Computer Networks 52 (2008) 2819–2830

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Scheduling algorithms for conducting conflict-free measurements in overlay networks

M. Fraiwan *, G. Manimaran

Real-Time Computing and Networking Laboratory, Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, United States

ARTICLE INFO

Article history:
Available online 13 June 2008

Keywords:
Measurement scheduling
Overlay networks
Measurement conflict
Network monitoring

ABSTRACT

Network monitoring is essential to the correct and efficient operation of ISP networks and the kind of applications they support, and active measurement is a key design problem in network monitoring. Unfortunately, almost all active probing algorithms ignore the measurement conflict problem: Active measurements conflict with each other – due to the nature of these measurements, the associated overhead, and the network topology – which leads to reporting incorrect results. In this paper, we consider the problem of scheduling periodic QoS measurement tasks in overlay networks. We first show that this problem is NP-hard, and then propose two conflict-aware scheduling algorithms for uniform and non-uniform tasks whose goal is to maximize the number of measurement tasks that can run concurrently, based on a well-known approximation algorithm. We incorporate topological information to devise a topology-aware scheduling algorithm that reduces the number of time slots required to produce a feasible measurement schedule. Also, we study the conflict-causing overlap among overlay paths using various real-life Internet topologies of the two major service carriers in the US. Experiments conducted using the PlanetLab testbed confirm that measurement conflict is a real issue that needs to be addressed. Simulation results show that our algorithms achieve at least 25% better schedulability over an existing algorithm. Finally, we discuss various practical considerations, and identify several interesting research problems in this context.

1. Introduction

In computer networks, the traditional measurement problem has typically focused on ways to measure a given path (or link) QoS parameters (e.g., bandwidth, delay, and jitter) to the best possible accuracy. Measurement methods are categorized as either active or passive. Passive measurements are non-intrusive, but have their limitations [1]. On the other hand, active measurements inject a non-negligible amount of probe traffic [2]. Such approaches have generally overlooked the limited capacities of the network links and the complex connectivity overlaps among different paths [3,4]. Hence, causing concurrent measurements to compete for the network resources (e.g., computation and communication resources), interfere with each other, and produce inaccurate results. In addition, Service Level Agreements (SLAs) usually limit the amount of monitoring traffic that can exist in the network at any point in time [3]. Thus, there is a need to coordinate simultaneous conflicting measurement activities in a way that the accuracy and timeliness properties are satisfied.

In this paper, we consider the measurement conflict problem in the context of overlay networks. The proliferation of the Internet has led to many measurement-sensitive network applications. Such applications include overlay routing (e.g., RON [5]), content distribution systems (e.g., Akamai [6]), end-system multicast (e.g., Narada [7]), and security monitoring (e.g., denial of service attacks

* Corresponding author. Tel.: +1 515 708 2567.
E-mail addresses: mfraiwan@iastate.edu (M. Fraiwan), gmani@iastate.edu (G. Manimaran).

1389-1286/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2008.06.006

detection). In addition, Internet service providers (ISPs) are deploying measurement overlays to monitor the health of their networks.

A key aspect of the measurement conflict problem is the overlap among measurement paths. This overlap is an obvious observation at the IP-level of the network. However, measurements and the applications they support are typically deployed using overlays with specifically designed topologies on top of the IP-level network. *The point to be noted here is that two seemingly disjoint overlay paths may actually be joined at the IP-level. This dependence may also change with time as the IP-level routing and the mapping to the overlay paths change.*

Consider the example in Fig. 1, two measurement tasks T_1 and T_2, are being conducted concurrently. Task T_1 is running from source overlay node A to destination overlay node F, while task T_2 is running from source overlay node B to destination overlay node E. The overlay (or IP) routing has resulted in these two tasks sharing the overlay (or IP) link $C{-}D$. Now, depending on the bandwidth and computational requirements of each task and with sufficient synchronization, a conflict may occur wherein both tasks will inaccurately measure the bandwidth, loss, or latency of their respective paths due to their resource consumption at the shared link. Experiments conducted using PlanetLab [8] with various measurement tools [9,10] confirm that such conflict does happen with a substantial effect on the measurement accuracy.

