}+state_{i_k}} + B_{rep_{i_k}})}{\tau(m_{i_k}, m_{i_0})}) + \\ (x_{i_k} + r_{i_k}) \end{cases}) \quad (3.13)$$

For $k = 1, 2, ..., n$

Where $\delta(m_{i_0}, m_{i_k})$ represents the network delay between m_{i_0}, m_{i_k}, and $\lambda \geq 1$ represents queuing delay factor that arises from congestion and router processing delays especially if more than one energy meter is probed for its reading. That is the time elapsed until the last meter sends its reading data to the server in the client server architecture is the time of completing the reading of all meters. This time is calculated provided that all meters where probed for their reading at the same time.

However for large and dense metering systems, this time is making large bandwidth usage since all meters will be communicating with the server which increases latencies and queuing times $\lambda * \delta(m_{i_0}, m_{i_k})$ and also the processing and response time of the server.

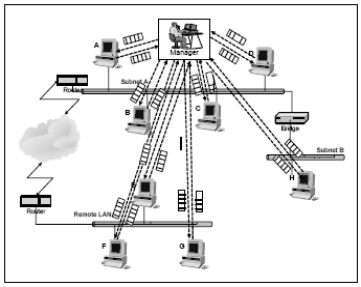

Fig 2. Client Server AMR

In contrast to the MAMR approach, the network bandwidth is free in most of the parts of the network except the path during MAMR moving from one meter to another.

3.2 Multiple MAMR System

An interesting case where multiple MAMRs are initiated and dispatched to collect and monitor remote meters is of concern. This case can be considered as parallel and distributed meter reading, hence the time used in reading all meters will be less than the time used in the single MAMR case, however the consumed bandwidth will increase since multiple sessions will be opened in different parts of the network. Also the case where each meter has a MAMR is also interesting where this is comparable to the client server case with the difference that more time is used in creating, sending and then processing all MAMRs.

Accordingly, the number of MAMRs, the execution time and the used bandwidths are significant performance factors in MAMR systems. The energy company can generate multiple MAMRs for achieving the energy meters reading and management tasks. Sending fewer MAMRs causes lower network traffic and consumes less bandwidth. MAMRs consume network bandwidth when they travel over the designated set of nodes. Badly scheduled MAMRs' itineraries can cause longer execution times as a result of higher routing costs. The number of MAMRs created for a task also influences the total routing cost. Hence intelligent MAMRS are of more importance here so as to determine the best routes to be selected while moving through a set of energy meters to perform its required and scheduled tasks. Clearly, the greater the number of MAMRs created , the higher the overall reading cost.

A tour for the Mobile Agent Automatic Reader Problem consists of specifying the order in which to read the meters, namely a permutation $< i_1, i_2, ..., i_n >$ of 1 through n . With multiple MAMRs, this permutation can be partitioned into ζ sub connected energy meters so that each MAMR is responsible of performing a tour on one of these sub group of connected meters, that is reading and managing one of $< i_1, i_2, ..., i_{\zeta_1} >$, $< i_1, i_2, ..., i_{\zeta_2} >$,…, $< i_1, i_2, ..., i_{\zeta_\Psi} >$,...,and $< i_1, i_2, ..., i_{\zeta_\Phi} >$ permutations which represents the sets of all the existing energy meters in the energy company enterprise, $m_0, m_1, ..., m_n$.

The expected time to complete the tasks and read all meters provided that all meters are reachable, for one of the existing tours $< i_1, i_2, ..., i_{\zeta_\Psi} >$ representing an arbitrary permutation of a subset of the existing energy meters can be written as:

$$T_{MAMR} = \begin{cases} 2\delta(m_{i_0}, m_{i_1}) + \dfrac{\sum_{s=0}^{v} B_{bc_s} + (1-\sigma)B_{rep_{i_1}}}{\tau(m_{i_0}, m_{i_1})} + x_{i_1} + r_{i_1} + \\ \sum_{k=2}^{\zeta_\Psi} \left\{ \left(\prod_{j=1}^{j=k} p_{i_j} \right) \right\} \cdot \left\{ \begin{array}{l} 2\delta(m_{i_j}, m_{i_k}) + \\ \dfrac{(\sum_{s=0}^{v} B_{bc_s} + B_{data_j + state_{i_k}} + (1-\sigma)B_{rep_{i_k}})}{\tau(m_{i_j}, m_{i_k})} + \\ x_{i_k} + r_{i_k} \end{array} \right\} + \\ \left(\prod_{j=1}^{j=z} p_{i_j} \right) (2\delta(m_{i_z}, m_{i_0}) + \dfrac{\sum_{s=0}^{v} B_{bc_s} + B_{data_{i_z} + state_{i_z}} + (1-\sigma)B_{rep_{i_0}}}{\tau(m_{i_0}, m_{i_1})} + x_{i_0}) \end{cases} \quad (3.14)$$

