Performance Evaluation 70 (2013) 889–899

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

A simple index rule for efficient traffic splitting over parallel wireless networks with partial information

J.W. Bosman [a,*], G.J. Hoekstra [b,a], R.D. van der Mei [a,c], S. Bhulai [c]

[a] CWI, Stochastics, Amsterdam, The Netherlands
[b] Thales, Innovation Research & Technology, Huizen, The Netherlands
[c] VU University Amsterdam, Department of Mathematics, The Netherlands

ARTICLE INFO

Article history:
Available online 29 August 2013

Keywords:
Traffic splitting
Processor sharing
Concurrent access
Flow-level performance
Partial information model
Heuristic

ABSTRACT

Multi-path communication solutions provide a promising means to improve the network performance in areas covered by multiple wireless access networks. Today, little is known about how to effectively exploit this potential. We study a model where flows are transferred over multiple parallel networks, each of which is modeled as a processor sharing node. The goal is to minimize the expected transfer time of elastic data traffic by smartly dispatching the flows to the networks, based on partial information about the numbers of foreground and background flows in each of the nodes. In the case of full state information, the optimal policy can be derived via standard MDP-techniques, but for models with partial information an optimal solution is hard to obtain. An important requirement is that the splitting algorithm is efficient, yet simple, easy-to-implement, scalable in the number of parallel networks and robust against changes in the parameter settings. We propose a simple index rule for splitting traffic streams based on partial information, and benchmark the results against the optimal solution in the case of full state information. Extensive simulations with real networks show that this method performs extremely well under practical circumstances for a wide range of realistic parameter settings.

1. Introduction

Today, many wireless networks have already closely approached the Shannon limit on channel capacity, leaving complex signal processing techniques room for only modest improvements in the data transmission rate [1]. A powerful alternative to increase the overall data rate then becomes one in which multiple, likely different, networks are used concurrently because (a) the spectrum is regulated among various frequency bands and corresponding communication network standards, and (b) the overall spectrum usage remained to be relatively low over a wide range of frequencies [2]. The concurrent access to multiple networks simultaneously has opened up enormous possibilities for increasing bandwidth, improving reliability, and enhancing Quality of Service (QoS) in areas that are covered by multiple wireless access networks. Currently, the efficient use of multiple networks concurrently is an active area of both research [3] and standardization efforts [4]. However, despite the enormous potential for quality improvement, only little is known about how to fully exploit this potential. This raises the need for new splitting algorithms for concurrent access that are simple, easy to implement, yet effective.

The effectiveness of splitting data traffic streams is generally believed to increase when detailed information about the state of the system (e.g., the number of flows, measured round-trip times and the network load) is available. In practice,

* Corresponding author.
E-mail address: J.W.Bosman@cwi.nl (J.W. Bosman).

0166-5316/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.peva.2013.08.016

however, there is often no such detailed information available, or at best only some coarse-grained and aggregated statistics. Consequently, a key challenge is to achieve efficient network utilization levels and good end-user application performance, based on information that is only partially available. At the same time, for the practical usefulness the splitting algorithms are required to be simple, easy-to-implement, scalable in the number of access networks and robust against changes in the parameter settings. We emphasize the importance of this requirement: 'optimal' splitting algorithms based on idealized situations where all detailed information is available – if possible at all – are often too complicated to be practically feasible.

Processor Sharing (PS) models provide a powerful means to model the bandwidth sharing behavior of elastic traffic streams in TCP-based data networks. A particularly attractive feature of these models is that they abstract from the complex packet-level details of the network, but at the same time maintain the essential factors that determine the data transfer-time performance of elastic data flows. Moreover, the theory of PS models is well-matured and has been successfully applied to model the flow-level behavior of a variety of communication networks, including CDMA 1xEV-DO [5], WLAN [6], UMTS-HSDPA [7] and ADSL [8]. In [6], an analytic flow-level model was presented that explicitly translates the complex and detailed packet-level dynamics of the FTP/TCP/IP-stack over a WLAN into a $M/G/1$-PS model for the flow-level performance of data transfers.

In the literature, a variety of fundamental and applied studies have been focused to the splitting and scheduling flows to multiple nodes. The available results and techniques are outlined below. In the context of telecommunication systems, the concurrent use of multiple network resources in parallel was already described for a Public Switched Digital Network (PSDN) [9], where inverse multiplexing was proposed as a technique to perform the aggregation of multiple independent information channels across a network to create a single higher-rate information channel. Various approaches have appeared to exploit the multiple transmission paths in parallel. For example, by using multi-element antennas, as adopted by the IEEE 802.11n standard [10], at the physical layer or by switching datagrams at the link layer [11,12], and also by using multiple TCP sessions in parallel to a file server [13]. In the latter case, each available network transports part of the requested data in a separate TCP session. Previous work has indicated that downloading from multiple networks concurrently may not always be beneficial [14], but in general significant performance improvements can be realized [15–18]. Under these circumstances of using a combination of different network types, in particular, the transport layer-approaches, have shown their applicability [16,18] as they allow appropriate link layer adaptations for each TCP session.

