WUJNS
Wuhan University Journal of Natural Sciences

Vol.9 No.4 2004 429-434

Article ID: 1007-1202(2004)04-0429-06

The Distributed Network Monitoring Model with Bounded Delay Constraints

□ LIU Xiang-hui[1], YIN Jian-ping[1†], LU Xi-cheng[1], CAI Zhi-ping[1], ZHAO Jian-min[2]

1. School of Computer Science, National University of Defense Technology, Changsha 410073, Hunan, China;
2. School of Computer Science, Zhejiang Normal University, Jinhua 321004, Zhejiang, China

Abstract: We address the problem of optimizing a distributed monitoring system and the goal of the optimization is to reduce the cost of deployment of the monitoring infrastructure by identifying a minimum aggregating set subject to delay constraint on the aggregating path. We show that this problem is NP-hard and propose approximation algorithm proving the approximation ratio with $\ln m+1$, where is the number of monitoring nodes. At last we extend our modal with more constraint of bounded delay variation.

Key words: network; distributed monitoring; delay constraint; NP-hard

CLC number: TP 393

Received date: 2004-03-10

Foundation item: Supported by the National Natural Science Foundation of China (60373023)

Biography: LIU Xiang-hui(1973-), male, Ph. D candidate, research direction: algorithm complexity analysis, QoS in Internet. E-mail: LiuXH@tom.com

† To whom correspondence should be addressed

0 Introduction

The explosive growth of Internet has created a massive increase in demand for bandwidth, performance, and predictable Quality of Service (QoS). Simultaneously, the need has emerged for monitoring technology that will support this growth by providing IP network managers with effective tools for monitoring network utilization and performance. Knowledge of the up-to-date performance information is critical for numerous important network management tasks, including proactive and reactive resource management and traffic engineering, as well as providing and verifying QoS guarantees for end to end user application[1-3].

Monitoring of the network-wide state is usually achieved through the use of the Simple Network Management Protocol (SNMP). SNMP has two kinds of entities: one management station and some management agents running on network nodes. The management station sends SNMP commands to the management agents to obtain information about the network. Traditionally, the management station function is performed by a centralized component responsible for aggregating all network nodes where management agents are on them. Such processing SNMP queries have some inherent weaknesses. Firstly it can adversely impact router performance and SNMP data transfers can result in significant volumes of additional network traffic. In particular, when the network monitoring process requires more data to be collected and at much higher frequencies, the overhead that the aggregating procedure imposes on the underlying network can be significant. In addition to traffic limits, the monitoring scheme has different demands in terms of bandwidth, reliability, delay and jitter. For example, one key property of aggregating procedure is its time dependency. The support of knowledge of the up-to-date performance information requires the establishment of reliable, low delay and low cost aggregating routes. So much research should be made to develop strategies that aggregates information within its delay bound, thereby allows a tradeoff between delay and cost[2,4].

To improve the scalability of SNMP, several enhancements have been made to the SNMP protocol itself by improving efficiency of basic SNMP primitives and more effort have been made to propose distributed monitoring framework[5-7]. As being pointed out that in a centralized monitoring system, although the central provides a network-wide view but has some inherent weaknesses, this scheme is not suitable for large scale network[8]. Taking into account the issues of scalability and network-wide view for large service provider networks, an ideal monitoring architecture is a hierarchical system which implied that there is a central manager but the resource intensive tasks such as polling are distributed among a set of monitoring nodes. Between the central manager and the monitoring nodes, there exists a set of aggregating nodes. The aggregating nodes are distributed and each node is responsible for a aggregating domain consisting of a subset of the network nodes. Information gathered from the individual monitoring nodes is then aggregated. The condensed information in aggregators is then sent to the central manager that provides an overall view of network behavior. Such a hierarchical architecture overcomes the weaknesses while still maintaining a network-wide view. Figure 1 depicts this hierarchical distributed monitoring model.

