

# Integrated Resource Management in Cognitive Radio

Vuk Marojevic, Nemanja Vucevic, Xavier Revés, and Antoni Gelonch

**Abstract**—The today's radio environment features a heterogeneous mix of radio access technologies. The technology that facilitates accessing different air interfaces at different times with the same radio equipment is known as software-defined radio (SDR). A cognitive radio system manages the operational modes of SDR equipment as a function of the radio environment. This paper introduces a novel approach to resource management in cognitive radio. We coin it integrated resource management (IRM) because it integrates the management of radio and computing resources. The corresponding system then features two cognitive cycles, the computing cycle and the radio cycle.

We demonstrate the suitability of our proposal in a heterogeneous radio and computing scenario. In particular, simulation results show that two simple IRM algorithms can considerably reduce the number of lost user sessions due to inappropriate reconfiguration decisions. The paper concludes that such an integrated resource management, when embedded in a cognitive radio system, can intelligently distribute the radio and computing loads, while adapting to environmental changes.

**Index Terms**—Cognitive radio, integrated resource management, software-defined radio (SDR).

## I. INTRODUCTION

A high degree of heterogeneity in the wireless access characterizes the radio environment of today and tomorrow. In such an environment many wireless standards, or radio access technologies (RATs), coexist. In the near future the mostly used technologies will be the 3G-RATs and WLAN followed by emerging technologies, such as WIMAX or 4G. Therefore we argue that a proper cooperation and integration of a set of, dynamically evolving, radio access technologies with different capabilities is required to fully exploit their capacities.

Nowadays, many multi-mode mobile terminals are capable to manage several radio access technologies. These terminals contain specifically designed and power efficient application-specific integrated circuits (ASICs). In contrast, base stations are built around a cluster of application-specific and general purpose processors (GPPs). This mix of ASICs and GPPs reflects the trade-off between efficiency and flexibility. We observe a tendency towards flexible

hardware architectures, where an array of GPPs replaces, as much as possible, the array of ASICs.

The increasing computing and power efficiency of the latest microprocessors, the development of powerful design tools that integrate the hardware and software issues, and the currently available hardware abstraction layers, such as SCA (software communications architecture) [1] and PHAL (platform and hardware abstraction layer) [2], facilitate the introduction of software-reconfigurable base stations and mobile terminals. This concept is known as software-defined radio (SDR). It characterizes hardware platforms that principally consist of an array of programmable processors, such as general-purpose processors, digital signal processors (DSPs), and field-programmable gate arrays (FPGAs). These SDR platforms will provide the necessary computational power to execute an SDR application, which stands for a RAT that is implemented in software following an object oriented approach [3] [4].

Although the users change their mobile terminals every 2 or 3 years, the benefit of software-defined radio is not limited to base stations: The possibility of reconfiguring a mobile terminal facilitates to immediately obtain a new and previously unexpected user service the market may offer. This requires a dynamic management of the emerging SDR equipment, be it mobile or fixed.

These new levels of flexibility in radio communications encourage the introduction of intelligent management systems that address the complete set of resources: the scarce radio resources and the limited computing capabilities of SDR equipment. Such systems should observe the environment and react intelligently upon its changes in order to optimize the complete system behavior according to the established policies. The basic idea is known as cognitive radio [5]–[7], although actual research focuses merely on the joint radio resource management (JRRM).

This paper introduces a novel approach to resource management in cognitive radio. We coin it integrated resource management (IRM) because it integrates the management of radio and computing resources (section II). We simulate a heterogeneous radio and computing scenario to demonstrate the importance of our proposal (sections III and IV).

## II. INTEGRATED RESOURCE MANAGEMENT IN COGNITIVE RADIO

Cognitive radio, introduced in the late 90s, is an ambient aware system that reacts upon changes in the radio environment. It incorporates a learning mechanism and is, thus, an intelligent system. Currently, its objective is to optimize the usage of radio resources [7].

Manuscript received January 31, 2007; revised May 15, 2007. This work has been supported by the DURSI of the Catalanian Government and the CYCIT (Spanish National Science Council) under grant TEC2006-09109, which is partially financed from the European Community through the FEDER program.

