The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)

An optimal architecture for a multi-standard reconfigurable radio: A network theory re-formulation

Virgilio RODRIGUEZ, Christophe MOY, Jacques PALICOT
SCEE Laboratory
IETR, Supélec
Cesson-Sevigné, France
e-mail: vr <at> ieee.org, {christophe.moy, jacques.palicot}@supelec.fr

ABSTRACT

We provide a procedure for identifying an architecture for a multistandard reconfigurable radio that is optimal in view of cost and performance (latency) considerations. We examine the trade-off between installing complex self-contained components providing high performance at a high cost (as well as size and weight), versus invoking simpler, reusable low level modules, which reduces cost but increases system latency. In the present work, we show that the the problem of finding an optimal design for a multi-standard reconfigurable radio can be reformulated as a "network design problem". This reformulation provides a wealth of results, algorithms, and experience already available in the scientific literature. We explain the reformulation, give a simple but realistic design example, and discuss some algorithms available in the network design literature.

I. INTRODUCTION

We view the design of a multi-standard reconfigurable radio as choosing the optimal point between two extremes. At one extreme is the "Velcro" solution: to support several communication standards through a few self-contained complex communication components, each exclusively dedicated to a given standard. At the other extreme, we can attempt to support all desired standards through very simple, primitive operators (adders, multipliers, MAC, etc) by invoking them repeatedly in order to perform the various communication tasks necessitated by the standards. The Velcro solution will generally provide the best performance, but at the highest manufacturing cost (and probably greatest size and weight). Conversely, by going to the other extreme, we can minimise the (monetary) cost (and possibly the size and weight) of the radio, but at a performance level that may be unacceptable for practical applications. Thus, we need to find the right level of complexity at which to use self-contained components. This is the level that gives the best trade-off between performance and cost.

Recently, we have introduced a mathematical model to identify the optimal architecture for a multi-standards reconfigurable radio [1, 2]. We model the radio as a graph of progressively simpler functional modules. Each of these modules can either be implemented via a self-contained component, or can depend on the execution of simpler (lower level) modules. A self-contained component is an optimised hardware/software combination built to perform a specific task in the most efficient way. One such module could be an equaliser, for example, or a fast Fourier transform (FFT) operator. Simpler, lower-level components are generally less expensive to build, and can be reused by several upper-level modules inside and across standards. The basic trade-off we examine is that of the (monetary) cost of a multi-standard reconfigurable radio versus its performance. The use of lower-level modules reduces the manufacturing cost, and quite possibly the size and the weight of the radio. But when a task is performed by invoking lower level components, the execution time of the task increases. As a consequence, the latency of the system may exceed practical limitations.

With our formulation, we can, in principle, find an architecture for a reconfigurable radio which is optimal in view of both economic cost and performance (computational time)[1, 2]. In [1] we discuss two possible computational approaches: finding (i) the *exact* optimum through exhaustive search ("brute force"), or (ii) an approximate solution through simulated annealing. The "brute force" approach is computationally impractical when there is a very large number of design alternatives. The second approach utilises a "generic" algorithm not specifically oriented to the problem at hand, and cannot guarantee a solution near the "true" optimum. A third alternative is to seek a solution through an algorithm which fully utilises the special structure of the graph to efficiently search the solution space. In the present work we recast our problem as a single-source"network design problem". The network design problem is well-known, and counts with a very extensive literature, which offers algorithms, computational complexity results, and many practical applications in a variety of contexts[3]. Below, we show how to perform the problem reformulation, and give a specific example grounded on a practical design. We do not have space to fully apply a network design algorithm. But we do discuss several specific algorithms which seem particularly promising [4, 5, 6].

At the foundation of this study is the "common operators approach" to the design of a reconfigurable radio. Its main principle is the identification and (re)use of common operators that can each match several processing contexts by a simple parameter adjustment [7]. This approach can lead to an improved design of a multi-standard reconfigurable radio, both in terms of manufacturing cost, and of the speed of reconfiguration during operation. This approach and its advantages are discussed more extensively in [7]. The scientific literature provides other interesting approaches to the design of reconfigurable radios, which we review in [1, 2].

