This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

Priority Based Selection to Improve Contents Consistency for Mobile Overlay Network

Zhou Su, Jiro Katto, and Yasuhiko Yasuda, *Members, IEEE*

Contact Email zhousu@asagi.waseda.jp

***Abstract*—With the growing use of dynamic content by mobile content distribution systems, how to manage dynamically changing files has become an important issue, since the cached replicas on different mobile sites must be updated if the originals have been changed. Therefore, this paper proposes a priority based selection method to enhance the efficient utilization of network resource and support the client mobility for Mobile Contents Delivery Network (M-CDN). On one hand, a consistency priority is calculated by analyzing the characteristics of mobile surrogates. If a given content which has been changed on its original node, only the replicas with the high consistency priority instead of all replicas are updated. On the other hand, an update priority is also proposed. If a replica has been selected for update, the latest version will be sent from the site decided by the update consistency. Simulation results show that the proposed new approach outperforms other conventional methods.**

***Index Terms*— Mobile Contents Delivery, Consistency Algorithm, Web Performance, Network Traffic**

I. Introduction

Recently, a Mobile Contents Delivery Network (M-CDN) has appeared to facilitate a set of moving nodes to communicate with each other via short-range wireless transmission protocols such as IEEE 802.11, Blue-tooth or Ultra Wide Band (UWB). A moving node receives information from its neighbors, or from remote nodes by multi-hop transmission relayed by intermediated moving nodes in the M-CDN [1][2].

Lots of attentions and applications have been attracted nowadays: for example, the Nokia mobile server and the Helix mobile server led by Real Networks have been optimized for delivery of all major file formats to any wireless network, with the broadest support for mobile delivery standards. Furthermore, more and more mobile contents are being delivered over the mobile network, such as Mobile streaming TV supported by Digital Multimedia Broadcasting (DMB), and mobile games to allow people to play a game on a mobile handset. However, although the above technologies advocate their wireless support for contents delivery, how to improve the content consistency have not yet clarified nor verified.

Zhou SU, Jiro KATTO, Yasuhiko YASUDA are with Department of Computer Science, Waseda Umiversity, Okubo 3-4-1, Shinjuku-ku, Tokyo 169-8555, JAPAN

To deliver mobile contents, the consistency control is an important issue to be resolved. That is to say: the cached replicas on different sites must be updated if the originals change. If a requested object can be provided to the user with the latest version, the user will not need to contact the remote origin server to get the updated object. Therefore, if the appropriate consistency control of server replicas can be carried out, it will benefit content providers by reducing latency for their clients, and benefits ISPs by reducing bandwidth consumption and transmission cost.

Although there are some conventional methods to keep consistency within the wired network, these methods can not be efficiently carried out in mobile environments. The reason is: in contrast to conventional CDN systems in wired networks that consist of surrogates, the mobile CDN are subjected to the limitations of long-lived nature of typical multimedia sessions, wireless bandwidth, and the dynamically changed network topology. Challenges arise in how to improve the data consistency in a mobile CDN.

In this paper, we therefore propose a novel algorithm for controlling Web consistency in mobile contents delivery overlay. Firstly, the Web access of M-CDN is analyzed. A consistency priority is proposed to judge whether a replica of mobile content should be updated or not when its original data changes. Secondly, an update priority is proposed to select the node to fetch the latest version of the updated contents from. Finally, we test our proposal under the dynamic mobile network environment with varied parameters. And the results show that our proposal get better and more stable performance than other conventional methods.

II. Previous Work

Content Delivery Networks (CDNs) have been playing an important role in managing content distribution to large numbers of users in wired networks. However, unlimited in the wired network, with the development of wireless technology, the demands for mobile content delivery to dynamically changing client, has appeared in recent years.

[11] talked about the importance of the Web consistency and also proposed a method of Consistency-preserving Caching for Dynamic Database Content. However, the above work doesn't

978-1-4244-2948-6/09/$25.00 ©2009 IEEE

support the wireless communication. In [12], an optimized query planning of continuous aggregation queries in dynamic data dissemination networks was presented. This method of query execution can be implemented using schemes similar to that used in CDNs. However, how to optimally control the query to improve consistency is still not resolved.

