Adaptive Information Provisioning in Distributed Context Centric Architectures

Jamie Walters, Theo Kanter and Rahim Rahmani
Department of Computer and System Sciences, Stockholm University
Forum 100, Kista, Sweden

Abstract
The provisioning of user context information between service endpoints is central to realizing massive immersive participation on an Internet of Things. This information must in turn be provisioned to endpoints with minimal overhead costs. Where this is achieved through centralized repositories of context information there arises issues of scalability and availability. Where distributed approaches have been proposed, information dissemination has been optimized relative to the underlying network properties. In this paper we extend the Distributed Context Protocol (DCXP) to support subscriptions relative to an entity-application-entity triple, minimizing the number of subscriptions required and through application specific optimization minimize the overall cost of delivering user context information to service endpoints.

Keywords: *Context-Awareness, Immersive Participation, Context, Context Models, Internet Of Things, Context Proximity, Sensor Information, P2P Context.*

1. Introduction

Immersing users in massive participation scenarios varying from theatrical performances to online pervasive gaming is enriched through the availability of the context information driving the interactions among the underlying collection of connected *things*. Experiences such as Google Ingress [1] and Maryam [2] and are enabled by open solutions such as [3] and [4] require an Internet of Things (IoT) infrastructure that is capable of delivering this additional information thus defining our pervasive realities and providing the support for understanding the dynamic relationships that exist between a vast global collection of people, places and things.

Collecting these diverse kinds of context information has been addressed though the extensive use of wireless sensor networks supported by the increasing availability of network connections such as mobile Internet and fibre. This creates large volumes of user generated context information and raises issues with provisioning this information to service endpoints within specific real time constraints. Provisioning this information has taken two general approaches: centralized and cloud based infrastructures such as SenseWeb [5] and the IP MultiMedia Subsystem (IMS) or distributed approaches such as MediaSense [6] and SCOPE [7].

By aggregating centrally, approaches such as IMS reduce the cost of provisioning information to remote endpoints through algorithms and rules for selectively filtering mediating context information between user endpoints. One such approach, the address book based filtering is described by Petras et al. in [8]. Here, an entity watches or subscribes to other entities contained within its address book. Petras et. al estimated the average address book estimated to be no greater than *0.005 * global population* [8]. While this method indeed reduces the amount of information that must be mediated collections of context end points, it subsequently places hard limits on the ability to discover entities of interest with which to establish common context relationships. This, as relationships are limited in the first instance to those that are defined by the user's address book. Examining relationships outside of these hard limit hard limits would pose severe load constraints on such centralized services. This would in turn render these approaches non-scalable and simply and pruning the message queue as suggested by Petras et al. would offer little guarantee with regards to the quality and accuracy of the disseminated context information.

Additionally, such centralized approaches continually raise issues centered on security and privacy that are more greatly emphasized where vast amounts of information describing a user's behavior are available through vast singular repositories. Centralization further raises issues of bulk data corruption and information loss associated with server configuration errors and maintenance issues. Furthermore these approaches are dependent on DNS as a means of locating service portals, users and applications. Issues with DNS availability due to DoS attacks and configuration errors raises questions about its continued suitability and prompting research into Distributed Hash Table (DHT) based overlays possible replacements [8].

Solutions such as MediaSense [6] and SCOPE [7] approached the problem of massive information provisioning through the use of DHT's and peer-to-peer distributed algorithms. These approaches realised response times on par with UDP and deemed adequate enough to support real-time context dependent services. These approaches further showed that distributed platforms were more capable of meeting real-time provisioning constraints than corresponding centralized approaches. Distributed access to context information, however does not in itself adequately address the issues of accessing and provisioning context information while minimizing overheads costs. The distributed nature of the context information itself places hard requirements on end nodes' ability to locate context information over vast heterogeneous networks.

Yoo et. al in [9] and Santa et. al in [10] suggested that publish-subscribe approaches provided for the most suitable method to provisioning multi-dimensional context information. Kanter et. al in [11] further showed that such approaches realized communication performance on par with UDP signally used in SIP implementations, while benefiting from the scalability of being decentralized. Frey and Roman in [12] extended this approach to provide for event driven subscriptions in context networks. However, such approaches are centered on events rather than the underpinning context information itself.

These solutions however do not adequately address the issues associated with the distribution of the context information itself. Other approaches such as Kiani et. al in [13] and Sridevi et. al. in [14] while exploiting distributed approaches for provisioning user context information rely on single end points for disseminating context information to subscribers. Such single end points do not solve the issues associated with localized congestion and where the number of subscribers to a piece of context information is large, would have to resort to pruning subscriptions lists in order to scale with demand. Additionally, such end points are subjected to questions of availability being tightly coupled to the underlying network layers.

One approach to scaling the distribution of information across de-centralized architectures is through the use of multi-cast trees. Here, nodes on a P2P network participate both as consumers and distributors of information ensuring that the information is broadcasted to all interested end points while minimizing overhead costs. The construction of such multi-cast trees is achievable through multiple algorithmic approaches considering network location, round trip times, information relevance, or user groups. These have shown to provide good results in prioritizing the delivery of information while minimizing overall costs.

Seok et. al., in [15] implemented such a P2P multi-cast approach on top of Scribe [16] . Here, each entity creates a topic for each context dimension, which can then be subscribed to by the other entities across the platform. Here, all entities subscribe to interested end points and receive context information updates as changes occur. This is similar in approach to most centralized solutions with the added benefit that the dissemination costs are distributed by the overlay.

Multicast solutions optimized around the properties of the underlying network layer however undermine a move towards content centric networking and limits the ability to extend SDN type approaches to realizing context centric networks, where context information is provisioned relative to the relationships between entities existing at service end points.

This notion of relevance or proximity between context bearing entities is described by Zimmermann et al. in [17] as the overarching factor in establishing context relationships. This subsumed the earlier address book approaches and moved towards realizing truly context-centric networks where interactions, discovery and relationships are underpinned by the degrees of relationships between entities over their underlying context information. Zimmermann et al. equated the notion of proximity to spatiality, essentially disregarding the types of higher level relational proximity expected by Hong et al. in [18].