The authors are with the Dept. of Signal Theory and Communications, Universitat Politècnica de Catalunya (UPC), C/ Jordi Girona 1-3, 08034 Barcelona, Spain (e-mail: marojevic@tsc.upc.edu, vucevic@tsc.upc.edu, xavier.reves@tsc.upc.edu, antoni@tsc.upc.edu).

We already argued for the importance of computing resource management in software-defined radio [8]. This paper correspondingly extends the scope of cognitive radio, introducing the term *integrated resource management*, which stands for an integrated radio and computing resource management. Some arguments for this integration are the expected commercialization of software-defined radio (SDR), the numerous 4G-RATs that partially overlap in space and time, and the vast variety of SDR equipment.

Fig. 1 shows our implementation of a cognitive radio system, which consists of two cognitive cycles, the *computing cycle* and the *radio cycle*. The two cognitive cycles continuously scan the computing and radio environments and provide the information of any change to the IRM entity (IRME). The IRME is the brain of the system. It initiates the appropriate actions as a function of the system's current state. Each action modifies the computing and radio environments.

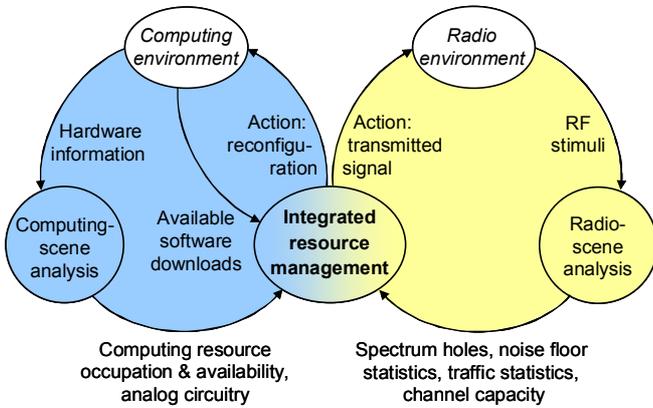


Fig. 1. Integrated resource management joining two cognitive cycles.

In this paragraph we exemplify the above concepts and therefore consider a heterogeneous radio scenario that supports several radio access technologies within the coverage area of the cognitive radio system. The radio scene analysis reveals that the radio resources of RAT  $X$  are overused while those of RAT  $Y$  are not. In other words, RAT  $X$  is running out of radio resources, whereas RAT  $Y$  can accept additional users. The computing cycle, in parallel, observes that many SDR- $X$  mobile terminals (SDR terminals that currently implement RAT  $X$ ) support RAT  $Y$ . The network informs about the availability of the required software for accessing RAT  $Y$ . After processing these inputs, the IRME would probably initiate the corresponding software downloads and the reconfiguration of some SDR- $X$  terminals to become SDR- $Y$  terminals.

Fig. 1 places the two cognitive cycles in the context of the intelligent reconfiguration of SDR mobile terminals. The above concepts are, though, not limited to SDR mobile terminals, but rather valid for any software-reconfigurable radio-access or network element.

The IRME, in general, runs an algorithm that jointly manages the computing and radio resources of its surrounding. Based on the inputs of the two cognitive cycles and the accumulated knowledge, it may request the reconfiguration of selected SDR equipment. Irrespectively of the particular decision, it constantly learns from observing its effects. Al-

though some knowledge may be available a priori, it is the learning mechanism that makes it an intelligent entity.

Radio and computing resources are and will always be limited. Therefore, we argue that such an integrated resource management is necessary to jointly optimize their usage. Moreover, we believe that the lack of an integrated resource management and not the resource scarcity itself hamper an anytime, anywhere, and anyhow wireless access. Future work may additionally integrate application-oriented issues or energy aspects.

The rest of the paper examines the influence of integrating the computing and the radio resource management in a single management entity. Particularly, we study the IRME's influence on the radio load balancing and therefore consider the following operation of the two cycles: The radio cycle informs the IRME about the supported RATs and their load in the cell or cells that are within the coverage area of the cognitive radio system. The computing cycle provides the information about the computing capabilities of the SDR mobile terminals and the available software downloads. Section II-A further concretizes this scenario. Section II-B describes the IRM algorithms.

#### A. Case Study

The computing scene analysis of the cognitive radio system's coverage area shows that many of the active SDR-GPRS terminals with UMTS capabilities have currently unoccupied computing resources. This potentiates their reconfigurations to UMTS, the more computing-complex RAT. The network, furthermore, informs about the availability of software downloads for different UMTS realizations. At the same time, the radio scene analysis reveals a high number of active GPRS sessions and a low number of active UMTS sessions within the system's boundaries. Combining these inputs, the IRME tries to alleviate the misbalance in the usage of radio resources of these two RATs while exploiting the terminals' computing capabilities. Among the terminals that penetrate the region controlled by the cognitive radio system (*coverage area*), the IRME examines the reconfiguration of 10000 active SDR-GPRS terminals, which specify our simulation time span.

#### B. IRM Algorithms

We consider the two IRM algorithms Alg-1 and Alg-2. Both define the thresholds  $T_{384}$ ,  $T_{128}$ , and  $T_{64}$ , which specify the maximum achievable bit-rate as a function of the total processing capacity  $C_T$  of the SDR platform:

- 384 kbps UMTS if  $C_T \geq T_{384}$ ,
- 128 kbps UMTS if  $C_T \geq T_{128}$ ,
- 64 kbps UMTS if  $C_T \geq T_{64}$ .

The algorithms try to provide the best possible service a terminal can support. Thus, if  $C_T \geq T_{384}$  the system initiates the download of the 384 kbps UMTS receiver processing chain, which is locally mapped to the available computing resources of the SDR platform. If this mapping fails (*infeasible mapping*), which means a lost user session, Alg-1 tries to reconfigure the next terminal of the same characteristics to the 128 kbps UMTS realization, whereas Alg-2 chooses a bit-rate of 64 kbps and, if successful, tries to map the 128 kbps receiver the next time.

In case that the 128 kbps UMTS processing chain is unfeasibly mapped to the available computing resources, both algorithms try the 64 kbps processing chain the next time such a terminal appears, and if this reconfiguration also fails, this kind of terminal is declared infeasible for running in UMTS mode and is not again considered for reconfiguration.

The baseline algorithm, Alg-0, does not process any hardware information and tries to reconfigure all 10000 SDR-GPRS mobile terminals to operate in the 384 kbps UMTS mode.

### III. SIMULATIONS

#### A. SDR Platforms and Applications

Future SDR platforms will be of different architectures and computing capacities. To simulate this (in a reduced manner), we assume the three platform types of Fig. 2 with the variable computing and bandwidth capacities  $C$  and  $B$ . Each platform consists of 3 fully connected processors. A completely homogeneous, a semi-heterogeneous, and a completely heterogeneous architecture characterize these platforms, which represent a small excerpt of the vast variety of possible SDR mobile terminals. The first 5000 examined SDR-GPRS mobile terminals are of types I and II, being equiprobable, whereas the next 5000 terminals are of types I, II, and III with a probability of 0.33 each. This simulates the dynamism in the introduction of new mobile terminals.

Each SDR mobile terminal's computing capacity consists of the total processing capacity  $C_T = 3 \cdot C$  and the total bandwidth  $B_T = 6 \cdot B$ .  $C$  and  $B$  stand for the available processing and bandwidth resources after the demapping of the SDR-GPRS transceiver functions. They take random values in  $[5000, 5050, \dots, 6200]$  Mops and  $[250, 350, \dots, 1450]$  Mbps.

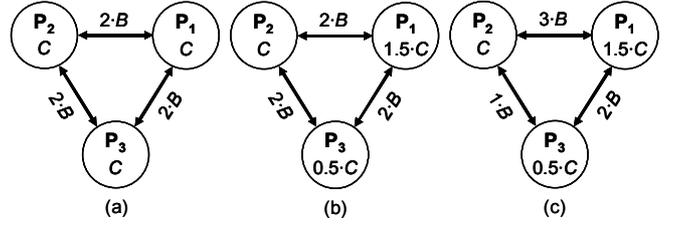


Fig. 2. SDR platform types I-III.

We assume that all 10000 SDR-GPRS platforms contain the analog RF circuitry that is necessary to access the WCDMA air interface. We can make this assumption because not doing so would not affect the performance of our IRM approach, which is aware about the available RF circuitry (Fig. 1). We address the digital signal processing part at the physical layer of a UMTS downlink receiver and assume that additional computing resource for implementing the higher OSI layers and the UMTS uplink transmitter are sufficiently available.

Figs. 3 and 4 capture the computing requirements of the chip- and bit-rate processing chains of a UMTS downlink receiver supporting the raw data rates of 64, 128, and 384 kbps. We obtained these computing demands from [9]–[12] and from available implementations, considering that a processing requirement is the number of multiply-accumulate operations times the required processing frequency. Similarly, the bandwidth requirement is the sampling frequency  $f_s$  multiplied by the number of bits per sample.

The three complete processing chains require a minimum processing capacity of 14994, 15162, and 16264 Mops. The IRME specifies the thresholds  $T_{64}$ ,  $T_{128}$  and  $T_{384}$  as 15000, 15300, and 16350 Mops, which is, given the granularity of  $C$ , theoretically just enough to execute the corresponding receiver realization. We introduce  $T'_{384} = T_{384} / 3$ ,  $T'_{128} = T_{128} / 3$ , and  $T'_{64} = T_{64} / 3$  for simplifying the notation below.

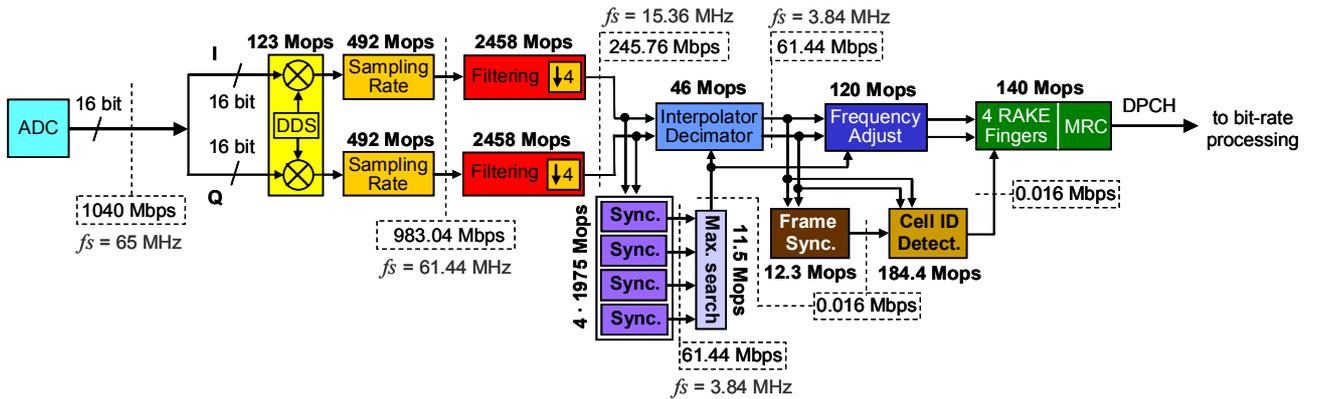


Fig. 3. Chip-rate processing requirements of a UMTS downlink receiver.

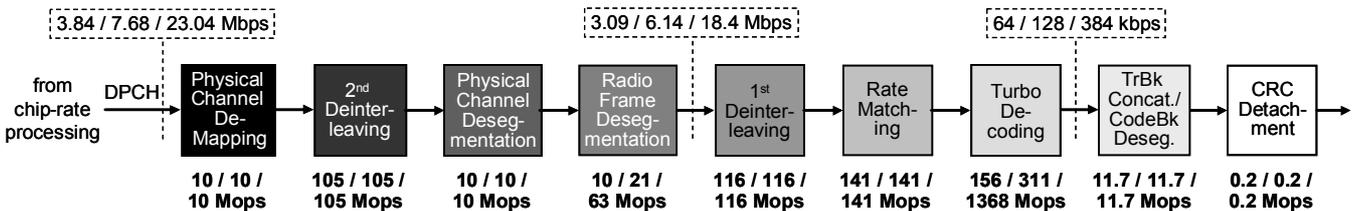


Fig. 4. Bit-rate processing requirements of a 64 / 128 / 384 kbps UMTS downlink receiver.

