

Syddansk Universitet

## Multi-Scale Dissemination of Time Series Data

Guo, Qingsong; Zhou, Yongluan; Su, Li

*Published in:*

Proceedings of the 25th International Conference on Scientific and Statistical Database Management

*Publication date:*

2013

*Document version*

Peer reviewed version

*Citation for published version (APA):*

Guo, Q., Zhou, Y., & Su, L. (2013). Multi-Scale Dissemination of Time Series Data. In A. Szalay, T. Budavari, M. Balazinska, A. Meliou, & A. Sacan (Eds.), Proceedings of the 25th International Conference on Scientific and Statistical Database Management Association for Computing Machinery.

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Multi-Scale Dissemination of Time Series Data

Qingsong Guo  
Department of Mathematics  
and Computer Science,  
University of Southern  
Denmark  
qguo@iamada.sdu.dk

Yongluan Zhou  
Department of Mathematics  
and Computer Science,  
University of Southern  
Denmark  
zhou@imada.sdu.dk

Li Su  
Department of Mathematics  
and Computer Science,  
University of Southern  
Denmark  
lsu@imada.sdu.dk

## ABSTRACT

In this paper, we consider the problem of continuous dissemination of time series data, such as sensor measurements, to a large number of subscribers. These subscribers fall into multiple subscription levels, where each subscription level is specified by the bandwidth constraint of a subscriber, which is an abstract indicator for both the physical limits and the amount of data that the subscriber would like to handle. To handle this problem, we propose a system framework for multi-scale time series data dissemination that employs a typical tree-based dissemination network and existing time-series compression models. Due to the bandwidth limits regarding to potentially sheer speed of data, it is inevitable to compress and re-compress data along the dissemination paths according to the subscription level of each node. Compression would caused the accuracy loss of data, thus we devise several algorithms to optimize the average accuracies of the data received by all subscribers within the dissemination network. Finally, we have conducted extensive experiments to study the performance of the algorithms.

## Categories and Subject Descriptors

H.2.4 [Database]: Scientific Data Management

## General Terms

Multi-Scale Dissemination

## Keywords

Time Series Data, Data Dissemination, Piecewise Linear Approximation

## 1. INTRODUCTION

In many cases, scientific data generated from sensors and various scientific equipments often take the form of time-series, i.e. streams of data points measured at successive time instants with uniform time intervals. In the context of Data-as-a-Service (DaaS) or large-scale cooperated research projects, there would be a large number of parties to the data. For instance, Microsoft's SensorMap project [23] establishes a portal to enable collection and sharing of sensor network data from various deployments of public institutes, companies as well as individuals. Data collected by the SensorMap

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

SSDBM '13, July 29 - 31 2013, Baltimore, MD, USA  
Copyright 2013 ACM 978-1-4503-1921-8/13/07 \$15.00

should be disseminated to a great number of receivers who have subscribed to them. This framework can be organized as a publish/subscribe system, where each subscriber is an interested party.

With the technology development, the rates of data generated by many advanced devices are prohibitively high. For example, there are around 600 million collisions per second within LHC (Large Hadron Collider) [2], which are observed by 4 detectors and produce a data rate of approximately 300 GB/s and the data size is over 300 MB/s even after pre-filtering. As another example, consider underwater sensor network [33] for monitoring the health of river and marine environments, where the overall data rate scales according to the size and the density of the deployment of sensors. Note that scientists often tend to obtain measurements with the largest scale and finest grain allowed by the technology. Assume we monitor an area of  $100km \times 100km$  which are divided into  $1m \times 1m$  grids, and each grid deployed with a sensor. Suppose each node produces one measurement every second and each measurement has 4 bytes of data. This will give a total data rate at around 40 GB/s.

Due to the sheer speed of such data, it is challenging to disseminate them to a large number of subscribers. Furthermore, different organizations or individuals would have different physical capacities and demands on data and hence required different granularities or accuracies of the data. Sending the raw data to all the subscribers are not only resource consuming but also undesirable. Supporting different subscription levels with different data granularities or accuracies would be a favorable feature for both the data providers and the subscribers.

In this paper, we consider the problem of disseminating time-series data to a large number of subscribers with multiple subscription levels. These nodes are organized as a provide/subscribe system, or dissemination network, where a node has been designated as the primitive data provider and each subscriber is under a *bandwidth constraint*. Bandwidth constraint is introduced to describe the subscription level of a node, it is an abstract parameter indicates not only the volume of data that a subscriber would prefer to consume but also some physical capacities, such as bandwidth of communication link. This problem suffers from the challenges of the sheer speed of data as well as bandwidth constrains in the system. The basic idea is to reduce the volume of data and to optimize the quality of service by compression, as done in other publish/subscribe systems [32].

Similar with the architecture of End System Multicast [13], all ends has been organized into an overlay tree, our dissemination network

also is constructed as a tree on the application layer. As we know, a End-to-End communication probably goes through other nodes. It means that extra packages are inevitable for these intermediate nodes if data are transmitted from the source node, which make the challenge even worse as the bandwidth is limited regarding to the data rate. We have to optimize our dissemination network. A basic idea is that data for each subscriber originates from the data of its parent other than the source. Therefore, in our system, each node plays two roles: a subscriber that requests a particular scale of compressed data and a broker that provides multi-scaled compressed data to its child nodes.

In our context, data for subscribers are compressed into multiple scales according to their bandwidth constraints. Various approaches have been proposed to approximate and compress time-series data, such as discrete Fourier Transform [8, 26], discrete Wavelet Transform [10], Singular Value Decomposition [14], Piecewise Aggregate Approximation [18], and Piecewise Linear Approximation (PLA) [16, 20], etc. In this paper, we adopt PLA as the compression model, i.e. series of line segments are generated and disseminated instead the raw data. Apart from its simplicity and wide applicability, PLA modelling can efficiently performed in an online fashion. In our approach, each node within the dissemination tree receives PLA models from its parent and then forwards them to its children, where a re-compression is called when the bandwidth constraint of any child is smaller than the size of models it received. The whole procedure is referred to as *model-based dissemination*. In summary, we have made the following contributions in this paper:

- We formulate the problem of model-based time-series data dissemination with multi-levels of bandwidth constraints as an optimization problem to maximize the average accuracy of the data received by all the nodes in the dissemination network.
- Due to the expensiveness of data compression operations and the need of the optimization algorithms to estimate the data accuracies with many different compression ratios, we propose an approach to estimate the expected data accuracy without actually performing the compressions. Experiments show that the approach is effective and robust.
- We proposed three algorithms to generate an optimized dissemination plan. The greedy algorithm tries to maximize the total volume of models that are disseminated to all the subscribers. However, this approach may incur a lot of re-compressions over the network and hence can incur high data inaccuracies. We further proposed two algorithms to address this problem, namely the Randomized algorithm and the Most Frequent Bandwidth First (MFBF) algorithm.
- We conducted an extensive experimental study on the performance of the proposed algorithms. We have conducted our experiments on 19 real datasets, among which the representative results of 5 datasets are presented in this paper. The experimental results show that our cost model can accurately estimate the model accuracies. Furthermore, the results suggest that none of the three proposed algorithms can outperform the others in all the situations. Our extensive experiments can help making guidelines for real deployments.

The rest of this paper is organized as follows. In section 2 we formulate the problem studied in this paper. In section 3 we propose

a cost model to estimate the accuracies with arbitrary compression ratios. In section 4 we design three approximate algorithms to optimize the problem and we experimentally evaluate the proposed methods in Section 5. Finally, in section 6 we present some related work and we conclude the paper in Section 7.

## 2. RELATED WORK

**Approximate representation models:** There exist many mathematical and statistical models to approximate and compress time-series data. They are designed to for different purposes and hence may suit different applications. For examples, Discrete Fourier Transform [8] is for mapping time-series data to frequency domains and often used for feature extraction; [30] present a framework (Cypress) to compress FFT represented time-series data to multi-scales for efficient archival and querying; Discrete Wavelet Transform (DWT)[10] is mainly used for efficient similarity search over time-series data; Singular Value Decomposition [14] is usually used for dimensionality reduction in similarity searching; Piecewise Linear Approximation (PLA) [16, 20, 17, 22, 15, 19] is a very popular approximate model and has been used in a variate of purpose, e.g. fast exact similarity search, fuzzy query, dynamic time warping and relevance feedback etc. Due to PLA’s simplicity and wide applicability, a variety of algorithms have been proposed to perform PLA modelling. Most segmentation algorithms can be divided into three categories: Sliding Windows, Top-Down, and Bottom-Up [16, 17], which are batch based algorithm. In [16], the authors proposed an online algorithm, called SWAB (Sliding Window and Bottom-up), for representing time series data by combine sliding windows and bottom-up algorithms.

The multi-resolution feature of DWT also enable it to achieve multi-scale compression. Unfortunately, the compression scales for DWT are fixed which is inefficient for compression at arbitrary scales. Given a sequence with length of  $n$ , then  $\lceil \log_2 n \rceil$  scales can be obtained by Haar Transform, i.e.  $\{\frac{1}{n}, \frac{2}{n}, \dots, \frac{2^i}{n}, \dots, 1\}$  ( $i = 1, 2, \dots, \lceil \log_2 n \rceil$ ). PLA has no such limitation on compression scales as we can see from the rest of the paper.

**Data Dissemination and Publish/Subscribe Systems.** There are many previous works on data dissemination and publish/subscribe systems, e.g. [9, 31, 35, 34]. In such system, data are continuously disseminated from the sources to distributed subscribers. To enhance the dissemination efficiency, these subscribers or brokers are typically structured into dissemination trees. In [31, 34], the authors studied the problem of optimizing dissemination trees to reduce the latency of data dissemination and hence minimize the average loss of fidelity. Their techniques of tree constructions can be adapted to construct the dissemination tree in our system.

In [35], the authors proposed a general framework to disseminate statistical models rather than raw data to the data receivers. However, their focus is on minimizing the volumes of models that are transferred over the network and hence they proposed some model routing algorithms to achieve this objective. Due to the radical difference in the application scenarios and problem formulations, their results are not applicable to this paper.

**Data as a Service:** In recently years, data has increasingly become an important commodity. For examples, Xignite [7] sells various financial data; Gnip [1] provides data from social media (e.g. Twitter and Facebook), PatientsLikeMe [6] provides more than 150,000 self-reported patient statistical data; AggData [4] sells various location data, such as the locations of all the Walmart stores. Many

cloud services that support and facilitate the setup of online data marketing have emerged, such as Windows Azure Marketplace [3] and Infochimps [5]. Furthermore, pricing of data has become an interesting research problem, e.g. Paraschos et al. have proposed a query-based data pricing model in [21]. In some sense, we also proposed a pricing model over time-series data in this paper, i.e. priced based on the compression scale.

### 3. PROBLEM FORMULATION

#### 3.1 System Model

In our system, a data source  $N_0$  generates an infinite sequence of numerical values,  $(\dots, v_{i-1}, v_i, v_{i+1}, \dots)$ , which will be disseminated to a large number of data receivers,  $\{N_1, N_2, \dots, N_n\}$ . Periodically,  $N_0$  will collect a subsequence of the data and disseminate it to the receivers.

In this paper, we consider a tree-based dissemination architecture, which is a widely adopted approach [35, 9, 31, 34]. In this architecture, all nodes are organized into a dissemination tree with the root  $N_0$  acting as the source. Data is disseminated from  $N_0$  and relayed by the internal nodes to all the nodes in the network.

**Example 1:** Figure 1 shows an example of dissemination tree  $\mathcal{T}$ , which involves 9 nodes,  $\{N_0, N_1, \dots, N_8\}$ . Each node, other than  $N_0$ , is under a bandwidth constraint respectively, i.e.  $\{c_1, c_1, \dots, c_8\}$ . An instance of bandwidth constraints is also given, which is specified by the integer attached to each node.

**DEFINITION 1.** The dissemination path to  $N_i$ , denoted as  $\mathcal{P}(N_i)$ , is defined as a sequence of nodes,  $\langle p_k(N_i), \dots, p_2(N_i), p_1(N_i), N_i \rangle$ . Where  $p_j(N_i)$  ( $1 \leq j \leq k$ ) denotes the  $j$ -th preceding node of  $N_i$  on  $\mathcal{P}(N_i)$ , specially  $p_1(N_i)$  is  $N_i$ 's parent node and  $p_k(N_i)$  is the root node. All these preceding nodes form a set  $\varphi(N_i)$ .

For example, as shown in Figure 1, the dissemination path to  $N_8$ , denoted as  $\mathcal{P}(N_8)$ , is the sequence of nodes  $\langle N_3, N_1, N_0 \rangle$ .

As we addressed earlier, each node in the dissemination network embedded with some constraints, such as physical bandwidth of its communication link, demands on granularity or accuracy of data, etc. Here, we use an abstract parameter *bandwidth constraint* of  $N_i$  to represent the constraint on the volume of data could be transmitted to  $N_i$  per time unit. For example, as shown in Figure 1, the bandwidth constraint of  $N_1$  is  $c_1$ , so the data volume sent to  $N_1$  from  $N_0$  per time unit should not exceed  $c_1$ .

In practice, a child node may has a higher bandwidth constraint than its parent node, e.g.  $c_3 > c_1$ . However, the volume of data received is no more than its parent node as such data originates from the data received by its parent. It means that the volume of data transmitting along any dissemination path is non-strictly monotonously decreasing. We refer to this truth as *volume decreasing property*.

#### 3.2 Model-based Dissemination

**Piecewise Linear Models.** In PLA, the data sequence would be partitioned into a number of non-overlapping and consecutive subsequences and each subsequence is represented with a linear function [16]. Assume a sequence of time-series data  $\mathcal{D} = (v_1, v_2, \dots, v_k)$  has been divided into  $\iota$  subsequences  $\mathcal{S} = (s_1, s_2, \dots, s_\iota)$ , where  $s_j$  ( $j \leq \iota$ ) consists of a subsequence of data points with

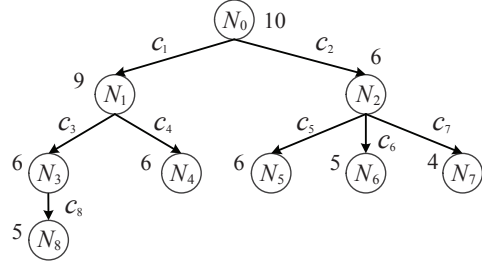


Figure 1: Dissemination Tree  $\mathcal{T}$

consecutive indices  $(v_{t_j}, v_{t_j+1}, \dots, v_{t_j+\tau})$  and  $\tau$  is the size of  $s_j$ . Each subsequence  $s_j$  is approximated with a linear function  $\ell_j$ . All these linear functions  $M = (\ell_1, \ell_2, \dots, \ell_\iota)$  compose a piecewise linear approximate representation of the original data sequence  $\mathcal{D}$ .

The segmentation of data sequences is an optimization problem that has been extensively studied in previous work, e.g. [16, 11], and is out of the scope of this paper. In our implementation, we adopt the SWAB algorithm proposed in [16], which is an online algorithm and suitable for our scenario.

**DEFINITION 2.** The volume (or size) of a sequence of models  $M = (\ell_1, \ell_2, \dots, \ell_\iota)$  is defined as the number of linear functions that it contains, denoted as  $|M|$ .

**Example 2.** Figure 2 shows an example of using different number of linear piecewise models to approximate a sequence of dynamic data. Where a) is a sequence of raw Electrocardiogram data (ECG) originates from the dataset in [11], and in b), c) and d) we use 23, 13 and 3 linear segments to represent the raw data respectively.

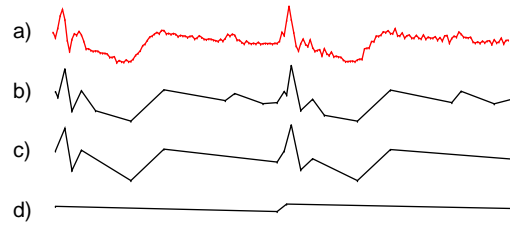


Figure 2: Piecewise Linear Approximation

**Models' Accuracy.** As above example shows, we can use various sequences of models to approximate the raw data. So we need some criteria to evaluate the quality for a sequence of models, i.e. *error* and *accuracy*.

Suppose models  $M$  is a approximate representation of original data  $\mathcal{D}$ , and  $\mathcal{D}' = (v'_1, v'_2, \dots, v'_k)$  is the corresponding time-series data of  $M$ , where  $v'_j$ ,  $1 \leq j \leq k$ , is the vertical value of  $M$  at the time instance of  $v_j$ . In some of the literatures [17, 25], the *error* of using  $M$  to represent  $\mathcal{D}$  is defined by the sum of vertical differences between the  $\mathcal{D}'$  and  $\mathcal{D}$  of at all data points, i.e.  $\sum_{j=1}^k |v_j - v'_j|$ . In our work, it has been modified as a relative error, modification on this definition, where normalize the error into a value in range  $[0, 1]$ .

DEFINITION 3. Given a sequence of dynamic data  $\mathcal{D} = (v_1, v_2, \dots, v_k)$ , a sequence of models  $M$  and its corresponding data  $\mathcal{D}' = (v'_1, v'_2, \dots, v'_k)$ , the error of using  $M$  to approximate represent  $\mathcal{D}$  is defined as

$$\varepsilon = \frac{\sum_{j=1}^k |v_j - v'_j|}{\sum_{j=1}^k |v_j|}.$$

$\varepsilon$  is the relative deviation between  $M$  and  $\mathcal{D}$ , which also reflects the quality of using  $M$  to represent  $\mathcal{D}$ . For convenience of discussion, we define the accuracy of a sequence of models.

DEFINITION 4. The accuracy of models  $M$  received by node  $N_i$ ,  $f(N_i)$ , is defined as

$$f(M) = \begin{cases} 1 - \frac{\varepsilon}{\delta}, & \text{if } 0 \leq \varepsilon < \delta \\ 0, & \text{if } \varepsilon \geq \delta \end{cases}$$

where  $f(M) \in [0, 1]$ ,  $\varepsilon$  is the error of  $M$ , and  $\delta$  is a user defined constant.

As we can see in Figure 2, various models can be used to represent the same source data  $\mathcal{D}$ . However, some models may seriously deviate from  $\mathcal{D}$ , which are not suitable to representing  $\mathcal{D}$ . Those models can be excluded from the user's choice by adjusting factor  $\delta$ . With a lower  $\delta$  value, more models' accuracies will become 0 and hence it tends to exclude more models from the solution.  $\delta$  can be set according to the application's requirements and can be set to a relative smaller value while users need high quality models.

