Perturbative time and frequency allocations for RFID reader networks

Vinay Deolalikar, Malena Mesarina, John Recker, Devaraj Das, Salil Pradhan
HP Laboratories Palo Alto

HPL-2005-162
September 16, 2005*

scheduling,
perturbations,
RFID

RFID reader networks often have to operate in frequency and time constrained regimes. One approach to the allocation of frequency and time to various readers in such regimes is to perturb the network slightly so as to ease the constraints. We investigate how to perform these perturbations in a manner that is profitable from time and frequency allocation point of view.

Perturbative time and frequency allocations for RFID reader networks

Vinay Deolalikar[1], Malena Mesarina[2], John Recker[2], Devaraj Das[2], and Salil Pradhan[2]

[1] Information Theory Research Group
vinayd@hpl.hp.com
[2] Mobile and Multimedia Systems Labs
{mesarina, jrecker, devaraj1, salil}@hpl.hp.com
Hewlett-Packard Labs, Palo Alto CA 94304

Abstract. RFID reader networks often have to operate in frequency and time constrained regimes. One approach to the allocation of frequency and time to various readers in such regimes is to perturb the network slightly so as to ease the constraints. We investigate how to perform these perturbations in a manner that is profitable from time and frequency allocation point of view.

1 Introduction

The recent years have witnessed an enormous amount of technical and commercial development of radio frequency identification (RFID) technology. Research attention has quickly followed, with an immediate requirement for algorithms that will enable the deployment of RFID systems in various application scenarios. One of the foremost challenges in any such algorithm is the scheduling problem. Roughly, the scheduling problem asks the question - what is the optimal time sequence for firing the various RFID readers in the network. While the details of this problem are clearly dependent on the application at handIn this paper, there is a central core of design criteria that pervades all application dependent solutions. We seek to add to this body of knowledge by examining perturbative solutions to the scheduling problem. This aspect of the scheduling problem has hitherto not been investigated in literature.

First, let us try to understand the broad constraints that shape any solution to the scheduling problem. The scheduling problem is rendered nontrivial because of two major types of interactions between the readers. The first is collision between readers. Readers lying in each other's field can collide with each other, resulting in faulty or missing reads. The second is correlation. Readings of various readers are correlated not just to the event, but to each other. The challenge then is to optimize the functioning of the network across both these interactions.

[1] Contact author.

There are two ways to avoid collision between a pair of RFID leaders with overlapping fields. The spectral approach to avoiding collision assigns different frequencies to readers that have overlapping fields. However, a prerequisite for this approach to collision avoidance is that the readers must be capable of operating on multiple frequencies. A large number of RFID readers in the market today are single frequency. Thus, a frequency allocation scheme is no longer possible for collision avoidance for these readers.

The second method of collision avoidance is time scheduling. In this approach, readers with overlapping fields are fired at different times so that they do not collide.

Both of these allocations - spectral and temporal - can be modelled as graph partitioning (or equivalently, as graph coloring) problems. This has already been discussed in the context of frequency allocation. However, in several cases of practical importance, there is not enough resource (frequency or time) for a complete allocation. This might happen, for instance, where the readers have few frequency channels, making optimal frequency allocation impossible. It might also happen in the time domain when tags are moving through the system so quickly that there is not enough time to schedule all the readers up to their saturation time.

In such resource constrained cases, we must resort to suboptimal resource allocation. This is the context into which the notion of perturbation of the network is introduced. The idea is to perturb the network slightly so that in the perturbed network, resource allocation is easier. We explore this theme in the rest of the paper.

2 The frequency and time allocation problems

2.1 Graphical formulation

An (undirected) graph G is an ordered pair $G = (V, E)$ where V is a set of vertices or nodes and E is a set of unordered pairs of distinct vertices, called edges. Denote the minimum and maximum degrees of the vertices in G by by $\delta(G)$ and $\Delta(G)$, respectively. A subset of V is called an independent set if there are no edges between vertices in V. An independent set is said to be maximal if the addition of any more vertices will result in a set that is not independent. Denote the size of the maximum independent set of G by $\beta_0(G)$. A graph is said to be complete when there is an edge between every pair of vertices in it. A complete subgraph of G is called a clique. The clique number $\omega(G)$ is the cardinality of the largest clique in G. If $\overline{G}$ is the complementary graph of G, then clearly $\omega(\overline{G}) = \beta_0(G)$.

