An Open Source Product-Oriented LTE Network Simulator based on ns-3

Nicola Baldo, Marco Miozzo, Manuel Requena-Esteso, Jaume Nin-Guerrero
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)
Av. Carl Friedrich Gauss 7, 08860 Castelldefels (Barcelona), Spain
{nbaldo, mmiozzo, mrequena, jnin}@cttc.es

ABSTRACT

In this paper we present a new simulation module for ns-3 aimed at the simulation of LTE networks. This module has been designed with a product-oriented perspective in order to allow LTE equipment manufacturers to test RRM/SON algorithms in a simulation environment before they are deployed in the field. First, we describe the design of our simulation module, highlighting its novel aspects. Subsequently, we discuss the testing methodology that we adopted to validate its output. Finally, we present some experimental result to assess its performance in terms of execution time and memory usage.

Categories and Subject Descriptors

I.6.5 [**Simulation and Modeling**]: Model Development—*Modeling methodologies*; I.6.7 [**Simulation and Modeling**]: Simulation Support Systems—*Environments*

General Terms

Algorithm, Design, Performance, Verification

Keywords

3GPP, LTE, E-UTRA, Scheduling, Simulator, ns-3

1. INTRODUCTION

3GPP Long Term Evolution (LTE) is the most promising standard for the upcoming fourth generation (4G) of mobile wireless communication systems. Very briefly, the innovative aspects included in LTE are a high speed radio interface based on OFDMA/SC-FDMA called Evolved Universal Terrestrial Radio Access (E-UTRA), an all-IP core network called Evolved Packet Core (EPC), and, most importantly, Self Organized Network (SON) capabilities, i.e., the ability of the base stations – called eNodeBs (eNBs) using the LTE nomenclature – to form a self configuring, self-optimizing and self-healing network. SONs are seen as a crucial enhancement with respect to prior mobile technologies, since they allow higher density network deployments that yield a significant increase in system capacity, and at the same time they permit a reduction in the operational expenditure of mobile network operators. In addition, the development of SON capabilities is an opportunity for LTE vendors to add value to their product and to be more competitive in the market. In this process, LTE simulators are expected to play a crucial role [1].

In this paper we describe a new software module for ns-3 that we are developing. The model partially leverages on the previous work of [2], in that the Channel model and part of the Physical layer model are reused. However, the rest of the E-UTRA model has been totally redesigned to be more representative of the LTE standard, as well as closer to real-world implementations. In particular, the LTE MAC Scheduler API published by the FemtoForum [3] has been adopted in the simulator. This interface is used by femtocell manufacturers for the implementation of scheduling and Radio Resource Management (RRM) algorithms. By introducing support for this interface in the simulator, we make it possible for LTE equipment vendors and operators to test in a simulative environment exactly the same algorithms that would be deployed in a real system.

The ultimate objective of our LTE simulator module is to become a widely accepted reference evaluation platform for LTE-based SON systems. In order to achieve this, we pay a particular attention to two important aspects. The first one is a thorough test and validation procedure that aims at guaranteeing that the output produced by the simulator represents the performance of LTE systems with a satisfactory level of accuracy. The second one is the open source nature of the simulator: by being GPL-licensed and by having a development process which is open to the community, we expect not only to foster early adoption and contributions by industrial and academic partners, but also to enhance the overall quality of the simulator by leveraging on a wide base of users and developers. The interested reader can download the source code from [4].

2. DESCRIPTION OF THE LTE SIMULATION MODEL

2.1 Design Criteria

The LTE module that we present in this paper has the objective of supporting the evaluation of the following aspects of LTE systems: 1) Radio Resource Management, 2) QoS-aware Packet Scheduling, 3) Inter-cell Interference Coordination and 4) Dynamic Spectrum Access. In order to

MSWiM'11, October 31–November 4, 2011, Miami, Florida, USA.

model LTE systems to a level of detail that is sufficient to allow a correct evaluation of the above mentioned aspects, the following requirements have been considered:

