A Tactical Information Management Middleware for Resource-constrained Mobile P2P Networks*

Bo Xu[1], Mark Linderman[2], Sanjay Madria[3] and Ouri Wolfson[1]

[1]Department of Computer Science, University of Illinois at Chicago, USA
[2]Air Force Research Lab, Rome, USA
[3]Department of Computer Science, Missouri University of Science and Technology, USA

Abstract— **In this paper we provide an architecture for Tactical Information Middleware for bandwidth constrained information management. We propose the ideas of rank-based data dissemination, and the use of a tactical information management query language. These ideas will deal with dynamic changes in bandwidth and explore opportunistic data dissemination. Thus, will lead to a cross layer design of a system capable of handling the dynamic data management issues relevant in many mission critical applications.**

I. INTRODUCTION

Air force tactical missions such as collaborative reconnaissance involve Unmanned Air Vehicles (UAVs), ground stations, and dismounted users. Such missions often run applications which disseminate video/imagery, requiring stringent levels of quality of service (QoS). Furthermore, this tactical level QoS for information management needs to be achieved under environmental and operational constraints such as intermittent network connectivity, low bandwidth, limited CPU and battery power. Tactical Information Management (TIM) is the capability of providing QoS in a dynamic topology under bandwidth constraints. Similar conditions will exist in future mobile P2P environments that involve vehicles, mobile devices such as cellular phones carried by pedestrians, and stationary computers.

We propose to address the TIM needs by building a **Tactical Information Middleware (TIMW)** that uses a declarative language like SQL, specifically designed for intermittent and low/dynamic connectivity environments. The TIMW will receive queries from the application layer, and it will collaborate with the Radio-Communication Layer to process them in a mobile, distributed fashion.

Current technology such as OMG Data Distribution Service (DDS), used in many DoD applications, promises to deliver the QoS assurances but is limited in its capability to address the resource limitations of tactical missions. The proposed ideas will enhance the current functionality of systems such as AIMS [15].

The rest of the paper is organized as follows. In section II we introduce the environment. In section III we describe the architecture of the TIMW. In sections IV to VI we discuss each individual component of TIMW respectively.

[1] Research supported by NSF DGE-0549489, IIS-0957394, and IIS-0847680.

II. ENVIRONMENT

The environment is a *mesh network* which has a wired part and a wireless part. The nodes (henceforth called peers) in the wired part are referred to as *wired peers*. Wired peers are static and are connected via wirelines. Any pair of wired peers can communicate with each other all the time at high bandwidth. The peers in the wireless part operate within a mobile P2P network and are referred to as *wireless peers*. Wireless peers can be static or mobile and they communicate with each other via peer-to-peer radio communication. The peer-to-peer radio capability is associated with a *transmission range r*, which is the maximum physical distance between communicating wireless peers. Wireless peers that are within transmission-range are called *neighbors*. Some wired or wireless nodes in the mesh network that are long-lived and have extended storage and are called *servers*.

The wired and wireless networks communicate with each other via *gateways*, which are peers that are both wired and wireless. For example, a gateway may be an access point (also called a hotspot) which is a wired peer capable of peer-to-peer radio communication.

At any point in time, the *network topology* of the mesh network is an undirected graph, where each peer is a vertex; there is a link between each pair of wired peers, and between each pair of neighboring wireless peers (see Figure 1).

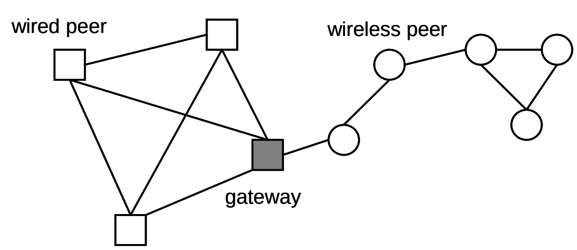

Figure 1. An example network topology of the mesh network

Mobile peers move according to trajectories. A *trajectory* of a peer defines the location of the peer as a function of time. It may be fixed or random. Some fixed trajectories may be known *a priori* to some or all of the peers. For example, an airborne platform orbiting in a racetrack pattern may have a trajectory known to many peers, while an airplane traversing through unexpectedly may not.