In a queueing-theoretical context, only few papers study partial information models. Bellman [19] was the first to study decision problems with a transition law that is not completely known. He observed that the problem could be transformed into an equivalent full observation problem by augmenting the state space with the set of probability distributions defined on the domain of the unknown quantity (i.e., the unobserved state, or the unknown parameter) and updating it by Bayes' rule. The transformation of the partial information problem to the complete information model, however, comes with added computational difficulties, since policies are defined over a continuum of states. This is the fundamental problem in developing algorithms for computing optimal policies [20]. There is some work in the theoretical domain to characterize the structure of the optimal policy (see, e.g., [21–24]). Even then, finding the optimal policy computationally for a general Bayesian decision problem is intractable. Approaches dealing with this are to be satisfied with suboptimal solutions or to develop algorithms that can exploit problem characteristics (see, e.g., [25–30]). We refer to [31–34] for some surveys on computational techniques.

In the present paper, we study a model consisting of N non-identical parallel networks that are modeled as PS nodes that serve $N+1$ streams of (file) flows. Node i has processing speed C_i. Stream 0 is called the foreground stream, and streams $1, \ldots, N$ are called the background streams. Flows of background stream i are served exclusively at PS node i. Each flow of the foreground stream has to be routed to exactly one of the PS nodes by the dispatcher. Flow sizes are assumed to be exponentially distributed. The goal is to develop a dynamic dispatching policy that minimizes the expected sojourn time of foreground flows based on information about the numbers of foreground and background flows at each of the PS nodes. Based on practice, we assume that the dispatcher is not able to distinguish the number of foreground and background flows in the network, but instead only has information about the *total* number of flows.

The model under consideration (see Section 2 below for details) was also studied in [35], where we addressed this problem through a learning mechanism, in which the dispatcher makes a statistical inference on the distribution of the numbers of foreground and background flows after the each decision. This Bayesian splitting algorithm in [35] was found to be highly effective in dealing with partial information, and its performance was found to be close to the performance of the optimal policy under full state information. However, the Bayesian approach has two main drawbacks: (1) the method is quite complicated and requires in-depth knowledge about stochastic models, which limits its practical usefulness, and (2) the method is not scalable in the number of parallel access networks, N, because it needs the full-state information MDP solution, which suffers from the curse of dimensionality. This limits the applicability of the Bayesian methods due to memory constraints.

The method presented in this paper is a simple index rule called the convex combination (CC) method. The CC method comprises a convex combination of techniques that are found to work well in different extreme cases: (1) the Weighted Join Shortest Queue (WJSQ) policy that routes the foreground flow arrivals to the node where the total number of flows, normalized by the node speed, is minimized, and (2) the Conditional Sojourn Time (CST) approach where the expected sojourn time, conditioned on the total numbers of flows at each of the networks, is minimized. The WJSQ policy is particularly effective when the foreground traffic load tends to saturate the nodes, whereas the CST policy is expected to perform well

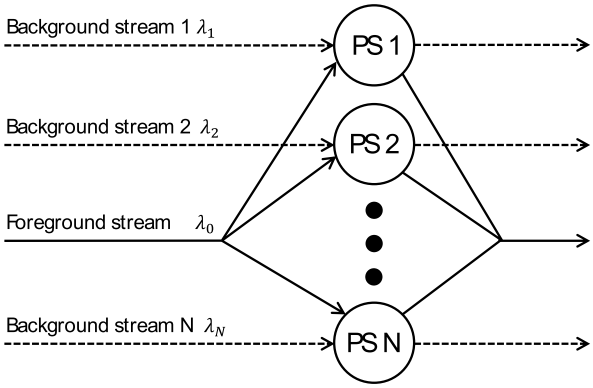

Fig. 1. The model.

in systems with low load, or low foreground load situations. The interpolating factor, denoted α ($0 < \alpha < 1$), represents the ratio of the foreground load and the remaining amount of capacity. When $\alpha \approx 0$ the CST policy is expected to perform well, whereas for $\alpha \approx 1$ the WJSQ policy is expected to perform well. To assess the effectiveness of the CC method, we have performed extensive simulation experiments in a real network simulator, called OPNET [36], that implements the full wireless protocols stack. The results show that the CC method leads to close-to-optimal performance for a wide range of realistic parameter settings.

We emphasize that the main contribution of this paper lies in (1) its practical usefulness, providing a simple but very effective means to (near-)optimally split elastic traffic streams over wireless networks based on the limited information about the state of the system, (2) its scalability with respect to the number of parallel access networks, and (3) the fact the efficiency of the splitting approach is extensively validated by a wide range of real network simulations (rather than simplified queueing simulations) implementing the complex dynamics of full wireless protocol stacks.