- At the radio level, the granularity of the model should be at least that of the Resource Block (RB). In fact, this is the fundamental unit being used for resource allocation. Without this minimum level of granularity, it is not possible to model accurately packet scheduling and inter-cell-interference.[1]
- The simulator should scale up to tens of eNBs and hundreds of User Equipments (UEs). This rules out the use of a link level simulator, i.e., a simulator whose radio interface is modeled with a granularity up to the symbol level. This is because, to have a symbol level model, it would be necessary to implement all the PHY layer signal processing, whose huge computational complexity would severely limit simulation scalability. In fact, link-level simulators are normally limited to a single eNB and one or a few UEs.
- It should be possible within the simulation to configure different cells so that they use different carrier frequencies and system bandwidths. The bandwidth used by different cells should be allowed to overlap, in order to support dynamic spectrum licensing solutions such as those described in [5,6]. The calculation of interference should handle appropriately these configurations.
- The LTE simulation module should contain its own implementation of the FemtoForum LTE MAC Scheduler Interface Specification [3]. Neither binary nor data structure compatibility with vendor-specific implementations of the same interface should be supported; hence, a compatibility layer should be interposed whenever a vendor-specific MAC scheduler is to be used with the simulator. This requirement is necessary to allow the simulator to be independent from vendor-specific implementations of this interface specification.[2]

2.2 Implemented MAC Functionality

In this section we describe the MAC features that are implemented in the simulator. All these features are implemented after models from the literature. The reason for this choice is twofold: first, well-known models are easily understood by developers and are a convenient starting code base for the development of more advanced solutions; second, they can be used as reference algorithm when doing performance evaluation. With this latter respect, we stress that the use of a publicly available implementation of models (in particular scheduler implementations) as the reference for a performance evaluation study is beneficial to the authoritativeness of the study itself.

2.2.1 Resource Allocation Model

We now briefly describe how resource allocation is handled in LTE, clarifying how it is implemented in the simulator. The scheduler is in charge of generating specific structures called Data Control Indication (DCI) which are then transmitted by the PHY of the eNB to the connected UEs, in order to inform them of the resource allocation on a per subframe basis. In doing this, the scheduler has to fill some specific fields of the DCI structure with all the information, such as: the Modulation and Coding Scheme (MCS) to be used, the MAC Transport Block (TB) size, and the allocation bitmap which identifies which RBs will contain the data transmitted by the eNB to each user.[3] The latter bitmap can be coded in different formats; in this implementation, we considered the *Allocation Type 0* defined in [8], according to which the RBs are grouped in Resource Block Groups (RBG) of different size determined as a function of the Transmission Bandwidth Configuration[4] in use.

2.2.2 Adaptive Modulation and Coding

The Adaptive Modulation and Coding (AMC) model provided by the simulator is a modified version of the model described in [2], which in turn is inspired from [10]. Our version is described in the following. Let i denote the generic user, and let γ_i be its SINR. We get the spectral efficiency η_i of user i using the following equations:

$$\Gamma = \frac{-\ln(5 * \mathrm{BER})}{1.5} \tag{1}$$

$$\eta_i = \log_2\left(1 + \frac{\gamma_i}{\Gamma}\right) \tag{2}$$

where BER = 0.00005. The procedure described in [11] is used to perform AMC. The spectral efficiency is quantized based on the table reported in [11] (rounding to the lowest value) and is mapped to the corresponding MCS scheme.

2.2.3 Round Robin (RR) Scheduler

The Round Robin (RR) scheduler is probably the simplest scheduler found in the literature. It works by dividing the available resources among the active flows, i.e., those logical channels which have a non-empty RLC queues. If the number of RBGs is greater than the number of active flows, all the flows can be allocated in the same subframe. Otherwise, the excesses flows that cannot be allocated in a subframe will be considered in the subsequent one in a circular fashion. The MCS to be adopted for each user is evaluated according to the received wideband CQIs.

2.2.4 Proportional Fair (PF) Scheduler

The Proportional Fair (PF) scheduler [7] works by scheduling a user when its instantaneous channel quality is high relative to its own average channel condition over time. Let i, j denote generic users; let t be the subframe index, and k be the resource block index; let $M_{i,k}(t)$ be MCS usable by user i on resource block k according to what the AMC model reported (see Section 2.2.2); finally, let $S(M, B)$ be the TB size in bits as defined in [8] for the case where a number B of resource blocks is used. The achievable rate $R_i(k, t)$ in

[1]The reason is that, since packet scheduling is done on a per-RB basis, an eNB might transmit on a subset only of all the available RBs, hence interfering with other eNBs only on those RBs where it is transmitting.