If all code blocks of the MAMR is to be transferred between m_{i_j} and m_{i_k} . However if only a list of this code is to be transferred then this time expressed as:

$$T'_{MAMR} = \begin{cases} 2\delta(m_{i_0}, m_{i_1}) + \dfrac{\sum_{s=0}^{v} P_s(B_{bc_s} + B_{cr}) + B_{list_{i_1}} + (1-\sigma)B_{rep_{i_1}}}{\tau(m_{i_0}, m_{i_1})} + x_{i_1} + r_{i_1} + \\ \sum_{k=2}^{\zeta_\Psi} \left\{ \left(\prod_{j=1}^{j=k} p_{i_j} \right) \right\} \cdot \left\{ \begin{array}{l} 2\delta(m_{i_j}, m_{i_k}) + 2(\prod_{s=0}^{v}(1-P_s)\delta(m_{i_j}, m_{i_k})) + \\ \dfrac{(\sum_{s=0}^{v} P_s(B_{bc_s} + B_{cr}) + B_{list_{i_k}} + B_{data_j + state_j} + (1-\sigma)B_{rep_{i_k}})}{\tau(m_{i_j}, m_{i_k})} + \\ x_{i_k} + r_{i_k} \end{array} \right\} + \\ \left(\prod_{j=1}^{j=z} p_{i_j} \right) (2\delta(m_{i_z}, m_{i_0}) + \\ \dfrac{\sum_{s=0}^{v} P_s(B_{bc_s} + B_{cr}) + B_{list_{i_0}} + B_{data_{i_z} + state_{i_z}} + (1-\sigma)B_{rep_{i_0}}}{\tau(m_{i_z}, m_{i_0})} + x_{i_0}) \end{cases} \quad (3.15)$$

It should be noted that $z = \zeta_\Psi$ means that all the meters in that sub group are read and the $MAMR_{\zeta_\Psi}$ should return to the energy company represented as m_{i_0} . Now we can calculate the total reading and managing time for the multiple MAMRs with and without all code transfer as:

$$T_{Multiple_{MAMR}} = Max\,(\lambda_{\zeta_\Psi} * T_{MAMR_{\zeta_\Psi}}) \quad \text{for all } 1 \le \Psi \le \Phi \qquad (3.16)$$

$$T'_{Multiple_{MAMR}} = Max\,(\lambda_{\zeta_\Psi} * T'_{MAMR_{\zeta_\Psi}}) \quad \text{for all } 1 \le \Psi \le \Phi \qquad (3.17)$$

Where $\lambda_{\zeta_\Psi} \ge 1$ represents queuing delay factor that arises from congestion and router processing delays for $MAMR_{\zeta_\Psi}$ especially if more than one MAMR is accessing certain network node. It is clear that λ_{ζ_Ψ} can be minimized if good MAMR tour planning and energy metering groups' selection is performed. Another very important question is how many MAMRs are required to perform the reading and managing certain tasks within a certain limit of time or bandwidth constraints. However these issues are not within the scope of this work and can be left as open questions and for future work.

3.3 AMR and MAMR Networks Simulations using OPNET

OPNET is a powerful network simulation tool that can be used to model communication systems and predict network performance. Its accuracy and ease-of-use make it a valuable tool for network planners and administrators[14]. We used the OPNET Simulator to model:

- Networks of power metering systems for part of Hebron city in (Figures 3,4)
- Modeling the Power Meter
- Modeling the LAN of Power Meter in a building.
- Modeling the WAN of Power Meter LANs Sites.
- Modeling the Connection to the Central Office.