The rest of the paper proceeds as follows. First, we present

1-4244-0330-8/06/$20.00

our graph-theoretic model, and give an example of a graph corresponding to a "realistic" design problem. Next, we discuss the graph/network reformulation and apply it to the previously mentioned example. Subsequently, we identify and briefly discuss some promising algorithms already available in the network design literature. Then, we discuss some key issues involved in the identification of an optimal architecture for a multi-standard reconfigurable radio. We end with a discussion of our approach, and future research directions.

II. Mathematical model

A. Graph-theoretic representation

As illustrated by figure 1(a), we can represent a reconfigurable radio as a hypergraph of progressively simpler functional modules. Each node represents a functional module that can either be implemented via a dedicated hardware/software component (an ASIC, or a software intellectual-property core, for example), or can be achieved by invoking lower-level modules. The hyperarcs leaving a node (parent) specify the simpler modules (descendants) that could provide the required functionality through multiple calls. The roots of this graph, at the highest level, may represent communication standards that the reconfigurable radio needs to support.

An OR arc (direct arrow leaving from the node itself) means that only one of the descendant nodes is necessary to implement the functionality of the parent node. An AND arc (leaving from another arc, which itself leaves a node) means that *all* descendant nodes are needed to implement the functionality of the parent node. For instance, the functionality of node $S3$ (the full communication standard) can be provided either (i) through a dedicated self-contained component implementing the complete standard, $S3$, or (ii) through *both* modules $A4$ and $A5$. Note that in some case, a parent node may have both AND and OR dependencies with its descendants. For example, to implement $S2$, one has, besides the obvious choice of a dedicated component, 2 additional choices: (i) providing the functionality of node $A4$, which is sufficient, *or* (ii) providing *both* $A2$ and $A3$.

Descendant nodes may not all be at the same level. For example, both $A1$ and $B1$ are direct descendants of $S1$, but are at different levels.

The fact that the graph is, by construction, acyclic simplifies the analysis.

B. A realistic hypergraph

Figure 1(b) shows a sub-graph corresponding to the decomposition of several processing elements (equalisation, multi-channel, OFDM) that could be part of a multi-standard reconfigurable radio. The sub-graph is *not* intended to show *all* the possible alternative implementations for each of the considered processing elements. Root nodes (standards) are *not* shown.

The equalisation block compensates for the multi-path impairment typical of wireless channels. It can be either implemented through a finite-impulse response filter (FIR) or through the FFT operator. The implementation of the equalisation in the frequency domain is particularly attractive for channels that exhibit long impulse responses, which leads to FIR filters with a very high number of taps [8]. Notice that the inverse FFT (IFFT) is computationally equivalent to the FFT, which we capture by attaching a multiplicative factor of 2 to the arc pointing from the equaliser to the FFT.

Multi-channel refers to the channelisation function of a cellular base station. This can be accomplished via the "classical" channel per channel procedure, or by proceeding in parallel, through a filter-bank channeliser (which can be implemented via FFT). Other lower-level modules correspond to well-known signal processing constructs.

The graph shows that at least the FFT operator is needed to implement OFDM modulation. The graph also shows that both equalisation and OFDM could employ the same FFT operator, although it is optional for FIR-based equalisation. A reconfigurable FFT component could provide the functionality of FFT operators of different orders.

The numerical values near the bottom right of a block represent cost / time associated with the component that could provide that functionality. The units of measurements are immaterial for our purposes. We only need relative figures, to be able to compare one design alternative to another. But the time figures must be consistent with those in which the deadlines are expressed. At the top left there is a numerical identifier for the corresponding module.

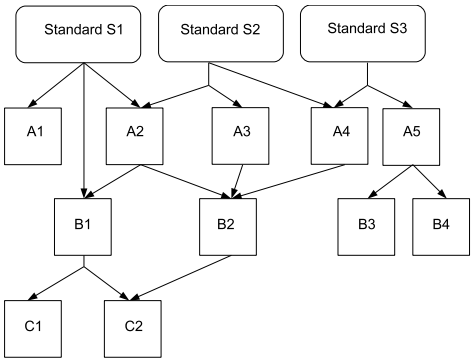
The arcs are tagged with a number of calls (NoC) figure. When a node is needed several times by a higher level module, it is called several times, and not physically replicated. Accordingly, the multiple calls affect the latency of the system, but not its monetary cost.