[2]We note that [3] is a logical specification only, and its implementation (e.g., translation to some specific programming language) is left to the vendors.

[3]For the mapping of resources to physical RBs, we adopt a *localized mapping* approach (see [7], Section 9.2.2.1); hence in a given subframe each RB is always allocated to the same user in both slots.

[4]We note that the Transmission Bandwidth Configuration, defined in [9], indicates the total number of RBs which can be used by an eNB. The allowed values are 6, 15, 25, 50, 75 and 100.

bit/s for user i on resource block k at subframe t is defined as

$$R_i(k,t) = \frac{S\left(M_{i,k}(t), 1\right)}{\tau} \quad (3)$$

where τ is the TTI duration. At the start of each subframe t, each RB is assigned to a certain user. In detail, the index $\widehat{i}_k(t)$ to which RB k is assigned at time t is determined as

$$\widehat{i}_k(t) = \underset{j=1,\ldots,N}{\operatorname{argmax}} \left(\frac{R_j(k,t)}{T_{\mathrm{j}}(t)} \right) \quad (4)$$

where $T_j(t)$ is the past throughput performance perceived by the user j. According to the above scheduling algorithm, a user can be allocated to different RBGs, which can be either adjacent or not, depending on the current condition of the channel and the past throughput performance $T_j(t)$. The latter is determined at the end of the subframe t using the following exponential moving average approach:

$$T_j(t) = (1 - \frac{1}{\alpha})T_j(t-1) + \frac{1}{\alpha}\widehat{T}_j(t) \quad (5)$$

where α is the time constant (in number of subframes) of the exponential moving average, and $\widehat{T}_j(t)$ is the actual throughput achievable by the user i in the subframe t. $\widehat{T}_j(t)$ is measured according to the following procedure. First we determine the MCS $\widehat{M}_j(t)$ actually used by user j as the minimum one among the allocated RBGs. Then we determine the total number $\widehat{B}_j(t)$ of RBs allocated to user j:

$$\widehat{B}_j(t) = \left| \{ k : \widehat{i}_k(t) = j \} \right| \quad (6)$$

where $|\cdot|$ indicates the cardinality of the set; finally,

$$\widehat{T}_j(t) = \frac{S\left(\widehat{M}_j(t), \widehat{B}_j(t)\right)}{\tau} \quad (7)$$

3. TESTING AND VALIDATION

To test and validate the ns-3 LTE module, several provided test suites are integrated with the ns-3 test framework. The most important ones are described in the following subsections.

3.1 Adaptive Modulation and Coding Test

The test suite `lte-link-adaptation` provides system test cases recreating scenarios with a single eNB and a single UE. Different test cases are created corresponding to different SINR values perceived by the UE. The aim of the test is to check that the chosen MCS corresponds to some known reference values. These reference values are obtained by re-implementing the model described in Section 2.2.2 in Octave. The resulting test vector is represented in figure 1. The test passes if both the following conditions are verified:

- the UE-perceived SINR calculated by the simulator corresponds to the value intended for the given test case within a tolerance of 10^{-7}. The tolerance is meant to account for the approximation errors typical of floating point arithmetic.

- the MCS index chosen by the scheduler matches exactly with the MCS index in the test vector, determined using the above described procedure.

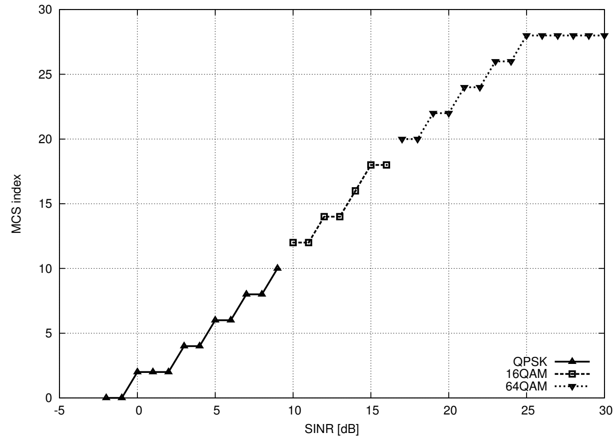

Figure 1: Test vector for the Link Adaptation test

3.2 Round Robin scheduler performance

The test suite `lte-rr-ff-mac-scheduler` creates different test cases with a single eNB and several UEs, all having the same Radio Bearer specification. In each test case, all UEs experience the same inferference and noise impairements from the eNB, i.e. same SINR; different test cases are implemented by using different distances among UEs and the eNB (therefore having different SINR values) and different numbers of UEs. The test consists on checking that the obtained throughput performance is equal among users and matches a reference throughput value obtained according to the SINR perceived within a given tolerance.