The organization of the paper is as follows. In Section 2 we describe the model and introduce the notation. In Section 3 we discuss the full-state information model and present our simple index-rule based heuristic. In Section 4 we discuss the results of extensive numerical evaluation of the heuristic in realistic network simulations with OPNET [36], where full wireless protocol stack is implemented.

2. Model

We study a model consisting of N non-identical parallel networks that are modeled as PS nodes, depicted in Fig. 1 that serve $N+1$ streams of flows (we refer to [6] for details on the validation and the parameterization of PS models for modeling wireless networks). Stream 0 is called the *foreground* stream, and streams $1, \ldots, N$ are called the *background* streams. From each stream flows arrive according to a Poisson process with arrival rate λ_i ($i = 0, 1, \ldots, N$). Flows from background stream i are served exclusively at PS node i. Each flow from the foreground stream has to be dispatched to one of the PS nodes on the basis of information on the *total* number of flows (thus, number of foreground flows *plus* the number of background flows) at each of the nodes, such that the expected sojourn time $E[S_0]$ for an arbitrary foreground flow is minimized. Flow sizes are assumed to be exponentially distributed with rate μ, and each node has processing speed C_i, so that server i can handle $C_i\mu$ flows per time unit. Without loss of generality, the node capacities are normalized such that $C_1 + \cdots + C_N = 1$. For each node i, the offered background load is given by $\rho_i := \lambda_i/C_i\mu$ ($i = 1, \ldots, N$), and the foreground load is $\rho_0 := \lambda_0/\mu$. Considering all arriving flows, the total offered load is given by $\rho := \rho_0 + \sum_{i=1}^{N} \rho_i C_i$. For stability reasons, we assume $\rho < 1$. The fraction foreground load compared to the total load is denoted by $\beta := \rho_0/\rho$.

In general, for each given splitting policy that bases its routing decision on the full state information, the model can be described as a CTMC with state space $\mathcal{S} = \mathbb{N}_0^{2N}$, where a state $s \in \mathcal{S}$ can be written as $s = (\bar{x}, \bar{y})$, with $\bar{x} = (x_1, \ldots, x_N)$ the number of foreground flows on the nodes and $\bar{y} = (y_1, \ldots, y_N)$ the number of background flows. In this paper, it is assumed that the dispatcher has only access to *partial information* in the sense that it has knowledge of $\bar{n} := \bar{x}+\bar{y}$, i.e. the *total* number of flows on each of the nodes. Recall that in the case of full state information the dispatcher has knowledge of $(\bar{x}, \bar{y})$. Based on the above information, there is a central decision maker that has to decide on the distribution of the foreground flows over the N servers. In doing so, the aim is to have a decision policy that minimizes $E[S_0]$, where S_0 is the sojourn time of an arbitrary foreground flow in the system.

3. Splitting algorithms

In this section we describe a number of splitting algorithms, which will be evaluated in the next section. In Section 3.1 we describe the MDP model for the case of full state information (see [37] for details on MDP's), which will be used as a benchmark to assess the efficiency of the index rule for the case of partial information, which is discussed in Section 3.2.

3.1. Full state information

In this subsection we assume that the dispatcher has full state information, and formulate the optimal dispatching problem as a Markov decision process (MDP). More specifically, the dispatching decisions are based on the state description $(\bar{x}, \bar{y})$. Let $\mathcal{I} = \{1, \ldots, N\}$ be the set of nodes and $\mathcal{A} = \{1, \ldots, N\}$ be the set of actions, where the action i means that the dispatcher forwards the flow to node $i \in \mathcal{A}$. The goal is to minimize the total expected response time by minimizing the total number of active foreground flows. Note that the MDP will not directly obtain the expected sojourn time for foreground flows but by using Little's Law, $\lambda_0 \mathbb{E}[S_0] = \mathbb{E}[N_0]$, we can obtain the average response time from $\mathbb{E}[N_0]$, the average number of foreground flows. The reward function, corresponding to the total number of foreground flows, is defined as $r(\bar{x}, \bar{y}) = x_1 + \cdots + x_N$, where $(\bar{x}, \bar{y}) \in \mathcal{S}$. Let $V(\bar{x}, \bar{y})$ be the value function operator and g be the long term expected average reward. In this case g/λ_0 corresponds to the expected response time for the requests that can be obtained using backward recursion, defined by the following equations (see [37] for details):

$$V(\bar{x}, \bar{y}) + g = \min_{i \in \mathcal{A}} \{\lambda_0 V(\bar{x} + \bar{e}_i, \bar{y})\} \tag{1a}$$

$$+ \sum_{i=1}^{N} y_i + \lambda_i V(\bar{x}, \bar{y} + \bar{e}_i) \tag{1b}$$

$$+ \sum_{i=1}^{N} \mu C_i \frac{x_i}{x_i + y_i} V(\bar{x} - \bar{e}_i, \bar{y}) \tag{1c}$$

$$+ \sum_{i=1}^{N} \mu C_i \frac{y_i}{x_i + y_i} V(\bar{x}, \bar{y} - \bar{e}_i). \tag{1d}$$

In the backward recursion (1a) corresponds to the foreground flow arrivals that have to be optimized, (1b) corresponds to arrivals of cross-traffic flows, (1c) corresponds to departures of foreground flows, and (1d) corresponds to departures of cross traffic flows.

Applying the backward recursion results in an optimal policy $R^* \in \mathcal{A}^{|\mathbb{S}|}$. An optimal policy contains optimal decisions depending on the number of flows on the PS-nodes. For each state $s = (\bar{x}, \bar{y})$, the policy found by the backward recursion $R^*(s) \in \mathcal{A}$ will give an optimal action.

3.2. Partial state information

In this section we propose a heuristic policy for near-optimal dispatching in the case of partial information, i.e. the dispatcher only has knowledge of the total numbers of (foreground plus background) flows at each node. The policy is based on the combination of two policies that perform well on complementary sets of parameter combinations (see also the discussion below): (1) the Weighted Join the Shortest Queue (WJSQ) policy, and (2) the Conditional Sojourn Time (CST) policy.

The WJSQ policy routes an arriving foreground flow to the node where the total number of flows (normalized by the node speed) is minimal. Thus, the WJSQ forwards an incoming foreground flow to node i^*, such that

$$\gamma_{i^*}^{(\mathrm{WJSQ})} = \min\left\{\gamma_1^{(\mathrm{WJSQ})}, \ldots, \gamma_N^{(\mathrm{WJSQ})}\right\}, \quad \text{where } \gamma_i^{(\mathrm{WJSQ})} := \frac{n_i}{C_i} \ (i = 1, \ldots, N). \tag{2}$$

In other words, the WJSQ routes foreground flows to the node with the smallest n_i/C_i ratio. Ties are randomized according to the ratio of node speeds (i.e. $C_1 : C_2 : \cdots : C_N$). The WJSQ may be expected to work particularly well when the total load to the system is large (i.e., $\rho \approx 1$) and the foreground load represents a significant fraction of the total load offered to the system (i.e., $\beta \approx 1$).

The CST approach routes an incoming flow to the node for which the expected sojourn time, conditioned on the fact that there are n_i other flows at node i at that moment, is minimal. Using a well-known result for the conditional expected sojourn time in an $M/M/1$-PS queue [38], the CST policy forwards an incoming foreground flow to node i^*, such that

$$\gamma_{i^*}^{(\mathrm{CST})} = \min\left\{\gamma_1^{(\mathrm{CST})}, \ldots, \gamma_N^{(\mathrm{CST})}\right\}, \quad \text{where } \gamma_i^{(\mathrm{CST})} := \frac{n_i + 2}{2\mu C_i - \lambda_i} \ (i = 1, \ldots, N). \tag{3}$$

The CST approach may be expected to work well if the foreground load is negligible compared to the total load (i.e., $\beta \approx 0$). If the foreground load is large, then the dynamic decision making will induce a correlation between the number of flows in a PS node and the combined arrival process into that node, which leads to a violation of the Poisson assumption that underlies (3).

Both the WJSQ and the CST policies generate a switching curve given the total number of flows n_i on each node. We aim to develop a method that works well for the whole range of foreground and background load values. To this end, we propose

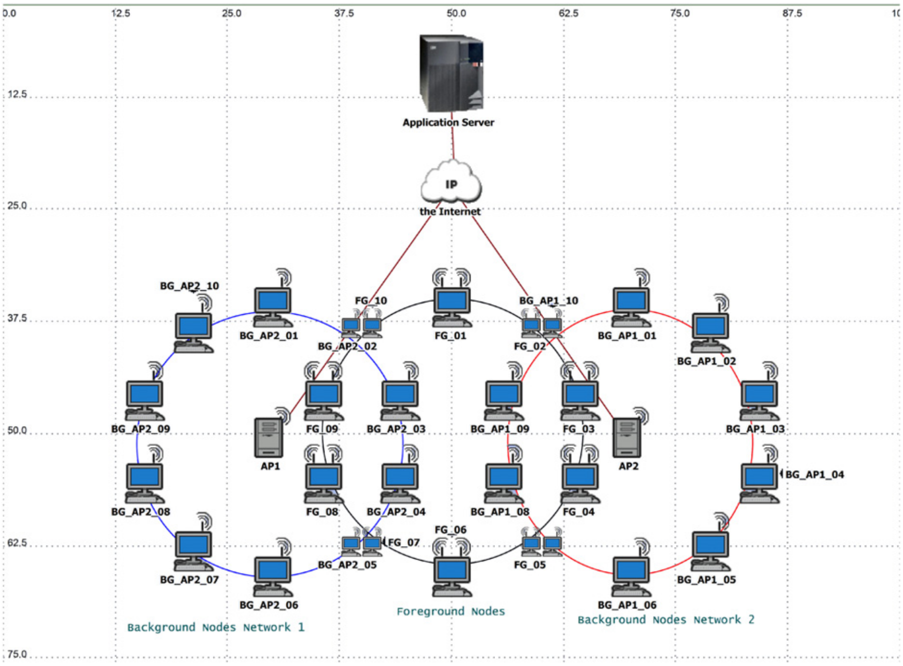

Fig. 2. The experimental setup.