While semantic approaches such that described by Dobslaw et al. in [19], Toninelli et al in [20] and Liu et al. in [21] offer some support towards this problem, Adomavicius et al. in [22] suggested that these types of approaches are limited and should be complemented by metric type approaches thus realizing the ability to answer the question of *"nearness"* as posited by both Schilit et al. in [23] and Dey and [24]. This further characterization would permit us to better identify and establish context relations between related entities which, according to Hong et al. in [18], is critical in realizing applications and services that can discover nearby sensors or points of information.

Padovitz in [25] demonstrated that context relationships are defined and established within the universe of discourse of an application and as such do no exist outside the bounds of such a space. Schmohl in [26] views this as a well defined hypesphere where or boundary where context relationships are defined relative to an n-dimensional proximity value. Citing this we in [27] defined an approach to establishing context centric relationships between entities on an Internet of Things. Here, relationships are established between entities over the similarity of their underlying context behaviors relative to an application space. This extends the work of Zimmermann et al. in [17] towards a context-centric model while subsuming it with respects to expressiveness. This satisfies the initial requirement of a context relational model capable of supporting the establishing, adjusting and exploiting of implicit context based relationships in massive immersive environments. With this approach, we are capable of identifying and discovering candidate entities that can be fused to realize new user experiences and deliver more expressive applications and services

In this paper we introduce an adaptive approach to provisioning information on context centric architectures. To achieve this, we extend our existing publish-subscribe approach to introduce a subscription triple between two entities, executing this triple and its associated constraints on distributed end points. We introduce several new constraints that can be applied to such triples which respond to the changes in the context behavior between end points, In Chapter 2 we summarize existing background information and approaches in the area. In Chapter 3, we introduce our solution and discuss the proposed adaptive approaches. Chapter 4 discusses early analysis and results. Chapter 4 completes with a conclusion and discussion.

2. Background

In [28] we presented the MediaSense platform, a distributed peer to peer platform for provisioning context information on an Internet of Things. MediaSense consists of peer-to-peer nodes organized around a supporting distributed hash table (DHT) and implementing a Distributed Content Exchange Protocol (DCXP) for creating, discovering and provisioning user context information between distributed service endpoints.

2.1 The Distributed Content Exchange Protocol

DCXP is an application level P2P protocol used for offering reliable communication among the endpoints in the MediaSense platform [28]. An end point may, through registering with the MediaSense platform share context information. Context information is addressable using a naming scheme that is similar in structure and format to the URIs used in SIP.

The DCXP naming scheme uses Universal Context Identifiers (UCIs) to refer to Context Information such as sensors that are stored in the DCXP network. DCXP thus enables the exchange of context information between sources and sinks residing at endpoints and referred to as context user agents or CUA. In order to achieve this, DCXP employs six primitives, shown in Table 1.

An endpoint wishing to subscribe to the context information of an entity residing at another remote endpoint, issues a subscribe of the format:

$$SUBSCRIBE\ dcxp://Jamie@dsv.su.se/temperature$$

REGISTER	A CUA uses REGISTER to register the UCI.
RESOLVE	A RESOLVE is used to identify the endpoint where a piece of context information resides.
GET	A GET is used to retrieve the latest value associated with a UCI.
SET	A SET is used to modify the value of an actuator at an end point
NOTIFY	Provides notification about the latest information to subscribing CUAs every time an update occurs or if asked for an immediate update with GET.
SUBSCRIBE	SUBSCRIBE enables the CUA to start a subscription to a specified UCI, only receiving new information when the UCI is updated.

Table 1: DCXP Messaging Primitives

2.2 Relational Publish Subscribe

The DCXP protocol described in [29] relies on the address book approach to locating context entities with which to establish relationships. Disseminating context information relative to the underlying relational organization requires a new approach towards publish-subscribe solutions in context aware systems. Our previous work on the publish subscribe approach detailed in [6] supported primitives for getting or subscribing to the context information of an entity.

In contrast to Petras et al. in [8], such a solution was distributed minimizing issues of scalability. However like Petras et al. a watcher cannot subscribe to greater than the size of its address book. This is estimated in [8] to be around 0.005*N*, the number of global presentities. This however poses a hard limit on our ability to discovery related context entities as the watcher's address book should not ideally determine the number of presentities it watches, but rather be determined by the number of entities with which it can potentially establish a context centric relationship. As with centralized approaches the maximum possible number of subscriptions for each watcher would therefore be *N,* the global number of entities. This would in turn not scale well.

By considering the relation between context entities as a factor for driving the discovery and subsequent establishing of context relationships, we are able to make two key changes to the way in which context information is provisioned on distributed context based architectures. Firstly, we move away from the address book approach proposed by Petras et al. [8] to create a solution that is

capable of discovering related context based entities while minimizing subscription costs.
To this end this we extended the publish-subscribe approaches to enable subscriptions to relationships and areas of interest as defined by an underlying proximity function. A *watcher* issues a subscription for all **(a_{min}, a_{max})** of the underlying application space. This is issued to *range_brokers*, distributed nodes responsible for brokering ranges of values between *watchers* and their *presentities*. The *range_sub* or *range_get* is routed by the underlying support to the node or nodes in the data space that are responsible for the range of values in **(a_{min}, a_{max}).**

Each *presentity* subsequently publishes its current context value to the corresponding *range_broker*. The *range_broker* in turn sends the current set of states to the watcher. The *watcher* evaluates the list of states over its proximity function and establishes a context relationship with selected *presentities*. In order to maintain the relationship, the *watcher* then subscribes to the *presentities* and continually evaluates the relationship with each new context states received. The *watcher* also maintains a subscription with the range_*broker*, which continually send lists of context states matching the range subscription to the watcher. Context entities publish new context states as the supporting context information changes; this can range from very frequently in highly dynamic situations to seldomly in lesser dynamic situations.