The test vector is obtained according to the values of TB size reported in table 7.1.7.2.1-1 of [8], considering an equal distribution of the physical resource blocks among the users using Resource Allocation Type 0 as defined in Section 7.1.6.1 of [8]. Let τ be the TTI duration, N be the number of UEs, B the transmission bandwidth configuration in number of RBs, G the RBG size, M the modulation and coding scheme in use at the given SINR, X the number of RBs assigned to a given user and $S(M,X)$ the corresponding TB size in bits as defined in [8]. We first calculate the number of RBGs allocated to each user as $L = \lfloor B/NG \rfloor$. The reference throughput T in bit/s achieved by each UE is then calculated as $T = S(M, LG)/\tau$. The test passes if the measured throughput matches with T within a relative tolerance of 0.1. This tolerance is needed to account for the transient behavior at the beginning of the simulation (e.g., CQI feedback is only available after a few subframes) as well as for the accuracy of the estimator of the average throughput performance over the chosen simulation time (0.4s). This choice of the simulation time is justified by the need to follow the ns-3 guidelines of keeping the total execution time of the test suite low, in spite of the high number of test cases. In any case, we note that a lower value of the tolerance can be used when longer simulations are run.

In Figure 2, the curves labeled "RR" represent the test values calculated for the RR scheduler test, as a function of the number of UEs and of the MCS index being used in each test case.

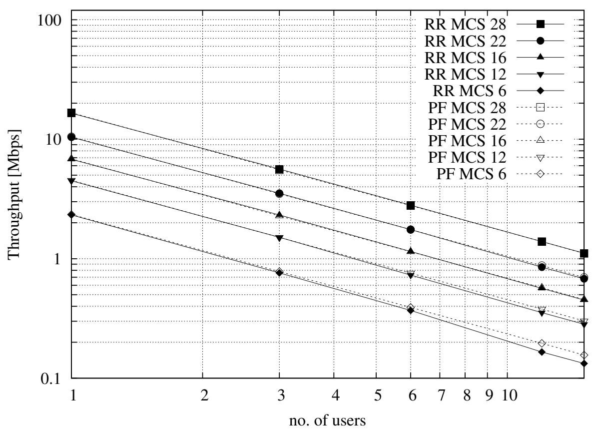

Figure 2: Test vectors for the RR and PF Scheduler in the downlink in a scenario where all UEs use the same MCS.

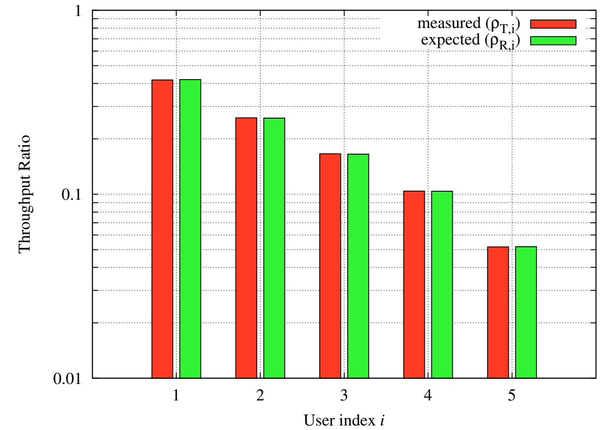

Figure 3: Throughput ratio evaluation for the PF scheduler in a scenario where the UEs have MCS index $28, 24, 16, 12, 6$

3.3 Proportional Fair scheduler performance

The test suite `lte-pf-ff-mac-scheduler` creates different test cases with a single eNB, using the Proportional Fair (PF) scheduler, and several UEs, all having the same Radio Bearer specification. The test cases are grouped in two categories that validate respectively the adaptation to the channel conditions and the fairness among UEs.