a method where both switching curves are combined using a convex combination of these curves. The convex combination (CC) approach forwards an incoming foreground flow to node i^*, such that

$$\gamma_{i^*}^{(CC)} = \min\left\{\gamma_1^{(CC)}, \ldots, \gamma_N^{(CC)}\right\}, \quad \text{where } \gamma_i^{(CC)} := \alpha\frac{n_i}{C_i} + (1-\alpha)\frac{n_i+2}{2\mu C_i - \lambda_i}, \tag{4}$$

and α $(0 \le \alpha \le 1)$ is given by:

$$\alpha := \frac{\rho_0}{\sum_{i=1}^{N} C_i(1-\rho_i)} \quad (i = 1, \ldots, N). \tag{5}$$

Thus, the CST method is expected to work well when $\alpha \approx 0$, whereas the WJSQ method is expected to work well when $\alpha \approx 1$. In the next section the CC-approach defined in (4)–(5) will be evaluated.

For sake of completeness, we also give a brief sketch of the Bayesian partial information approach proposed in [35]. The basic idea of this approach is that one keeps track of so-called *belief states*. Based on the (partial) knowledge of the total number of flows at each of the nodes (i.e., $\overline{n} = \overline{x} + \overline{y}$), and observations on recent arrivals and departures, one can estimate the conditional distribution of the full state $(\overline{x}, \overline{y})$. This distribution, in turn, is used to identify the state $(\overline{x}^*, \overline{y}^*)$ that corresponds to the maximum likelihood. This state is plugged into the solution for the full MDP, discussed briefly in Section 3.1. For more details about the Bayesian policy, we refer to [35].

In the next section, the performance of each of the policies discussed above is evaluated by simulations.

4. Numerical experiments

In Section 3 we defined index rules for efficiently assigning downloads with concurrent access based on partial state information. To assess the performance of these index rules, we have performed an extensive experimentation with a state-of-the-art network simulation package OPNET [36], using an implementation for FTP file transfers via TCP/IP over two parallel WLANs. We have performed a large number of experiments with a wide range of parameter settings. The results are outlined below.

4.1. Experimental configuration

In the experimental setup, depicted in Fig. 2 all wireless terminals download files from an application server, which may also be a dispatcher in front of several application servers (not shown). The application server has information about the number of ongoing downloads over each of the WLAN access networks, AP1 and AP2, but is unable to distinguish between

Table 1
IEEE 802.11b MAC parameters.

Parameter	Value	Parameter	Value
mac	224 bits	ack	112 bits
difs	50 μs	R_b	$\{1, 11\} \cdot 10^6$ bps
sifs	10 μs	$Cw_{\min}$	31 slots
eifs	364 μs	phy	192 μs
δ	1 μs	τ	20 μs
R_c	10^6 bps		

Table 2
Network and application settings.

Variable	Setting
X_{FTPget}	4096 bits
$X_{FTPclose}$	64 bits
TCP_{stack}	Full-featured
X_{MSS}	11,584 bits
$X_{tcp/ip}$	416 bits
w	70,080 bits (8760 bytes)
X_{file}	$1.6 \cdot 10^6$ bits

the multi-homed and the single homed terminals, because there is no binding between both network addresses of the multi-homed terminals. Both WLAN access points operate on non-overlapping frequency channels to establish two non-interfering parallel paths to the application server from the multi-homed systems. The transmission links from the access points towards the application server are considered to incur a negligible delay and loss to packets from and to the access points. This assumption is motivated by the much higher capacities and reliability offered in contemporary fixed-line carrier-grade Internet connections in comparison to the IEEE 802.11b access networks. The analytic model from [6] captures the combined dynamics and protocol overhead of the 802.11 MAC, IP, TCP and application-layer into an explicit expression for the effective service time of a file download. Based on the effective service time, the effective load can be determined of the file transfers in our simulated WLAN networks with a flow-level $M/G/1$ PS model.

In the simulated network there are ten multi-homed terminals (named $FG\_01 - FG\_10$) that generate download requests (that are considered foreground flows in the queueing model) with arrival rate λ_0. These foreground terminals are positioned between both access points in a circle with a radius of 15 m. In addition there are ten single-homed terminals (named with prefix $BG\_AP1\_$) that generate background traffic in network 1 with file downloads arriving with rate λ_1 to the first network. The remaining ten single-homed terminals (named with prefix $BG\_AP2\_$) generate background traffic at rate λ_2 in network 2 in a similar fashion. All background terminals are positioned at an equal distance of 15 m from their respective access point. The file download requests arrive according to an independent Poisson process and may have multiple file transfers in progress.

The MAC/PHY parameters of the WLAN stations are set in accordance to the widely deployed IEEE 802.11b standard amendment as it relies on the same MAC protocol basis as the contemporary higher rate (IEEE 802.11 $a/g/n$) amendments and has lower computational requirements for high-load network simulations. Table 1 summarizes the IEEE 802.11 MAC parameters used in our analytic model to calculate the effective load values for the simulation runs.

In this table, mac is the number of bits of overhead bits associated to a MAC data frame. The difs, sifs, eifs are the DCF, short and extended interframe spacing times, respectively. The δ is the propagation delay that is assumed in our analytic model. R_c is the transmission rate for WLAN acknowledgments of size ack bits, and R_b is the WLAN transmission rate for MAC data frames that is set to 1 or 11 Mbps. $Cw_{\min}$ corresponds to the minimum contention window in slots. Phy is the physical layer overhead, and τ is the slot time. In addition to the WLAN MAC, specific settings apply to the higher protocol layers and are outlined in Table 2.

In Table 2, X_{FTPget} is the size of the FTP GET-command that is issued for initiating a file download, $X_{FTPclose}$ is the size of the FTP CLOSE-command that concludes the file transfer at the application. The TCP stack used in our experiments is characterized in OPNET as 'Full-Featured', which is an enhanced version of TCP Reno that uses Selective Acknowledgments (SACK) [39] and has a slightly smaller MSS, X_{MSS} (in bits), due to the use of timestamps to fit in the 1500 bytes that are used as the WLAN data frame payload. The number of TCP/IP overhead bits per segment is $X_{tcp/ip}$ bits. The maximum TCP receiver window size is indicated as w (in bits), and the file size as X_{file} (in bits). Based on the parameter setting from Tables 1 and 2 and respecting the engineering guidelines from [6] we can assume that the mean download response times in our simulation model can be accurately predicted from the effective load of the network using the $M/G/1$ PS model.

4.2. Experimental results

In our simulation study we have considered three scenarios. For the first two scenarios the OPNET simulations for the experimental results have been run with approximately 322,000 foreground flows and the background flows ranging from

Table 3
Comparison of ($\mathbb{E}[S_0|\text{JSQ}]$, $\mathbb{E}[S_0|\text{full MDP}]$, $\Delta\%$) in OPNET for $\rho_0 = 0.1$.

ρ_1	ρ_2				
	0.1	0.3	0.5	0.7	0.8
0.1	(0.356, 0.354, 0.6%)	(0.380, 0.369, 3.0%)	(0.401, 0.385, 4.2%)	(0.419, 0.400, 4.7%)	(0.422, 0.406, 4.0%)
0.2		(0.402, 0.395, 1.8%)	(0.435, 0.420, 3.7%)	(0.463, 0.446, 3.8%)	(0.471, 0.455, 3.5%)
0.3		(0.422, 0.421, 0.4%)	(0.469, 0.458, 2.3%)	(0.513, 0.498, 2.9%)	(0.533, 0.519, 2.8%)
0.4			(0.506, 0.501, 1.0%)	(0.574, 0.564, 1.8%)	(0.610, 0.599, 1.9%)
0.5			(0.548, 0.547, 0.0%)	(0.649, 0.639, 1.6%)	(0.706, 0.696, 1.5%)
0.6				(0.744, 0.737, 1.0%)	(0.840, 0.828, 1.5%)
0.7				(0.867, 0.865, 0.2%)	(1.030, 1.018, 1.2%)
0.8					(1.319, 1.319, 0.0%)

Table 4
Comparison of ($\mathbb{E}[S_0|\text{CC}]$, $\mathbb{E}[S_0|\text{full MDP}]$, $\Delta\%$) in OPNET for $\rho_0 = 0.1$.

ρ_1	ρ_2				
	0.1	0.3	0.5	0.7	0.8
0.1	(0.356, 0.354, 0.6%)	(0.371, 0.369, 0.7%)	(0.386, 0.385, 0.2%)	(0.401, 0.400, 0.4%)	(0.408, 0.406, 0.5%)
0.2		(0.397, 0.395, 0.5%)	(0.422, 0.420, 0.5%)	(0.447, 0.446, 0.1%)	(0.459, 0.455, 0.9%)
0.3		(0.422, 0.421, 0.4%)	(0.460, 0.458, 0.4%)	(0.501, 0.498, 0.6%)	(0.523, 0.519, 0.8%)
0.4			(0.503, 0.501, 0.4%)	(0.568, 0.564, 0.8%)	(0.599, 0.599, 0.1%)
0.5			(0.548, 0.547, 0.0%)	(0.644, 0.639, 0.8%)	(0.702, 0.696, 0.9%)
0.6				(0.738, 0.737, 0.1%)	(0.835, 0.828, 0.8%)
0.7				(0.867, 0.865, 0.2%)	(1.022, 1.018, 0.4%)
0.8					(1.319, 1.319, 0.0%)

roughly 644,000 flows to 5.1 million flows depending on the load. The first simulation scenario considers equal capacity networks in which all terminals are configured to use a WLAN transmission rate of 11 Mbps. For simulating a scenario in which the network capacity of both access network is unequal, the WLAN transmission rate used in AP2 is lowered to 1 Mbps, which reduces the medium capacity for processing file transfers by a factor of 5.79. In this second scenario, the background load applied to AP2 is based on the lower capacity, whereas the foreground traffic intensity remains the same as for the equal capacity network. The third scenario considers again unequal capacities for the two networks while the fraction of foreground load is varied and the total load is kept fixed. We have executed 48 runs for the equal first scenario (24 runs for the fully observed MDP and 24 runs for the heuristics) and 80 runs for the second scenario. In the third scenario we have performed $4 \times 7 \times 3$ (84) runs. All runs of the first and second scenario have completed a total simulation time of 300 h per run of which 1 h is the warm-up time leading to a wall clock time of approximately 75 h per run. This experimental setup is sufficient to derive a 99% confidence interval of approximately 0.7% with respect to the point estimates. For the third scenario the total simulation time depends on the foreground traffic intensity and varies between 300 and 6000 h, and is sufficient to derive a 99% confidence interval of approximately 4.7%.

To assess the efficiency of the different partial-information policies, we have simulated the mean transfer time of an arbitrary foreground flow, $E[S_0]$, for different policies, and compare the outcome to the full MDP case. For given policy π and benchmark comparison policy π', the relative error is defined as follows:

For $\pi, \pi' \in \{\text{WJSQ, CST, CC, Bayes, full MDP}\}$,

$$\Delta\% = \frac{\mathbb{E}[S_0|\pi] - \mathbb{E}\left[S_0|\pi'\right]}{\mathbb{E}[S_0|\pi']} \times 100\%. \tag{6}$$

4.2.1. The case of equal capacities

We first consider the case where both access networks have the same (normalized) capacity, i.e., $C_1 = C_2$. The results of the experiments are outlined in Tables 3–5, for $\rho_0 = 0.1$ and a number of combinations ρ_1 and ρ_2. Tables 3 and 4 show the results for the (W)JSQ and the CC policies, benchmarked against the full MDP policy. Table 5 shows a comparison between the CC policy and the Bayesian policy [35]. Note that the parameter values are obtained according to the parameterization as defined and validated in [6].

The results in Table 3 show that the JSQ policy performs quite well, with a maximum error up to 4.7%. However, the results in Table 4 show that the CC policy strongly outperforms JSQ, with a maximum error of 0.9%. The difference in performance between JSQ and CC manifests itself mainly when the background load values are strongly asymmetric. In those cases the JSQ policy become highly inaccurate. To illustrate this, consider a two-node system where node 1 has high background load and node 2 has low background load. If $n_1 < n_2$ then the JSQ policy will route an incoming flow T to node 1. In this situation, it may well occur that this decision is not optimal, because the sojourn time of T is likely to be stretched due to the background flow arrivals at node 1. Table 5 shows that the CC policy performs comparably well to the Bayesian policy, despite the fact that the Bayesian policy has a much higher computational complexity. We re-emphasize that the computational complexity of the CC rule is negligible.

Table 5
Comparison of $(\mathbb{E}[S|\text{Bayes}], \mathbb{E}[S|\text{CC}], \Delta\%)$ in OPNET for $\rho_0 = 0.1$.

ρ_1	ρ_2				
	0.1	0.3	0.5	0.7	0.8
0.1	(0.355, 0.356, −0.3%)	(0.370, 0.371, −0.4%)	(0.385, 0.386, −0.2%)	(0.402, 0.401, 0.1%)	(0.408, 0.408, −0.1%)
0.2		(0.396, 0.397, −0.3%)	(0.420, 0.422, −0.4%)	(0.446, 0.447, 0.0%)	(0.456, 0.459, −0.8%)
0.3		(0.421, 0.422, −0.3%)	(0.460, 0.460, −0.1%)	(0.502, 0.501, 0.0%)	(0.521, 0.523, −0.3%)
0.4			(0.503, 0.503, 0.0%)	(0.565, 0.568, −0.6%)	(0.601, 0.599, 0.3%)
0.5			(0.551, 0.548, 0.6%)	(0.643, 0.644, −0.1%)	(0.700, 0.702, −0.3%)
0.6				(0.742, 0.738, 0.6%)	(0.831, 0.835, −0.5%)
0.7				(0.867, 0.867, 0.0%)	(1.028, 1.022, 0.6%)
0.8					(1.322, 1.319, 0.2%)

Table 6
Comparison of $(\mathbb{E}[S_0|\text{WJSQ}], \mathbb{E}[S_0|\text{full MDP}], \Delta\%)$ in OPNET for $\rho_0 = 0.1$ with $C_1 = 1$ and $C_2 = 0.17$.