2.3 Context Relational Model

Padovitz in [30] describes a general model for defining context spaces and interactions between context bearing entities. This universe of discourse of an application is the subset of all global context information considered relevant to all interactions relative to problem domain or application and supports the delivery of any application or service relative to this domain. .

This is modeled as an *n-dimensional* hyperspace, where dimension corresponds to a type, value subset and the domain for an element of context information. This domain is modeled as:

$$\forall \mathcal{D} = \{(d_1^{min} < d_1 > d_1^{max}), (d_2^{min} < d_2 > d_2^{max}), \ldots, (d_i^{min} < d_i > d_i^{max})\} \quad 1$$

In our immersive participation environment, such a domain could be the play "*Maryam*". A domain could be daily commuting. This domain universe is partitioned into situations or activities. Each sub-space represents an acceptable range of context information defining a real world situation or activity such that:

$$\forall \mathcal{A}^{\mathcal{D}} = \{(a_1^{min} < a_1 > a_1^{max}, \omega_1), (a_2^{min} < a_2, > a_2^{max}, \omega_2), \ldots, (a_i^{min} < a_i > a_i^{max}, \omega_i)\} \quad 2$$
$$\vdots\ (a_i^{min}, a_i, a_i^{max} \in d_i)$$

For the domain *Maryam*, activities could be *Scene 1, Scene 2, Scene 3*, etc. Activities definitions are not mutually exclusive and therefore several activities could overlap in their sub-space definition.

Finally, each situation contains context states; a combination of unique attribute values within a situation or activity space such that:

$$\forall S^{\mathcal{A}} = \{s_1, s_2, \ldots s_i\} \vdots\ s_i \in \mathcal{A}_i \quad 3$$

Each state corresponds to a context observation made on an entity. For the domain *Maryam,* a state would be the context information recorded from body sensors at *Scene 1*. A state may be occupied by one or more entities, each of which continually transits states within the context space. An entity within an application space is classified according to its current state information in order to determine the most likely situational space being occupied at a given point.

We extended this in [27], to introduced a dynamic heterogeneous approach to context relationships where the notion of context proximity is one that considers the situation, attributes, relations, accuracy and heterogeneity of both the underlying information and the vast array of requirements for metrics supporting application domains.

We define a similarity matrix between the activities within an application or domain space as shown in Table 2. This is an MxN matrix of real values between 0.0 and 1.0 conveying the similarity between activities in a domain space; the ease with which one activity can be transformed into another. Here an activity is not simply confined to primitive determinations such as walking, running, sitting laying but rather encompasses higher level notions of activities such as *going to work*, *going home*, *shopping*, *watching television*, *washing*, *cooking*.

These higher-level activities are not necessarily discernable from raw context information but can be derived by applying learning methods, human annotation and assumptions. The underlying context information could be very similar or even identical while the perceived higher-level activities are not.

Relational proximity is derived between the states of entities as observed over a time window W. For solving this, we used the Earth Movers Distance as described by Rubner et al. in [31] setting the distributions as the sets of observable context states for each window W, the weighted edges being the activity similarity between P and

Q and the ground distance d_{ij} being the distance between pairs of states s_i, s_j derivable as:

$$d_{ij}(s_i, s_j) = \frac{\left(\sum_{k=1}^{n}\left[w_a * \left|\mathcal{F}_a^{\mathcal{D}}(a_i, a_j)\right|^r\right]_k\right)^{\frac{1}{r}}}{\left(\sum_{k=1}^{n}\left[w_a * \left|\mathcal{F}_a^{\mathcal{D}}(a_i, a_j)_{max}\right|^r\right]_k\right)^{\frac{1}{r}}} \qquad 4$$

where $a_i \in \mathcal{A}_i^{\mathcal{D}}, a_j \in \mathcal{A}_j^{\mathcal{D}}$

Here, w is the weighting for each attribute. The value of r can be adjusted to reflect the perceived distance between P and Q as shown by Shahid et al. in [32]. The distance is normalized with respects to the maximum distance between states in the encompassing application space. Our measure of proximity therefore logically subsumes and extends existing $Lp - norm$ approaches.

The EMD algorithm is then applied to derive the largest possible transformation between P and Q that minimizes the overall context transformation cost, where:

$$WORK(P \rightarrow Q, \boldsymbol{F}) = \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{ij} \qquad 5$$

Subjected to the following constraints:

1. $f_{ij} \geq 0 \quad 1 \leq i \leq m,\ 1 \leq j \leq n$
2. $\sum_{i=1}^{m} f_{ij} \leq P \quad 1 \leq i \leq m$ 6
3. $\sum_{j=1}^{n} f_{ij} \leq Q \quad 1 \leq j \leq n$
4. $\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = min(\sum_{i=1}^{m} P, \sum_{j=1}^{n} Q)$

The first constraint permits the transformation and hence the proximity from $P \rightarrow Q$ and not the opposite. The second and third constraints limit the transformation $P \rightarrow Q$ to the maximum number of context observations made for P or Q. The final constraint forces the maximum transformation possible between both entities. The context proximity, $\delta_{(P,Q)}$, is therefore earthmover's distance normalized by the total flow.

$$\delta_{(P,Q)} = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{ij}\right) * \left(\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}\right)^{-1} \qquad 7$$

By using the maximum possible flow between P and Q. It is important to note, that $\delta_{(P,Q)}$ is indifferent to the size of both sets of observations and permits partial similarity where the behavior of P is subsumed by the behaviour of Q. Therefore $\delta_{(P,Q)} \mid w = \delta_{(P,Q)} \mid \frac{1}{2} w$. This is a distinct advantage of our approach and excess observations are inherently discarded.

3. Adaptive Context Provisioning

Citing the contributions of Padovitz in [25] and Schmohl in [26] we further extend the publish-subscribe model in the existing MediaSense platform to supporting subscriptions to entities relative to the domain of discourse of the overarching applications and services. Firstly an entity P discovers an entity Q through the range subscription extensions described Section 2.2. On evaluating the context relationship using the approach detailed in 3, entity P then subscribes to entity Q relative to the application A and subject to a set of constraints C This defines a subscription triple between entities and applications.