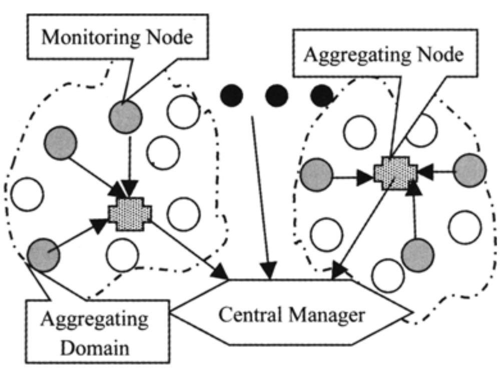

Fig. 1 Illustration for hierarchical distributed monitoring model

In particular, the most recently works addresses the problem of minimizing the number of aggregating nodes while keeping the aggregating bandwidth within predefined limits[8]. The limits of it rest with its sole consideration of bandwidth limits. In addition to bandwidth limits, the monitoring scheme has different demands in terms of bandwidth, reliability, delay and jitter. Nevertheless no research has developed a strategy that aggregates information within its delay bound, thereby allows a tradeoff between delay and cost. Thus the objective of this paper is to produce a minimal cost distributed monitoring model which guarantees bounded aggregating delay.

Our work focuses on optimizing a distributed monitoring model for a service provider network that supports QoS. The main contributions of our work are as follows: We show that the problem of minimizing the number of aggregating nodes in a given network subject to delay constraints is NP-hard. Then we propose algorithm using heuristics policy based on the maximum assignment of monitoring nodes to a pickup aggregating node. At last we extend subject of our model to bounded delay variation.

1 System Model and Problem Formulation

Before outlining the system model, we introduce some necessary definition illustrated as Table 1.

Table 1 Notation in the paper

Symbol	Semantics
$G(V,E)$	Network topology, where denotes the set of nodes, represents the set of edges between two nodes
$E(v)$	$E(v)=\{(v,u)\mid(v,u)\in E\}$ denotes the (sub)sets of edges(liks) incident on v
$E(S)$	Given $G(V,E)$ and $S\subseteq V$ $E(S)=\{(u,v)\mid(u,v)\in E,u,v\in S\}$ denotes the subset of edges that both endpoints in S
Path(u,v)	Given $u\in V\wedge v\in V$, Path(u,v) , denotes an unique path between node u and v that is aggregating route
$E($Path$(u,v))$	Represent the relative links (edges) of Path (u,v)

Firstly we define edge-delay function $D: E\rightarrow \mathrm{R}^+$ which assigns a non-negative weight to each link in the network. The value $D(e)$ associated with edge $e\in E$ is a measure (estimate) of total delay that packets experience on the link, including the queuing, transmission, and propagation components. Let the set $E(\text{Path}(u,v)) = \{e_1, e_2, \cdots, e_m\}$, we have $\text{Delay}(\text{Path}(u,v)) = \sum_{i=1}^{m} D(e_i)(e_i \in (\text{Path}(u,v)))$.

Our model for the monitoring domain is an undirected graph $G(V,E)$, where $V=\{v_1, v_2, \cdots, v_n\}$, is the set of all nodes or routers that are in the monitoring domain and. E represents the set of edges. The node set S_m ($S_m \subseteq V \wedge S_m \neq \varnothing$) represents the monitoring nodes in the monitoring domain.

The optimal aggregating node location and monitoring node assignment problem can therefore be stated as follows: Given a network $G(V,E)$, determine

① a minimum subset of nodes S_a ($S_a \subseteq V$) on which to place aggregating node such that the delay constraint on every node $v(v \in S_m)$ satisfy $\text{Delay}(\text{Path}(v,w)) \leqslant \delta$, where δ is the maximum delay that can be used for aggregating by a node denoted as w.

② a mapping λ which maps a monitoring node to its aggregating node. That is, for any node $v(v \in S_m)$, if $\lambda(v)=w$, then node v is assigned to the aggregating node w. Note in some situation, we can use additional constraints to decide whether the monitoring node v can be aggregated by itself.