The contributions of this paper are as follows:

- We have formulated the measurement conflict problem as a scheduling problem of periodic QoS real-time tasks, and shown that the complexity of the problem is NP-hard [4].
- We have proposed two conflict-aware heuristic scheduling algorithms for uniform and no-uniform measurement tasks, based on graph partitioning concepts [4].
- We propose a topology-aware heuristic scheduling algorithm that increases the efficiency of producing feasible measurement schedules.
- We study the overlap among overlay paths using various real-life Internet topologies of the two major service carriers in the US.
- We show the existence and the effect of measurement conflict by constructing a globally distributed measurement overlay on top of PlanetLab and using various measurement tools.

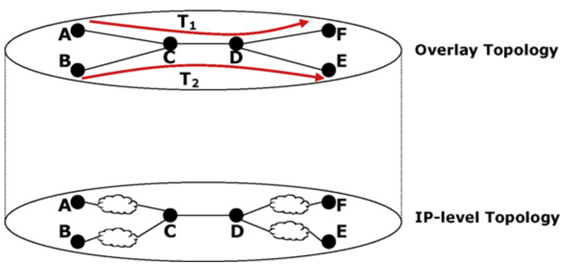

Fig. 1. A measurement example, tasks T_1 and T_2 sharing overlay link $C{-}D$, which could have also been an IP link.

The remainder of this paper proceeds as follows. Section 2 presents our network monitoring model, its assumptions, and computational complexity. Section 3 discusses the related work and motivation. Section 4 goes into the details of the conflict-aware scheduling algorithms. Section 5 presents the topology-aware scheduling approach and algorithm. We discuss some implementation issues in Section 6. Section 7 presents our performance evaluation results. We conclude in Section 8.

2. Network monitoring model and complexity

2.1. Network monitoring model

We generalize the network model to include overlay networks, as well as IP-level networks. As we have mentioned earlier, network monitoring is considered part of a network measurement infrastructure (NMI) employed by ISPs (Internet service providers) at the IP-level network. While the rise of overlay networks and their varying application requires regular monitoring to achieve efficient and correct operation of these overlays.

Given an overlay network undirected graph $G_o = (V_o, E_o)$, where V_o is the set of overlay nodes and E_o is the set of overlay edges. The corresponding IP-level graph $G_{IP} = (V_{IP}, E_{IP})$ is also available, where V_{IP} is the set of physical nodes and E_{IP} is the set of physical edges. Note that $V_o \subseteq V_{IP}$, but $E_o \not\subseteq E_{IP}$ in general. Several previous studies have assumed and justified the knowledge of the underlying IP-level topology by the overlay network operator [11].

Let $T = \{T_1, \ldots, T_n\}$ be the set of tasks to be scheduled. $T_i = (s_i, d_i, c_i, p_i, \text{tool}_i)$, where s_i is the source of the measurement task, d_i is destination of the same task, c_i is the running time of the task, p_i is the period of task T_i, and tool_i is the tool used by the task. The deadline of the task is the same as its period. We also define matrix $M = [m_{ij}]$ to be an $n \times n$ 0–1 matrix, representing the possible conflict between tasks if they are run at the same time, where $m_{ij} = 1$ if task i conflicts with task j, and 0 otherwise. The conflict matrix M captures the conflict among tasks based on the set of tasks and the computational and communicational conflict factors.

2.2. Problem definition

We start by defining three important terms related to this problem:

- Feasibility: A feasible schedule is a schedule in which all tasks meet their deadlines and no two tasks are scheduled in a conflicting manner.
- Optimality: A scheduling algorithm is said to be optimal if no other algorithm can find a feasible schedule for a task set that this algorithm has failed to find a feasible schedule for.
- Schedulability: This defines the efficiency of the scheduling algorithm in terms of the ability to find a feasible schedule.

The measurement conflict scheduling problem can be defined as follows: