

Bottlenecks in Two-Hop Ad Hoc Networks Dividing Radio Capacity in a Smart Way

A. Remke^a, B.R. Haverkort^a, L. Cloth^a

^a Design and Analysis of Communication Systems,
University of Twente,
[anne,brh,lucia]@ewi.utwente.nl

Abstract

In two-hop ad hoc networks the available radio capacity tends to be equally shared among the contending stations, which may lead to bottleneck situations in case of unbalanced traffic routing. We propose a generic model for evaluating adaptive capacity sharing strategies. We use infinite-state stochastic Petri nets for modeling the system and use the logic CSRL for specifying the measures of interest.

1 Introduction

The availability of cheap yet powerful wireless access technology, most notably the IEEE 802.11 (“wireless LAN”), has given an impulse to the development of wireless ad hoc networks. In such networks, the stations (nodes) that are in reach of each other, facilitate connectivity by forwarding traffic, e.g., to obtain access to the fixed internet. In an 802.11 ad hoc network, the stations that are in mutual reach, help each other in obtaining and maintaining connectivity. At the same time they are also competitors, as they all contend for the same resource, i.e., the shared ether as transmission medium. The medium access control of 802.11 is based on CSMA/CA and is commonly referred to as the distributed coordination function (DCF) [1, 2]. Research has shown that, effectively, the DCF tends to equally share the capacity among contending stations, thus leading to a processor sharing type of scheduling [3, 5]. Although this appears to be a nice fairness property, this fairness may lead to undesirable situations in case one of the nodes happens to function as a bridge toward either another group of nodes, or to the fixed internet as illustrated in Figure 1, in such cases of unbalanced traffic routing.

2 Earlier work

Earlier work on the performance of IEEE 802.11 ad hoc networks considers a variety of scenarios, cf. [6], however, none of the papers mentioned there, explicitly addresses the delays or throughputs in a *multihop* ad hoc network. In [6], a two-hop ad hoc network is considered, where the second hop has to forward the traffic of many sources (the first hops), thus forming a bottleneck, since all active stations have to share the transmission capacity. [6] yields explicit (closed-form) equations for the expected overall delay and the expected delay at the bottleneck, by translating the model at hand into a generalized processor sharing queuing model, as studied by Cohen [7]. Although the analysis is approximate, good results are obtained, as confirmed by simulations. However, this evaluation approach is limited in that it only allows for an equal sharing of transmission capacity between active stations.

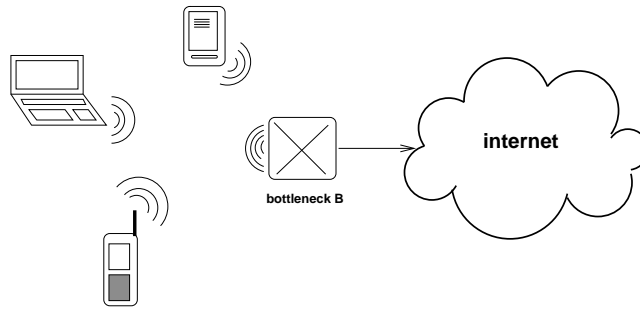


Figure 1: Wireless ad hoc network connected to the fixed internet

3 Our recent work

We follow the same line of modeling as in [6], however, we allow for alternative capacity sharing strategies as well; such strategies are made possible through the recent QoS-extension of the IEEE 802.11 standard, e.g., through the EDCA (“E”) version [8]. In doing so, we can study the impact of adaptive capacity sharing strategies that recognize potential bottlenecks and adapt accordingly.

In [10] we already modeled and analyzed two alternative capacity sharing strategies. To achieve flexibility in modeling and at the same time more modeling convenience, we specified our models as infinite-state stochastic Petri nets (iSPNs) [9]. The underlying infinite-state Markov chain, which can be automatically generated from the iSPN, then obeys a quasi-birth-death (QBD) structure [9]. Equipping these QBDs with rewards and using the logic CSRL (continuous stochastic reward logic [11]) to specify measures, we analyzed these alternative capacity sharing strategies using new model checking algorithms for CSRL on QBDs. The two adaptive capacity sharing strategies as analyzed in [10] are the so-called *buffer-related threshold (BRT)* and *source-related threshold (SRT)*.

The BRT model distinguishes two modes: in the *low occupancy* mode the buffer content is less than a given value τ and the bottleneck node and each active source receives an equal share of the radio capacity C . BRT enters the *high occupancy* mode whenever there are at least τ packets waiting in the buffer. The bottleneck node then receives 50% of the radio capacity and all active sources equally share the remaining 50%.

SRT distinguishes three modes: in the *start up* mode, when less than m sources are active, the bottleneck node and each active source receives an equal share of the radio capacity. As soon as at least m sources are active, SRT enters the *run* mode and the bottleneck node receives m times as much radio capacity as every single active source. When the number of active sources drops again below m , SRT enters the *clearance* mode where the allocation of radio capacity stays the same as in the *run* mode, to assure that the buffer will be emptied completely. When the buffer is completely empty, SRT switches back to *start up* mode.

4 A framework for adaptive capacity sharing models

Modeling adaptive capacity sharing strategies as QBDs provides more flexibility than that has been explored in [10]. In essence, our modeling approach allows us to specify and analyze a whole range of adaptive capacity sharing strategies, that can be characterized along three

dimensions:

1. The *number of modes* specifies how many different allocation regimes will be distinguished.
2. A corresponding number of *threshold values* specifies when a change of operational model is in place. Furthermore, it should be specified whether the threshold is to be reached from above or below, and whether it is buffer- or source-related.
3. We have to decide on a *resource allocation in the different modes*.

The earlier models we proposed, i.e., BRT and SRT, as well as the original processor sharing model are special cases of the above. The above approach allows, however, for more general sharing techniques. The number of modes is, in principle, unlimited, however, distinguishing few modes only does well in practice. Furthermore, in case of even more modes, selecting the threshold values and radio capacity allocation shares will be a more difficult problem (too many combinations possible to sensibly choose from).

From our study [10], it appears that an allocation of half of the radio capacity to the bottleneck and the other half to the active sources is very fair: it benefits the bottleneck and has little impact on the number of active sources. It will be interesting to combine this fixed resource allocation with a source-related threshold and to study whether there are still better fixed resource allocations. As our work on the SRT model has shown [10], such adaptive capacity sharing strategies are not easy to understand and their performance can be rather counter-intuitive. Thus, evaluating a wide variety of adaptive capacity sharing strategies seems to be essential to come up with an effective design.

The model specification (as above) is done separately from the specification of the measures of interest; for the latter purpose we use the logic CSRL. In doing so, it is straightforward to compute the packet length distribution in the bottleneck station, the fraction of time spent in the various modes, as well as the distribution of the amount of work done (the number of packets handled by the bottleneck) in a given interval of length t . The general model checking procedure that we advocate allows us to compute these measures almost completely automatically.

Model checking CSRL formulas for the mentioned measures relies on algorithms for the computation of the steady-state probabilities, the transient probabilities, the expected reward and the distribution of the accumulated reward. The most involved algorithms are required for the accumulated reward distribution: they have a time complexity that is cubic in the number of considered QBD states and grows at least quadratically with increasing accuracy [4].

In the final paper (and workshop presentation) we will present a variety of (new) capacity sharing models and evaluate their (relative) performance, from which we conclude with recommendations for future capacity sharing techniques in ad hoc networks.

Bibliography

- [1] J. Kurose and K. Ross, *Computer Networking*. Addison-Wesley, 2005.
- [2] J. Schiller, *Mobile Communications*. Addison-Wesley, 2003.

- [3] G. Bianchi, “Performance analysis of the IEEE 802.11 Distributed Coordination Function,” *IEEE Journal on Selected Areas in Communications* **18**: 535–547, 2000.
- [4] L. Cloth and B. R. Haverkort.”Five performability algorithms: A comparison.” *Markov Anniversary Meeting*: 39–54. Bosen Books, 2006.
- [5] R. Litjens, R. Roijers, J.L. van den Berg, R. Boucherie, and M. Fleuren, “Analysis of flow transfer times in IEEE 802.11 wireless LANs,” *Annals of Telecommunications* **59**: 1407–1432, 2004.
- [6] J.L. van den Berg, M. Mandjes, and F. Roijers, “Performance modeling of a bottleneck node in an IEEE 802.11 ad-hoc network,” in *ADHOC NOW*, 2006.
- [7] J. Cohen, “The multiple phase service network with Generalized Processor Sharing,” *Acta Informatica* **12**: 245–284, 1979.
- [8] IEEE 802.11e/D13.0, “Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS),” *draft supp. to IEEE 802.11 std.*, 2005.
- [9] A. Ost, *Performance of communication Systems*. Springer, 2001.
- [10] A. Remke, B.R. Haverkort, and L. Cloth, “A versatile infinite-state Markov reward model to study bottlenecks in 2-hop ad hoc networks,” in *QEST '06*: 63–72. IEEE Computer Society: , 2006.
- [11] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen, “On the logical characterisation of performability properties,” *LNCS* **1853**: 780–792, 2000.