Now we define some variable to describe the integer program formulation about the problem. The binary variable x_{ij} indicates whether monitoring node v_i is aggregated by node v_j, where $v_i \in S_m$ and $v_j \in V$. The binary variable b_e^{ij} indicates whether edge e belongs to the Path (v_i, v_j) between node v_i and v_j. The binary variable y_j indicates whether node v_j is a aggregating node or not. The problem minimizing the number of aggregating nodes in a given network subject to delay constraints can naturally expressed as a decision problem:

The objective is: Minimize $\sum_{j=1}^{|V|} y_j$ and the subjects are below:

$$\sum_{j=1}^{|V|} x_{ij} = 1 \ (\forall v_i \in S_m) \tag{1}$$

$$x_{ij} \leqslant y_j (\forall v_i \in S_m, \forall v_j \in V) \tag{2}$$

$$\sum_{e \in E} b_e^{ij} D(e) x_{ij} \leqslant \delta \ (\forall v_i \in S_m, \forall v_j \in V) \tag{3}$$

$$x_{ij} \in \{0,1\} \ (\forall v_i \in S_m, \forall v_j \in V) \tag{4}$$

$$y_j \in \{0,1\} \ (\forall v_j \in V) \tag{5}$$

The first constraint makes sure that each monitoring node v_i is aggregated by exactly one aggregating node. The second constraint guarantees that a node v_j must be an aggregating node if some other monitoring node v_i is assigned to (aggregated by) it. The third constraint ensures that delay during aggregating procedure not exceeds the delay constraint on the path between each monitoring node and its aggregating node.

In order to prove the problem minimizing the number of aggregating nodes in a given network subject to delay constraints is NP-hard, firstly we consider a simplified form. Suppose that the delay value which assigns to each link is 1 and the delay bound δ for every node $v(v \in S_m)$ is $1+\varepsilon$, where ε is very small positive value, while the set of monitoring nodes is equivalent to V. This is to say that for any node $v(v \in S_m)$ can and only can be aggregated by the itself or by the node $u(u \in V)$ satisfying $(v,u) \in E$.

Lemma 1 The simplified form as described as upwards is NP-Complete.

Proof The decision form of problem is: given a graph $G(V,E)$ and integer K, does G contains a node set S of size at most K, satisfying that for each vertex $v(v \in V)$ either is in or has a relative node $u(u \in S)$ with $(v,u) \in E$.

We will show this problem is NP-Complete by reducing the well known NP-Complete Problem Vertex Cover to it. The vertex cover problem is as follows: given Graph $G'(V',E')$ and integer K', does G' contains a node set S' of size at most K', satisfying that for each edge $(u',v') \in E'$ at least one of u' and v' belongs to S'.

The class of decision problem which can be checked by a non-deterministic Turing machines, so it belongs to NP class.

For the reduction, we create G as follow. For each edge $(u,v) \in E'$ add a new vertex named uv and add edges (uv,u) and (uv,v). And we have claim: There exists a vertex cover in G' of size K' if and only if there exists a node set S of size at most K' in $G(V,E)$, satisfying that for each vertex $v(v \in V)$ either is in S or has a relative node $u(u \in S)$ with $(v,u) \in E$. And it's obvious that the reduction can be implemented in polynomial time.

Because if S' is a vertex cover in G' of size K' then S' also satisfies that for each vertex $v(v \in V)$ either is in S or has a relative node $u(u \in S)$ with $(v,u) \in E$. For a vertex of G that is also in G', if the vertex has one edge incident on it, then the other end of the edge is in S' if the vertex itself is not in S'. For a new vertex uv, we know that either u or v is in S'. Thus for any node $v(v \in$

V) in graph G, $v \in S'$ or has a relative node $v(v \in V)$ with $(v,u) \in E$. To prove the converse, note that if S is a vertex set in G of size K, satisfying that for each vertex v ($v \in V$) either is in S or has a relative node $u(u \in S)$ with $(v,u) \in E$, then without loss of generality vertexes in S are also in G'. If a vertex of form uv is the S, then we can remove uv and add u (or v) to S and S still satisfies that for each vertex $v(v \in V)$ either is in S or has a relative node $u(u \in S)$ with (v,u). Now note that S is a vertex cover in G' since for each edge $(v,u) \in E'$, since for node uv we know that either u or v is in S. This completes the proof.