## B. Application Mapping

In this case study, the SDR mobile terminals are equipped with the  $t$ -mapping algorithm [8], which maps the downloaded SDR-UMTS processing chain to the platform's available computing resources. The SDR functions are mapped one by one to all processors minimizing the sum between the computation and the communication costs. A bandwidth is reserved if two communicating processes are mapped to different processors; otherwise, the data is directly obtained from the processor's local memory. The mapping of each SDR function to all  $N$  processors results in  $N$  different mapping options of the entire processing chain. The algorithm then chooses the least-cost solution.

The low complexity of this mapping approach greatly facilitates its implementation. Since being non-optimal, it does not always find a feasible solution. A feasible solution, or *feasible mapping*, reserves no more than 100% of any computing resource and, hence, meets the RAT-specific computing constraints. In particular, a feasible mapping can meet the timing constraints because time is considered as an implicit resource [8].

## C. Simulation Results

Fig. 5 shows the number of lost sessions as a function of the SDR platform. Alg-0, which does not process any hardware information, constantly loses user sessions due to inappropriate reconfiguration decisions. Alg-1 and Alg-2 saturate the first time after having examined between 3000 and 4000 SDR platforms. This means that the algorithms have learned which of the terminals of types I and II cannot be feasibly reconfigured to operate in the UMTS mode and, thus, do not request their reconfiguration any more. With the introduction of the third terminal type, the learning process for this terminal starts. Alg-1 and Alg-2 finally saturate at about 1000 and 750 lost sessions.

Figs. 7-9 illustrate the accumulated mapping information due to the learning process of Alg-1 and Alg-2. First we observe that, although the UMTS realization is selected as a function of a platform's processing capacity (section II-B), the mapping algorithm cannot always find a feasible solution. Figs. 7-9 also show that the mapping results are a function of the platform type so that each type requires its own learning process. For practical cases this means that the cognitive learning process does not terminate after several cognitive cycles; the dynamism in the introduction of new SDR platforms, amongst others, requires that the two cognitive cycles of Fig. 1 run continuously.

Alg-0 either feasibly reconfigures the SDR-GPRS mobile terminal to the 384 kbps UMTS realization or loses the current user session. Alg-1 and Alg-2 lose much less sessions than Alg-0 because they either maintain the terminals that cannot be feasibly reconfigured to operate in the 384 kbps UMTS mode in GPRS mode or reconfigure them to a lower bit-rate UMTS realization (Fig. 6). Alg-2 loses fewer sessions than Alg-1 because its prime objective is to test whether a terminal that cannot operate at the theoretically possible speed can operate in the UMTS mode at all. Alg-1, on the other hand, prioritizes finding a feasible implementation of the *best* UMTS realization as soon as possible. Consequently, Alg-1 reconfigures more terminals to the 128 kbps and fewer to the 64 kbps UMTS mode in respect to

Alg-2. Figs. 6b and c, however, show that the corresponding results only negligibly differentiate the two algorithms. The considerably better performance of Alg-2 regarding the number of lost sessions makes it the better choice in this scenario.

Alg-1 and Alg-2 feasibly reconfigure 5791 SDR-GPRS mobile terminals ( $\sim 58\%$  in Figs. 6b and c), whereas Alg-0 reconfigures only 3656, or 36.6% (Fig. 6a), feasibly. Hence, both IRM algorithms clearly outperform the baseline algorithm in maximizing the number of feasible reconfigurations and minimizing the number of infeasible reconfigurations, or lost sessions. This reflects the achievable improvements of the integrated resource management over the joint radio resource management by itself.