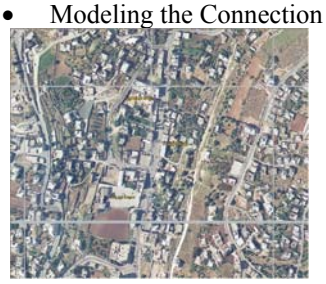

Fig 3. Part of Hebron city used in Our Simulation Model

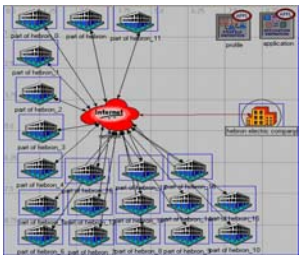

Fig 4. The whole network which connects the facilities in each part of Hebron (20 parts)

However, simulating the whole network of the city (20 parts) requires very large amount of memory, accordingly we have chosen to simulate the power meters network using 1 or 2 parts of the city with both wired and wireless scenarios. The Energy/Power company is assumed to have a T1 (DS0 1.544Mbps) Leased Line, each part of the city (~ 50-100 homes) is connected using ADSL/DSLAM (16kbps/64kbps). The homes are connected through a wired or wireless LAN, Figure 6. More than 50 simulation sceneries were considered in the simulation for Utilization (%), Throughput (packet/sec) and Queuing delay (sec) performance parameters. For example, Figure 7 shows the utilization for the DS1 link for Energy Company for one region and two regions connected. Figure 8 shows the delay for packets passing through this link. Figures 9, 10 shows the utilization and delay of the xDSL link for the scenario where all meters are sending their readings using FTP or e-mail and compared to the case where only one meter is getting all readings and sending it on behalf of the other meters (the size of the sent data is much larger). The later case represents the MAMR idea, where the former one represents the client- server case. This case is repeated for wireless energy meters.

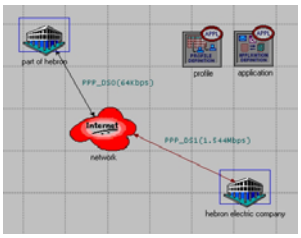

Fig 5. Configuration of a connected part of the city through DS0/DS1

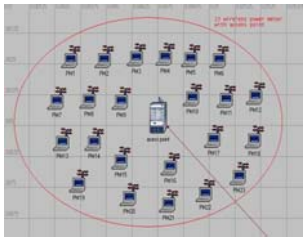

Fig 6. The homes' Energy meters connected through a LAN

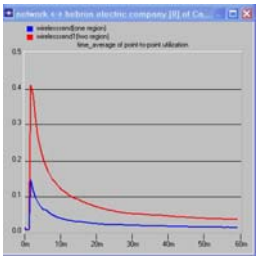

Fig 7 DS1 Utilization: one and two regions, wired.

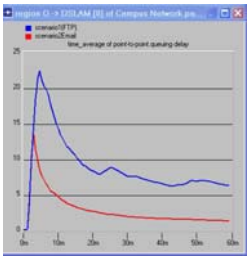

Fig 8 Queuing delay: FTP& Email on xDSL (16Kbps), wired.

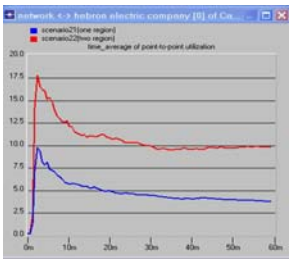

Fig 9 Utilization: Client-Server versus MAMR.

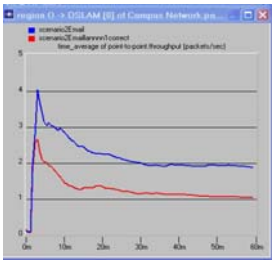

Fig 10 Queuing delay: Client- Server versus MAMR.

Through the simulation results and different scenarios we have noticed the following observations:

- Increasing the number of power meters in a region, leads to high utilization and delay which requires higher data rates for xDSL and DS1 links.

- The MAMR shows better performance parameters (link utilization, queuing delay, and throughput) over the case of client-server in the xDSL link. This is true for both wired and wireless networked energy meters. We could not measure it for the DS1 link because of RAM limitations.
- For the Multiple MAMR, we could simulate two MAMRS only and not very clear enhancement in performance parameters was observed over the single MAMR case. This may not be the case for more MAMRS, but more RAM is needed for simulation.
- For a typical xDSL communication link configuration between power meters and power company , we observed that maximum number of energy meters in each site preferred to be limited to 50 meters, other wise a higher xDSL bandwidth should be used.