In the first category of test cases, the UEs are all placed at the same distance from the eNB, and hence all have the same SINR. Different test cases are implemented by using a different distance value and a different number of UEs. The test consists on checking that the obtained throughput performance matches with the known reference throughput up to a given tolerance. The expected behavior of the PF scheduler when all UEs have the same SINR is that each UE should get an equal fraction of the throughput obtainable by a single UE when using all the RBs. We calculate the reference throughput value by dividing the throughput achievable by a single UE at the given SINR by the total number of UEs. Using the same notation introduced in Section 3.2, the reference throughput T in bit/s is expressed as $T = S(M, B)/\tau N$.

The second category of tests aims at verifying the fairness of the PF scheduler in a more realistic simulation scenario where the UEs have a different SINR (constant for the whole simulation). In these conditions, the PF scheduler will give to each user a share of the system bandwidth that is proportional to the capacity achievable by a single user alone considered its SINR. In detail, let M_i be the modulation and coding scheme being used by each UE (which is a deterministic function of the SINR of the UE, and is hence known in this scenario). Based on the MCS, we determine the achievable rate R_i for each user i using the procedure described in Section 2.2.4. We then define the achievable rate ratio $\rho_{R,i}$ of each user i as

$$\rho_{R,i} = \frac{R_i}{\sum_{j=1}^{N} R_j} \tag{8}$$

Let now T_i be the throughput actually obtained by the UE i, which is obtained as part of the simulation output. We define the obtained throughput ratio $\rho_{T,i}$ of UE i as

$$\rho_{T,i} = \frac{T_i}{\sum_{j=1}^{N} T_j} \tag{9}$$

The test consists of checking that the condition $\rho_{R,i} = \rho_{T,i}$ is verified. If so, it means that the throughput obtained by each UE over the whole simulation matches with the steady-state throughput expected by the PF scheduler according to theory. In particular, the test condition can be derived from [12] as follows. From Section 3 of [12], we know that, $\forall i,\ T_i/R_i = c$, where c is a constant. By substituting the above into the definition of $\rho_{T,i}$ given previously, we get

$$\frac{T_i}{\sum_{j=1}^{N} T_j} = \frac{cR_i}{\sum_{j=1}^{N} cR_j} = \frac{R_i}{\sum_{j=1}^{N} R_j} \tag{10}$$

which is exactly the test condition.

Figure 3 presents the results obtained from the simulator in a test case with 5 UEs $i = 1, \ldots, 5$ that are using respectively the MCS index $28, 24, 16, 12, 6$. From the figure, we note that, as expected, the obtained throughput is proportional to the achievable rate.

4. EXECUTION TIME AND MEMORY CONSUMPTION

In order to provide an evaluation of the execution time and memory consumption, a reference simulation program (`examples/profiling-reference`) has been developed. This program simulates a scenario composed by a set of eNBs, and a set of UEs attached to each eNB. All eNBs have the same number of attached UEs. Communications are performed both in downlink and in uplink using a saturation model (i.e., each RLC instance always has a PDU to transmit). The eNBs are distributed in a line with a spacing of 140m; each UE is placed at the same position of the eNB it is attached to. The total simulation time is set to 60s. Using this simulation program, we ran a simulation campaign varying the number of eNBs as well as the number of UEs per eNB. For each simulation, we measured the execution time using the `time` shell command in linux, and the memory consumption by looking at the information in

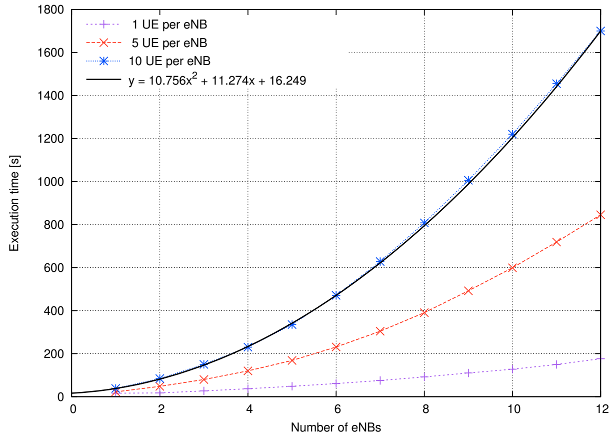

Figure 4: Execution time of the reference program.