[4] proposed a mobile surrogate-selection scheme that can significantly reduce the number of server-handoffs necessary for mobile multimedia contents delivery via a high density M-CDN. In this method, all surrogates are divided into 2 tiers: Lower-tier servers are topologically closer to the clients, and hence can deliver better QoS in terms of end-to-end delay and jitter. On the other hand, higher tier servers have a larger coverage area and hence their clients incur fewer server handoffs. However, the criterion for dividing surrogates into different tiers is still not clearly presented. Furthermore, the cooperation among different tiers is not talked about.

[1] proposed a M-CDN model for the replication in enterprise networks, which takes client demand variations into account to dynamically add/remove content replicas to/from the network in order to minimize the total traffic over the backbone. It provided two solutions: an offline optimal and a heuristic online solution. Although the proposed model is helpful to contents replication, its analysis is only based on the network traffic. The corresponding analysis about network bandwidth and topology should be carried out.

In [8], it concluded that using the fixed caching proxies to serve mobile nodes may defeat the entire purpose of using caches for traffic localization, therefore it is important that the mobile node can switch to appropriate caching proxy as it moves. Their results showed that delay experienced by mobile nodes could be significantly reduced if mobile surrogate keeps switching to nearest caching proxy. Thought this work proved that keeping the replica on different proxies is helpful, how to keep the consistency of the replicas on different proxy has not been mentioned.

A cooperative caching and data consistency scheme [2], called Proximity Regions for Caching in Cooperative (PReCinCt), has been proposed to support scalable data retrieval in large-scale Mobile Peer to Peer (MP2P) networks. This cooperative caching scheme enables peers in a region to share their data, thus providing a unified view of the cache. This helps in alleviating the message latency and limited accessibility problems in MP2P networks. The caching scheme considers data popularity, data size and region-distance during replacement to optimize cache content of peers. The above method used a parameter called Time-to-Refresh (*TTR*) to keep the consistency among the replicas, however, how to decide an optimal value of Time-to-Refresh (*TTR*) was not resolved.

In *Propagation* method, the updated version of a document is delivered to all copies whenever a change is made to the document on the origin server. Although the copies always keep the latest version of the originals by the *Propagation*, this method may generate significant levels of unnecessary traffic if documents are updated more frequently than accessed.

In *Invalidation* method [7], an invalidation message is sent to all copies when a document is changed on the origin server. This method does not make full use of the delivery network for content delivery and each replica needs to fetch an updated version individually at a later time. Therefore, the user-delay may get worse if a frequently accessed document can not be updated on time.

We ourselves proposed an integrated pre-fetching and searching algorithm for mobile P2P model, where the query can be efficiently used [9]. We also presented a scheme to manage web consistency in a contents delivery network, in which all nodes are wired without mobile contents [10]. The former is limited to P2P network and the later is designed for wired network, which are both different from the proposal in this paper.

III. Theory Analysis

In this section, we analyze the Web consistency over the M-CDN. For the readers' convenient, Table 1 presents a list of parameters used in our network model.

Table 1: Client Workload and Model Parameters

Parameters	Definition
$o(j)$	Server originally stores content *j*
$k(i,j)$	Sorrogate which has the latest version of content *j* when it is requested from sorrogate *i*
w_j	Ranking of request times of content *j*
λ_i	Aggregate request rate to surrogate *i*
B_j	Data size of content *j*
ca_i	Coverage area of surrogate *i*
P_j	Request probability of content *j*
r_j	Relative residence of a client within surrogate *i*
$r_{q,i}$	Residence time value for client *q* in surrogate *i* 's zone
$f_{k(I,j)}$	Free degree of surrogate $k(I,j)$
$RSS_{k(i,j),i}$	Received signal strength on surrogate $k(i,j)$
$l_{k(i,j),i}$	Probability that surrogate *k*'s *RSS* is beyond the threshold
$S_{k(i,j),i}$	User's Satisfaction with getting content *j* from surrogate $k(i,j)$
$t_{k(i,j),i}$	User-delay ($t_{k(i,j),i}$) to obtain content *j* from surrogate $k(i,j)$
$D_{k(i,j),i}$	Shortest distance (hop count) from surrogate $k(i,j)$ to surrogate *i*
$C_{k(i,j),i}$	Average bandwidth (per hop) during the path from surrogate $k(i,j)$ to surrogate *i*
$\theta_{i,j}$	Consistency priority of content *j* on the surrogate *i*
$\Delta_{k(i,j),i}$	Update priority of content *j (*requested from surrogate *i)* on the surrogate $k(i,j)$

A. Contents Selection

We consider an overlay network that has *I* mobile surrogates. Each surrogate *i* (*i*=1,..., *I*), has a λ_i (bytes/second) which denotes an aggregate request rate from clients to the server.