3.4 AMR and MAMR Networks Simulations using JADE

The power meter platform PMP is assumed to be a power meter running an agent platform that is capable of hosting mobile agents, MAMRs. In this simulation we installed the JADE environment on one of the computers representing the energy company platform. The energy/power meters are simulated using computers running only Java Run Time Machine (JVM) with a small library of functions that can be installed on each computer[14]. The size of the JVM and the JAVA library does not exceed 1 Mbyte which is acceptable for practical purposes if the energy meters will be running such environment in the future. Each energy meter is running a process that simulates reading a multiple power load and storing the consumption value of each load in a local database. The computer simulating the energy company creates the MAMR and sends it to the first computer (power meter), the MAMR reads the local database and moves to the next computer and so on. The last computer send back the MAMR to the energy company computer with all the collected readings from each meter and hence the MAMR updates the energy company database with the read data through its tour.

The testing environment consisted of a lab with 12 computers using one of them as the energy company platform and the other 11 computers as the energy meters to be visited. On the other hand, the same system is used to let each meter send its data to the energy company computer/server using traditional client-server techniques after opening a database connection with the energy company server. The stored data of both methods were compared and checked for correctness. The design and implementation of this system is too large to be presented in this paper. However the results showed the workability of this system 100%. We are trying to test this system with the company platform being in another network using a routed WAN link while increasing the number of computers representing the energy meters to be in different labs and networks. Hence performance parameters can be tested and measured for this enlarged environment.

5. Security in Power Metering Systems

It is clear that the Power metering system is n-tier enterprise architecture. Hence, general security requirements and planning applies for this system. [5] Presents similar projects that can be used to design and implement such secure systems.

5.1. MAMR Security Issues

There are many security issues that should be considered in designing MAMR security protocols. This consists of masquerading, denial of service, eavesdropping, and alteration. Common mechanisms addressing these issues include cryptographic authentication and integrity checks, code signing and encryption, etc. The main threats in such a system consist of the Power Meter Platform (PMP) to MAMR threat, the MAMR to PMP threat and the MAMR to MAMR threat. The main security tools used in a PMP and MAMR system are encryption, signatures, and hashing. The system will use both asymmetric public key infrastructure and a symmetric private key infrastructure. These security tools will be used to attain confidentiality, integrity, availability and accountability (logging) of the power meter readings collection process.

5.2. Related MA Security Protocols

Mobile Agent systems raise a well-known set of security issues [6,7,9] that have to be addressed. These security concerns can be classified in two broad categories, according to whether the agent's or the platform's security is of concern [8].

Many protocols address the confidentiality and integrity problems with different degrees of success [8]. In any case, most of these protocols are based on standard cryptographic schemes, often relying on public key infrastructures, via platform driven protection mechanisms. Unfortunately, these security solutions often rely on a static prospect, implying the modification of all the involved platforms.

5.3. Suggested TNMP Protocol

The already existing MA data collection protocols solves problems like protecting data or agent code and most of them wait until the end to discover if the agent or data has been altered or tampered with, others discover this earlier but all do destroy both the data and MA when they discover this tampering. This may cause low data collection utilization since the collected data will be

rejected even if 1 bit of huge collected data is tampered and a new MA is sent again to collect the data, which may return back with the same problem. Our protocol depends basically on not moving the MAMR to another host unless it is sure that the new host is not malicious and is a trusted one. This is achieved by sending an inspection agent IMA to check the status of the next PMP before moving to it. Since this step is taken from a current MAMR running on a trusted host, it will not depend on a third party check and this decision is taken by the collecting MAMR before moving to the next PMP. We will apply the TNMP to the Power Meter Readings collection Problem using MAMR running at the current trusted Power meter PMP_i and checking the next PMP_{i+1} for safety conditions before moving to it. Although the way of protecting and collecting data looks similar to that in the self protecting MA protocol [8], there are many improvements and new ideas in achieving this way of data collection in a secure manner.

5.3.1. *Power Company Facility*

For the purpose of secure data collection, we assume that the power company facility is by itself trusted, secure and protected. And it is responsible for creation of the MAMR, and IMA agents and receiving the MAMR with the final collected encrypted data, Figure 11. This facility is also responsible of certifying each power meter and the server software installed on each power meter and issues a certificate for each meter using the power company's private key and storing this certificate in the FPGA card attached to each power meter.

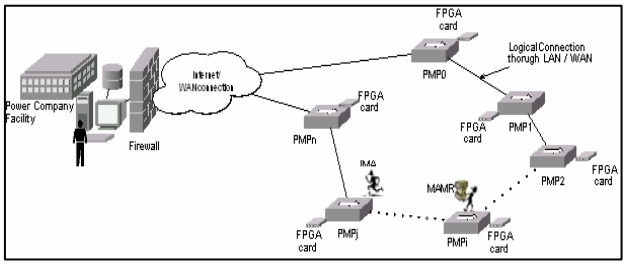

Figure 11 Secure Collection using PMP, MAMR and IMA

5.3.2. *Power Meters Platform (PMP)*

Each power meter platform PMP_i contains the following components as shown in Figure 12. The FPGA card forms an interface point that is capable of executing all required security and compression functions like encryption, decryption, hashing, digital signing and random number generation. It also stores a certificate $cert_i$ that is generated by the power company facility. This certificate includes the power meter identity and the hashing of the basic software functions code of each PMP server. That is the server code in each PMP is hashed and signed using the power company private key.

$$cert_i = S_{e_{pc}}(H(Code_{PMP_i}, ID_{PMP_i})) \quad (5.1)$$

$S_{e_{Pc}}$ Is a signing function computed using the private key of the power company facility. $H()$ Is a collision free one-way hashing function, $Code_{pmp_i}$ is the basic code of the functions stored at the power meter platform with identity ID_{PMP_i} .

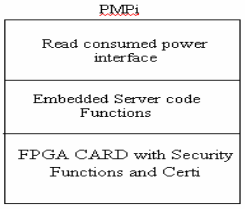

Figure 12 PMP Components

5.3.3. *MAMR Structure*

MAMR is used in collecting the power meter readings from each trusted PMP. The agent can be represented as a $(C_{MAMR}, R, cert_{MAMR})$, where R is split into platform-specific data chunks mainly composed of the Power Meter Readings:

$$cert_{MAMR} = S_{e_{pc}}(H(C_{MAMR}))$$

$$R = R_0, \ldots, R_n \quad (5.2)$$

$$R_i = E_{d_{pc}}(r_i, T_i, ID_{PMP_i}, S_{e_{session}}(H(r_i, T_i, ID_{PMP_i})))$$

Where $cert_{MAMR}$ denotes the MAMR code certificate signed by the Power company private key e_{pc} . $S_{e_{session}}(H(d_i, T_i, ID_{PMP_i}))$ Denotes the meter-reading certificate signed by the session private key together with the meter identification number and the time stamp T_i at which the reading was taken. This session private and public keys pair $(S_{e_{session}}, S_{d_{session}})$ will be generated before moving the MAMR to the next meter. It is clear that IMA and MAMR are capable of executing any of the library functions stored in the FPGA card which is a secure hardware electronic library.

5.3.4. *Inspection MA structure (IMA)*

IMA is a mobile agent used in inspecting the next PMP before moving the MAMR to it. IMA can be represented as $(C_{IMA}, D_{IMA}, cert_{IMA}, d_{session})$ where

$cert_{IMA}$ is the IMA certificate signed by the power company private key e_{pc}. And D_{IMA} are the encrypted session parameters using the session private key. That is:

$$cert_{IMA} = S_{e_{pc}}(H(C_{IMA}))$$
$$D_{IMA} = E_{e_{session}}(T_i, ID_{PMP_i}, S_{e_{session}}(H(T_{session}, ID_{PMP_i}))) \quad (5.3)$$

5.3.5. TNMP modeling

- The Power Company creates the MAMR and the IMA with all the tokens related to both agents. MAMR, IMA, PMP_0 takes the following actions:
- MAMR generates $(S_{e_{session}}, S_{d_{session}})$ pair. And calculates:

$$D_{IMA} = E_{e_{session}}(T_i, ID_{PMP_{pc}}, S_{e_{session}}(H(T_{session}, ID_{PMP_{pc}}))) \quad (5.4)$$