Some of the wireless peers are mobile, and thus the network topology changes over time depending on the trajectories of the peers. When a peer knows or can estimate the trajectories of other peers in the network, it is able to predict a portion of the network topology for some time into the future.

Occasionally, a peer P_i produces a *Managed Information Object (MIO)* which comprises a payload (e.g., document, imagery, or video clip) and metadata (e.g., topic, time stamp, location stamp, and possibly a description of the video content). Each P_i has a *local MIO database*, denoted $MIODB_i$, which stores some of the MIOs that P_i has produced or has received from neighbors by P2P communication[2]. We define the global MIO database $MIODB = \bigcup_{i=1}^{n} MIODB_i$. Thus, at any point in time each $MIODB_i$ is a subset of the MIOs in $MIODB$, and since MIOs may be replicated, the different $MIODB_i$s may overlap.

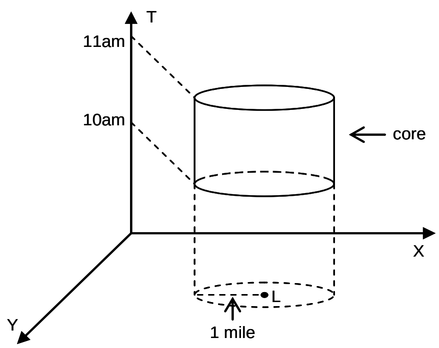

Figure 2. An illustrative example of core.

From time to time, each peer P_i may produce queries that express its interest in information. Each query has a temporal *extent* that restricts the age of MIOs in which it is interested. In addition, each query has a *lifetime* that is the interval of time during which it is interested in receiving results. A traditional (*ad hoc*) query may have a temporal extent [–∞, now] and a lifetime of [now, now+60s], and standing query may have a temporal extent of [now-60s, ∞] and a lifetime of [now, now+3600s]. Furthermore the boundaries of the lifetime and extent parameters may involve predicates and events. For example, the lifetime of the query may be [now, next-missile-shooting]. In this sense, the full power of a temporal logic may be used (see e.g. [19]).

A peer may also have a *core* that is a spatio-temporal extent representing its desire for MIOs. For example, a node may have a standing query for its core that solicits all the MIOs whose location stamps are within 1 mile from a location L and whose time stamps are between 10am and 11am. In this case, the core corresponds to a cylinder in a 3D (ignoring altitude) space as shown in Figure 2. A core may be fixed, or it may vary over time. For example, a peer is responsible for a particular area that may evolve based on its trajectory. The core can help in optimizing query results as peers may have only partially intersecting cores which will reduce the number of results as well as reduce the latency. Also, peers can pass their cores to their neighbors in case they plan on disconnecting to handle failures. Core concept can also address memory constraint which may exist at each peer.

When two wireless peers come within transmission range of each other, they *interact* by following a data dissemination protocol to exchange portions of their local databases. The data dissemination protocol will be discussed in section IV. Each exchange is assigned a *communication budget* [10] which limits the number of bytes that can be communicated during the exchange. The communication budget may be determined based on bandwidth constraint or energy constraint or the combination of the two. In addition, in order to save bandwidth and latency in exchanging information, two peers can share their bloom filters and based on that they exchange information which are unique to each peer.

III. Architecture of TIMW

The architecture of TIMW is shown in Figure 3. TIMW resides between the radio/wire communication layer and the application layer. Queries received from the application via the application programming interface are expressed in a language called *Tactical Information Query Language* (TIQL), which is an extension to SQL. The extension has special constructs that deal with resource constraints, particularly low bandwidth, and with multimedia, spatio-temporal information, and moving objects. The query processing utilizes a data dissemination component called *Rank-based Opportunistic Dissemination*. This component ranks MIOs so that the most important ones are exchanged within the assigned communication budget. The *Bandwidth Prediction* component predicts the available bandwidth of a link. The estimation is used to setup the communication budget. In sections IV to VI we discuss each of these components respectively.

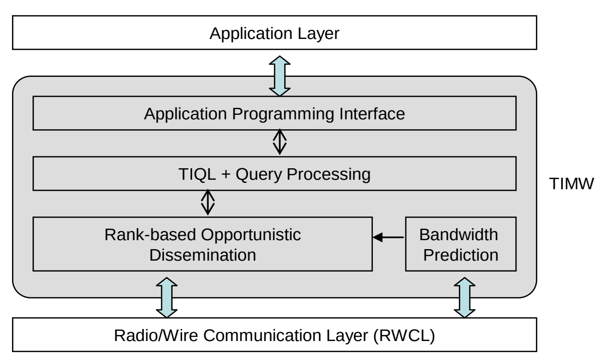

Figure 3. The architecture of TIMW on one peer.

IV. Rank-based Opportunistic Data Dissemination

Mobile peers impose challenges on data dissemination due to their mobility and resource constrains (bandwidth, memory, and energy). Thus in this section we focus on data dissemination among mobile peers or between mobile peers and static peers. In order to cope with peer mobility and

[2] Due to storage constraints, not all the produced and received MIOs are stored.

network disconnections, we adopt an opportunistic data dissemination paradigm. In this paradigm, when two peers come in contact, they answer queries (traditional as well as standing) carried by each other. Often in low-bandwidth environments, only metadata and possibly small video segments or still images can be exchanged [12]. So the set of answers to each query may be compressed in this fashion. Still, all results to the answered queries in an interaction may not fit in the communication budget. If so, the set of MIOs are ranked, and the most important ones are transmitted.

The rest of this section is organized as follows. In subsection IV.A we discuss the ranking of MIOs. In subsection IV.B we discuss data dissemination strategies in terms of push versus pull.

A. *Ranking of MIOs*

The rank of a MIO *m* represents the relevance *m* to the mesh network overall, and is computed by a ranking function which is specified *a priori* by the user to the TIMW, based on parameters known at the time of exchange. Some examples of ranking parameters are:

1. Relevance to queries. The degree to which *m* satisfies each of the queries answered in the interaction.

2. Query priority. The priorities of the queries that *m* satisfies.

3. Query lifetime. Each query may be associated with a time duration for which it is effective. If the queries that *m* satisfies are expected to expire prior to successful delivery, then the rank of *m* should be low.

4. Age. How current is *m*?

5. Access cost. The access cost in turn may depend on parameters such as the size of the MIO.

6. Source-Reliability. How reliable is the source that produced the MIO.

7. Replication. The more widely the MIO has been replicated, the lower the rank. The degree of replication may be inferred by a machine learning algorithm from parameters such as the age of the MIO and the number of times it has been received by a peer. This is an approach we investigated in prior work [1]. The locations at which the MIO is replicated are also important. A higher the availability of these locations implies a lower the rank, because it is likely that the MIO can be obtained from other locations.

The ranking function is specified using a query language which will be discussed in section V. Some ranking functions have been developed in [1,2,13,14], each based on some of the above parameters.

The relevance of an MIO to a query may be learned by relevance feedback. The feedback may be provided automatically or with a human involvement. The feedback may be provided in one hop or in multi-hops. In the one hop case, the feedback is provided by the receiver immediately after the reception of the MIO. In the multi-hop case, the feedback is provided by the receiver after it moves out of the transmission range of the sender. In this case, the feedback may be returned to the sender by multi-hop transmission, or it may be sent to the sender upon the next encounter between the sender and the receiver. The feedback can be in the form of receipts which can provide the information.

B. *Opportunistic Data Dissemination Strategies*

Opportunistic data dissemination may adopt push (data-to-query) or pull (query-to-data). Push means that in each interaction, after answering queries, if additional communication budget is available, a peer will push additional data that is not queried. This type of push has been studied in [13]. Pull means that in each interaction, a peer only transmits MIOs that are queried even though there is additional communication budget available. Peers may use pull in order to, for example, save energy. Observe that there can be several variants of push. For example, push may occur before answering queries so that high priority MIOs may be transmitted first. Furthermore, due to size-differences, the metadata and the payload may be pushed independently. For example, a peer may push 10 metadata descriptions rather than 1 complete MIO. Since metadata descriptions are smaller than their payload counterparts, they propagate faster and possibly meet more queries within a given time period. The purpose of disseminating metadata descriptions will become clear shortly.

