Scheduling Cooperative Emergency Response (or how the Meek shall overcome the Greedy)

Carol Dottin
Department of Math.& Computer Science
John Jay College, CUNY
New York, 10016
cdottin@jjay.cuny.edu

Bilal Khan
Department of Math.& Computer Science
John Jay College, CUNY
New York, 10016
bkhan@jjay.cuny.edu

ABSTRACT

We consider the problem of scheduling emergency responders to geospatially located finite duration temporally bounded tasks. We consider two different schedulers, Greedy and Meek. schedulers: the Greedy algorithm schedules the closest available qualified emergency responder to a task, while the Meek algorithm, assigns the qualified emergency responder that minimizes the expected future cost of the Greedy algorithm. We show that to be effective emergency response scheduling must take into consideration future costs (as the Meek algorithm does), and not merely instantaneous costs (as the Greedy algorithm does).

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design–*wireless communication*

General Terms

Algorithms, Performance

Keywords

Ad hoc networks, cooperative mobility

1. INTRODUCTION

In the event of natural or man-made disasters, public emergency response services play a critical role in saving lives and reducing property losses. In such crisis situations, each emergency response team member must fuse data from their immediate vicinity with relevant information obtained from other rescuers and data warehouses, in order to maintain situational awareness to execute effective actions which would lead to mission objectives. This paper concerns the design and development of components of a system that is capable of efficiently handling the tasking requirements of emergency first responders in disaster-recovery settings.

IWCMC '09 , June 21-24, 2009, Leipzig, GERMANY

2. BACKGROUND

Timing, execution and allocation of resources become even more important during emergencies. Disaster response and recovery efforts require timely interaction and coordination of all resources in order to be effective [2]. Recent research efforts have sought to address the challenge of allocating resources in disasters, in a manner that is both efficient and effective. Some researchers have identified the data recovery scheduling problem as an optimization problem, in which the goal is to minimize the financial penalties due to downtime, data loss [4] and vulnerability to subsequent failures [1]. They have presented several methods for finding optimal or near-optimal solutions, including priority-based, randomized and genetic algorithm-guided ad hoc heuristics [1], as well as strategy for mapping a network using a collection of cooperating mobile agents [3].

3. PROBLEM FORMULATION

The system to be designed will be presented with a sequence of tasks and must manage the allocation of resources (a set of first responders) effectively towards the fulfilment of these tasks – to the extent that such an objective is attainable. In what follows, we will identify each first responder with the responder's mobile phone. This is not an unreasonable supposition, since we assume that the mobile phone is the only reliable way for the system to exchange information with the emergency response personnel. Each task has a duration and a window of time during which the task needs to be performed. Clearly, the window is always greater than or equal to the duration. Each task in general will require one or more capabilities (on the part of the responder) in order for the task to be executed.

This system is part of a larger research initiative at John Jay College of Criminal Justice (City University of New York), known as the Cooperative Networking for Emergency Response Teams (CONCERT) Project, whose objective is to develop a distributed mobile phone application that addresses the problem of context and interest-based data acquisition and delivery for teams of disaster recovery responders using effective algorithms for optimal agent scheduling. The system may choose to reject a task (e.g. because it determines that it does not have any qualified available responders to perform it). If the system rejects the task, then the submitter is free to seek alternate means to meet his objectives (i.e. using resources outside the purview of the CONCERT system). The system must however, provide quality of service (QoS) assurance in the following sense: *If the system accepts a task, then the task must indeed be*

The expected general trend that acceptance rate will increase as the number of phones increases is supported by the graph in Figure 5. A interesting point on the graph is where the number of phones available for assignment is 5. At that point all three Controllers accepted approximately 50% of the tasks presented to them. As the number of capabilities available was two and any phone had a 50% probability of having one of the 2 features and arriving tasks also had a 50% probability of having the required capability, we can conclude that of the 5 phones available, half met the feasibility criteria to execute any task that arrived. As the number of phones increased beyond this, the Meek algorithm was more conservative in its acceptance of tasks and thus had a lower acceptance rate than the Greedy algorithm. However, when resources were limited, the Meek algorithm was more effective at assigning tasks.

6.4 Task Duration

Since the duration of each task equals ϵW, to analyze the influence of task duration we considered how changes to slack (ϵ) affected the acceptance rate and the average movement cost of the three algorithms when varying ϵ in the range of $0.1 \leq \epsilon \leq 1.0$. The set of capabilities was taken as $m = |\mathbb{C}| = 2$, and the set of phones $\mathbb{P}$ was taken to be of size $n = 2$. Each phone $i \in \mathbb{P}$ has $|C_i| = 1$, $\upsilon_i = 1.0 \div 3600$. Each task T_i was constructed using the default specification.