- IMA is sent to PMP_0 using a VPN, IPSEC or SSL session.
- PMP_0 Authenticates the IMA and checks for it's code integrity by using d_{pc} public keys and calculating h'_{IMA}, h''_{IMA}:

$$h'_{IMA} = Dec_{d_{pc}}(S_{e_{pc}}(H(C_{IMA}))) \quad (5.5)$$

$$h''_{IMA} = H(C_{IMA}) \quad (5.6)$$

- If both are equal then IMA code is correct, and then it uses $d_{session}$ to decrypt the received D_{IMA} and extract the other parameters to make sure that it is coming from the intended previous PMP and using the session time to check for non-replay attacks.
- Now since IMA is ok, it starts checking the platform by hashing its code and identity together with any other parameters that were used in creating the certificate of the PMP (that is the parameters used by the power company to certify the PMP), then the IMA encrypts the calculated hash, the stored PMP certificate (in the FPGA card), and other parameters like time and identification and sends it using VPN, back to the MAMR (which is assumed to be running on the previous trusted PMP). That is it sends:

$$E_{d_{session}}(H(Code_{PMP_i}, ID_{PMP_0}), ID_{PMP_0}, T_{session}, cert_i, S_{d_{session}}(H(Code_{PMP_i}, ID_{PMP_0}))) \quad (5.7)$$

- When MAMR receives this information, it decrypts it using $e_{session}$ and extract the information the $cert_i$ using the d_{pc} key and performs the necessary comparisons to make sure that the next PMP is trusted and not tampered and its basic code functions are safe. It then decides to move to PMP_i (if MAMR is currently at the power company facility it will move to PMP_0) since it is safe. The move is done using VPN session between the PMP's (that is PMP_{i-1} to PMP_i). Before moving MAMR destroys the unneeded parameters like the session keys which should be generated each time it send the IMA to the next PMP.
- Once MAMR is at PMP_i, it will be authenticated and checked for integrity for both data and code in the same manner as the IMA was checked.
- MAMR now gets PMP_i readings and encapsulates it in the R part using:

$$R_i = E_{d_{pc}}(r_i, T_i, ID_{PMP_i}, S_{e_{session}}(H(d_i, T_i, ID_{PMP_i}))) \quad (5.8)$$

- MAMR can perform other management routines on the PMP as required by the power company and prepares for the next trusted move.
- If the next PMP_i is not functioning or it has been tampered or a time out for not receiving the sent IMA occurs, the MAMR decides not to move to this un-trusted PMP_i and prepares for the next move to PMP_{i+1}. It can also keep a report that PMP_i was down or tampered until it gets back to the power company. The TNMP guarantees that the collected data until PMP_{i-1} were correct, hence no need to discard this collected readings or abort the MAMR mission as done by many other protocols.

When the MAMR finishes collecting the readings it returns back to the power company facility with R set of collected readings encrypted using the Power Company's public key. The power company is the only one who can get this data after checking the MAMR authenticity and integrity. Also it gets the report of malfunctioning PMPs and takes the required actions to repair and recertify them.

5.3.6. TNMP Simulations using JADE

In this simulation we used the testing environment described in section 3.4. PMPs are simulated using computers running only Java Run Time Machine (JVM) with a library of required security functions that is installed on each computer [14]. This library is assumed to be secure and can't be changed during running the protocol to simulate the FPGA functionality. Also we assume that each library (instead of FPGA) stores a digital certificate that identifies each computer (power meter) from another. The design and implementation of this protocol is too large to be presented in this paper. However the results showed the workability of the TNMP

100%. Future work will be done to test the efficiency of the TNMP and compare it with other protocols.

Lastly the following properties can be summarized for the TNMP:

- Supports confidentiality, authenticity, data integrity, truncation resilience, insertion resilience, and non-repudiaple attacks.
- Self protecting, that is code and readings of IMA and MAMR are protected while migration.
- No need for updates, since rounds of periodic readings is decided by the Power Company facility.
- Collects only trusted data so no need for destruction or ignorance of collected data.
- Report of malfunctioning PMPs and/or malicious hosts to be fixed by the power company.
- All data transfer between PMPs can be done using VPN, SSL, and IPSEC techniques so attacks can be minimized and detected.
- Collected data readings should be higher than older collected data so proof of not tampering is increasing the trust of read data.