A simple undirected graph $G = (V, E)$ is called bipartite if there exists a disjoint partition of the vertex set $V = V_1 \cup V_2$ $V_1 \cap V_2 = \emptyset$ into independent sets V_1 and V_2. In general, a graph is called k-partite when its vertex set V can be partitioned into k disjoint independent sets $V = V_1 \cup V_2 \ldots \cup V_k$. Clearly a graph that is k-partite is also l-partite for $|V| \geq l \geq k$. We will denote the minimum

k for which a graph G is k-partite by $\chi(G)$. This is the same as the chromatic number of G since the notion of partitions is equivalent to colorability. Clearly $\chi(G) \geq \omega(G)$.

It is not hard to show that $\chi(G) \leq 1 + \Delta(G)$. Brook's theorem tightens this general result.

Theorem 1. *If $\Delta(G) \geq 2$, then $\chi(G) = \Delta(G)$ unless*

1. *$\Delta(G) = 2$ and G contains a cycle of odd length.*
2. *$\Delta(G) > 2$ and G contains a clique of size $\Delta(G) + 1$.*

A cycle in a graph is a sequence of adjacent edges that begins and ends at the same vertex. The length of a cycle is the number of edges it contains. König's theorem provides an elegant characterization of bipartite graphs.

Theorem 2. *A graph is bipartite if and only if it has no cycles of odd length.*

The girth $g(G)$ of graph G is the length of its shortest cycle. The following counterintuitive result is due to Erdös [7].

Theorem 3. *For any positive integers g and k, there exists a graph with $g(G) \geq g$ and $\chi(G) \geq k$.*

In other words, high partiteness does not necessarily result only from short cycles. Of course, this result is only true for the class of non-planar graphs.

We denote by K_p the complete graph on p vertices, and by $K_{p,q}$ the complete bipartite graph on independent sets of p and q vertices. Kuratowski's Theorem gives the necessary and sufficient condition for a graph to be planar.

Theorem 4. *A finite graph G is planar if and only if it has no subgraph that is edge-contractable to K_5 or $K_{3,3}$.*

The chromatic numbers of planar graphs are the subject of the storied four color theorem.

Theorem 5. *Every planar graph can be colored with 4 colors. Equivalently, $\chi(G) = 4$ for every planar graph.*

More can be said about planar graphs having no triangles (i.e., cycles of length three).

Theorem 6. *$\chi(G) = 3$ for every planar graph without four triangles.*

The following theorem is also due to Erdös [6].

Theorem 7. *Consider the function f from graphs to integers such that $f(G)$ is the maximum number of edges among all the bipartite subgraphs of G. Then*

$$f(G) > \frac{|E|}{2}.$$

Notice that the MAX-CUT problem for a graph G seeks precisely to evaluate $f(G)$. MAX-CUT in general is known to be NP-Hard.

The following theorem [4] provides a result on a related function.

Theorem 8. *Consider the function h from integers to integers such that $h(e)$ is the largest integer having the property that every graph with e edges has a bipartite subgraph with at least $h(e)$ edges. Then*

$$h(e) \geq \frac{e}{2} + \frac{-1 + \sqrt{8e+1}}{8}.$$

Now that we have stated the graph theoretic results that we need, we proceed to formulate a graphical model for an RFID network. Given a collection of RFID readers laid out in some topology, we associate to it a $G = (V, E)$ as follows. The vertex set V is in bijection with the set of RFID readers and there is an edge $e \in E$ joining v_1 and v_2 if the RFID readers corresponding to v_1 and v_2 have overlapping fields. This graph will be called the collision graph for the RFID network. It will be the main subject of analysis for the rest of the paper.

2.2 Time scheduling and Frequency allocation - similarities and differences

Let us first revisit the frequency allocation problem for networks of RFID readers. The goal is to allocate frequencies to various readers such that if two readers lie in each other's interference region, they are given different frequencies. Clearly this problem reduces to coloring a graph where the vertices correspond to readers, with an edge between vertices when the readers lie in each other's interference region. The graph coloring problem is also at the root of several other allocation problems in wireless communication.