3.1 The Context Subscription Triple

The context subscription triple is a further extension of the DCXP *SUBSCRIBE* primitive to create a subscription triple where each triple is of the type:

$$SUBSCRIBE(Q, A, C)$$

This holds with Padovitz et. al in [30] limiting interactions to defined application spaces. This creates a subscription that can be evaluated over the relationship between (P, Q) subject to the constraints C of the application A. Where A defines an application space bounded within an n-dimensional hypercube. All interactions between interactions between P and Q are subsequently exist only in and are confined to one or more application spaces. This n-dimensional space is described in Section 2.3.

3.2 The Context Subscription Constraints

To support the adaptive provisioning of context information we introduce some basic application proximity constraints that can be applied to each subscription triple to adaptively mediate user context information between entities residing at service endpoints.

Context Proximity

We define context proximity as:

The amount of work required to transform the context behavior of one entity into that of over the characteristics of their current underlying context states

The context proximity is derived as the perceived multi-relational distance between the observed behaviors of pairwise context bearing entities. This is detailed in Section 2.3 and derived as the Earth Mover's Distance between sets of context states. As a constraint, the context proximity is used to adaptively provision context information relative to the perceived closeness in behaviors. While the proximity of two entities, P, Q are closer to 1 the context information is updated at a lower rate, which then increases as their proximities merge to 0. The maximum rate at which this can be done is the max rate of context updates of both entities.

Given that entity P has a rate of P^r and entity Q has a rate of Q^r, then the max application rate A^r_{max} = max (P^r, Q^r). Each application maintains a minimum rate A^r_{min} such that:

$$\lim_{\delta_{(P,Q)}\to 1}(A^r) = A^r_{min}$$

and:

$$\lim_{\delta_{(P,Q)}\to 0}(A^r) = \max(P^r, Q^r)$$

and the application rate is determined as:

$$(A^r) = \max(P^r, Q^r)$$

Entities that are further away are updated at a lower rate, which progressively increases as the entities merge. With this approach we adaptively update the context proximity with a frequency as a function of the last known proximity.

Convergence Factor

The convergence factor Cf is defined as the rate at which the proximity $\delta_{(P,Q)}$ between two entities changes with respect to time. Entities that are show little or no convergence are deemed as stable entities with respect to their proximities and the application rate (A^r) can subsequently be adjusted to reflect this such that:

$$\lim_{|Cf_{(P,Q)}|\to 1}(A^r) = \max(P^r, Q^r)$$

and:

$$\lim_{|Cf_{(P,Q)}|\to 0}(A^r) = A^r_{min}$$

Dimension Noise Tolerance

Dimension noise ratio (DNT) refers to the degree to which a dimension must change before it can be propagated to a state change and subsequently used to compute a new proximity and dispatched to the related end points. Each application subscribes to the dimension by specifying the DNT value. This is particularly important for dimensions such as location, where interference in GPS or course location techniques can result in a varying in dimension values with no real change in the underlying user's behavior.

Proximity Noise Tolerance

The proximity to noise ratio (PNT) between two entities defines the accuracy of the currently observed proximity between two entities. Noise from sensor information can serve to distort the derived proximity from what is considered the true proximity. This permits applications to filter updates that can be deemed as noise and as such offer no application proximity updates to end points when the proximity changes within this specified tolerance threshold.

State Change Tolerance

State change tolerance determines the number of dimensions that must be changed before an application space recognizes a state change valid to relative to the application. This occurs in scenarios where altitude of entity might change, however a state change is limited to a wider change in location including latitude, longitude and altitude. This change could therefore be excluded until all dimensions have changed above a DNT level to propagate a state change and a new application proximity value.

Activity Threshold

Applications can use the higher-level activity as the basis for evaluating the relationships between two entities. Here, proximities are not calculated and disseminated unless a change in activity relative to the application space has occurred. This satisfies scenarios where the underlying context information is less valuable to the realization of the applications or service. Instead the more human approachable activities are more meaningful. Given that:

A defines an application activity enclosing hyperspace A_v
A is comprised of dimensions $\{d_1, d_2, d_3 \ldots d_x\}$
That a user P updates her context states at a rate λ.
Then:

$$\lim_{A_v\to 0}(A^r) = \lambda$$

and

$$\lim_{A_v\to\infty}(A^r) = A^r_{min}$$

Proximity Threshold

A proximity threshold (PT) value provides for a minimum change in proximity before endpoints are notified.

3.3 Initialization

Each context entity P joins the MediaSense platform by associating with and residing at an endpoint. When entity joins, it firstly creates a rendezvous points for each of its context dimensions at an endpoint on the distributed platform.

This is achieved by routing the UCI of the context dimension on the underlying overlay network, assigning the rendezvous point to the terminal node.

Each dimension is therefore addressable by its UCI such that temperature would be:

$$dcxp://jamie@dsv.su.se/temp$$

When an entity Q wishes to subscribe to a UCI owned by entity P, it routes a $SUBSCRIBE$ to the UCI. This subscription message in turn sent to the rendezvous point from which accepts the subscription and records it in the subscription lists for each UCI.

A change in the value associated with one or more context dimensions results in the existence of a new context state associated with the entity P as described by Padovitz et. al in [30]. This new context state is sent to all the dimensions that are changed between time $t-1$ and t. This is achieved by routing the new context state the endpoint. The rendezvous point in turn publishes the context state to all subscribing end points.

3.4 Subscription

Firstly entity P discovers entity Q using the range subscription approached detailed in 2.2. Upon evaluating the context relationship using the approach detailed in 3, entity P then subscribes to entity Q relative to the application A. This is achieved by routing the subscription (P, Q, A) to the pseudo UCI determined as:

$$dcxp://P@dsv.su.se - dcxp://Q@dsv.su.se$$

Additionally, entity Q subscribing to entity P would route to the same UCI. This removes redundancies and the overhead associated with multiple subscriptions.