As for the contents, we assume that there are *J* different contents in our M-CDN. A parameter P_j defines the request probability for content *j* (*j*=1,..., *J*), with a data size B_j.

According to the Zipf distribution which the distribution of Web accesses follows, the probability that the content *j* is requested can be obtained as follows:

$$P_j = \frac{V}{w_j^{\alpha}} \tag{1}$$

where V , α are parameters of the Zipf distribution, and w_j is the ranking of request times.

We assume that there is a Request Routing (RR) function in each surrogate. The RR maintains the residence time value $r_{q,i}$ for each client *q* in surrogate *i* 's zone.

Then, the relative residence of a client within surrogate i can be

$$r_i = \frac{\sum_q r_{q,i}}{\sum_i \sum_q r_{q,i}} \quad (2)$$

Assume that a client firstly enters in the area of surrogate i and then requests contents j, we can get the probability that the content j is requested from surrogate i to be:

$$\theta_{i,j} = \frac{V}{w_j^{\alpha}} \cdot \frac{\sum_q r_{q,i}}{\sum_i \sum_q r_{q,i}} \quad (3)$$

We call the above *Consistency Priority* $\theta_{i,j}$. Note that this priority can support node-mobility since both the residence time and the entering area are taken into consideration. The concrete usage of it will be introduced in later sections.

B. Surrogate Update

In a conventional method, if a surrogate decides to fetch the latest version of a modified document, it will contact the modified document's original server to fetch its latest version. However, this method causes a bad load balance since all requests are going to the original server. Furthermore, fetching the contents from the original server (which is far from the client) also increases the extra network traffic.

It has become an important issue to select a proper surrogate to provide the latest version of the requested content to the client. Here, we think the proper surrogate should at least satisfy the following three requirements: 1) be reachable: The received signal strength (RSS) of the surrogate should be strong enough to let the requested content arrive at the client. 2) be satisfied: The client should be satisfied with the quality (such as user delay) after receiving the updated content from the surrogate. 3) be available: The surrogate should have enough free sessions to receive the request when a latest version is wanted from the client.

We make analyses of the above requirements as follows:

1) Analysis of the Reachable Ability

Assume that content j is originally stored in server $o(j)$ and $D_{k(i,j),i}$ is the shortest distance (hop count) from surrogate $k(i,j)$ to surrogate i. Here, surrogate i means the node where the client sends the request of content j, and surrogate $k(i,j)$ is a node which has the latest version of content j at this time. Surrogate i's received signal strength (RSS) from surrogate $k(i,j)$ can be expressed as [13]

$$RSS_{k(i,j),i} = RSS_0 - 10 \cdot \infty \cdot \log(D_{k(i,j),i} / D_0) + \psi_{k(i,j),i} \quad (4)$$

Here, D_0 is the unit distance from source surrogate. RSS_0 is the received signal strength when the distance between the surrogate and the source is D_0. ∞ is the attenuation factor while $\psi_{k(i,j),i}$ denotes a zero mean.

The above equation can be transformed to the one which is not in dB as follows.

$$RSS'_{k(i,j),i} = \frac{M \cdot \eta_{k(i,j),i}}{\left(\frac{D_{k(i,j),i}}{D_0}\right)^{\infty}} \quad (5)$$

Where

$$RSS'_{k(i,j),i} = 10^{RSS_{k(i,j),i}/10} \quad (6)$$

$$M = 10^{RSS_0/10} \quad (7)$$

$$\eta_{k(i,j),i} = 10^{\psi_{k(i,j),i}/10} \quad (8)$$

Then we define the probability that its signal is better than the threshold as follows:

$$\begin{aligned} l_{k(i,j),i} &= P\left\{RSS'_{k(i,j),i} \geq Thresh\right\} \\ &= P\left\{RSS'_{k(i,j),i} \,\middle|\, RSS'_{k(i,j),i} \geq Thresh\right\} / T \end{aligned} \quad (9)$$

where *Thresh* is the set threshold and T is the total detecting times within the testing period.