### 3.3 Dissemination Plan

**Dissemination Plan.** A dissemination plan  $\mathcal{X} = (r_{N_0}, r_{N_1}, r_{N_2}, \dots, r_{N_n})$  is a vector of compression ratios, which defined how many models will be generated and transmitted to each node per time unit.

Assume models received by node  $N_i$  is  $M_i$ , and  $M_0$  is the initial models. Then, the volume of  $M_i$  is  $|M_i| = r_{N_i} \cdot |M_0|$ .

Take Figure 1 as an example. Assume  $(c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8) = (10, 9, 6, 6, 7, 6, 5, 4, 5)$ . Then,  $\mathcal{X}_1 = (1, 0.9, 0.6, 0.6, 0.7, 0.6, 0.5, 0.4, 0.5)$  is a dissemination plan. Note that  $\mathcal{X}_1$  is feasible as it satisfies both the bandwidth constraints and volume decreasing property. However, plan  $\mathcal{X}_2 = (1, 0.9, 0.6, 0.6, 0.7, 0.6, 0.5, 0.4, 0.6)$  is infeasible, as it violates the bandwidth constraints. Furthermore,  $\mathcal{X}_3 = (1, 0.9, 0.6, 0.4, 0.7, 0.6, 0.5, 0.4, 0.5)$  is also infeasible, because it does not satisfy the volume decreasing property. The volume of models received by  $N_8$  is larger than  $N_3$ 's, which is impossible as  $N_3$  is  $N_8$ 's preceding node on the path  $\mathcal{P}(N_8) = \langle N_0, N_1, N_3, N_8 \rangle$ .

**Remodelling.** Due to the bandwidth constraints in the dissemination network, it is inevitable a dissemination plan  $\mathcal{X}$  may state that a child node should receive less data than its parent node. Suppose  $N_i$  is an internal node and  $N_j$  is its child node and  $r_{N_i} < r_{N_j}$ . Then, node  $N_i$  is responsible to compress its models to a new sequence of models with size of  $r_{N_j} \cdot |M_0|$ . This process is called *remodelling*. For each internal node, only single remodelling is required regardless the number of its children. Assuming  $N_i$  has more than one child embedded with diverse bandwidth constraints, and  $N_j$  is the smallest. By using PLA, models in each scale can be generated naturally when data is compressed at ratio of  $r_{N_j}$ .

Note that the accuracy of the models  $M_i$  transmitted to  $N_i$  is determined by the specific remodelling processes along the dissemination path  $\mathcal{P}(N_i)$  rather than solely on  $M_i$ 's volume. For example, as shown in a model-based dissemination network Figure 1,  $f(M_2)$  is different with  $f(M_3)$ , though the volumes of models received by both are equal (i.e. 6). Models  $M_2$  have higher accuracy as they are derived by from the original data ( $10 \rightarrow 6$ ), but the models for  $N_3$  are generated from models with lower accuracy ( $9 \rightarrow 6$ ).

### 3.4 Optimization Problem

Our goal is find out the optimal solution from all feasible dissemination plans. The objective function for this problem is defined as average accuracy of models received by all nodes within the dissemination network. More formally, given the set of models  $\mathcal{M} = (M_0, M_1, M_2, \dots, M_n)$  to be received by  $\mathcal{N} = (N_0, N_1, \dots, N_n)$ , the average accuracy  $\mathcal{F}(\mathcal{M})$  is defined as

$$\mathcal{F}(\mathcal{M}) = \frac{1}{n} \sum_{i=0}^n f(N_i)$$

The formal problem statement is as follow.

**MAXIMUM-MODEL-DISSEMINATION (MMD):** Given a dissemination tree  $\mathcal{T}$  composed by a set of nodes  $\mathcal{N}$ , the bandwidth constraints  $\mathcal{C}$  of all the nodes in  $\mathcal{N}$ , choose a dissemination plan  $\mathcal{X}$  that  $\mathcal{F}(\mathcal{M})$  is maximized subjecting to the following conditions:

$$r_{N_i} \leq \frac{c_{N_i}}{c_{N_0}} \wedge r_{N_i} \leq r_{N_j}, \text{ where } N_j \in \varphi(N_i)$$

### 3.5 Challenges

In this subsection, we will discuss the challenges of solving the aforementioned optimization problem.

**THEOREM 1.** The Maximum-Model-Dissemination problem is NP-hard.

Due to the page limit, we omit the proof.

**Additional Challenges.** To solve the MMD problem, we have to design algorithms to search the solution space and try to find the optimal plan with the minimum  $\mathcal{F}(\mathcal{M})$ . Besides the NP-Hardness of the problem, there are some additional challenges to be addressed. In the optimization algorithms, we have to generate many different dissemination plans and compare their costs. A straightforward way to estimate the cost of a dissemination plan will work like the following. First, we can use some historical data to generate the models to be disseminated to each node. Then we can calculate the accuracy of these models according to Definition 4. However, the modelling process is usually very expensive and it has to be done for every possible plan that is considered in the optimization algorithms. Hence, this approach is computationally prohibitive.

One may think that the models' accuracy is related to the number of models that are sent to each node and hence an alternative approach is to transform the problem to maximizing the number of models that are transmitted to each node. Unfortunately, the models' accuracy depends on not only the volume of models but also, more importantly, the actual remodelling processes within the dissemination network. For example, in Figure 1,  $(10, 9, 6, 6, 7, 6, 5, 4, 5)$  is a feasible plan with the largest total volume of models, but it is not an optimal plan. If we change  $x_1$  to 6, then  $f(N_1)$  would

be decreased while  $f(N_3)$ ,  $f(N_4)$  and  $f(N_8)$  would be increased. Therefore it is possible that the latter plan would have a higher average accuracy. This is also validated by our experiments presented later in the paper.

In summary, we need to find a method to calculate the models' accuracy that can take the remodelling processes into consideration but need not actually generate the models. The basic idea is to obtain a function of the models' accuracy that can produce the accuracy estimation without the actual models. Such a function would be multi-dimensional with the parameters being the compression ratios of the nodes on the dissemination path. Finding an efficient way to generate such a function is challenging.

## 4. MODEL-BASED DISSEMINATION

In this section, we highlight the overall dissemination procedure in our framework. Given a feasible plan  $\mathcal{X}$ , the model-based dissemination works as follows. It starts at the root of  $\mathcal{T}$  and end until all leaves received their models. At first, the initial models  $M_0$  will be generated from the raw data  $\mathcal{D}$  in the root node. With the PLA models, these initial models are simply line segments that connect pairs of adjacent points and hence has the same volume as the raw data.

For each internal node  $N$ , new sequences of models will be generated based on the models received by  $N$  according to  $\mathcal{X}$  and then are transmitted to the corresponding children. The detail operation executed at each node is illustrated in Algorithm 1. The *remodel()* (line 8) produces re-compress the input model sequence according a specified size.

In our current implementation, we use the Bottom-Up algorithm proposed in [16]. The Bottom-Up algorithm will create  $m/2$  segments to approximate represent the raw data initially. Then, the cost of merging each pair of adjacent segments is calculated, and it merge a pair with the lowest cost iteratively. The algorithm stop until some user-specified criteria is met, such as compression ratio. Of course, after one merge operation, the cost of merging the new longer segment with its left neighbor and right neighbor need be re-calculated.

This remodelling operations in an internal node can be further optimized according to the actual compression methods being used. For instance, with Bottom-Up approach, the new sequences of models for all the children can be generated by running one pass of the compression from models  $M(N_i)$  to  $M(nodes[k])$ , where  $nodes[k]$  is the child with the smallest compression ratio.

---

### Algorithm 1: Model-based Dissemination

---

**Input:** The current node  $N_i$ , Models  $M(N_i)$  received by this node, Plan  $\mathcal{X}$

```

1  $nodes[0] \leftarrow N_i$ ;
2 append all child nodes  $nodes$  ;
3 sort  $nodes$  in descending order by their compression ratios;
4 for  $j = 1$  to  $sizeof(nodes)$  do
5   if  $r_{nodes[j]} < r_{nodes[j-1]}$  then
6      $M(nodes[j]) \leftarrow remodel(M(nodes[0]), r_{nodes[j]})$ ;
7   else
8      $M(nodes[j]) \leftarrow M(nodes[j-1])$ ;
9   transmit  $M(nodes[j])$  to  $nodes[j]$ ;
```

---

## 5. COST MODEL

As mentioned earlier, it is very costly to estimate the models' accuracy. Our general approach is to try to find a proper form of the accuracy function and use historical data and curve fitting methods to find out the appropriate coefficients of the function.

As the accuracy of the models send to  $N_i$  is determined by the remodelling processes in its dissemination path, we can define a general form of the accuracy function,  $f(N_i)$ , as follows.

**DEFINITION 5.**  $f(N_i)$ , the accuracy of node  $N_i$ , is a multidimensional function of the compression ratios of all the nodes within  $\mathcal{P}(N_i)$ :

$$f(N_i) = \tilde{h}(r_{N_0}, r_{p_{k-1}(N_i)}, \dots, r_{p_1(N_i)}, r_{N_i}) \quad (1)$$

where  $p_{k-1}(N_i) \in \varphi(N_i)$ , and  $r_{p_{k-1}(N_i)}$  is its compression ratio and  $r_{N_0} = 1$ .