Each subscription (Q, A, C) arrives at a rendezvous point R which examines the application space and then issues a standard DCXP $SUBSCRIBE$ to all the composite dimensions of the application space. For each subscription triple that arrives at R a subscription is therefore made to each unsubscribed dimension in order to satisfy the dimensional constraints of each application. It is added to a dispatch queue Q with a priority of:

$$\delta_{(P\to Q)} * (1 - Cf_{(P,Q)})$$

Where $\delta_{(P\to Q)}$ is the current context proximity and $Cf_{(P,Q)}$is the rate of change of context proximity between both entities. Initially these values are set to 0 and the node placed at a random position in the queue. Using this approach the priority with which a node's proximity is calculated and dispatched is a function of the current proximity between both nodes and the rate at which they are converging or diverging. Here, stable nodes with larger proximities are evaluated and delivered last, while closer, more dynamic nodes are evaluated and delivered with the highest priority. More distant stable nodes are subsequently more readily pruned in order to keep the queue at an optimal.

3.5 Publication

An entity on observing a change in any of its underlying context dimensions issues a $NOTIFY$ message to its rendezvous point associated with the most recently changed dimension. Endpoints may add additional constraints such as error correction to determine what constitutes a state change such that a change in context state may not directly correlate with each change in context information.

Such an example would be an entity that enforces error thresholds on sensor data or where the change of context information is not significant enough to effect a change in the supported context relationships. Equally so are cases where the entity periodically updates its context state at regular intervals such as every 30seconds. When the entity publishes a context state change, it routes the new state p to the rendezvous point associated with each dimension changed such that:

$$ROUTE(dcxp://jamie@dsv.su.se/temp\ , p)$$

The rendezvous points each route p to each of the subscribing application rendezvous points. At each rendezvous point, the queue Q is processed in order where each triple is evaluated with respects to the applied constraints $C \in A$. The proximity is computed for each triple and satisfying all constraints, the proximity is then dispatched to both P and Q and the subscription updated in the queue with the new priority.

4. Results and Discussion

We implemented our approach on our distributed platform using a P2P overlay and the MediaSense publish subscribe interface. We simulated the proximity based adaptive queuing of context information dissemination in service end points by setting the average transmission delay of 10ms per subscription delivery reflecting UDP times. The simulation consisted of 1000 subscribers to each of 1000 context source points. This created a total subscription base of 10^6 across the entire MediaSense platform. Each node then published its context information. The results are shown in Figure **1**.

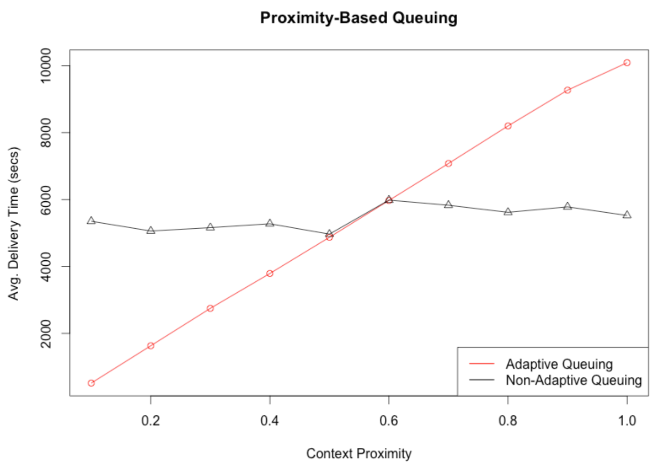

Figure 1 – Proximity Based Queuing

As shown in the results, with approaches such as Scribe where no adaptive queuing is applied, entities achieved the same delay in receiving context information. This equates delivery times to critical end points to non-critical end points and offered no guarantee of delivery where the context information is deemed more relevant or critical to the execution of an application or service. By applying our adaptive provisioning algorithm, we adjusted the provisioning times to reflect the proximity of the entities to each other, thus offering better guarantees with respect to information delivery times to entities that have a more similar context relationship.

Secondly, using this approach we measure the number of subscribers maintained at each node, as the number of application scales. We ran the simulation with 1000 nodes, and 1000 subscribers, randomly subscribing to nodes across the platform. We started with one application per node, and gradually increased the load until each subscriber had 20 applications. Each application had a random selection of context attributes constituting its application space. As shown in Figure **2**, the number of subscriptions per node increased sharply as the number of applications grew from 1 to 3.

This results as there is no existing dimension subscription at each node, and the node is initializing or stabilizing with respect to the number of maintained subscriptions. As the number of applications residing at each node increases, the number of subscriptions gradually increases, remaining relatively constant where duplications in dimension subscriptions occur. This reduces the load on each node and is a result of our approach used to map application spaces between entities. This removes unnecessary duplication on the platform as both entities within the same application reside on the same node realizing context derivation from a common set of subscriptions. This is in contrast to existing approaches where applications reside on end nodes and therefore introduce overheads through duplication.

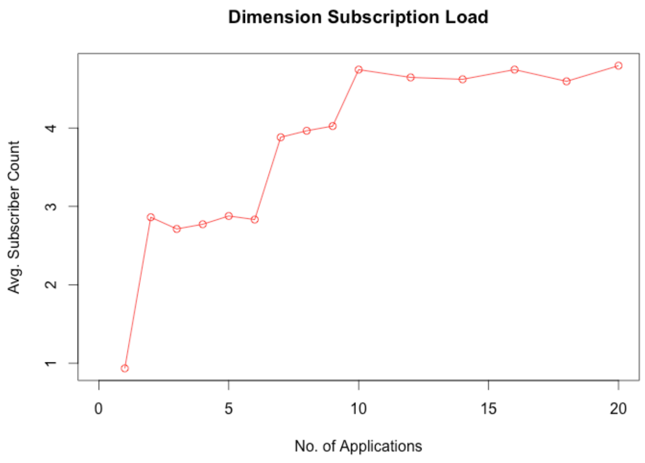

Figure 2 - Dimension Subscription Load

We additionally, measure the distribution of the number of subscribers as the number of applications increased. As shown in Figure **3**, with a single application running across each subscriber, the number of subscriptions is fairly evenly distributed across nodes, with a half of the nodes having no active subscriptions. As the number of applications increase across the nodes, the number of subscribers remain well distributed with the number of nodes registering no subscriptions showing a marked decrease.

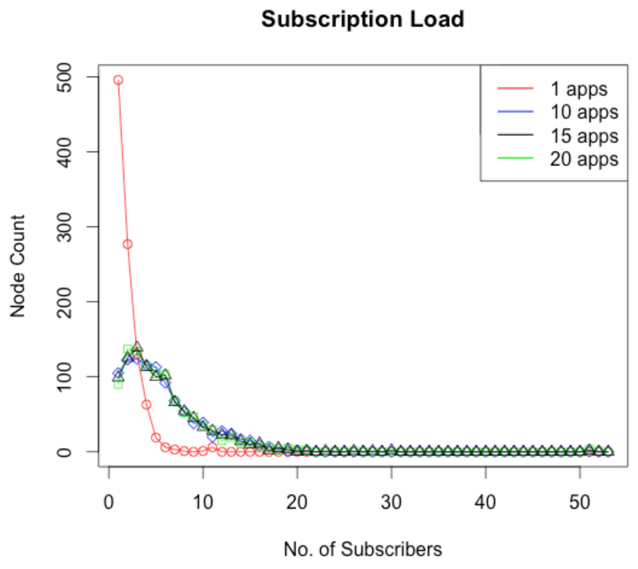

Figure 3 - Subscription Load

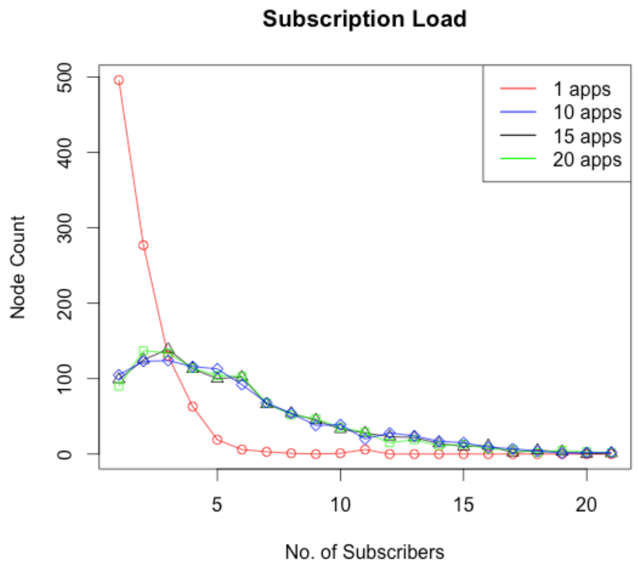

Figure 4 - Subscription Load

This shows that the application rendezvous points are well distributed as the number of applications grows and slowly move towards being evenly distributed over the platform. These results are highlighted in Figure **4**. This effectively reduces the communication costs on nodes within the overlay and distributes the cost of realizing application centric context provisioning across the overlay as opposed to residing in the service end points only.

We further simulated the volume of context information communicated when the proximity of the entities are considered to adaptively adjust the rate at which the context information is provisioned. Firstly we compare the rate of context information between non-adaptive updates and adaptive updates for all known entities in an address book. Each entity is assigned a GPS location at the beginning of the simulation. The entities are evaluated and relationships created between the host entity P and each of the other 100 entities Q_i. Each entity changed its location at small rate randomly drawn from the uniform distribution:

$$\mathcal{U}(0.0, 0.00250)$$

degrees for both latitude and longitude, moving with each simulation round. With each movement, the proximity is calculated and used to determine the average number of updates sent to P. The number of updates is derived as from a Poisson distribution where $\lambda = \mathcal{U}\,(0.0, 0.00250)$. The small value of 0.0025 was chosen to reflect the small changes in distance that could be observed from a human moving.

As shown in Figure **5**, where the adaptive approach is not used, the number of updates remained constant and equal to the number of event updates per minute. Where the adaptive algorithm is applied, the number of updates per minute rapidly decreases. This as the first update has no proximity value and therefore must be calculated. After the first calculation, the rate is reduced over the next calculations as the entities begin to move and the proximity values change. As more entities move close to P the rate increases however as the entities gradually move away over time, the rate at which updates are sent is reduce significantly.

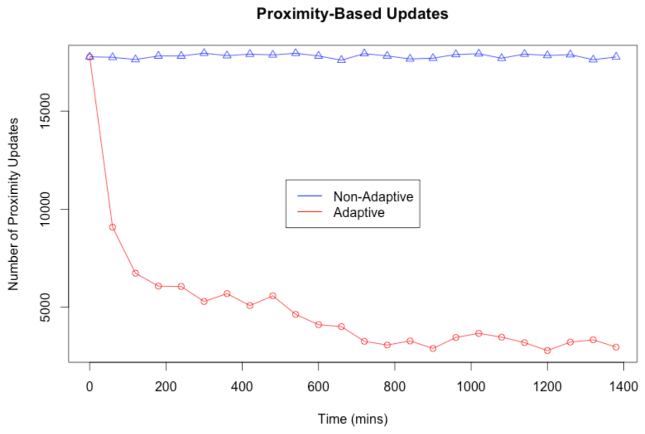

Figure 5 - Proximity-Based Updates

We further modified the adaptive algorithms to remove entities that had proximities outside of the interest area of the application triple and therefore required no updates to entity P. This is shown in Figure **6**. Filtering non-related entities rather than all subscribed entities reduces the overall communication costs as entities remain dynamic and continually changing location and context proximity.

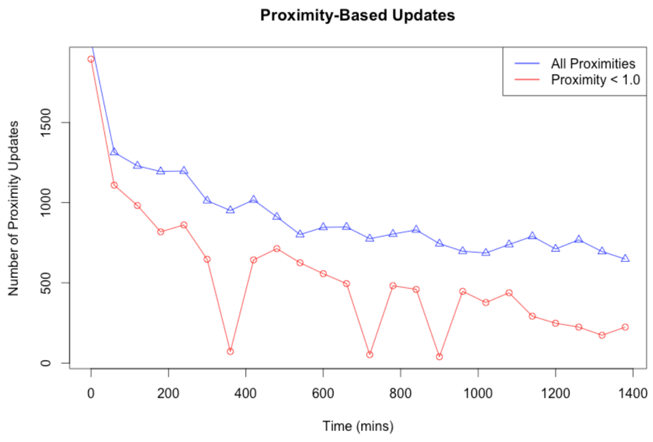

Figure 6 - Proximity-Based Updates From Related Entities

We further implemented our adaptive approaches using the Opportunity Challenge [33] context information dataset, first issuing updates to remote end points as the context information changes as event based-updates. An update is sent to the rendezvous point with each state update, the proximity is then calculated and sent to the endpoints. Calculating at the rate of context dissemination, which occurred at an average rate of 18000 updates per minute from all the sensor information generated. This is shown in Figure **7**. The volume of information transmitted remained constant with the updates from each entity. There was a sharp decrease noticed at 420secs, this was due to no information being transmitted from the sensors.

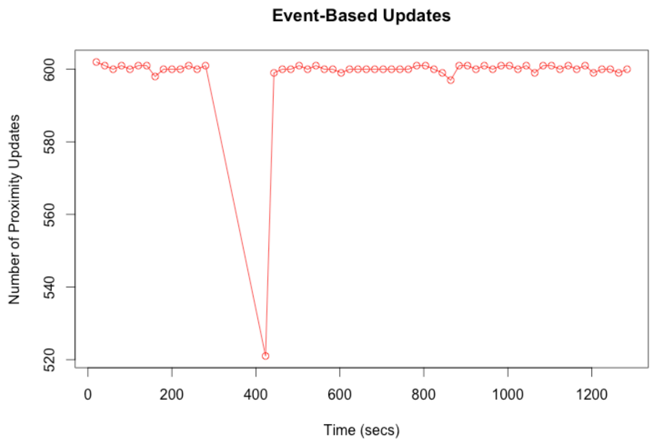

Figure 7 - Event Based Updates

We applied our activity-based adaptive approach, updating endpoints where the activity had changed and therefore merited a revaluation of the proximity or the degree of relationship between the two points. This is applicable in scenarios that are largely activity based, where the higher level meaning of the context states determines the underlying relationships between the two entities. As shown in Figure **8**, the number of updates varied between 0 and 8 updates per minute for more general activities such as standing or walking. This as the classification of activity states within the application space minimizes the need to send multiple updates to end points. Multiple changes in context states, or multiple context changing events exist within the discourse of an activity and where applications are realized over the higher level activities and interactions, such states can be ignored until the higher level activities evolve or change.

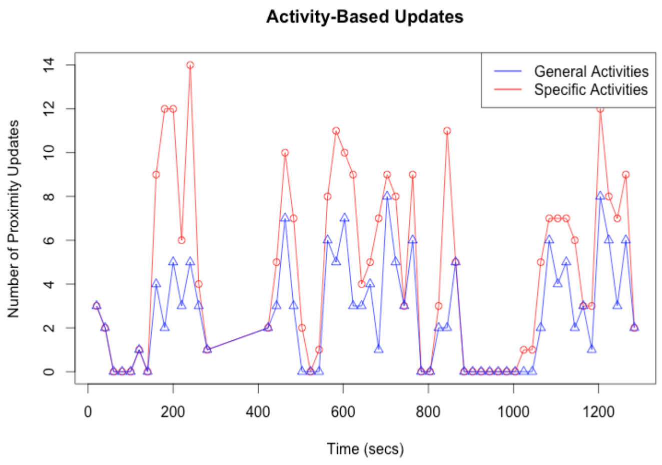

Figure 8 - Activity-Based Updates with Varying Granularity

Consequently, as the bounded space defining the activity becomes better defined, such as standing and opening the door, the number of updates increased to cover this increased granularity realizing more context updates per minute as shown. Applications can therefore find optimal values for the granularity of the activity spaces vs. the volume of context information needed to support this.

4. Conclusion

In this paper described our approach to adaptively provisioning user context information on distributed context centric architectures. Through this approach we can realize a connected things infrastructure that considers the relationships as well as the application domain within which entities interact. This is relevant to the prioritization of information delivery and maximizing real-time properties while reducing overall communication costs.

We firstly extended the existing publish-subscribe approach in MediaSense to introduce a context subscription triple comprising of an entity, an application space, and a set of provisioning constraints. This was implemented relative to the MediaSense platform. Each triple is mapped to a unique key and routed to random endpoint. This in turn decomposes the triple and creates subscription messages to each dimension. These are in turn routed to the rendezvous node for each entity's dimension. This effectively realized a multicast structure within which nodes progressively evaluated subscriptions based on the context triple, the context centric relationship between the entities. We are therefore able to constraints to each application triple, in response to the relationships between entities and adaptively provision updates on the relationships between entities at remote end points.

We simulated our approach and showed that we realized significant reductions in the overall communication costs to end points while maintaining and application that scales well. This approach can be further extended to create multiple entities in a single triple and further applied to software-defined networks to adaptively route context information between endpoints relative to their current relationship.

References

[1] "Ingress." [Online]. Available: http://www.ingress.com/. [Accessed: 26-Feb-2013].

[2] R. Forsberg and W. Sauter, "Digital theater in favor of audiences?," Stockholm, Sweden.

[3] V. Monfort and S. Cherif, "Bridging the Gap between Technical Heterogeneity of Context-Aware Platforms : Experimenting a Service Based Connectivity between Adaptable Android , WComp and OpenORB," vol. 8, no. 4, pp. 1–12, 2011.