Acknowledgment

The authors would like to express their cordial thanks to PPU University administration and students for helping us in using the labs and implementing the testing systems.

References

[1] R. Tahboub, E. Abu-Garbyeh, V. Lazarescu, “Micro Web-Servers for Remote Device Control”, 6th JIEEEC, Jordan, March 14-16-2006.

[2] E. Abu-Garbyeh, R. Tahboub, V. Lazarescu, “Home Automation Using Micro Web-Servers”, 6th JIEEEC, IEEE, Jordan, March 14-16-2006.

[3] R. Tahboub, D. Lazarescu, V. lazarescu, I. A. Saroit, "Intelligent Secure Management of Electric Power Organizations: Data Collection Using Mobile Agents", 5th IBIMA2005, Egypt, Dec 13-15, 2005.

[4] R. Tahboub, V. Lazarescu, “Mobile Agents in Remote Energy Meter Reading and Management Systems”, ECAI 2005 - International Conference, Electronics, Computers and Artificial Intelligence, Piteşti , Romania, July 1-2, 2005.

[5] R. Tahboub, M. Alsaheb, “Security in Database Web Access”, Sixth Scientific Conference on IT, Baghdad, Iraq, November 2000.

[6] Man Young Rhee, “Internet Security, Cryptographic Principles Algorithms and Protocols”, John Wiley & Sons Ltd, 2003.

[7] Carlos A. Osorio Urzúa, “Bits of Power: The Involvement of Municipal Electric Utilities in Broadband Services”, Massachusetts Institute of Technology, June 2004.

[8] J. Ametller, S. Robles, J. A. Ortega-Ruiz, “Self-Protected Mobile Agents”, AAMAS'04, July 19-23, 2004.

[9] Sergio Loureiro, Refik Molva, and Alain Pannetrat. "Secure Data Collection with Updates.", Electronic Commerce Research Journal, vol.1, No 2., February/March 2001.

[10] Robert Gray, David Kotz, George Cybenko and Daniela Rus."Mobile agents: Motivations and state-of-the-art systems", Thayer School of Engineering / Department of Computer Science- Dartmouth College. Hanover, April 19, 2000.

[11] M. Bakhouya, J. Gaber and A. Koukam, "Observations on Client-Server and Mobile Agent Paradigms for Resource Allocation". Proceedings of the International Parallel and Distributed Processing Symposium IEEE (IPDPS.02).2002.

[12] Brewington, R. Gray, K. Moizumi, D. Kotz, G. Cybenko, and D. Rus, Mobile agents for distributed information retrieval. In Mathias Klusch, editor, Intelligent Information Agents, chapter 15, pp. 355-395, Springer-Verlag, 1999.

[13] Katsuhiro Moizumi. Thayer, “Mobile Agent Planning Problems”. A Thesis Submitted to the Faculty in partial fulfillment of the requirements for the degree of Doctor of Philosophy. School of Engineering Dartmouth College. Hanover, New Hampshire. October 23, 1998.

[14] www.opnet.com, www.fipa.org, www.sun.com .

Radwan Tahboub: Received the B.sc. degree in EE Eng. from METU and M.sc. in CSE from Bilkent university in Ankara Turkey, 1992. Currently he is a PHD student at the Applied Electronics and Information Engineering, PUB. He works as a lecturer at PPU University since 1994, and worked as an IT consultant for many organizations including HEPCO, SEDCO power companies. His research interests include networking, security, operating systems and MM.

Dan Lazarescu: received B.Sc. and M.Sc. in electronic engineering from Politehnica University of Bucharest, Romania, in 2000. He is currently Ph.D. student at the Applied Electronics and Information Engineering, PUB. From 2000 he is with Robert Bosch GmbH, Germany. His interests include architectures and algorithms for automotive signal processing, hardware and software code sign for microprocessor applications.

Vasile N. Lazarescu received the B.Sc., M.Sc. (1970) and Ph.D. (1983) degrees in electronic engineering from Politehnica University of Bucharest. He is currently professor with the Applied Electronics and Information Engineering Dept., PUB, Romania. He has authored /coauthored 12 books and published more then 100 refereed journal and conference papers. His research and teaching interests include signal processing, computational intelligence, data acquisition and processing, computer architecture.