One can assume a particular form of  $f(N_i)$ , e.g. a multidimensional quadratic function, and then we can find out the co-efficients by curve fitting. However, curve fitting for multidimensional functions are usually very expensive and less accurate. In the following subsections, we try to simplify such a general function to a two-dimensional function based on several realistic observations and assumptions.

### 5.1 Observations and Assumptions

In this subsection, we present a few observations and assumptions that will be used to simplify the accuracy function.

**ASSUMPTION 1.** For two distinct arbitrary nodes  $N_i$  and  $N_j$  in  $\mathcal{T}$ , if  $r_{N_i} = r_{N_j}$  and  $f(N_i) = f(N_j)$ , then the sequence of models received by  $N_j$  and  $N_i$ , i.e.  $M_i$  and  $M_j$ , are equivalent.

For two equivalent sequences of models  $M_i$  and  $M_j$ , they can be replaced by each other.

**ASSUMPTION 2.** For an arbitrary node  $N_i$ ,  $0 \leq f(N_i) \leq 1$  holds.  $f(N_i) = 1$  if and only if  $r_{N_i} = 1$ .

Assumption 2 states that a node's accuracy is a positive value that falls in  $[0, 1]$  and it is equal to 1 if and only if the data it received do not undergo any remodeling, i.e. the original data.

**ASSUMPTION 3.** For an arbitrary node  $N_i$ ,  $r_{N_i} \leq r_{p_1(N_i)}$  and  $f(N_i) \leq f(p_1(N_i))$  hold.  $f(N_i) = f(p_1(N_i))$  if and only if  $r_{p_1(N_i)} = r_{N_i}$ .

Assumption 3 means that the accuracy of an arbitrary node is no more than the accuracy of its ancestor nodes and the compression ratios fulfill the *volume decreasing property* on a dissemination path. The equality holds if and only if there is no remodeling occurs while models transmitted from  $p_j(N_i)$  to  $N_i$  over the path  $\mathcal{P}(N_i)$ .

The next two assumptions are about two distinct arbitrary nodes in  $\mathcal{T}$ ,  $N_i$  and  $N_j$ . Figure 3 shows their dissemination paths.

**ASSUMPTION 4.** If  $r_{p_1(N_i)} = r_{p_1(N_j)}$ ,  $r_{N_i} = r_{N_j}$ , and  $f(p_1(N_i)) > f(p_1(N_j))$ , then  $f(N_i) > f(N_j)$  holds.

Assumption 4 states that, if  $N_i$  and  $N_j$  are equal in their compression ratios and so are their parents, their accuracies are determined, respectively, by their parents' accuracies.

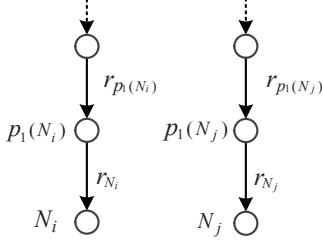


Figure 3: Two Dissemination Paths

ASSUMPTION 5. If  $r_{p_1(N_i)} = r_{p_1(N_j)}$ ,  $f(p_1(N_i)) = f(p_1(N_j))$  and  $r_{N_i} > r_{N_j}$ , then  $f(N_i) > f(N_j)$ .

According to Assumption 1, the model sequences received by  $p_1(N_i)$  and  $p_1(N_j)$  are equivalent under the conditions of Assumption 5. Simply put, Assumption 5 states that the accuracies of the new models generated from two sequence of equivalent models only depends on its compression ratio.

ASSUMPTION 6. The accuracy difference between nodes  $p_1(N_i)$  and  $N_i$ , i.e.  $f(p_1(N_i)) - f(N_i)$ , is a binary quadratic function of  $r_{p_1(N_i)}$  and  $r_{N_i}$ :

$$f(p_1(N_i)) - f(N_i) = \alpha_1 r_{N_i}^2 + \beta_1 r_{N_i} + \alpha_2 r_{p_1(N_i)}^2 + \beta_2 r_{p_1(N_i)} + \gamma r_{N_i} r_{p_1(N_i)} + \omega \quad (2)$$

where  $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma$  and  $\omega$  are constant coefficients.

Assumption 6 is to simplify the accuracy function from a multi-dimensional one to a two-dimensional quadratic function. One can of course assume a more complicated function. But, as we will see later in the experiment results, this function performs quite well in practice and hence we will use it due to its simplicity.

## 5.2 Accuracy Function

We can further reduce the coefficients in Eqn. (2) by applying the equivalent condition in Assumption 3. More specifically, according to Assumption 3, if  $r_{p_1(N_i)} = r_{N_i}$ , then  $f(N_i) = f(p_1(N_i))$ . We can derive that

$$0 = \alpha_1 r_{N_i}^2 + \alpha_2 r_{N_i}^2 + \gamma r_{N_i}^2 + \beta_1 r_{N_i} + \beta_2 r_{N_i} + \omega$$

Therefore,  $\beta_1 = -\beta_2$ ,  $\alpha_1 + \alpha_2 + \gamma = 0$  and  $\omega = 0$ .

By substitution, we can simplify Eqn. (2) as follows:

$$f(p_1(N_i)) - f(N_i) = \alpha_1 r_{N_i}^2 + \alpha_2 r_{p_1(N_i)}^2 + \beta_1 r_{N_i} - \beta_1 r_{p_1(N_i)} - (\alpha_1 + \alpha_2) r_i r_{p_1(N_i)} \quad (3)$$

LEMMA 1. The accuracy  $f(N_i)$  defined by Eqn.(3) satisfies Assumption 2 to 5.

PROOF. It is easy to prove Eqn. (3) satisfies Assumption 2 and 3. According to Assumption 6, we know that

$$\begin{aligned} f(N_0) - f(p_{k-1}(N_i)) &\geq 0, \\ &\dots \\ f(p_1(N_i)) - f(N_i) &\geq 0, \\ f(N_i) &\geq 0. \end{aligned}$$

Thus,  $1 = f(N_0) \geq \dots \geq f(p_1(N_i)) \geq f(N_i) \geq 0$ . Obviously,  $f(p_1(N_i)) = f(N_i)$  holds if and only if  $r_{p_1(N_i)} = r_{N_i}$ . In addition,  $r_{N_i} \leq r_{p_1(N_i)}$  is satisfied by the *volume decreasing property*. Therefore, (3) satisfies Assumption 2 and 3.

Consider the path  $\mathcal{P}(N_j)$  in Figure 3, according to function (3), we have:

$$f(p_1(N_j)) - f(N_j) = \alpha_1 r_{N_j}^2 + \alpha_2 r_{p_1(N_j)}^2 + \beta_1 r_{N_j} - \beta_1 r_{p_1(N_j)} - (\alpha_1 + \alpha_2) r_{N_j} r_{p_1(N_j)} \quad (4)$$

For Assumption 4, let  $r_{p(N_i)} = r_{p(N_j)}$  and  $r_{N_i} = r_{N_j}$ , then

$$(3) - (4) \Rightarrow f(N_i) - f(N_j) = f(p_1(N_i)) - f(p_1(N_j)).$$

Thus, if  $f(p_1(N_i)) \geq f(p_1(N_j))$ , obviously  $f(N_i) \geq f(N_j)$  also holds.

For Assumption 5, as  $r_{p_1(N_i)} = r_{p_1(N_j)}$  and  $f(p_1(N_i)) = f(p_1(N_j))$ , we can derive (4) - (3) as

$$\begin{aligned} f(N_i) - f(N_j) &= \alpha_1 (r_{N_j}^2 - r_{N_i}^2) + \beta_1 (r_{N_j} - r_{N_i}) - (\alpha_1 + \alpha_2) r_{p_1(N_i)} (r_{N_j} - r_{N_i}) \\ &= (r_{N_j} - r_{N_i}) [\alpha_1 (r_{N_i} + r_{N_j}) + \beta_1 - (\alpha_1 + \alpha_2) r_{p_1(N_i)}] \end{aligned} \quad (5)$$

It is easy to make  $\alpha_1 (r_{N_i} + r_{N_j}) + \beta_1 - (\alpha_1 + \alpha_2) r_{p_1(N_i)} \leq 0$  holds by adjusting  $\alpha_1, \alpha_2$  and  $\beta_1$ . In addition  $r_{N_i} \geq r_{N_j}$ , thus  $f(N_i) \leq f(N_j)$ .

According to above analysis, it is clear that the function satisfies all assumptions.  $\square$

If we set  $r_{p_1(N_i)} = 1$  (i.e.  $f(p_1(N_i)) = 1$ ) in Eqn. (3), then

$$f(N_i) = -\alpha_1 r_{N_i}^2 + (\alpha_1 + \alpha_2 - \beta_1) r_{N_i} + (1 + \beta_1 - \alpha_2) \quad (6)$$

This means that if the models sequence  $M_i$  is obtained by direct modelling over the original data, then the accuracy is a quadratic function of its compression ratio. This is also validated in our experiments in Section 7.