In our numerical exercises, we have assumed that the channeliser handles 25 channels, that the FFT is of order 64, and that to implement this FFT through a butterfly, 192 calls are needed.

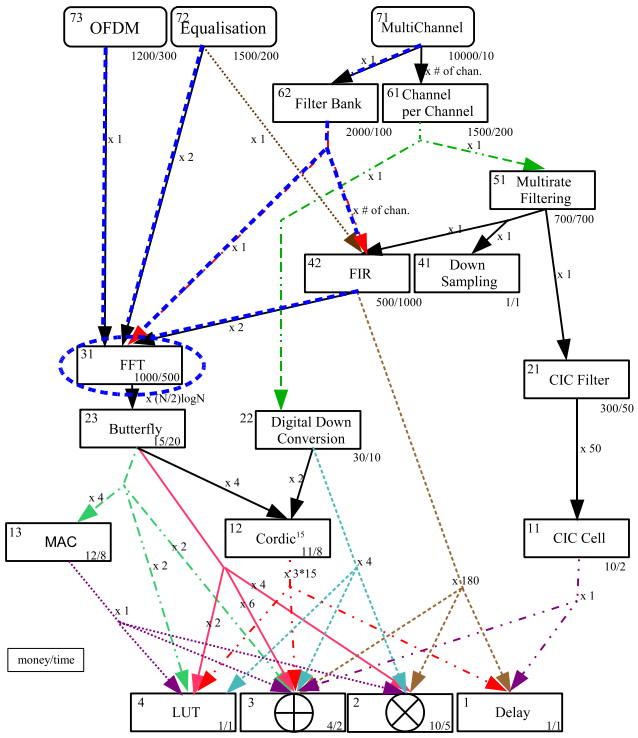
III. The network-design reformulation

A. "Network design" problem

Figure 2(a) provides a simple illustration of the single-source network design problem. Suppose there is an initial point called the origin, o, and three "terminals" (destinations), t_1, t_2 and t_3. We wish to connect o to each t_i in a way that satisfies certain optimality criteria. There may also be some intermediate nodes, which themselves may be (partially) interconnected. There are many conceivable solutions. The most obvious one is simply to build three direct "roads", one from o to t_1, a different one from o to t_2 and a third one from o to t_3. This solution is certainly plausible, and probably provides the shortest "travel times", but it is unlikely to be ideal. By building dedicated roads we are possibly missing the savings resulting from having "common segments" which may be used by all traffic regardless of the destination. For example, we could build a road from o to some intermediate point, A, plus dedicated segments from A to each t_i. Because the segment $o \rightarrow A$ is used by all traffic, certain savings may result (although this may lengthen the travel time to all traffic).

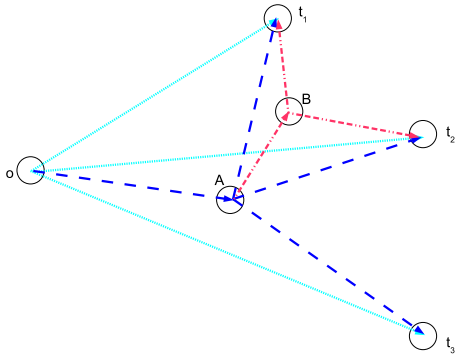


(a) A plausible hypergraph corresponding to some conceivable tri-standard reconfigurable radio

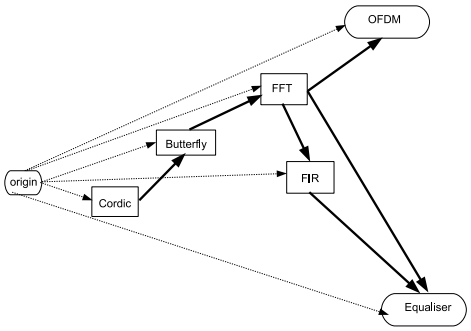


(b) Hypergraph of a realistic "sub-design" with some parameters, and a plausible solution (indicated by thick broken blue line).