In terms of the dissemination of queries, again there are several options. One option is that a peer only transmits its own queries in an interaction. Another option is that it transmits other peers' queries as well. In this case, the dissemination of metadata descriptions helps the peer prioritize other peers' queries; the queries that have matching metadata descriptions have higher priority to be transmitted. This is because these queries will trigger the transmission of the answer MIOs when meeting them.

In general, the balance between push and pull depends on factors such as the MIO generation rate, the query generation rate, and the overlap among queries from different peers. Intuitively, push is more effective when the MIO generation rate is relatively low compared to the query generation rate, or when there are big overlaps among queries from different peers (so that a dissemination of a MIO can satisfy many queries). It has been demonstrated that the combination of push and pull is superior to pure push and pure pull, in a static environment (see e.g., [17]). In this case, the dissemination of MIOs/queries follows a geometric structure, such as line segments or trees, based on the query generation rate and the MIO generation rate. For example, in [17], each peer pushes its data to a certain neighborhood resembling a needle and the query is disseminated only to a subset of the network resembling a comb. Combs are finer and needles shorter when the query generation rate is relatively low compared with the MIO generation rate, and vice versa. Such structure-based methods do not work in mobile peer-to-peer networks due to mobility and disconnections. The open problem is how to choose between or combine push and pull in opportunistic data dissemination. Possible separation between payloads and metadata further complicates the problem.

V. TACTICAL INFORMATION QUERY LANGUAGE

Queries are posed in extension to SQL called Tactical Information Query Language (TIQL). The extension has special constructs that deal with resource constraints, particularly low bandwidth, and with multimedia, spatio-temporal information, and moving objects. Multimedia, spatio-temporal, and moving objects constructs have been studied previously (see e.g., [6,7,8]), and thus in this section we elaborate on low bandwidth constraints. TIQL has two types of constructs: SQL query clauses and directives for configuration of the TIWM layer. In subsections V.A and V.B we discuss these two construct types, respectively.

A. *SQL Clauses*

Some existing SQL constructs can be used for coping with low bandwidth, although they were not designed for this purpose. For example, the *order by* clause of SQL allows a user to specify the criteria for ranking the answer set of the query. The ranking function takes as a parameter one or more attributes (e.g., physical distance) as discussed in section IV. An example SQL sentence is as follows.

select ** **from** MIOS **where** distance<1 mile **order by** distance/(age+1)*(1-degree_of_replication)*

This sentence asks for MIOs that are produced within one mile away from the query originator. It specifies the ranking function as distance/(age+1)*(1–degree_of_replication). distance/(age+1) computes the relevance of a MIO to the query under the condition that the MIO is new to (i.e., has not been received by) the query originator. degree_of_replication is the (estimated) fraction of peers that have received the MIO and thus 1–degree_of_replication computes the probability that the MIO is new to the query originator. Thus distance/(age+1)*(1–degree_of_replication) represents the expected relevance of the MIO. An example of such a query is "how many peers have reported traffic-jam within one mile distance".

Another way of dealing with limited bandwidth is to *compress* each answer MIO. The compression discussed here is based on the payload. *Payload-based compression* selects a still image or a segment of a video, or a low-resolution version of the image. *Metadata-based aggregation* also merges multiple answer MIOs into one based on the context recorded in the metadata. For example, two MIOs may be merged into one if the payload are similar, the distance between their location stamps is smaller than 100 feet and the difference between their time stamps is smaller than 10 seconds. These are possible parameters of the compress clause of TIQL. In the following we give an example for payload based compression and an example for aggregation.

select ** **from** MIOS **where** type=image **and** distance<1 mile **payload_compress** 1 meters/pixel.*

This sentence asks for image MIOs captured within 1 mile away from the query originator. It requests that answer images be compressed to resolution 1 meters/pixel.

select ** **from** MIOS **where** type=image **and** distance<1 mile **aggregate when distance<**100m **and time <**10 seconds **and** similarity>85%*

This sentence requests that MIOs be aggregated if: (1) image data is similar, and (2) if the distance between their location stamps is smaller than 100m, and (3) the difference between their time stamps is smaller than 10 seconds.

select ** **from** MIOS **where** type=image **and** distance<1 mile **decimate when distance<**100m **and time<**3600 seconds*

This sentence requests a set of "canonical" MIOs such that collections of MIOs falling within the bounding box (100m and 3600s) are replaced by a member of the collection that will serve as a canonical member.