LEMMA 2. For a dissemination path  $\mathcal{P}(N_i)$  to  $N_i$ ,  $f(N_i)$  is a quadratic function of  $r_{N_i}$  if all the compression ratios of  $N_i$ 's preceding nodes on  $\mathcal{P}(N_i)$  are given.

Lemma 2 shows a way to obtained the specific form of the accuracy function. If we fix all the compression ratios of the preceding nodes of  $N_i$  on  $\mathcal{P}(N_i)$ , then we can use a quadratic function to fit the  $N_i$ 's actual accuracy by using the historical data. Through the experiments, we found that a quadratic polynomial function can fit all the tested datasets pretty well.

## 5.3 Computing Model Accuracy

With the cost model, model accuracy can be calculated without source data. It simplifies the calculation of model accuracy. It can also enormously improve the efficiency of calculation, which is crucial for low-latency dissemination.

As stated earlier, the accuracy of a sequence of models  $M_i$  is determined by the specific remodelling processes on the dissemination path  $\mathcal{P}(N_i)$ . It is impossible to enumerate all possible remodelling



processes of a dissemination network and hence we can not pre-compute all possible models' accuracies. Alternatively, the compression ratios ( or volume ) of models are classified into several scales according to the bandwidth constraints (subscription levels) of the receivers. All the possible remodelling paths will form a *Remodelling Tree (RT)* as illustrated in Figure 4 as an example, where the compression ratios are classified into 10 scales. A complete *remodelling path* in *RT*, from the root to a leaf, is a sequence of strictly monotonically decreasing compression ratio values, which starts at compression ratio of 1 and end at 0.1. Furthermore, the models' accuracies on arbitrary possible remodelling paths are pre-computed regarding to these scales.

With a *RT*, models' accuracies can be approximately calculated regardless of its dissemination path. In order to estimate models' accuracies on an actual dissemination path  $\mathcal{P}(N_i)$ , we should find the closest remodelling path for  $\mathcal{P}(N_i)$  in the *RT*. For example, given a dissemination path  $\langle 1, 0.9, 0.9, 0.75, 0.43, 0.1 \rangle$ , the closest path in *RT* is  $\langle 1, 0.9, 0.8, 0.4, 0.1 \rangle$ . Then, the accuracies for each node on  $\mathcal{P}(N_i)$  are approximated by the accuracies on the remodelling path. Due to the space limit, we will not discuss the details about the algorithm, which is quite straightforward.

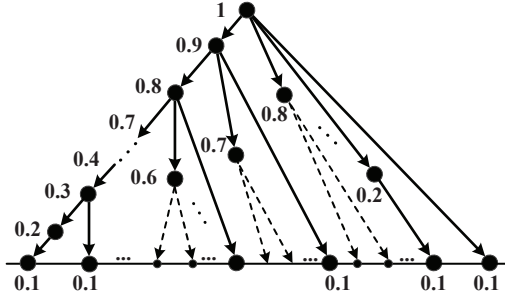


Figure 4: Remodelling Tree *RT*

## 6. OPTIMIZATION ALGORITHMS

With the cost model developed above, we will, in this section, design algorithms to optimize the dissemination plan  $\mathcal{X}$ . As stated, *MMD* is NP-hard. Therefore, we will design several heuristic-based algorithms to solve this problem.

### 6.1 Greedy Algorithm

As analyzed earlier, the accuracy of a sequence of models is related to its size. Therefore, in the first algorithm, we try to maximize the number of models that are sent out from each internal node. In other words, we set the compression ratio of each node as large as possible. This algorithm is presented in Algorithm 2 and works as follows. It starts from the root node  $N_0$  and set its compression ratio as 1. Then it traverse the tree in a breadth first manner. For each internal node  $N_i$  and its child node  $N_j$ , according to the volume decreasing property and bandwidth constraint,  $r_{N_j}$  is set as  $\text{minimum}(\frac{c_{N_j}}{|M_0|}, r_{N_i})$ . Algorithm stops when all the nodes' compression ratio is set. Greedy algorithm rjust need one traversal of  $\mathcal{T}$  and hence its complexity is  $\mathcal{O}(n)$ .

### 6.2 Randomized Algorithm

By a closer investigation, we identify two potential problems of the above greedy algorithm. First, in the greedy algorithm, we try to maximize the number of models that are sent from each parent to its

---

### Algorithm 2: Greedy Algorithm

---

**Input:** Dissemination tree  $\mathcal{T}$ , bandwidth constraints  $\mathcal{C}$ , initial models  $M_0$

**Output:** dissemination plan  $\mathcal{X}$

- 1 Let *NodeQueue* be a queue for preserving nodes and  $t$  be the current visiting node;
  - 2 Initialization:  $r_{N_0} \leftarrow 1$ ;
  - 3 *enqueue*(*NodeQueue*,  $N_0$ );
  - 4 **while** *NodeQueue* is not empty **do**
  - 5      $t \leftarrow \text{dequeue}(\text{NodeQueue})$ ;
  - 6      $\{t_1, \dots, t_k\} \leftarrow$  find out all children of  $t$  in  $\mathcal{T}$ ;
  - 7     **for**  $i = 1$  **to**  $k$  **do**
  - 8          $\text{enqueue}(\text{NodeQueue}, t_i)$ ;
  - 9          $r_{t_i} \leftarrow \text{minimum}(r_t, \frac{c_{t_i}}{|M_0|})$ ;
- 

children. However this may not provide the optimal solution. For example, in the feasible plan depicted in Figure 1, if we decrease the volume of models received by  $N_1$  from 9 to 6, then we do not need to perform remodelling at  $N_1$  and hence the accuracies of the models sent to  $N_1$ 's descendants would be increased. This may actually end up with a better plan. Generally speaking, if we decrease  $r_{N_i}$ , then it is possible that some of its children's accuracies would be increased by avoiding the remodelling process. Second, the greedy algorithm only search for a limited part of the solution space and hence would easily produce a sub-optimal solution.

Based on the above observations, we try to design a new algorithm to address the above two potential problems. In this algorithm, for a internal node, its compression ratio is chosen within range  $[\text{min}C/|M_0|, \text{max}C/|M_0|]$ , where  $\text{min}C$  and  $\text{max}C$  are the minimum and maximum bandwidth constraints among its child nodes. This is to address the first problem stated above.

Furthermore, instead of using a greedy algorithm, we employ a randomized local search strategy. More specifically, we use the *stochastic hill climbing* approach which selects a neighbor at random. It will run in many iterations and within each iteration, the compression for a node will be altered to a value within range  $[\text{min}C/|M_0|, \text{max}C/|M_0|]$ . This procedure will be repeated a number of times until reached a local optimal and the corresponding plan will be chosen. The initial plan will be randomly generated by Algorithm 3.

In Algorithm 3, each node will be visited twice (except  $N_0$ ) within each iteration: one time for determining the compression ratio of its parent and another for itself. Besides, finding the  $\text{min}C$  and  $\text{max}C$  for a non-leaf node needs one scan of its children. Therefore the cost of running Algorithm 3 is  $2n + \sum_{i=0}^{n_0} \theta_i$ , where  $\theta_i$  is the number of child nodes of node  $N_i$ .

In addition, each non-leaf node  $N_i$  means an iteration for the whole algorithm and at  $N_i$  it would search at most  $\theta_i$  plans. Suppose the cost for calculate the accuracy for a plan is  $k$  times of the cardinality of the plan, i.e  $k \cdot n$ . Thus,

$$\vartheta(n) = (2n + \sum_{i=0}^{n_0} \theta_i) + \sum_{i=0}^{n_0} \theta_i \cdot (k \cdot n) \leq 3n + k \cdot n^2$$

The complexity of the whole randomized algorithm is  $\mathcal{O}(\vartheta(n)) = \mathcal{O}(n^2)$ .



---

**Algorithm 3:** Randomized Algorithm

---

**Input:** dissemination tree  $\mathcal{T}$ , nodes set  $N = \{N_0, \dots, N_n\}$ , bandwidth constraints  $\mathcal{C}$ , initial models  $M_0$

**Output:** initial plan  $\mathcal{X}_0$

```
1 Let  $t$  be current visiting node;
2 Initialization:  $r_{N_0} \leftarrow 1$ ;
3 for  $i = 1$  to  $n$  do
4    $t \leftarrow N_i$ ;
5   if  $t$  is not a leaf node then
6      $T \leftarrow$  find out all children of  $t$ ;
7      $min \leftarrow$  minimum bandwidth in  $T$ ;
8      $max \leftarrow$  maximum bandwidth in  $T$ ;
9      $rand \leftarrow$  random integer in  $[minC, maxC]$ ;
10  else
11     $rand \leftarrow c_t$ ;
12   $r_t \leftarrow \text{minimum}(r_{p_1(t)}, \frac{rand}{|M_0|})$ ;
```

---

---

**Algorithm 4:** MFBF Algorithm

---

**Input:** dissemination tree  $\mathcal{T}$ , nodes set  $N = \{N_0, \dots, N_n\}$ , bandwidth constraints  $\mathcal{C}$ , initial models  $M_0$

**Output:** dissemination plan  $\mathcal{X}$

```
1 Let  $U$  be the unvisited nodes set and  $V$  be the visited nodes set of  $\mathcal{T}$ ;
2 Initialization:  $r_{N_0} \leftarrow 1, U \leftarrow N - N_0, V \leftarrow N_0$ ;
3 while  $U \neq \emptyset$  do
4    $b \leftarrow$  the most frequent bandwidth of nodes in  $U$ ;
5    $\{t_1, \dots, t_k\} \leftarrow$  all nodes with bandwidth of  $b$  in  $U$ ;
6   for  $i = 1$  to  $k$  do
7      $\{p_1, \dots, p_m\} \leftarrow$  all preceding nodes of  $t_i$  in  $U$ ;
8      $p \leftarrow$  the lowest preceding node of  $t_i$  in  $V$ ;
9      $r_{t_i} \leftarrow \text{minimum}(r_p, \frac{b}{|M_0|})$ ;
10  for  $j = 1$  to  $m$  do
11     $r_{p_j} \leftarrow r_{t_i}$ ;
12    move  $p_j$  from  $U$  to  $V$ ;
```

---

### 6.3 MFBF Algorithm

The above randomized algorithm needs many iterations to obtain a good plan. So even though the algorithm still has a linear asymptotic complexity, constant coefficient may still be very big. In this subsection, we will design a more efficient algorithm that can still produce a good plan.