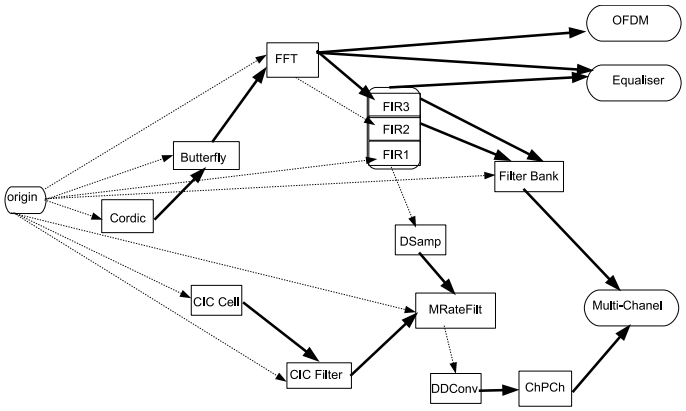
Figure 1: Design choices as hypergraphs



(a) A simple network-design problem



(b) The "network design" version of a section of the design graph. Solid arrows indicate "free" (pre-built) "algorithmic roads" that take time to travel, but require no monetary expenditure.



(c) The network design view in greater detail. Certain connections not shown.

Figure 2: The network design problem and the design of multistandard reconfigurable radios

By the same reasoning, building dedicated segments from A to each t_i may *not* be optimal. Perhaps we can build a dedicated road $A \rightarrow t_3$, but also a segment from A to some intermediate point B, followed by segments $B \rightarrow t_1$ and $B \rightarrow t_2$. In the latter proposal, all traffic would travel over the $o \rightarrow A$ segment, the t_3 traffic would go directly over $A \rightarrow t_3$ while both the traffic to t_1 and to t_2 would continue over $A \rightarrow B$, and from B over $B \rightarrow t_1$ and $B \rightarrow t_2$ respectively. There are many other possibilities.

B. *From a hypergraph to a network-design problem*

Our problem can be recast as a network design problem by proceeding as follows. The communication standards correspond to terminals we wish to reach. Building a direct road from the origin to each terminal corresponds to the "Velcro" solution: choosing a self-contained dedicated complex component to support each standard. This will provide the "fastest routes" to be sure, but probably not the ideal solution. Installing a self-contained component of "medium" complexity is the equivalent of building a road from the origin to the intermediate point A in the example above. For instance, [9] shows that many important tasks of a communication receiver can be implemented through the fast Fourier transform (FFT). In turn, the FFT functionality can be implemented with a butterfly operator. Thus, we can "build a road" from the origin to the FFT (that is, install a self-contained FFT component) to be shared by several tasks.

This will likely reduce the overall cost of the design, although it will generally "increase the travel time". But we could also "reach" the FFT node (that is, provide the functionality of the FFT operator), by first "building a road" from the origin to the intermediate junction "butterfly" and from there utilise an "existing road" to the FFT (that is, install a self-contained butterfly component, and invoke it repeatedly to provide the FFT functionality). Further explanation of the reformulation is given below through examples.

C. Illustration of the graph-to-network conversion

Figure 2(b) shows the network design problem corresponding to a section of the realistic hypergraph of design choices given by figure 1(b). The parts of the graph that concern lowest level components, and the channeliser have been ignored for pedagogical reasons.

Recall that now the objective is to build a "road network" that provides a path from the origin to each "terminal". The solid arrows mark "free" (pre-built) roads that take time to travel, but require no monetary expenditure. These represent known algorithms which can be used to provide a higher-level functionality. For example, from the FFT node there is a pre-built (free) "road" to OFDM, another to Equalisation, and yet another to the FIR node, because there are known algorithms that allow the implementation of equalisation, OFDM, and FIR through the FFT operator[9]. The dashed arrows indicate "possible roads" that definitely cost money to build (but possibly negligible time to travel). "Building a road" means installing a self-contained component to provide the functionality pointed by the arrow. For example, there is a dashed arrow from the origin to FFT (meaning we can provide the FFT functionality by installing a self-contained FFT component). There is also a "possible road" from the origin to Butterfly. If we build this "road" (i.e., if we install a self-contained butterfly component) we can "reach" the FFT node via a free (algorithmic) path (shown by a solid arrow from butterfly to FFT), because one can provide the FFT functionality by repeatedly invoking a butterfly operator.

Figure 2(c) shows in greater detail the "network design problem" corresponding to figure 1(b). The fact that the digital down conversion module can be done through CORDIC, and the lowest level modules are not considered, for clarity. And some obvious dashed arrows from the origin (for example, from *o* to Equaliser, meaning installing an equaliser component) are omitted to reduce clutter. The main purpose of figure 2(c) is to show that representing the AND dependencies is challenging in some cases, and may require some ingenuity. The FIR "triple node" illustrate some of the issues.