The time scheduling problem is subtly different. Here, the goal is to find the optimal sequence for firing the various RFID readers in the network. The constraints on this solution have two origins. The first is reader interference or collision. This precludes the firing of certain sets of readers simultaneously. At this point, this problem looks like the frequency allocation problem, except that the allocation is done along the time axis. Interfering readers are allotted non-overlapping periods of time so as to avoid collision between them. Thus it may appear that this problem also reduces to the graph coloring problem, with a coloring corresponding to a time scheduling. However, this is not the complete picture, since time scheduling is also affected by the speed at which tags are passed through the network, whereas the frequency allocation problem is not. This brings us to the notion of perfect and imperfect scheduling, which we define next.

2.3 Perfect and imperfect scheduling

Our model for each RFID reader assumes a fixed time of saturation, which is the time it takes to read its maximum capacity of tags from the tags that lie in its field. Up till this saturation time, a reader will read more tags given more time. After the saturation time, it will not read any more tags even if it is allowed to continue its read. The saturation times may vary with different readers.

Definition 1. *A scheduling scheme will be called perfect if every RFID reader is given enough time to saturate, each time that it is fired. A scheme will be called imperfect otherwise.*

One obvious desideratum for operation of reader networks is the ability to fire the RFID readers as frequently as possible without suffering collisions. The primary constraint on the simultaneous firing of a set of RFID readers is that the set of readers that can be fired simultaneously must be an independent set.

Furthermore let us assume that all the readers that can be fired simultaneously are fired, followed by the next such set, and so on, till every reader is fired at least once. Then this cycle is repeated. If this scheme is followed, then clearly we can do no better than fire the full set of vertices in the maximal partition of the graph, and then cycle through these sets. Under these assumptions, the period of this cycle will be least if the graph is bipartite. If the graph is bipartite, we fire all the readers in one partition first, followed by all the readers in the complement, and then repeat this cycle.

If the window of time for which a reader reads a tag is large enough, we can design a perfect scheduling scheme. In that case, the scheduling problem becomes isomorphic to the frequency allocation problem, and both of them reduce to the graph coloring problem.

However, if there is a constraint on time that precludes perfect scheduling, then we are left with the following tradeoff. We can switch off one of the groups early or we can let it saturate and possibly not give the other group time to read to its saturation limit. This leads to a possible imperfect scheduling solution. In this situation, one approach is to prune the network and then schedule the pruned network. This leads directly to the notion of perturbation of networks, which is the topic of the next section.

3 The Perturbative frequency and time allocation problem

The picture depicted so far is also incomplete on account of the second interaction between readers in a network, namely, correlation. In other words, physical interference is not the only way two readers are related to one another. Non interfering readers are still related to one another by means of their correlations. At this point we should make more precise what we mean by correlation between readers. Readers read subsets of tags from a box of tags that is in their field. Different readers, in general, read different subsets. The notion of correlation between two readers as a measure of the overlap between the subsets that these two readers read.

We next provide intuition as to why this affects the frequency allocation and time scheduling problems. We have already observed that one goal of the allocation scheme is to prevent colliding readers from reading simultaneously. However, that is not the only design constraint. Two non-colliding readers with a high correlation would be scheduled differently than two non-colliding readers with no correlation. This is because in a finite window of time, we wish to maximize the number of *distinct* reads made by the network. If two readers are highly correlated, it does not make sense in a resource constrained environment to give both of them separate frequencies in the spectral allocation problem, or both of them enough time to saturate in the time scheduling problem since our return on resource investment is scaled down by their cross correlation. Of course, there is a complex interplay here: we have to measure the cost of allowing both to saturate in the particular network topology. This consideration is often shaped significantly by the network topology at hand, and making general statements is difficult. For an evaluation of this tradeoff for certain particular network topologies, see [3].

3.1 The notion of perturbation of a RFID network

When there are not enough resources (frequency, time) to perform a perfect scheduling, one approach that presents itself is to work with a slightly altered network where this allocation might be easier. This leads directly to the notion of perturbation of a network. A network is perturbed by making alterations to it in a clever manner using some prior knowledge of what properties the altered network should possess so that the allocation problems in it are easier than in the original network.

Perturbation of a network could be desirable in other contexts as well. In some applications, it may be desirable to run only a subnetwork of the overall network. Here too, the question to be answered would be what subnetwork would be optimal from a resource allocation point of view. At HP Labs, we encountered these situations during practical deployments, but found that they had not been studied in literature. That provided the impetus for the present research.

We now formalize the notion of perturbation.

Definition 2. *Given a RFID network and its associated collision graph $G = (V, E)$, a perturbation is a sequence of steps each of which is of one of the two types given below.*

1. *Two readers whose fields overlap are moved apart so that this overlap vanishes.*
2. *A reader is switched off.*

We seek to reduce the problem of perturbation to a study of the subgraphs of the original network graph. We make the following simple observation. The first type of perturbation corresponds to the removal of the edge between these two reader vertices in G, while the second corresponds to the removal of the reader vertex and all its incident edges.

Now the perturbative resource allocation problem can be informally stated as follows. How to attain a subnetwork of the original network using the two steps above so that the resulting network is less constrained from a resource allocation point of view. It is a systematic study of this question that is the theme of this paper.

We proceed by studying separately the two kinds of perturbations, and how we can perform them in an intelligent fashion to ease the constraints on resource allocation in the network.

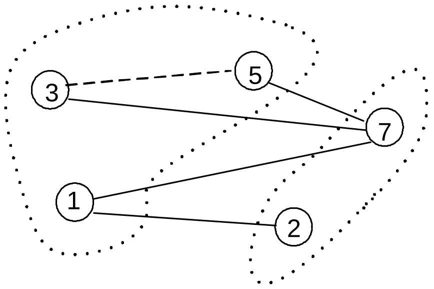

Fig. 1. Perturbation to induce bipartiteness. The original graph has a 3-cycle that prevents it from being bipartite. We displace readers to remove the 3-cycle so that the result is a bipartite graph

4 Perturbation by displacing readers

If we are not willing to perform a scheduling algorithm with a longer period, then one approach would be to eliminate edges in the collision graph till we arrive at a bipartite collision graph. This leads to the question of how many edges must be removed from the original collision graph to arrive at a bipartite graph. This is the edge deletion bipartization problem.

EDGE DELETION BIPARTIZATION: *Given a weighted graph $G = (V, E)$, what is the edge set of minimum weight whose deletion results in a bipartite graph?*

This problem is NP-complete for general graphs. For planar graphs, it is polynomial time [10].

A general result on the number of edges that will need to be deleted from a triangle free graph to make it bipartite is given by the following theorem due to Erdös et al [8].

Theorem 9. *A triangle free graph of n vertices and e edges can be made bipartite by removing at most*

$$Min\{\frac{e}{2} - \frac{(2e(2e^2 - n^3)}{n^2(n^2 - 2e)}), e - \frac{4e^2}{n^2}\}$$

edges.

Note that this theorem only assures us of the existence of a certain number of such edges, but does not tell us how to find them. One preliminary approach at finding such edges is provided by Theorem 2.

We could try to find the cycles of odd length. In most practical situations, the graphs would not be very densely connected, and it would be relatively easy to find cycles of length $3, 5$, and so on for smaller cycle length. We could then delete edges from these cycles in a judicious manner - first the common edges could be deleted, for instance.

In Figure 2 we illustrate the idea of edge deletion in order to make the collision graph bipartite.

Remark 1. The basic idea here is that we could reduce the partiteness of a graph by deleting edges, but we need not go all the way to a bipartite graph. We would, in general, need to delete fewer edges the lower the reduction in the original graph's partiteness. In other words, making a graph tripartite involves removing lesser (or equal) number of edges than making it bipartite. In general, making a graph k-partite involves reducing lesser number of edges for increasing values of k. This tradeoff is warranted in case, say, a specific application needs the presence of 3 RFID readers near each other, all lying in each other's collision range, and thus forming a cycle of length 3 which cannot be avoided. This sort of tradeoff is clearly application dependent.

Remark 2. In a low correlation regime, a high partite system might be appropriate as one would want input from a large number of readers. Thus in a low correlation regime, we would be biased towards a high partite system when we take the tradeoff decision of deleting edges to move towards a bipartite system. In other words, we are likely to stop this procedure earlier, and halt before we reach a bipartite graph.

In a high correlation regime, a low partite system might be more acceptable as one would want a few high correlation readers to fire as often as possible. The reasoning is that other readers likely have high cross-correlation with these few that are fired frequently, and therefore their outputs do not provide us with much additional information.

4.1 A heuristic algorithm for edge deletion

Our proposed algorithm for edge deletion is based on the following heuristic. Edges that correspond to pairs of readers with high cross-correlation with each other can be removed with the least loss of information. There are two ways of arguing for this heuristic. One is that a pair readers with high cross-correlation is effectively performing as a single reader, and displacing these two readers further apart is likely to render it into a genuine two-reader configuration. The second argument is that if any performance degradation does occur with the displaced reader, this does not affect the overall performance since the other reader was highly correlated with the original position of the first reader. Thus we retain access to a highly correlated version of the original information.

Thus our edge deletion algorithm is as follows.

Algorithm 1 Correlation based edge deletion

```
Weight graph $G = (V, E)$ by $w(e_{ij}) = r_{ij}$
while $G$ is not bipartite do
    Run algorithm vertex coloring
    From Defect of algorithm vertex coloring, identify edge of highest weight
    From the two RFID readers corresponding to this edge, displace the one with
    lower correlation with event
end while
$G$ is bipartite and can be perfectly scheduled
```

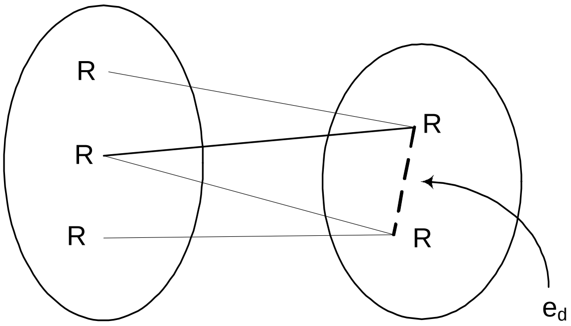

Fig. 2. Edge Deletion: If either of the readers that form the endpoints of the dotted edge are displaced sufficiently, the resulting network can be perfectly scheduled.

5 Perturbation by switching off readers

The other approach would be to remove vertices in the collision graph till we arrive at a bipartite collision graph. This leads to the question of how many nodes must be removed from the original collision graph to arrive at a bipartite graph. This is the node deletion bipartization problem.

NODE DELETION BIPARTIZATION: *Given a graph $G = (V, E)$, what is the smallest set of nodes whose removal results in a bipartite graph?*

In Figure 3 we illustrate the idea of switching off readers deletion in order to make the collision graph bipartite.

This problem is NP-hard for general graphs as well as for planar graphs. Heuristic algorithms exist that run in $O(n^2)$ [9].

An upper bound for the number of nodes that would have to be deleted is provided by the following theorem of Erdös [6] which tells us that by switching off at most half of the readers, we can arrive at a bipartite network, which can then be scheduled optimally.

Theorem 10. *Every graph has a bipartite subgraph on half of its vertices.*

Note that it is not known which readers, when switched off, lead to a bipartite network. However, for small networks, even a brute force search would be feasible.

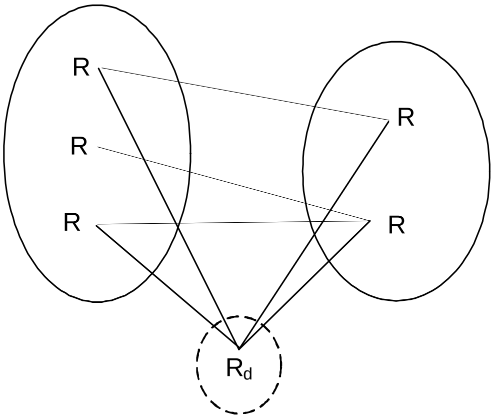

Fig. 3. Node Deletion. If reader R_d is switched off, the resulting network can be perfectly scheduled.

Situations where switching off readers might be a desirable approach include cases where there is high correlation between readers. If there exists a subset of readers which has high correlation with its complement, and which when switched off results in a bipartite network, then this would be a natural candidate to apply this approach.

5.1 A heuristic algorithm for node deletion

We begin with a result by Shearer [14] on triangle free graphs.