2) Analysis of the Satisfaction

Recent studies [6] show that user on surrogate i's satisfaction (or utility) with receiving the requested content j depends on the user-delay ($t_{k(i,j),i}$) to obtain content j from surrogate $k(i,j)$.

$$S_{k(i,j),i} = e^{-\omega \cdot t_{k(i,j),i}} \quad (10)$$

Here, ω is a satisfaction parameter that is varied according the user's sensitivity.

However, how to obtain the user delay has not been mentioned in Eq.10. As the user-delay is very important to reflect the user's satisfaction, in this paper we calculate the user delay, which is caused during the delivery from a surrogate to the client (requested surrogate) as follows.

Assume that $C_{k(i,j),i}$ is the average bandwidth (per hop) along the path from surrogate $k(i,j)$ to surrogate i. Then, when a request for content j happens at surrogate i and surrogate $k(i,j)$ sends the latest version to satisfy this request, the user delay during the delivery from surrogate $k(i,j)$ to surrogate i is given by

$$t_{k(i,j),i} = \frac{1}{\Lambda} \lambda_i \cdot B_j \cdot P_j \cdot D_{k(i,j),i} / C_{k(i,j),i} \quad (11)$$

If we continue to define:

$$G_j = \frac{1}{\Lambda} \cdot B_j \cdot P_j \quad (12)$$

$$U_{k(i,j),i} = D_{k(i,j),i} / C_{k(i,j),i} \quad (13)$$

Eq.(11) can be rewritten as follows:

$$t_{k(i,j),i} = \lambda_i \cdot G_j \cdot U_{k(i,j),i} \quad (14)$$

Then, we can obtain the user's satisfaction degree ($S_{k(i,j),i}$) when surrogate $k(i,j)$ provides the latest version of content j to surrogate i (where the user sends the request)

$$\begin{aligned} S_{k(i,j),i} &= e^{-\omega \cdot t_{k(i,j),i}} \\ &= e^{-\omega \cdot (\lambda_i \cdot G_j \cdot U_{k(i,j),i})} \end{aligned} \quad (15)$$

3) Analysis of the Availability

Assume that the surrogate i has a coverage area denoted by the radius of ca_i and its distance between other surrogate $k(i,j)$ is $D_{k(i,j),i}$.

For surrogate $k(i,j)$, if it is in the coverage area of surrogate i, then we can get:

$$D_{k(i,j),i} \le ca_i \qquad (16)$$

$$k(i,j)=\{1,2,\ldots I\} \;\&\; k(i,j) \neq i$$

Let $Z_{k(i,j)}$ denote the capacity (the number of simultaneous sessions) of surrogate $k(i,j)$, and Z is the number of total sessions happened in the whole network within a testing past period. Then the average sessions happened in surrogate $k(i,j)$ during the past period can be:

$$(\lambda_{k(i,j)} / \sum_{k(i,j)} \lambda_{k(i,j)}) \cdot Z) \qquad (17)$$

A free-degree is defined as follows to show the number of free sessions in surrogate k.

$$f_{k(i,j)} = (Z_{k(i,j)} - (\lambda_{k(i,j)} / \sum_{k(i,j)} \lambda_{k(i,j)}) \cdot Z)) \Big/ Z_{k(i,j)} \qquad (18)$$

4) Surrogate Update Priority

Let $X_{i,j,}$ be a parameter which takes a binary value of

$X_{i,j} = 1$ (if content j is stored in surrogate i)

$X_{i,j} = 0$ (otherwise) (19)

Then, we can get a matrix $\boldsymbol{X}$ of which element is $X_{i,j}$, which represents a placement pattern of contents.

For the mobile surrogates $k(i,j)$ ($k(i,j)$=1,…, K), which are in the coverage of surrogate i, and also have the replica of content j, we define an Update Priority as fllows:

$$\Delta_{k(i,j),i} = f_{k(i,j)} \cdot l_{k(i,j),i} \cdot S_{k(i,j),i} \qquad (20)$$

Since surrogate i and content j are fixed when we need to find a proper surrogate $k(i,j)$ to fetch the latest version of conetnt j to surrogate i, for simplicity, we use Δ_k to represent $\Delta_{k(i,j),i}$ in the following section.