A "triple" FIR node is used in figure 2(c) to show that there are three different "routes" to get the FIR functionality, and the route has an impact on what can be done from there. The "sub-node" FIR_3 concerns the case where the FFT is used to implement the FIR. The "sub-node" FIR_2 refers to the case when an FIR component is installed, even though the FFT functionality is present (this may be done to reduce execution time). FIR_1 denotes a case in which an FIR component is installed, and the FFT functionality is *not* available (perhaps OFDM and equaliser dedicated components have been installed, and the FFT module is unnecessary). In the cases FIR_2 and FIR_3 the functionality of the filter bank module can be achieved algorithmically, because both the FFT and the FIR are available. However, under FIR_1, there is no FFT module, thus the filter bank functionality would require an installed component. For this reason, there is no arrow pointing to the filter bank module from the FIR_1 sub-node. On the other hand, the solid arrow pointing to the equaliser from the FIR is drawn from the oval around the various FIR sub-nodes. This indicates that we can achieve equalisation through the FIR, irrespective of how we achieve the FIR functionality (contrary to the filter bank case).

Shown also on figure 2(c) is a simpler case of AND dependency, which involves achieving multi-rate filtering (MRateFilt) through *both* FIR *and* down sampling (DSamp). The dash line from the FIR oval to DSamp, followed by a solid arrow from DSamp to MRateFilt, shows that if we have the FIR functionality (no matter how), and we install ("build a road to") a down sampler, then we can achieve multi-rate filtering "for free" (algorithmically).

IV. An optimal architecture

A. Key optimisation parameters

The decision to provide a functionality via a self-contained component or by invoking multiple times simpler functional modules is determined by two key considerations: (monetary) cost and (execution) time.

The monetary cost of a component (which is paid only once during the useful life of the radio) represents the total cost of including the component in the design. In some architectures for reconfigurable radios, the monetary cost can be represented by (is proportional to) the number of logic units necessitated by an FPGA implementation. It is best to take the view point of a system integrator that "outsources" the components (software or hardware). Then, the monetary cost represents the fair-market value of acquiring the finished component from outsiders.

Execution time (incurred every time a component is employed throughout the life of the system) is also a critical consideration, because end-users have limited "patience" (tolerance to latency), and, in fact, communication standards impose hard time constraints for the completion of certain operations.

B. Considering both cost and latency: 2 approaches

There are at least two basic approaches to considering cost and latency in the design of reconfigurable equipment: (i) the "combined" cost function and (ii) the "deadline" approaches.

As in [1], we can combine economic and latency considerations into a single cost function (a weighted sum of both "costs"). The combined cost function has advantages and disadvantages. On one hand, (i) it adds the monetary cost (paid once by the designer) with the "delay costs" incurred by the user each time it executes a standard throughout the useful life of the equipment, (ii) it fails to account for the hard time constraints often arising from communication applications, and (iii) it makes the design highly dependent on the weights,

which are themselves arbitrary. On the other hand, the combined cost function (i) is a reasonable first step, which simplifies the exposition and the solution algorithm, and (ii) may be helpful in pre-design work, as it could allow the designer to gear a design to the "luxury" (high performance) or "value" market sectors.

As an alternative, we can proceed as in [2], where we minimise the economic cost only, and take latency into account through constraints ("deadlines") that cannot be exceeded while performing certain operations. However, determining the "deadline" to be observed while designing equipment to support a standard is nontrivial. For our purposes, we assume that the appropriate tolerances have been determined by others, and provided to the designers.

C. *Possible algorithms*

The reformulation of section III. provides us with several well-studied algorithms. We shall now identify and discuss some of these algorithms, which seem particularly promising.

Reference [4] considers a network design problem with 2 metrics: cost and "distance". This fits well with our formulation, with execution time as "distance". It follows the first approach discussed in subsection B., that is, it combines both cost and "distance" into a single objective. But this algorithm is probabilistic. Reference [5] modifies it to make it deterministic, utilising linear programming.