Another TIQL construct that addresses low bandwidth is the *lifetime* clause. This clause allows a user to specify whether the query is to retrieve only immediate results or the query represents the user's standing information need. The parameter of the life-time clause is a time duration which specifies how long the query persists and expected time to transmit. If lifetime is zero, then the query is evaluated against the current status of the local database, and the sequence of answers is returned subject to existing bandwidth constraints to provide results within the stipulated time. If lifetime is greater than zero, then the query is reevaluated within the lifetime as the local database changes. Upon reconnection of the query originator, the new answers are combined with the un-transmitted answers of the previous connection, and a new transmission sequence is created.

B. *Configuration Directives*

Previously we discussed the "order by" clause which enables sorting MIOs for a single query. During a peer-to-peer exchange, since there may be multiple queries to answer, several sorted result lists need to be merged to form a total order so that most important MIOs can be selected for transmission. Existing work such as CarTel [4] ranks MIOs across multiple queries by a *priority number* per-specified for each query. Answers to a higher priority query are ranked higher than those to a lower priority query. In our environment, the priority may not be an absolute number that can be determined at programming time, but may depend on what other queries need to be answered during an interaction. For example, the priority of a query becomes higher if it has significant overlap with other queries. This is because transmitting an answer to this query will satisfy many other queries.

The function for cross-query ranking is specified as a parameter for the TIWM layer instead of for an individual query. An example directive is as follows.

total_order by *relevance/size*

This directive specifies the cross-query ranking function to be relevance/size. Relevance is the sum of the relevance values computed by the "order by" clause (see section V.A) for all the incoming queries; size is the number of bytes occupied by MIO. According to this function, if a MIO satisfies many queries and it is small, then it has ranked high in the total order.

Another TIQL construct is *push-pull* which indicates how information push and pull are to be combined in order to overcome low bandwidth. An example directive is as follows.

set dissemination_mode *push* ***from*** *MIOS* ***where*** *distance<1 mile* ***order by*** *age*

This sentence sets the dissemination mode to push, which means that during each initial interaction, before answering queries, if additional communication budget is available, a peer will push data that is not queried. The data that will be pushed are the MIOs which are within 1 mile of the current location; these MIOs are ordered by age, such that if a disconnection occurs while pushing, the most recent MIOs have been transmitted.

VI. BANDWIDTH PREDICTION

Before payload transmission, it may be useful to estimate the available bandwidth between a sender and a receiver. Such estimation can control the number of connections made [11], and enable use of the available bandwidth by priority of the service. This is important for video transmission.

If bandwidth estimation is provided by the radio/wire communication layer (RWCL), the TIMW software will make use of this service. Otherwise, it will do the estimation itself by actively probing on per hop basis. By probing, each peer estimates the local bandwidth availability and compares the monitored data rate with a predefined value for the maximum rate. In the probing packet, the minimum spare bandwidth along the hop (or estimated path) can be stored and the source peer can be informed of this value. It can then decide the communication budget. An alternative is to monitor the effective bandwidth *while* transmitting the total order of MIO or simply specifying a upper limit on the time allotted for transmission. Bandwidth estimation may still play a role if choice of the prioritization policy is a function of available bandwidth [5].

Mechanisms that adapt the communication channel to be used may also depend on the density of peers. Using sensing and peer-to-peer messaging, a peer can determine other in-range peers and an ad hoc network can be established and used instead of the direct link to reach a server even if such a link is available. Again, this is a networking issue that may be addressed by the RWCL, but if not, the channel will be chosen by the TIMW layer.

VII. RELEVANT WORK

Mobile ad-hoc networks (MANETs) and delay/disruption tolerant networks (DTNs) [16]**.** Existing work on MANETs/DTNs focuses on routing, namely how to deliver a message from a source node to a destination node wherein the network id of the destination node is known to the source node. Our work, on the other hand, deals with intelligent query processing in MANETs/DTNs, which includes how to opportunistically disseminate queries/data to other peers, rank results and prioritize data in response to storing/queries considering the environmental constraints. Our problem is how to find answers without knowing the network ids of the answer producers.