For the mobile surrogates, which are in the coverage of surrogate i, and also have the replica of content j, we can select the mobile surrogate k with the maximum update index (Δ_k) by

$$\begin{cases} i \neq k \quad \& \; X_{k,j} = 1 \\ D_{i,k} \le ca_i \\ \max(\Delta_k) \end{cases} \qquad (21)$$

The usage of Update Priority will be introduced in later section.

IV. Proposal

We present our algorithm as follows:

- Step1: Content Selection (To decide whether a replica should be selected for update)

When a given content j has beenchanged at its original server $o(j)$, a consistency priority $\theta_{i,j}$ will be calculated according to Equa.3. For content j's each replica ($X_{i,j}$=1) that is distributed over the whole mobile network, only when its priority $\theta_{i,j}$ is beyond the threshold, the replica of content j at mobile surrogate i will be updated.

Otherwise, this replica will not be updated until a new request for content j happens at mobile surrogate i next time.

Therefore, when a given content j is changed at its original server, not all its replicas ($X_{i,j}$ = 1) over all overlay network will be updated according to the analysis of Web access distribution. Here, the threshold can be determined as TOP 10% contents based on the Zip'f distribution.

- Step2: Highest Satisfation Surrogate Update (To decide where to get the latest version)

Assume that there are K ($X_{k,j}=1$, $k=\{1,\ldots,K,\ldots,I\}$) replicas of content j selected to be updated by the Step1, then for a replica at mobile surrogate k , an update priority Δ_k will be calculated according to Eq.21.

The latest version of content j will be sent to the surrogate (which replica need to be replicated) from mobile surrogate k with the maximum Δ_k.

Therefore, the latest version will be sent from an algorithm-decided site instead of its original server resulting in the highest satisfaction.

V. Simulation Results

In the simulation scenario the M-CDN comprises 21 surrogates arranged by the Power-Law distribution [3]. For each surrogate, the radius of coverage area is 2/3 of its longest distance to other surrogates. *RSS* is set to be one of values of the set {1,2,3}, where 3 means the strongest value of *RSS*. The capacity of each surrogate is decided at random from 50 to 300 simultaneous sessions [4].

As for the request, it arrives according to a Poisson process. All clients' requests are always redirected to the closest server. Residence time is decided at random from 1 minute to 2.5 minutes at random [4].

There are 1000 different contents in the network. The update rate of a given object is decided at random. The total request times in the simulations are 100000.

There are 4 replication algorithms we will study:

1.Invalidation Policy [7]

2.TTR Policy

3.Proposal

Several researchers have observed that the distribution of web request from a fixed group of users follows a Zipf distribution. Besides, the value of α , a parameter of Zipf distribution, varies from trace to trace, ranging from 0.64 to 0.83 [6]. We then varied the Zipf parameter and get the results of Old Hit Ratio of these three algorithms in Fig.1 when the Zip'f parameter is changed. It can be found that our proposal can get the best Old Hit Ratio in Fig.1. The reason is that: the proposal selects parts of (not all of) contents to be updated as whenever its original is changed on the original site. Then, the replica (which should be updated for user's future request) can be updated on time to reduce Old Hit Ratio. Furthermore, when a replica is selected for update, the latest version of the contents is fetched from a surrogate chosen by the algorithm (which is close to the user and can provide short delay) instead of its original server. Then the Old Hit Ratio can be improved more.

Then we examined the three algorithms with different simulation times. As shown in Fig.2, each individual algorithm's Old Hit Ratio keeps increasing when we increase the total simulation times. However, obviously the increasing ratio of our proposal is the lowest compared with the other two. These experimental results verified that the Old Hit Ratio of our proposal is not only the lowest but also stable.

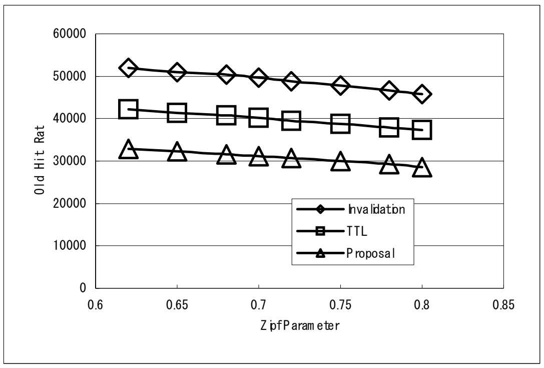

Fig.1: Comparison of Old Hit Ratio under Different Zipf Parameters.