Form Lemma 1, using restriction method, we have following theorem.

Theorem 1 The problem minimizing the number of aggregating nodes in a given network subject to delay constraints is NP-hard.

2 Approximation Algorithm

It is well-known that the Integer Programming formulation has an exponential running time in the worst case. In this section, we propose a greedy algorithm. Our greedy algorithm consists of two steps. In the first step our algorithm calculate out the maximum number of monitoring nodes satisfying the delay constraint when they are assigned to a aggregating node, and the set of these monitoring nodes is called candidate monitoring set of relative node. In the second step we greedily repeatedly picks an additional aggregating node (based on the greedy selection criteria described below) if there are any monitoring nodes still present in the network that does not have a aggregating node assigned to it. After an aggregating node is picked, the algorithm assigns candidate monitoring set to it without violating delay constraint. The repeat will interrupt when all monitoring nodes have been assigned, and the approximate aggregating node set includes all pickup additional aggregating nodes.

The formal description of candidate monitoring set of relative aggregating node is below:

Given a node $v(v \in V)$ the candidate monitoring set of node v, denoted as $C(v)$, satisfy that: $C(v) \subseteq S_m$, for any node $u(u \in C(v))$, there is a unique path between v and u and $\text{Delay}(\text{Path}(v,u)) \leqslant \delta$, where δ is delay tolerance. For any node in graph, we can use Breadth-First method to search maximum candidate monitoring set of it.

If we know the maximum candidate monitoring set of every node in the monitoring domain, then we can use the blow procedure to get approximate aggregating nodes set.

Procedure ComputeAggregatorSet$(V, E\{C(v) \mid v \in V\})$

{

Step 1 $S \leftarrow S_m$ /* Initiate the unassigned monitoring node as whole monitoring node set */

Step 2 $U \leftarrow \varnothing$ /* Initiate the aggregating node set as emptiness */

Step 3 while($S \neq \varnothing$){

Step 4 Pick a node $v(v \in V)$ satisfying that maximizes $(C(v) \cap S)$

Step 5 $U \leftarrow U \cap \{v\}$

Step 6 $S \leftarrow S / C(v)$

} /* End of While */

Step 7 return U } /* End of Procedure */

Let $v_1, v_2, \cdots, v_m$ $(m = |S_m|)$ be the order in which monitoring nodes are covered by above algorithm. Each monitoring node v_k $(v_k \in S_m)$ is first covered by some set T_i in step 4. Suppose that the cost of picking an additional aggregating nodes is 1, and define the cost of v_k, denotes $\text{cost}(v_k)$ as $1/|T_i \cap S_m|$. Let the aggregating node set computed by the algorithm be U, and the optimal (minimum) aggregating node set is U^*. We have the following lemma:

Lemma 2 For each $i(1 \leqslant i \leqslant m)$, we have $\text{cost}(v_i) \leqslant |U^*|/(m-i+1)$.

Proof Let $c = \{T_{i1}, T_{i2}, \cdots, T_{ir}\}$ be a minimal subcollection of U^* that covers $v_i, v_{i+1}, \cdots, v_m$. Let $D = \{v_1, v_2, \cdots, v_{i-1}\}$ and $D' = \{v_i, v_{i+1}, \cdots, v_m\}$. For $v_l \in D'(i \leqslant l \leqslant m)$ let T_{i_j} be the first set that contain v_l. And we define $f(v_l) = 1/|T_{i_j} - D - T_{i_1} - T_{i_2} - \cdots - T_{i_{j-1}}|$.

Note that $\sum_{l=i}^{m} f(v_l) = |C| \leqslant |U^*|$ and so there is some $v_l \in D'$ such that $f(v_l) \leqslant |U^*|/(m-i+1)$.