Theorem 11. *Consider the function f from graphs to integers such that $f(G)$ is the maximum number of edges among all the bipartite subgraphs of G. If G is a triangle free graph with e edges on n vertices with degrees $\{d_i\}_{i=1,\ldots,n}$, then*

$$f(G) \geq \frac{e}{2} + \frac{1}{8\sqrt{2}} \sum_{i=1}^{n} \sqrt{d_i}.$$

We now wish to make a heuristic argument that lies at the center of our proposed algorithm. Firstly, we observe that correlations between two RFID readers often arise from overlaps in their fields. Now consider a RFID reader that overlaps with a high number of other readers. From the preceding observation it follows that it is likely that a large part of the information supplied by this reader can be obtained from the readings of these other overlapping readers. Everything else being the same, the more the overlaps of a given RFID reader with other readers, the more information about its readings can be obtained from the readings of these overlapping readers. This leads to the intuition that removing readers with a higher number of overlaps leads to less loss of information than removing readers with fewer overlaps. At the same time, the formula for chromatic number in Theorem 11 also suggests that removing the nodes with highest degrees will result in the greatest drop in chromatic number (although the formula only gives a lower bound on chromatic number).

With this heuristic, and the formula for lower bounding the chromatic number, we propose the following algorithm.

Algorithm 2 Correlation based node deletion

```
while $G$ is not bipartite do
    Run algorithm vertex coloring
    From Defect of algorithm vertex coloring, identify node of highest degree
    Switch off the RFID reader corresponding to this node
end while
$G$ is bipartite and can be perfectly scheduled
```

Remark 3. While in this paper we have looked at perturbing the graph so as to make it bipartite, we observe that it may be profitable to lower the chromatic number, but not necessarily all the way to bipartiteness. In general, it would take a smaller perturbation to produce a smaller drop in chromatic number.

6 Conclusion

We have proposed and tackled a hitherto uninvestigated aspect of the operation of RFID (or general active sensor) networks, namely, intelligent perturbation of such networks based on knowledge of their subnetworks. We feel this is an area that needs more research focus, especially as RFID networks become more ubiquitous and support multiple applications in parallel. It is in those scenarios that sharing subnetworks across tasks will assume even more importance. Allocating resources to such subnetworks in an intelligent fashion requires an understanding of the perturbation problem. We have certainly found the insights presented in this paper to be useful in our RFID network test beds at HP Labs.

Acknowledgments

We thank Mehrban Jam and Christophe Gouguenheim for useful discussions.

References

1. V. Deolalikar, M. Mesarina, J. Recker, D. Das, and S. Pradhan Applications of graph partitioning to scheduling and clustering in active sensor networks, 2005
2. V.Deolalikar, M. Mesarina, J. Recker, and S. Pradhan Hybrid approaches to scheduling and clustering in active sensor networks, 2005

3. V. Deolalikar, M. Mesarina, J. Recker, and S. Pradhan Optimal switching strategies for active sensor networks, 2005
4. C. Edwards, “Some extremal properties of bipartite subgraphs,” Canad. J. Math. 3 (1973), pp. 475-485.
5. D. Engels, The Reader Collision Problem, Technical Report, available at http://www.autoidcenter.org/research/MIT-AUTOID-WH-007.pdf
6. P. Erdös, “On some extremal problems in graph theory,” Israel J. Math. 3 (1965), pp. 113-116.
7. P. Erdos, “Graph Theory and Probability - II.” Canad. J. Math. 13 (1961), pp. 346-352.
8. P. Erdös, R. Faudree, J. Pach, and J. Spencer, How to make a graph bipartite, J. Comb. Theory (B) 45 (1988), pp. 86-98.
9. M. Goemans and D. Williamson, “Primal-Dual approximation technique for constrained forest problems,” SIAM J. Computing 24 (1995), pp. 296-317.
10. F. Hadlock, “Finding a maximum cut of a planar graph in polynomial time,” SIAM J. Computing 4(3) 1975, pp. 221-225.
11. F. Harary, Graph Theory, Addison-Wesley, 1969.
12. I. Katzela and M. Naghshineh. Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey. IEEE Personal Communications, June 1996, pp. 10-31.
13. Ewa Malesinska. Graph-Theoretical Models for Frequency Assignment Problems. Ph.D. thesis, Technischen Universitt Berlin, 1997.
14. J. Shearer, “A note on bipartite subgraphs of triangle-free graphs,” Random Structures and Algorithms 3 (1992), pp. 223-226.
15. F. Zhao, J. Shin, and J. Reich, ”Information-Driven Dynamic Sensor Collaboration”, IEEE Signal Processing Magazine, March 2002.