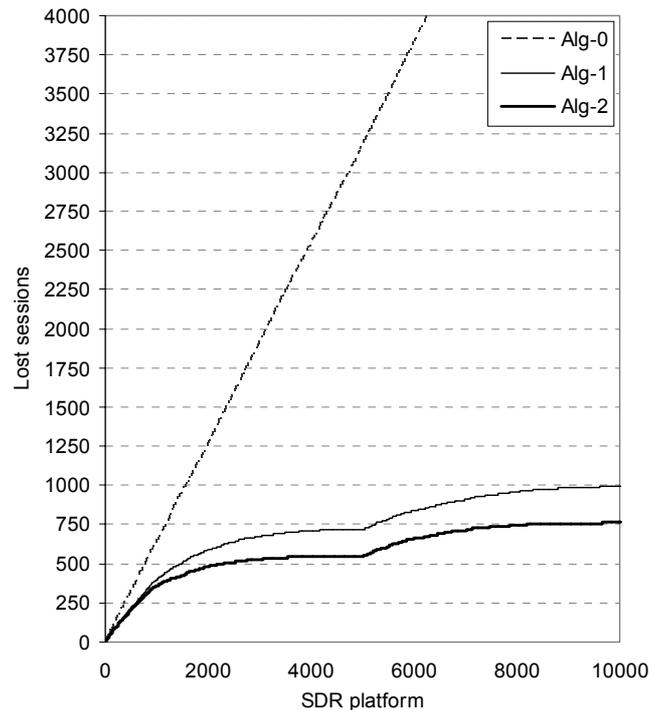


Fig. 5. Number of lost sessions as a function of the SDR platform.

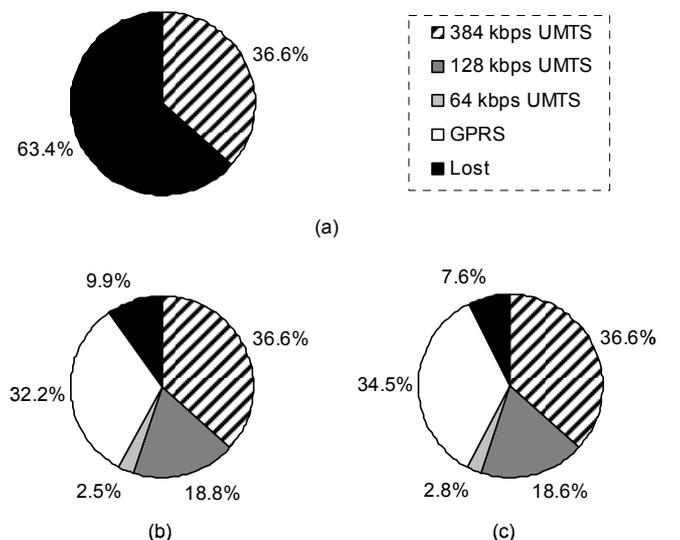


Fig. 6. Distribution of sessions among the 10000 SDR mobile terminals after applying Alg-0 (a), Alg-1 (b), and Alg-2 (c).

#### IV. CONCLUSIONS

This paper proposes to integrate the radio and computing resource management, coined as *integrated resource management*, in cognitive radio. The simulation results show that two simple IRM algorithms outperform an algorithm that is unaware of the SDR mobile terminals' architectures and computing capacities in a heterogeneous radio scenario.

The paper indicates that trading off computing against radio resources may facilitate the wireless access in dynamically evolving and heterogeneous radio and computing environments. We believe that the actual problem in modern radio communications is not the (computing and radio) resource scarcity but rather the lack of an integrated resource management framework and the corresponding algorithms. The proposed concepts may serve as the basis for developing such a framework and for introducing sophisticated IRM algorithms.