The algorithm is called *Most Frequent Bandwidth First* (MFBF). The basic idea of this algorithm is to minimize the number of re-modelling occurred in the dissemination tree. Therefore, we try to put higher priority to maximizing the accuracies of the nodes with more frequent bandwidth constraints. The details of MFBF is described in Algorithm 4.

As the first step, this algorithm will perform a traversal of the dissemination tree and if bandwidth constraint of a node is greater than the maximum  $maxP$  constraint on its dissemination path, then it will be re-set as  $maxP$ .

The algorithm runs in multiple iterations. Within each iteration, it follows the following procedures

- Find the most frequent bandwidth  $\kappa$  among the unvisited nodes and put the unvisited nodes with bandwidth of  $\kappa$  into a queue  $T$ .
- Iterate over  $T$  and, for each node  $t_i$  in  $T$ , set the compression ratio of  $t_i$  and all its *unvisited* ancestor nodes to a value  $r$ .  $r$  is calculated as the minimum value between  $c_{t_i}/|M_0|$  and the compression ratio of  $t_i$ 's lowest *visited* preceding nodes. After this step,  $t_i$  and all its unvisited ancestor nodes are considered visited.

The cost of MFBF algorithm mainly consists of three parts: 1) the cost of finding the most frequent bandwidth from the unvisited nodes  $U$ , denoted as  $h_1$ ; 2) the cost of finding  $N_i$ 's unvisited ancestor nodes from  $U$ , denoted as  $h_2$ ; 3) the cost of finding the lowest visited ancestor nodes, denoted as  $h_3$ . In 1), each execution of the finding the most frequent bandwidth just needs one pass scan of  $U$  and hence the complexity is  $\mathcal{O}(n)$ . Suppose that there are  $a$  distinct bandwidth constraints in  $\mathcal{T}$ , then  $h_1 \leq a \cdot n$ . In 2) and 3), arbitrary node appear as unvisited node or visited preceding node at most twice. In our implementation, we maintain a index point to parent node for each node, which means we can find a preceding node for  $N_i$  in a constant time, denoted as  $\epsilon$ . So,  $h_2 + h_3 \leq \epsilon \cdot 2n$ .

$$\begin{aligned} h(n) &= h_1 + h_2 + h_3 \\ &\leq a \cdot n + \epsilon \cdot 2n \\ &= n \cdot (a + 2 \cdot \epsilon) \end{aligned}$$

Thus,  $\mathcal{O}(h(n)) = \mathcal{O}(n)$ .

## 7. EXPERIMENTS AND EVALUATION

In this section, we will present our experimental results. We implemented the prototype system in Java. Due to the fact that our objective function is the accuracies of the data received by the subscribers, there is no difference between running it in a distributed system or simulating it on one single machine. Therefore, we simulate the whole system on one machine and hence can easily make tests with many different system parameters.

**Data.** We used a publicly available dataset [11], named as UCR time series data. UCR is a collections of time series data in mining/machine learning community. The data set we used is the *data I* in [11], which make up of 19 types time series data from diverse applications. We carry out our experiments on all of these data and present the results on five dataset. The results on the other data are similar to the reported ones. The specific time-series that we have used are: **50 Words** [29], **OSU Leaf** [12], and **Gun Point** [28, 27], **ECG** and **Wafer**[24]. Each data set is composed of hundreds of sequences, and there are hundreds of points in each sequence.

Bandwidth constraint and data rate for the source node were set to 1MBytes/s, which would be different in practice. For any other node, data rate was defined by its compression ratio (or bandwidth constraint), and its unit was omitted in latter context. The experiments can be divided into two parts:

- **Model Accuracy:** In this part, we show how to use curve-fitting approach to produce the cost model to estimate the model accuracies. Furthermore, we validate the cost model proposed in Section 5.

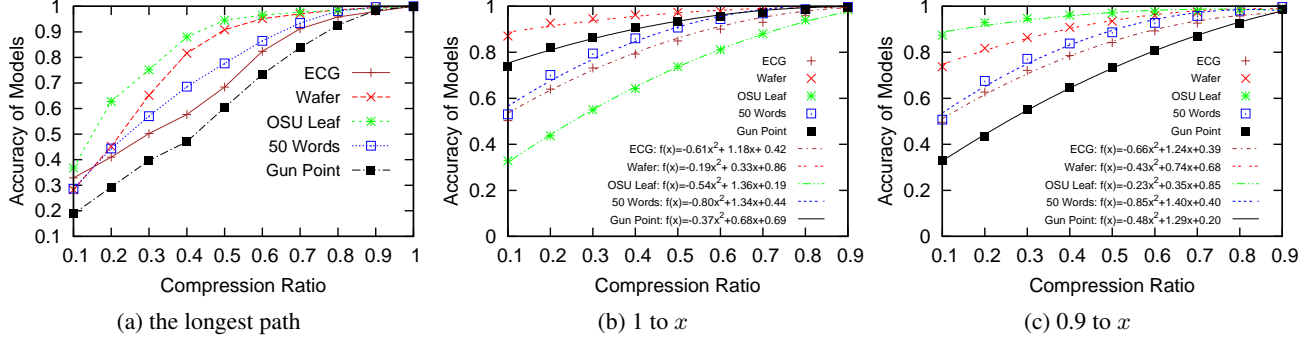


Figure 5: Model Accuracy and Corresponding Cost Model

- **Optimization Algorithms:** We construct a number of dissemination tree with multiple scales of bandwidth constraints and examine the performance of the optimization algorithms proposed in this paper.

## 7.1 Model Accuracy

We implemented the *Remodelling Tree (RT)*, as shown in Figure 4. In our experiments, we set the bandwidth constraints to ten scales, i.e.  $\{1, 0.9, 0.8, \dots, 0.1\}$ . We have also tested twenty scales, which produce similar results.

Suppose  $P = \langle 1, \dots, r_j, r_i, \dots, 0.1 \rangle$  is a remodelling path of *RT*, where  $r_j > r_i$ . According to Lemma 2, we construct a remodelling tree path for the remodelling processes on a dissemination path of  $\mathcal{T}$ , then we can get the cost model for each remodelling path by curve fitting. With the cost model, we can easily calculate the accuracy for each node.

$P_1 = \langle 1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1 \rangle$  is the longest path in *RT*. The models generated through  $P_1$  has the lowest accuracy among the models with the same compression ratio within *RT*. The results is illustrated in Figure 5 (a).

According to Lemma 2, if all compression ratios of  $N_i$ 's preceding nodes on path  $\mathcal{P}(N_i)$  are provided, then  $f(N_i)$  is a quadratic function of  $r_{N_i}$ . That means if we fix all the compression ratios before  $r_i$  in the remodelling path  $P$ , and  $r_i$  is within  $[0.1, r_j]$ , then we can get the cost model  $f(r_i)$  by curve fitting with a quadratic function.

Here, we choose two paths from *RT* to present our observation.  $P_2 = \langle 1, r_i, \dots, 0.1 \rangle$ , where  $r_i$  varies within  $\{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$ .  $P_3 = \langle 1, 0.9, r_i, \dots, 0.1 \rangle$ , and  $r_i$  varies in  $\{0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$ . As shown in Figure 5 (b) and (c), a quadratic function can fit the results quiet well, which validates our claims in Lemma 1 and 2. The other possible remodelling paths have similar results.

## 7.2 Performances of Optimization Strategies

We conduct an intensive study on the performance of the three optimization algorithms described in Section 6. The performances are measured on the average accuracy  $\mathcal{F}(\mathcal{M})$  of all nodes within the dissemination network  $\mathcal{T}$ .

We construct different dissemination tree  $\mathcal{T}$  randomly with two tunable parameters: the maximum fan-out  $\theta$  and the total number  $n$ . In our experiments, the setting of the two parameters as follows:

- $\theta$ : varying in 3 scales  $\{2, 5, 10\}$ ;
- $n$ : varying in  $\{100, 200, 500, 1000, 2000, 5000, 10000\}$ .