[4] H. Mcheick, "Developing Service Oriented Computing Model Based On Context-Aware Mohamed Dbouk Ahmad Karawash," vol. 9, no. 5, pp. 392–405, 2012.

[5] A. Kansal, S. Nath, J. Liu, and F. Zhao, "Senseweb: An infrastructure for shared sensing," *IEEE Multimed.*, vol. 14, no. 4, pp. 8–13, Oct. 2007.

[6] T. Kanter, P. Österberg, J. Walters, V. Kardeby, S. Forsström, and S. Pettersson, "The Mediasense Framework," in *2009 Fourth International Conference on Digital Telecommunications*, 2009, pp. 144–147.

[7] R. A. Baloch and N. Crespi, "Addressing context dependency using profile context in overlay networks," in *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, 2010, pp. 1–5.

[8] D. Petras, I. Baronak, and E. Chromy, "Presence Service in IMS.," *ScientificWorldJournal.*, vol. 2013, p. 606790, Jan. 2013.

[9] S. Yoo, J. H. Son, and M. H. Kim, "A scalable publish/subscribe system for large mobile ad hoc networks," *J. Syst. Softw.*, vol. 82, no. 7, pp. 1152–1162, Jul. 2009.

[10] J. Santa and A. F. Gomez-Skarmeta, "Sharing Context-Aware Road and Safety Information," *IEEE Pervasive Comput.*, vol. 8, no. 3, pp. 58–65, Jul. 2009.

[11] S. Forsström, V. Kardeby, J. Walters, and T. Kanter, "Location-Based Ubiquitous Context Exchange in Mobile Environments," *Mob. Networks Manag.*, pp. 177–187, 2011.

[12] D. Frey and G. C. Roman, "Context-aware publish subscribe in mobile ad hoc networks," in *Coordination Models and Languages*, 2007, vol. 4467, pp. 37–55.

[13] S. L. Kiani, M. Riaz, Y. Zhung, S. Lee, and Y. K. Lee, "A distributed middleware solution for context awareness in ubiquitous systems," in *Embedded and Real-Time Computing Systems and Applications, 2005. Proceedings. 11th IEEE International Conference on*, 2005, pp. 451–454.

[14] S. Sridevi, S. Bhattacharya, and R. Pitchiah, "Context Aware Framework," pp. 358–363, 2012.

[15] S. Seok, W. Song, and D. Choi, "Implementation of Pastry-based P2P system to share sensor data," *Int. J. Sens. Networks*, vol. 8, 2010.

[16] M. Castro, P. Druschel, a.-M. Kermarrec, and a. I. T. Rowstron, "Scribe: a large-scale and decentralized application-level multicast infrastructure," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 8, pp. 1489–1499, Oct. 2002.

[17] A. Zimmermann, A. Lorenz, and R. Oppermann, "An operational definition of context," in *Proceedings of the 6th international and interdisciplinary conference on Modeling and using context*, 2007, pp. 558–571.

[18] J. I. Hong and J. Landay, "An infrastructure approach to context-aware computing," *Human-Computer Interact.*, vol. 16, no. 2, pp. 287–303, Dec. 2001.

[19] F. Dobslaw, A. Larsson, T. Kanter, and J. Walters, "An Object-Oriented Model in Support of Context-Aware Mobile Applications," *Mob. Wirel. Middleware, Oper. Syst. Appl.*, pp. 205–220, 2010.

[20] A. Toninelli, S. Pantsar-Syväniemi, P. Bellavista, and E. Ovaska, *Supporting context awareness in smart environments*. New York, New York, USA: ACM Press, 2009, p. 1.

[21] L. Liu, F. Lecue, N. Mehandjiev, and L. Xu, "Using Context Similarity for Service Recommendation," *2010 IEEE Fourth Int. Conf. Semant. Comput.*, pp. 277–284, Sep. 2010.

[22] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Trans. Inf. Syst.*, vol. 23, no. 1, pp. 103–145, 2005.

[23] B. Schilit and N. Adams, "Context-aware computing applications," *Syst. Appl. 1994.*, pp. 85–90, 1994.

[24] A. K. Dey, "Understanding and Using Context," *Pers. Ubiquitous Comput.*, vol. 5, no. 1, pp. 4–7, Feb. 2001.

[25] A. Padovitz, "Context Management and Reasoning about Situations in Pervasive Computing," Monash University, 2006.

[26] R. Schmohl, "The Contextual Map," *deposit.ddb.de*, 2010.

[27] J. Walters, T. Kanter, and R. Rahmani, "Establishing Multi-Criteria Context Relations Supporting Ubiquitous Immersive Participation," *Int. J.*, vol. 4, no. 2, pp. 59–78, 2013.

[28] V. Kardeby, S. Forsström, and J. Walters, "The Updated MediaSense Framework," in *(ICDT), 2010 Fifth*, 2010.

[29] O. Akan, P. Bellavista, J. Cao, F. Dressler, D. Ferrari, M. Gerla, H. Kobayashi, S. Palazzo, S. Sahni, X. (Sherman) Shen, M. Stan, J. Xiaohua, A. Zomaya, G. Coulson, K. Pentikousis, O. Blume, R. Agüero Calvo, S. Papavassiliou, S. Forsström, V. Kardeby, J. Walters, R. Norling, and T. Kanter, *Mobile Networks and Management*, vol. 32. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 57–66–66.

[30] A. Padovitz, S. W. Loke, and A. Zaslavsky, "Towards a theory of context spaces," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, 2004, no. March, pp. 38–42.

[31] Y. Rubner, C. Tomasi, and L. J. Guibas, "A Metric for Distributions with Applications to Image Databases," p. 59, Jan. 1998.

[32] R. Shahid, S. Bertazzon, M. L. Knudtson, and W. a Ghali, "Comparison of distance measures in spatial analytical modeling for health service planning.," *BMC Health Serv. Res.*, vol. 9, p. 200, Jan. 2009.

[33] H. Sagha and S. Digumarti, "Benchmarking classification techniques using the Opportunity human activity dataset," *Syst. Man, …*, 2011.