Wireless sensor networks. Tinydb [18] addresses data-models and languages for sensors, but considers query processing in an environment of static peers. CarTel [4] addresses the translation of these abstractions to an environment in which vehicles transfer collected data to a central database via fixed access points.

Multimedia databases. Most work on multimedia databases has been on content based retrieval. In contrast, this paper addressed the issue of mobile peer-to-peer query processing. Our work is somewhat relevant to context-based multimedia retrieval (see e.g. [3]). In this type of retrieval, multimedia data is retrieved based on the context such as time and location in which they are created. However, existing work in this area assumes a central server model and concentrates on the matching algorithm.

REFERENCES

[1] Bo Xu, Ouri Wolfson. Flooding by Machine Learning in Mobile Ad Hoc Networks. *ACM Transactions on Adaptive and Autonomous Systems*, Vol. 4, No. 4, November 2009, pp. 1-36.

[2] T. Zhong, B. Xu, P. Szczurek, O. Wolfson. TRAFFICINFO: An Algorithm for VANET Dissemination of Real-time Traffic Information. 15th World Congress on Intelligent Transportation Systems, 2008.

[3] W. Viana, et al. Semantic Keyword-based Retrieval of Photos taken with Mobile Devices. *MoMM*, 2008.

[4] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, E. Shih, H. Balakrishnan, and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. In *Proc. ACM SenSys*, Nov. 2006.

[5] L. Bunch, J. Edmondson, J. Loyall, A. Sinclair, and M. Carvalho. Policy Specification and Enforcement for Quality-of-Service in Service-Oriented Information Management. Submitted to 11th International Middleware Conference, 2010.

[6] Zhongsheng Cao, Zongda Wu, Yuanzhen Wang. UMQL: A Unified Multimedia Query Language. *SITIS* 2007.

[7] Martin Erwig and Markus Schneider. STQL: A Spatio-Temporal Query Language. *Mining Spatio-Temporal Information Systems* (eds. R. Ladner, K. Shaw, and M. Abdelguerfi), 2002.

[8] R. H. Guting and M. Schneider. *Moving Objects Databases*. Morgan Kaufmann Publishers, 2005.

[9] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, E. Shih, H. Balakrishnan, and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. *SenSys* 2006.

[10] Anil Jade, Sanjay Kumar Madria, Mark Linderman: Incentive Based Routing Protocol for Mobile Peer to Peer Networks. Mobile Data Management 2009: 285-292.

[11] Hemanth Meka, Sanjay Kumar Madria, Mohan Kumar, Mark Linderman, Sharma Chakravarthy: Efficient Simulation Architecture for Routing and Replication in Mobile Peer to Peer Network of UAVs. Mobile Data Management 2010: 281-283.

[12] Kumar Abhinay Rathore, Sanjay Kumar Madria, Takahiro Hara: Adaptive searching and replication of images in mobile hierarchical peer-to-peer networks. Data Knowl. Eng. 63(3): 894-918 (2007).

[13] Bo Xu, Fatemeh Vafaee, Ouri Wolfson. In-network Query Processing in Mobile P2P Databases. *ACM GIS* 2009.

[14] F. Sailhan and V. Issarny. Energy-aware web caching for mobile terminals. *ICDCSW'02*.

[15] P. J. Ceccio. AIMS taking on roles to support tactical information dominance. *6578. International Society for Optical Engineering*, 2007.

[16] B. Burns, O. Brock, and B. N. Levine. MV routing and capacity building in disruption tolerant networks. *INFOCOM* 2005.

[17] X. Liu, Q. Huang, and Y. Zhang. Balancing Push and Pull for Efficient Information Discovery in Large-Scale Sensor Networks. *IEEE Trans. on Mobile Computing*, 6(3), 2007.

[18] S. R. Madden, M. Franklin, J. Hellerstein, W. Hong. TinyDB: an Acquisitional Query Processing System for Sensor Networks. *ACM Trans. on Database System*, 30(1), 2005, pp. 122-173.

[19] P. Sistla, O. Wolfson, S. Chamberlain, S. Dao. Modeling and Querying Moving Objects. *ICDE* 1997.