$\theta$  and  $n$  will affect the structure of  $\mathcal{T}$ , e.g. its height. The structure of  $\mathcal{T}$  determines the possible remodelling operations.  $\mathcal{T}$  is constructed with the following properties:

Each node in  $\mathcal{T}$  has  $num$  children, where  $num$  is a random integer within  $[0, \theta]$ . In addition, each node is associated with a bandwidth constraint whose scale fall within ten scales  $\{1, 0.9, \dots, 0.1\}$ . For a node  $N_i$  with a bandwidth constraint scale of  $i$ , its bandwidth constraint is equal to  $i \cdot |M_0|$ , where  $|M_0|$  is the expected volume of the time-series to be disseminated within each epoch. Furthermore, we simply set the bandwidth of the root to the volume of the initial models, i.e.  $c_{N_0} = |M_0|$ . In our experiments, we generate the bandwidths constraints of the nodes in  $\mathcal{T}$  based on five different distributions.

- **Level:** nodes at the same level within  $\mathcal{T}$  have the same bandwidth constraints.
- **Uniform:** the bandwidth constraints of each node obeys uniform distribution and independent on its location within  $\mathcal{T}$ .
- **Gaussian:** the bandwidth constraints obey Gaussian distribution with variance  $\sigma^2 = 1$  and mean  $\mu$  equals the median value of all the bandwidth constraints.
- **MaxZipf:** the bandwidth constraints are subject to *zipfian* distribution  $P(k) \sim \frac{1}{k^a}$ , where  $P(k)$  is the frequency of the  $k$ th bandwidth constraint and  $a = 1$ , and the larger bandwidth constraints have smaller  $k$  values.
- **MinZipf:** similar to MaxZipf except that the smaller bandwidth constraints have larger  $k$  values.

Furthermore, for each of the above given bandwidth distribution, we considered two schemes to assign these bandwidths to each node. In reality, this is determined by the organization of the dissemination tree, which is another complicated optimization problem and is out of the scope of this paper.

- **Ordered:** nodes at higher levels of the tree have larger bandwidth constraints.

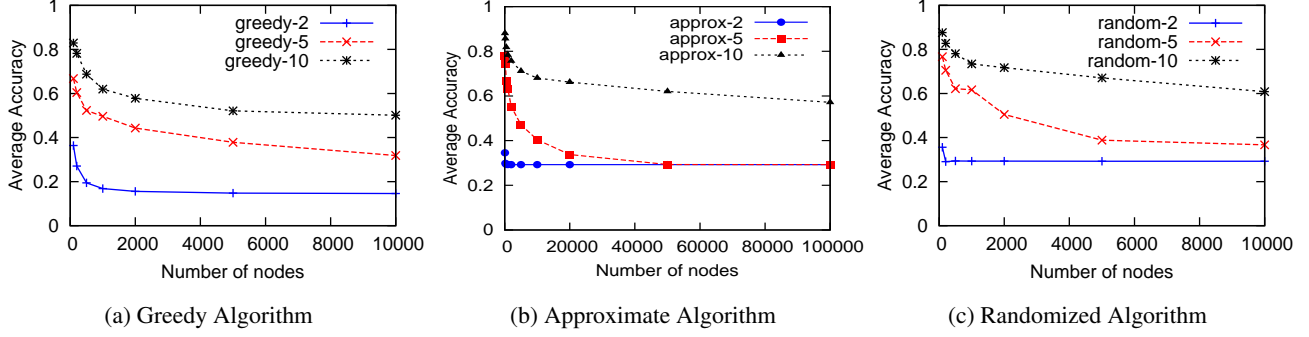


Figure 6: Horizontal Comparison

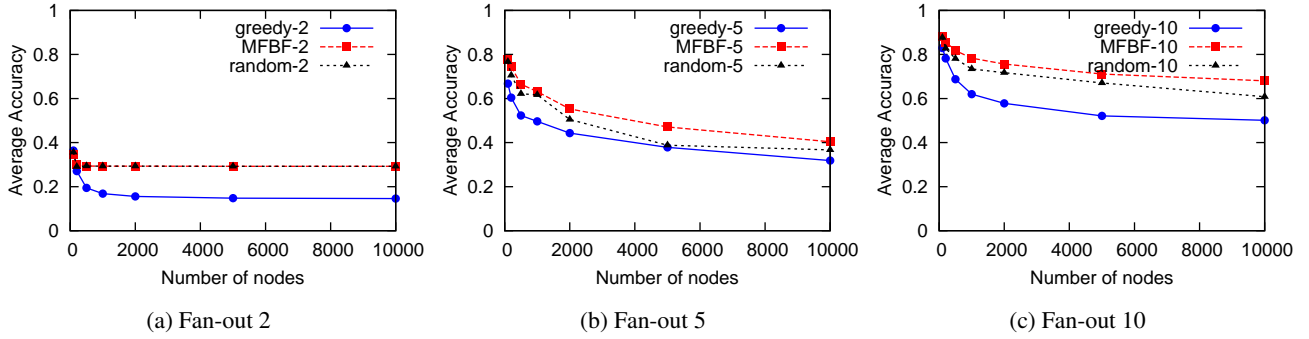


Figure 7: Vertical Comparison

- **Unordered:** no restriction on the bandwidths allocation within  $\mathcal{T}$ . Thus, we just randomly assign a bandwidths to a node  $N_i$  according to the specific distribution.

Note that we only consider the “Ordered” scheme for the “Level” distribution. Thus, there are 9 cases in total as shows in Table 1. We evaluate the performances of these algorithms with all the different distributions and the number of nodes varying from 100 to 10000. We repeat the each group of experiments for 100 times and take the mean of the average accuracies  $\mathcal{F}(\mathcal{M})$  as the final result.

Table 1: Possible Combinations

Distribution	Ordered	Unordered
Uniform	uniform-1	uniform-0
Gaussian	gaussian-1	gaussian-0
MaxZipf	maxzipf-1	maxzipf-0
MinZipf	minzipf-1	minzipf-0
Level	level	

### 7.3 Performances Analysis

The randomized searching algorithm applies the stochastic hill climbing search to optimize the dissemination, which is a costly procedure. For example, it takes 2 hours to complete the experiments with 10,000 nodes, while it takes only less than 1 minute for the other two algorithms. Due to the long running time of the randomized algorithm, we just present its results on the **Level** distribution, as shown in Figure 6 and Figure 7. For the other distributions, we only execute this algorithm on a small network scale and the results suggest similar conclusion as in the **Level** distribution. Therefore, in other 8 distributions, we only present and discuss the greedy and MFBF algorithms.

The results are illustrated in Figure 6 ~ Figure 15. We compare the algorithms both horizontally and vertically. The horizontal comparison is to compare the performances of the algorithms, as shown in Figure 6, while the vertical comparison is to compare the same algorithm with varying  $\theta$ , as shown in Figure 7. To illustrate, we plot the results on the *Level* distribution into 6 graphs. In the other cases, we put all these 9 results into one graphs to save space.

**Horizontal comparisons of the Algorithms.** As shown in Figure 6 and 14, MFBF performs better than greedy algorithm in *level* and *minzipf-0*. Furthermore, MFBF performs especially well in *level*. This is because, in *level* distribution, the bandwidth monotonically decreases along each dissemination path of the dissemination tree. Therefore, the greedy algorithm which attempts to maximize the bandwidth usage on each edge will incur excessive remodelling on all the dissemination paths. On the contrary, MFBF can significantly decrease the total number of remodelling processes by taking bandwidth frequency into consideration. This is also true in the *minzipf-0* distribution but with less extent. Moreover, in the *level* distribution, the randomized algorithm performs very close to the MFBF algorithm, which is also effective in reducing the remodelling processes. However, as mentioned earlier, it takes a much longer time to complete in comparing the other two algorithms.

In Figure 12 and 13, MFBF performs worse than the greedy algorithm. This is because the main idea of the MFBF algorithm is to reduce the remodelling processes by altering the bandwidth usage of the ancestor nodes of the nodes with more frequent bandwidths and it will be a great waste of bandwidths, especially when most of the bandwidths are very large, which is unfortunately the case of distribution *maxzipf-0* and *maxzipf-1*.

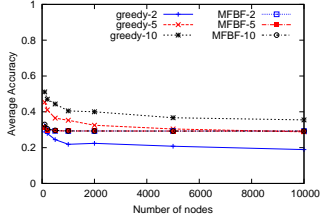


Figure 8: uniform-0

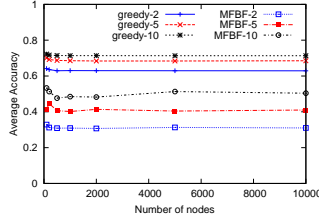


Figure 9: uniform-1

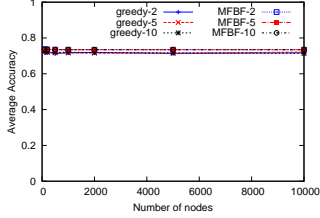


Figure 10: gaussian-0

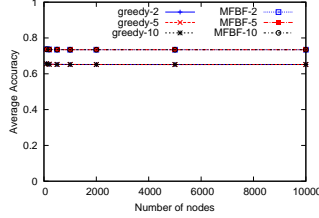


Figure 11: gaussian-1

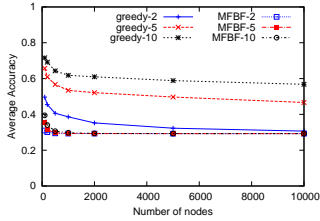


Figure 12: maxzipf-0

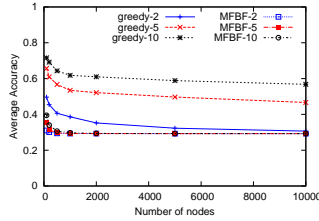


Figure 13: maxzipf-1

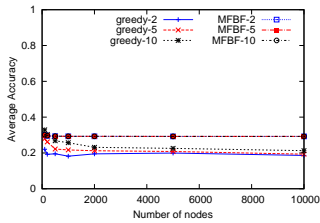


Figure 14: minzipf-0

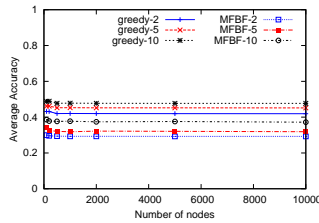


Figure 15: minzipf-1

As shown in Figure 8 and Figure 9, the greedy algorithm outperforms MFBF slightly in the uniform distributions, i.e. uniform-1 and uniform-0. In such cases, there is no significant difference in the bandwidth frequencies and hence there is limited opportunities for the MFBF algorithm to reduce the remodelling processes.