Just before the greedy algorithm covers v_l, none of the sets $T_{i_1}, T_{i_2}, \cdots, T_{i_r}$ have been picked by the algorithm, because nodes in D' are still uncovered. If at this stage T_{i_j} were chosen, it would assign to node v_l with cost: $\text{cost}(v_l) = 1/|T_{i_j} - D| \leqslant f(v_l) \leqslant |U^*|/(m-i+1)$.

Since the greedy algorithm chose to pick maximum candidate monitoring set and v_i is the next node, so we have $\text{cost}(v_i) \leqslant \text{cost}(v_l) \leqslant |U^*|/(m-i+1)$.

Using Lemma 2, we have the following theorem.

Theorem 2 $|U| \leqslant H(m)|U^*|$, where $H(x) = \sum_{i=1}^{x} 1/i$.

Proof

$$\begin{aligned}|U| &= \sum_{i=1}^{m}\text{cost}(v_i)\\ &\leqslant |U^*| \sum_{i=1}^{m} 1/(m-i+1)\\ &= H(m)|U^*|.\end{aligned}$$

And more, because of

$$H(d) \leqslant \int_1^d (1/x)\mathrm{d}x + 1 = \ln d + 1,$$

we can get

$$|U| \leqslant H(m)|U^*| \leqslant (\ln m + 1)|U^*|.$$

3 Extension of Model

Our goal of above model is to minimize the set of aggregating nodes for a given network topology subject to path delay tolerance δ, representing an upper bound on the acceptable end-to-end delay along any path from source to destination node. This parameter reflects the fact that the information carried by aggregating packets becomes stale δ time units after its transmission at source. Yet in some situation the synchronization window for various receivers is import and we should care about the delay variation tolerance. So we import a new parameters: delay variation tolerance ε, where $\varepsilon \in \mathrm{R}^+$. For a node in monitoring set S_m, there are two constraints on it.

① $\sum_{e \in \text{Path}(a,v)} D(e) \leqslant \delta$, where node $a(a \in V)$ is relative aggregating node of $v(v \in S_m)$.

② $\left|\sum_{e \in \text{Path}(a,v)} D(e) - \sum_{e' \in \text{Path}(a,u)} D(e')\right| \leqslant \varepsilon$, where the node $v(v \in S_m)$ and $u(u \in S_m)$ are in the same aggregating domain dominated by node $a(a \in V)$.

We call the second constraint as delay variation constraint. It seems that we can still use the algorithm described in section 2 if we determine the candidate monitoring set which satisfies these two constraints. Unfortunately below lemma tell us that determine the maximum the candidate monitoring set satisfying delay constraint and delay variation constraint is NP-hard. The efficient algorithm to solve the distributed monitoring model under delay constraint and delay variation constraint is our ulterior focus.

Lemma 3 Given a network $G(V,E)$, a aggregating node $a \in V$, a monitoring node set $S_m \subseteq V$, an edge delay function $D: E \to \mathrm{R}^+$, a delay tolerance δ and a delay variation tolerance ε, the question: Can we determine the maximum number of monitoring nodes $S(S \subseteq S_m)$ satisfying that for any node $v \in S$ there is a routing path between a and v that satisfy the delay constraint and delay variation constraint is NP-hard.

The question can be depicted as: Does there exists a tree $T_a = (V_a, E_a)$ spanning from a with the nodes in $S_m \cap V_a$ satisfying that $\sum_{e \in \text{Path}(a,v)} D(e) \leqslant \delta (\forall v \in S_m \cap V_a)$ and $\left|\sum_{e \in \text{Path}(a,u)} D(e) - \sum_{e' \in \text{Path}(a,v)} D(e')\right| \leqslant \varepsilon (\forall u, v \in S_m \cap V_a)$, and the $|S_m \cap V_a|$ is maximum. And the decision form of problem is: Given an integer k, with network $G(V,E)$ and monitoring node set $S_m \subseteq V$, does there exists a tree $T_a = (V_a, E_a)$ spanning from $a(a \in V)$ named aggregating node, satisfying the delay and delay variation constraints and $|S_m \cap V_a| = k$.