#### REFERENCES

- [1] N. Hayes, "The JTRS SCA specification - the past, the present, and the future," *Proc. IEEE Military Communications Conf. (MILCOM 2005)*, Atlantic City, NJ, 17–20 Oct. 2005, pp. 2713–2719.
- [2] X. Reves, Antoni Gelonch, Vuk Marojevic, Ramon Ferrús., "Software radios: unifying the reconfiguration process over heterogeneous platforms," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 16, Sept. 2005.
- [3] J. Mitola III, "The software radio architecture," *IEEE Commun. Mag.*, vol. 33, no. 5, pp. 26–38, May 1995.
- [4] W. H. W. Tuttlebee, "Software Defined Radio: facets of a developing technology," *IEEE Personal Commun.*, vol. 6, iss. 2, pp. 38–44, April 1999.
- [5] J. Mitola, G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, 1999.
- [6] J. Mitola, "Cognitive radio: an integrated agent architecture for software defined radio," Ph.D. dissertation, Royal Institute of Technology (KTH), Stockholm, Sweden, May 2000.
- [7] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE JSAC*, vol. 23, no. 2, pp. 201–220, Feb. 2005.
- [8] V. Marojevic, X. Revés, A. Gelonch, "Computing resource management for SDR platforms," *Proc. 16th IEEE Int'l Symp. Personal, Indoor and Mobile Radio Communications (PIMRC 2005)*, Berlin, Sept. 11–14, 2005, pp. 685–689.
- [9] R. Tanner, J. Woodard, *WCDMA – Requirements and Practical Design*. John Wiley and Sons Inc., 2004.
- [10] T. Faber, M. Schönle, "DSP-platform target report," SLATS Consortium, Project no. 27016, Deliverable D23, Dec. 1999.
- [11] Technical Specification Group Radio Access Network (3GPP), "TS 25.213 V5.5.0 – spreading and modulation (FDD)," Dec. 2003, [www.3gpp.org](http://www.3gpp.org)
- [12] Technical Specification Group Radio Access Network (3GPP), "TS 25.212 V6.4.0 – multiplexing and channel coding (FDD)," March 2005, [www.3gpp.org](http://www.3gpp.org)

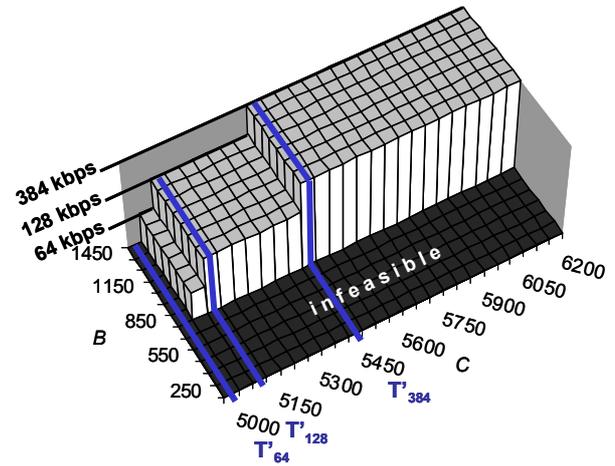


Fig. 7. Accumulated mapping information for SDR platform type I.

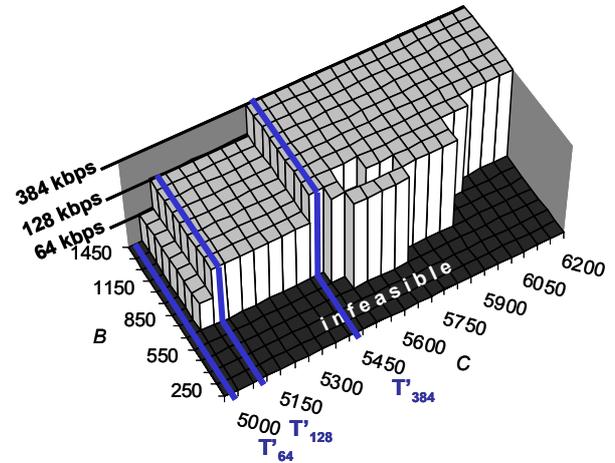


Fig. 8. Accumulated mapping information for SDR platform type II.

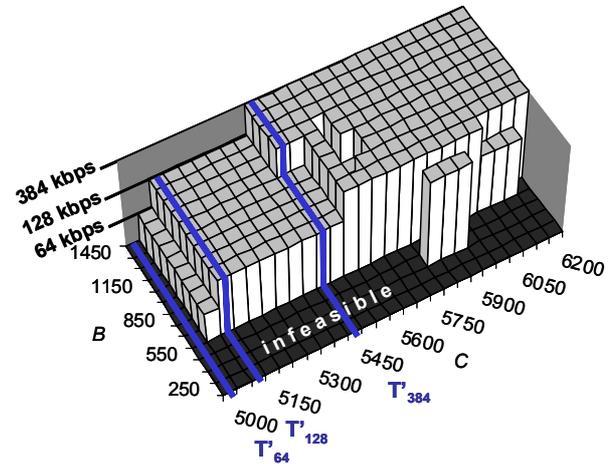


Fig. 9. Accumulated mapping information for SDR platform type III.