Furthermore, as shown in Figure 10 and Figure 11, the performance of greedy algorithm and MFBF are almost identical for the two Gaussian distributions. In these cases, most of the nodes have bandwidth constraints close to the median value. Therefore, the gain of accuracy by reducing the remodelling processes is almost equivalent to the lost of accuracy caused by the waste of bandwidth usages.

According to above analysis, we can conclude that MFBF performs very well when the dissemination tree is “well” organized as in our *level* and *minzipf-0* distribution. In such cases, there are more nodes with smaller bandwidth constraints and the nodes with higher bandwidth constraints are located higher in the dissemination tree. We conjecture that this is true in many real situations. If this is not true, then the greedy algorithm performs better.

**The general conclusion of the results.** The average accuracy  $\mathcal{F}(\mathcal{M})$  decreases progressively while we increase the total number of nodes in  $\mathcal{T}$  from 100 to 10000. The reason is that the remodelling is rare when the number of nodes is small, but it becomes more frequent with the increase of the number of nodes. Furthermore, when the network becomes greater, the average accuracies would become steady in all the cases.

**The sensitivity to  $\theta$ .** As shown in all the previous figures,  $\mathcal{F}(\mathcal{M})$  has a positive correlation with the maximum fan-out  $\theta$  of  $\mathcal{T}$ . The major reason of this is that the height of  $\mathcal{T}$  is determined by  $\theta$  when  $n$  is fixed and the height of  $\mathcal{T}$  determines the possible maximum remodelling processes on a dissemination path. In other words, when  $\theta$  increases, the number of remodelling within  $\mathcal{T}$  is reduced and consequently the average accuracies would increase.

**The influence of assignment schemes.** In terms of the assignment schemes, all the algorithms perform better in the ordered scheme than in the unordered scheme with the same distribution. Furthermore, the assignment schemes have a higher influence on greedy algorithm, while MFBF performed much more consistently.

## 8. CONCLUSIONS

In this paper, we have discussed about the problem of multi-scale time series data dissemination over a predefined tree-based network. We used bandwidth constraints as an abstract parameter to indicate the subscription levels of the subscribers involved in the dissemination network as well as the volumes of data that the subscribers would be able to consume. A framework of model-based dissemination is proposed. Based on this, we formulate an optimization problem to maximize the average accuracy of the data received by all the nodes in the dissemination network. One major challenges to solve this problem is the complexity to estimate the accuracy of a particular dissemination plan. To address the challenge, we have made a thorough study on the estimation of the accuracy of using PLA to represent time series data. In addition, we proposed three algorithms to optimize the dissemination plan. Meanwhile, an extensive experiments were carried out to evaluate the performances of these algorithms in various bandwidth distributions. The experimental results can be used as guidelines for choosing the optimization strategies according to the actual application environments.

There are some interesting directions to be studied in the future. First, the optimization of the structure of the dissemination tree is an interesting problem, which would have significant effects on the overall data accuracies. As we can see from our analysis and experimental results, the number of remodelling that are executed in the network is a major factor in the resulting data accuracies. On one hand, organizing the dissemination network to reduce the need of remodelling can help increase the data accuracies. On the other hand, the network organization should take into consideration of the actual physical network properties, such as transfer latency, localities etc. For example, connect one node  $N_i$  as a child node to another node  $N_j$  that has the same subscription level can avoid remodelling, but if  $N_j$  is far away from  $N_i$  and has a long network latency, then it is undesirable. Hence, a good trade-off should be found. Second, it is interesting to study using other models (e.g. DWT) to compress time series data. In general, our framework can accommodate different compression models as long as remodelling methods are provided. Furthermore, cost models and optimization algorithms should also be rethought and re-examined.

## 9. REFERENCES

- [1] <http://gnip.com/>.
- [2] <http://lhc.web.cern.ch/lhc/>.
- [3] <https://datamarket.azure.com/>.
- [4] <http://www.agpdata.com/>.
- [5] <http://www.infochimps.com/>.
- [6] <http://www.patientslikeme.com/>.
- [7] <http://www.xignite.com/>.
- [8] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. *The 4th International Conference on Foundations of Data Organization and Algorithms (FODO '93)*.
- [9] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems (TOCS)*.
- [10] F. K.-P. Chan, A. W. chee Fu, and C. Yu. Haar wavelets for efficient similarity search of time-series: With and without time warping. *IEEE Transactions on Knowledge and Data Engineering*.
- [11] K. E., Z. Q., H. B., H. Y., X. X., W. L., and R. C.A. The ucr time series classification/clustering homepage: [www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/). 2011.
- [12] A. Gandhi. Content-based image retrieval: Plant species identification. *Master thesis, Oregon State University*, September 2002.
- [13] Y. hua Chu, S. G. Rao, S. Seshan, and H. Zhang. A case for end system multicast. in *Proceedings of ACM Sigmetrics*.
- [14] K. Kanth, D. Agrawal, and A. Singh. Dimensionality reduction for similarity searching in dynamic databases. *Proceedings of the 1998 ACM SIGMOD international conference on Management of Data*.
- [15] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286, Aug 2000.
- [16] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *ICDM*, 2001.
- [17] E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: A survey and novel approach. *Data Mining in Time Series Databases*, 57:1–22, 2004.
- [18] E. Keogh and M. Pazzani. Scaling up dynamic time warping to massive dataset. *Proceeding of Third European Conference on Principles of Data Mining and Knowledge Discovery (PKDD '99)*.
- [19] E. J. Keogh and M. J. Pazzani. Relevance feedback retrieval of time series data. *International Conference on Research and Development in Information Retrieval (SIGIR)*, 3(3):183–190, Aug 1999.
- [20] A. Koski, M. Juhola, and M. Meriste. Syntactic recognition of ecg signals by attributed finite automata. *Pattern Recognition*.
- [21] P. Koutris, P. Upadhyaya, M. Balazinska, B. Howe, and D. Suciu. Query-based data pricing. *Proceedings of the 31st symposium on Principles of Database Systems*.
- [22] X. Liu, Z. Lin, and H. Wang. Novel online methods for time series segmentation. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 20(12):1616–1626, December 2008.
- [23] S. Nath, J. Liu, and F. Zhao. Sensormap for wide-area sensor webs. *IEEE Computer*.
- [24] R. T. Olszewski. Generalized feature extraction for structural pattern recognition in time-series data. *PhD thesis, Carnegie Mellon University, Pittsburgh, PA*, 2001.
- [25] T. Palpanas, M. Vlachos, and E. Keogh. Online amnesic approximation of streaming time series. *International Conference on Data Engineering (ICDE)*.
- [26] D. Rafiei and A. Mendelzon. Efficient retrieval of similar time sequences using dft.
- [27] C. A. Ratanamahatana and E. Keogh. Everything you know about dynamic time warping is wrong. *SIAM International Conference on Data Mining*, 2004.
- [28] C. A. Ratanamahatana and E. Keogh. Making time-series classification more accurate using learned constraints. *SIAM International Conference on Data Mining*, 2004.
- [29] T. M. Rath and R. Manmatha. Word image matching using dynamic time warping. *CVPR*, 2003.
- [30] G. Reeves, J. Liu, S. Nath, and F. Zhao. Managing massive time series streams with multi-scale compressed trickles. *Proceedings of 35th Conference on Very Large Data Bases*.
- [31] S. Shah, K. Ramamritham, and P. J. Shenoy. Resilient and coherence preserving dissemination of dynamic data using cooperating peers. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.
- [32] E. Skjervold, K. Lund, T. H. Bloebaum, and F. T. Johnsen. Bandwidth optimizations for standards-based publish/subscribe in disadvantaged grids. *Millitray Communications Conference (MILCOM 2012)*.
- [33] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data collection, storage, and retrieval with an underwater sensor network. *Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys 05)*.
- [34] B. C. O. Y. Zhou and K.-L. Tan. Disseminating streaming data in a dynamic environment: an adaptive and cost-based approach. *The International Journal on Very Large Data Bases (VLDB J.)*.
- [35] Y. Zhou, Z. Vagena, and J. Haustad. Dissemination of models over time-varying data. *Proceedings of the VLDB Endowment (PVLDB)*, 4, 2011.