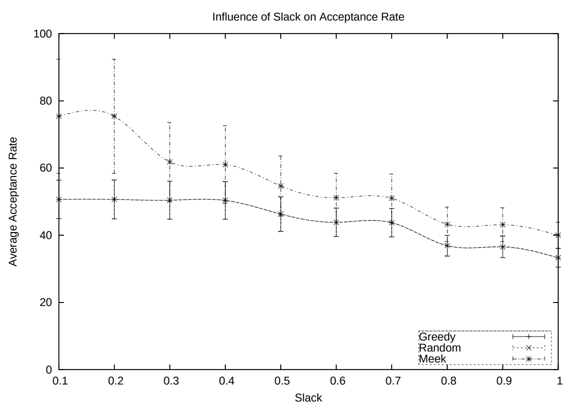

Figure 7: Acceptance Rate vs Slack

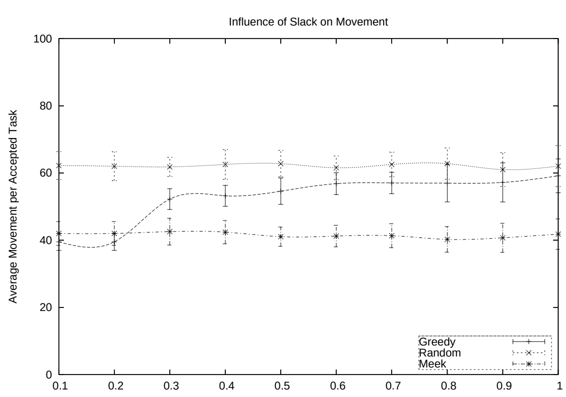

Figure 8: Movement Cost vs Slack

As ϵ increases, tasks take longer, resources are unavailable for longer periods of time, and so it is expected that the number of tasks accepted by each algorithm will decline (Figure 7). While all three algorithms show a decline in the acceptance rates, the Meek algorithm maintains a 10% higher acceptance rate compared to the Greedy algorithm. Once the duration of arriving tasks exceeds half the window of the task or greater (i.e. $\epsilon > 0.5$), the acceptance rate trends of the three algorithms coincide, though the Meek maintains a 10% higher acceptance rate compared to the greedy algorithm. The trend we expected for the average movement cost was that as the duration of tasks increased, the movement cost would also increase. However as shown in (Figure 8), the only algorithm that clearly exhibited this trend was the Greedy algorithm. The future awareness of the Meek algorithm's decisions made its performance scale favorable with respect to the ϵ parameter.

7. CONCLUSION

We considered the problem of scheduling emergency responders to geospatially located, finite duration, temporally bounded tasks, where both tasks and responders have a finite set of capabilities. We described two classes of strategies for scheduling emergency responders to tasks: the Greedy strategy schedules the closest available qualified emergency responder, while the Meek strategy assigns the qualified emergency responder that minimizes the expected future cost of the Greedy algorithm. We showed that the task acceptance rate for the Meek algorithm surpasses that of Greedy under almost all circumstances and yields lower per-task movement costs, and moreover, that the Meek algorithm's performance exhibits better scalability with respect to several key system parameters. Thus our results demonstrate that effective emergency response scheduling must take into consideration future costs (as the Meek algorithm does), and not merely instantaneous costs (as the Greedy algorithm does). The second author would like to acknowledge tha this research effort was partly supported by NSA grant H98230-07-1-0015.

8. REFERENCES

[1] Kimberly Keeton, Dirk Beyer, Ernesto Brau, Arif Merchant, Cipriano Santos, and Alex Zhang. On the road to recovery: restoring data after disasters. In *In Proceedings of the European Systems Conference*, pages 235–248. ACM Press, 2006.

[2] Andreas Meissner, Thomas Luckenbach, Thomas Risse, Thomas Kirste, and Holger Kirchner. Design challenges for an integrated disaster management communication and. In *Information SystemŤ, DIREN 2002 (co-located with IEEE INFOCOM 2002*, 2002.

[3] Nelson Minar, Kwindla Hultman Kramer, and Pattie Maes. Cooperating mobile agents for mapping networks. In *Proceedings of the First Hungarian National Conference on Agent Based Computing*, pages 34–41, 1998.

[4] Nuno Neves and W. Kent Fuchs. Adaptive recovery for mobile environments. *Communications of the ACM*, 40:68–74, 1997.