Considering a special case of the problem: Given an assured monitoring set $M(M \subseteq S_m)$, does there exists a tree $T_a^m = (V_a^m, E_a^m)$ spanning from a with the nodes in M satisfying that

$$\sum_{e \in \text{Path}(a,v)} D(e) \leqslant \delta (\forall v \in M)$$

and

$$\left|\sum_{e \in \text{Path}(a,u)} D(e) - \sum_{e' \in \text{Path}(a,v)} D(e')\right| \leqslant \varepsilon (\forall u, v \in M).$$

It is called the Delay and Delay variation bounded multicast Tree problem, which is proved NP-Complete by Rouskas G. N. and Baldine I. in Ref. [9]. So it's obvious that the maximum assignment of monitoring nodes to a pickup aggregating node with delay and delay variation constraints is NP-Hard.

4 Summarization

In this paper, we provide the integer programming formulation for finding the minimal set of aggregating nodes in a given network subject to bounded delay constraints. We show that this problem is NP-hard and propose approximation algorithm using heuristics policy that has approximation ratio $1 + \ln m$, where m is the number of monitoring nodes in the given graph. And more, with analyzing the situation when more subjects such as bounded delay variation are imported, we find that the problem the maximum assignment of monitoring nodes to a pickup aggregating node with delay and delay variation constraints is also NP-Hard.

Our goal of above model is to minimize the set of aggregating nodes for a given network topology subject to path delay and delay variation constraints. Yet in some situation the ideal object to obtain the lower bandwidth consumption during the aggregating procedure. Our future work is to increase the scope of our problem formulation by accounting for per-link bandwidth constraints.

Centralization and distribution need not be seen as opposing solutions. Rather, they are two end points on a scale with many intermediate points. Depending on the target system and the required performance the choice of architecture may vary on this scale. Ideally, the architecture should dynamically adapt to the user's requirements. Therefore, in the future we hope to adapt our hierarchical architecture to more general cases. In addition, we would also like to consider the situation with regard to asymmetric delay.

References

[1] Asgari A, Trimintzios P, Irons M, *et al.* A Scalable Real-time Monitoring System for Supporting Traffic Engineering. *Proceedings of the IEEE Workshop on IP Operations and Management* (IPOM 2002). Dallas: IEEE Press, 2002. 202-297.

[2] Breitbart Y, Chan C Y, Garofalakis M, *et al.* Efficiently Monitoring Bandwidth and Latency in IP Networks. *Proceedings of IEEE INFOCOM*, 2001, **20**(1): 933-942.

[3] Liu X H, Yin J P, Tang L L, *et al.* Analysis of Efficient Monitoring Method for the Network Flow. *Journal of Software*, 2003, **14**(2): 300-304(Ch).

[4] Breitgand D, Raz D, Shavitt Y, *et al.* SNMP GetPrev: An Efficient Way to Access Data in Large MIB Tables. *IEEE Journal of Selected Areas in Communication*, 2002, **20**(4): 656-667.

[5] Raz D, Shavitt Y. Active Networks for Efficient Distributed Network Management. *IEEE Communications Magazine*, 2000, **38**(3): 138-143.

[6] Jamin S, Jin C, Jin Y, *et al.* On the Placement of Internet Instrumentation. *IEEE INFOCOM*, 2000, **19**(1): 295-304.

[7] Awerbuch B, Du Y, Shavitt Y. The Effect of the Network Hierarchical Structure on Performance of ATM PNNI Hierarchical Routing. *Computer Communications*, 2000, **23**(10): 980-986.

[8] Li L, Thottan M, Yao B, *et al.* Distributed Network Monitoring with Bounded Link Utilization in IP Networks. *IEEE INFOCOM*, 2003, **22**(1): 1189-1198.

[9] Rouskas G N, Baldine I. Multicast Routing with End-to-End Delay and Delay Variation Constraints. *IEEE Journal on Selected Areas in Communications*, 1997, **15**(3): 346-356.

□