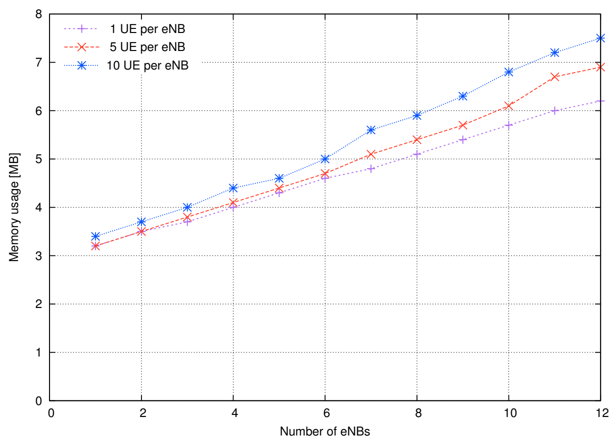

Figure 5: Memory usage of the reference program.

`/proc/{pid}/statm`. The reference hardware platform is an Intel Core2 Duo E8400 3.00GHz with 512 MB of RAM memory running a Fedora Core 10 distribution with kernel 2.6.27.5. The ns-3 build used in this experiment was configured with the options `-d optimized --enable-static`.

The results are reported in Figures 4 and 5. We note that the memory usage, as expected, primarily depends on the number of eNBs, and is in general quite low. The execution time depends significantly on both the number of eNBs and the number of UEs per eNB. For the case of 10 UEs per eNB, we also show that the experimental data can be fitted quite accurately by a quadratic function. We suggest that this behavior is due to the fact that the interference calculations, whose computational complexity is quadratic with respect to the number of eNBs, are dominant in the overall computational load.

5. CONCLUSIONS

In this paper we presented an open source product-oriented simulation model for ns-3 that is aimed at the performance evaluation of LTE SONs, discussing its design, validation and run-time performance.

As a final note, we would like to mention that development on this simulator is still ongoing, with the aim of including some functionalities which are still not supported at the moment. Among these features, we would like to mention the development of pathloss models, link error models, HARQ and MIMO support, as well as the introduction of those EPC features, such as the X2 interface, that are crucial for SONs.

6. ACKNOWLEDGMENTS

We would like to acknowledge Ubiquisys, the developer of UMTS and LTE small cells, for playing a major part in instigating and funding this project.

7. REFERENCES

[1] "Ubiquisys and CTTC Develop World's First Open Source Product-Oriented LTE Network Simulator," http://www.ubiquisys.com/femtocell-media-press-releases-id-203.htm, Feb. 2011.

[2] G. Piro, N. Baldo, and M. Miozzo, "An LTE module for the ns-3 network simulator," in *Proc. of the Workshop on NS-3 (in conjunction with SimuTOOLS 2011)*, Mar. 2011.

[3] FemtoForum, "LTE MAC Scheduler Interface Specification v1.11," http://www.femtoforum.org/femto/technical.php, Oct. 2010.

[4] "The LTE-EPC Network Simulator (LENA) project," http://iptechwiki.cttc.es/LTE-EPC_Network_Simulator_(LENA).

[5] Ofcom, "Consultation on assessment of future mobile competition and proposals for the award of 800 MHz and 2.6 GHz spectrum and related issues," Mar. 2011.

[6] RealWireless, "Low-power shared access to spectrum for mobile broadband," Ofcom Project MC/073, Tech. Rep., 2011.

[7] I. T. S. Sesia and M. Baker, *LTE - The UMTS Long Term Evolution*. Ed. Wiley, 2009.

[8] 3GPP, *Tech. Specif. Group Radio Access Network; Physical layer procedures (Release 8), 3GPP TS 36.213*.

[9] ——, *Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception (Release 8), 3GPP TS 36.101*.

[10] H. Seo and B. G. Lee, "A proportional-fair power allocation scheme for fair and efficient multiuser OFDM systems," in *Proc. of IEEE GLOBECOM*, Dallas, TX, USA, Dec. 2004.

[11] 3GPP, *Tech. Specif. Group Radio Access Network; Conveying MCS and TB size via PDCCH, 3GPP TSG-RAN WG1 R1-081483*.

[12] J. Holtzman, "CDMA forward link waterfilling power control," in *Proc. of IEEE VTC Spring*, May 2000.