The basic idea of [4] is as follows. The algorithm has several stages. At each stage it forms a matching ("pairing") of terminal nodes (not previously eliminated). Then, of each pair of nodes, it chooses a "centre". At this point, the non-centre nodes are removed from further consideration, and a new stage of the algorithm starts (with a new matching among the "centres" only). Because for us the terminals correspond to standards to be supported by the radio, and hence should be relatively few. Thus, this algorithm will give an answer in a relatively short number of stages.

For reasons given in subsection B., it may be preferable to utilise an algorithm which minimises cost, while keeping "length" (time/latency) under a specified budget. Reference [6] provides an algorithm that does exactly that. However, this algorithm appears more complex than the preceding ones.

V. Discussion and outlook

We represent a multistandard reconfigurable radio as a graph of progressively simpler functional modules. Recently, we have used this mathematical framework to find an architecture which is optimal in view of both cost and performance. Previously, we utilised two algorithms whose limitations we have discussed: (i) exhaustive search [1, 2] and (ii) simulated annealing [1].

Presently, we have recast our design effort as a "network design problem", which has an extensive literature, offering algorithms, computational complexity results, and many practical applications. We have illustrated our proposal through a simple but realistic design. We have not formally proved that *every conceivable* graph of design choices can be *exactly* represented as a network design problem. But even if this representation is not possible in specific cases, by ignoring certain non-essential connections, one may obtain a network design problem sufficiently close to reality to provide a good approximation to the optimal design (which perhaps can then be revised manually). As an alternative, one could study the inner workings of an appropriate network design algorithm and try to adapt it to the problem at hand. We lack space to apply here a particular network design algorithm, but we have identified and briefly discussed some specific algorithms which seem particularly promising. We will continue to explore the rich network design literature.

Important issues have not yet been addressed or need further attention. The list includes: building the hypergraph of all design choices, as well as considering (i) the time needed to reconfigure the radio while switching from a standard to another, (ii) the "travel time" of signals from a component to another, and (iii) the possible contention among high level modules for the service of the same lower-level module (which may be particularly important if the reconfigurable radio needs to support simultaneous communication over several standards). Additionally, we have not yet considered energy consumption issues. We hope to address these and other important issues in the near future.

Acknowledgement

Supported by the "Région Bretagne", France.

References

[1] C. Moy, J. Palicot, V. Rodriguez, and D. Giri, "Optimal determination of common operators for multi-standards software-defined radio," in *Proc. of 4th Karlsruhe Workshop on Software Radios*, (Karlsruhe, Germany), March 2006.

[2] V. Rodriguez, C. Moy, and J. Palicot, "An optimal architecture for a multi-standard reconfigurable radio: Cost-minimising common operators under latency constraints," in *15th IST Mobile and Wireless Comm. Summit*, (Myconos, Greece), June 2006.

[3] T. Magnanti and R. Wong, "Network design and transportation planning: models and algorithms," *Transportation Science*, vol. 18, no. 1, pp. 1–56, 1984.

[4] A. Meyerson, K. Munagala, and S. Plotkin, "Cost-distance: two metric network design," in *Foundations of Computer Science. Proceedings of the 41st Annual Symposium on*, (Redondo Beach, CA), pp. 624–630, 2000.

[5] C. Chekuri, S. Khanna, and J. Naor, "A deterministic algorithm for the cost-distance problem," in *Proc. of the twelfth annual ACM-SIAM symp. on discrete algorithms*, (Philadelphia, PA, USA), pp. 232–233, 2001.

[6] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt, "Bicriteria network design problems," *Journal of Algorithms*, vol. 28, pp. 142 – 171, 1998.

[7] J. Palicot, C. Roland, Y. Louet, and A. Al Ghouwayel, "A new parameterization technique for reconfigurable radio: the common operator approach." Submitted to Annales des Telecommunications, 2006.

[8] E. Bidet, J. Cardin, M. Kouam, C. Joanblanq, and J. Palicot, "FDF, a 512-TAP FIR filter using a mixed temporal-frequential approach," in *Proc. of the IEEE Custom Integrated Circuits Conference*, (Santa Clara, CA, USA), pp. 173–176, 1995.

[9] J. Palicot and C. Roland, "FFT: a basic function for a reconfigurable receiver," in *Telecommunications. 10th International Conference on*, vol. 1, pp. 898–902, 2003.