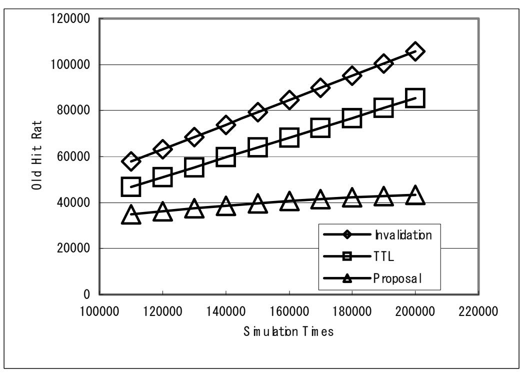

Fig.2: Comparison of Old Hit Ratio under Different Simulation Times.

VI. Conclusions and Further Work

In this paper, based on mathematical analysis we proposed an integrated algorithm to manage the consistency of different replicas in M-CDN. On one hand, the novel approach can decide whether a replica should be updated when its original content changes. On the other hand, if a replica has been decided for update, from which surrogate the latest version should be sent to the client can also be determined. Through intensive simulation, we verified that our design outperforms other conventional methods.

In our on-going efforts, we are planning to implement and evaluate our novel approach on top of emulation testbed.

References

[1] Wagner M. Aioffi, Geraldo R. Mateus, Jussara M. de Almeida, and Antonio A. F. Loureiro. "Dynamic Content Distribution for Mobile Enterprise Networks". IEEE Trans on Selected Areas in Communications, NO. 10, Oct 2005

[2] H.Shen, M.S.Joseph, M.Kumar, and S.K. Das, "PReCinCt: A Scheme for Cooperative Caching in Mobile Peer-to-Peer Systems", Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, 2005

[3] G. Antoniol, G. Casazza, G. Di Lucca, M. Di Penta, and E. Merlo, "Predicting Web Site Access: An Application of Time Series", Proceedings of IEEE the third International Workshop on Web Site Evolution, Florence, Nov.2001.

[4] M. Tariq, R. Jain and T. Kawahara: "Mobility Aware Server Selection for Mobile Streaming Multimedia Content Distribution Networks," Proceedings of International Web Caching Workshop (IWCW), Sep. 2003.

[5] L. Breslao, P. Cao, L. Fan, G. Phillips, and S. Shenker "Web Caching and Zip-like Distributions: Evidence and Implications" Proceedings of IEEE INFOCOM'99, New York, Apr. 1999.

[6] K.Nomura, K.Yamori, E.Takahashi, T.Miyoshi, and Y.Tanaka, "Waiting Time versus Utility to Download Images," Proceedings of 4th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT 2001), Kathmandu, Nepal / Atami, Japan

[7] P. Cao and C. Liu, "Maintaining Strong Cache Consistency in theWorld-Wide Web," Proceedings of the 17th Int'l Conf. Distributed ComputingSystems, May.1997.

[8] M. Tariq and A. Takeshita, "Management of cacheable streaming multimedia content in networks with mobile hosts," Proceedings of IEEE GLOBECOM, Taipei, Taiwan, 2002, pp. 2257–2258.

[9] Zhou SU, Jiro KATTO and Yasuhiko YASUDA, " An integrated Retrieval and Pre-fetching algorithms for Segmented Streaming in Mobile Peer-to-Peer Networks" Proceedings of IEEE Cluster2005, September, Boston, Massachusetts, USA, 2005

[10] Zhou SU, Jiro KATTO and Yasuhiko YASUDA, "Scalable Maintenance for Strong Web Consistency in Dynamic Content Delivery Overlays", Proceedings of 2007 IEEE International Conference on Communications (ICC 2007), Scotland, Jun, 2007

[11] Niraj Tolia and M. Satyanarayanan, "Consistency-preserving Caching of Dynamic Database Content", Proceedings of WWW2007, Banff, Canada, May, 2007

[12] Rajeev Gupta, Krithi Ramamritham "Optimized Query Planning of Continuous Aggregation Queries in Dynamic Data Dissemination Networks", Proceedings of WWW2007, Banff, Canada, May, 2007

[13] J.B.Anderson, T.S.Rappaport, and S.Yoshida, "A non-line-of-sight eoor mitigation algorithm in location estimation", IEEE Comm magazine, Vol.33-1, Sep 1995