ρ_1	ρ_2				
	0.1	0.3	0.5	0.7	0.8
0.1	(1.027, 0.415, 147.5%)	(0.962, 0.415, 131.7%)	(0.865, 0.416, 108.1%)	(0.865, 0.416, 108.1%)	(0.643, 0.415, 54.7%)
0.2	(1.094, 0.474, 130.7%)	(1.025, 0.475, 115.7%)	(0.930, 0.476, 95.4%)	(0.930, 0.476, 95.4%)	(0.709, 0.475, 49.2%)
0.3	(1.741, 0.555, 213.8%)	(1.106, 0.555, 99.3%)	(1.013, 0.551, 83.7%)	(1.013, 0.551, 83.7%)	(0.787, 0.555, 41.8%)
0.4	(1.247, 0.660, 88.9%)	(1.195, 0.664, 80.0%)	(1.111, 0.666, 66.9%)	(1.111, 0.666, 66.9%)	(0.900, 0.664, 35.6%)
0.5	(1.348, 0.806, 67.1%)	(1.315, 0.818, 60.8%)	(1.252, 0.826, 51.6%)	(1.252, 0.826, 51.6%)	(1.056, 0.833, 26.8%)
0.6	(1.498, 1.013, 47.8%)	(1.499, 1.047, 43.1%)	(1.463, 1.065, 37.4%)	(1.463, 1.065, 37.4%)	(1.320, 1.099, 20.1%)
0.7	(1.714, 1.305, 31.3%)	(1.775, 1.390, 27.7%)	(1.828, 1.476, 23.8%)	(1.828, 1.476, 23.8%)	(1.789, 1.595, 12.1%)
0.8	(2.093, 1.777, 17.8%)	(2.304, 2.013, 14.4%)	(2.545, 2.273, 12.0%)	(2.545, 2.273, 12.0%)	(2.966, 2.862, 3.6%)

Table 7
Comparison of $(\mathbb{E}[S_0|\text{CC}], \mathbb{E}[S_0|\text{full MDP}], \Delta\%)$ in OPNET for $\rho_0 = 0.1$ with $C_1 = 1$ and $C_2 = 0.17$.

ρ_1	ρ_2				
	0.1	0.3	0.5	0.7	0.8
0.1	(0.418, 0.415, 0.8%)	(0.415, 0.415, 0.0%)	(0.416, 0.416, 0.2%)	(0.416, 0.416, 0.2%)	(0.415, 0.415, 0.0%)
0.2	(0.476, 0.474, 0.3%)	(0.476, 0.475, 0.1%)	(0.477, 0.476, 0.2%)	(0.477, 0.476, 0.2%)	(0.475, 0.475, 0.0%)
0.3	(0.556, 0.555, 0.3%)	(0.556, 0.555, 0.2%)	(0.555, 0.551, 0.7%)	(0.555, 0.551, 0.7%)	(0.555, 0.555, 0.0%)
0.4	(0.664, 0.660, 0.6%)	(0.666, 0.664, 0.4%)	(0.667, 0.666, 0.2%)	(0.667, 0.666, 0.2%)	(0.667, 0.664, 0.4%)
0.5	(0.809, 0.806, 0.3%)	(0.821, 0.818, 0.3%)	(0.831, 0.826, 0.6%)	(0.831, 0.826, 0.6%)	(0.834, 0.833, 0.0%)
0.6	(1.019, 1.013, 0.6%)	(1.047, 1.047, 0.0%)	(1.070, 1.065, 0.5%)	(1.070, 1.065, 0.5%)	(1.100, 1.099, 0.1%)
0.7	(1.331, 1.305, 2.0%)	(1.399, 1.390, 0.6%)	(1.477, 1.476, 0.1%)	(1.477, 1.476, 0.1%)	(1.602, 1.595, 0.4%)
0.8	(1.841, 1.777, 3.6%)	(2.023, 2.013, 0.5%)	(2.313, 2.273, 1.8%)	(2.313, 2.273, 1.8%)	(2.866, 2.862, 0.1%)

Table 8
Comparison of $(\mathbb{E}[S_0|\text{Bayes}], \mathbb{E}[S_0|\text{CC}], \Delta\%)$ in OPNET for $\rho_0 = 0.1$ with $C_1 = 1$ and $C_2 = 0.17$.

ρ_1	ρ_2				
	0.1	0.3	0.5	0.7	0.8
0.1	(0.418, 0.417, 0.4%)	(0.415, 0.415, 0.0%)	(0.416, 0.416, 0.2%)	(0.416, 0.416, 0.2%)	(0.415, 0.415, 0.0%)
0.2	(0.476, 0.476, −0.1%)	(0.476, 0.477, −0.2%)	(0.477, 0.479, −0.4%)	(0.477, 0.479, −0.4%)	(0.475, 0.475, 0.0%)
0.3	(0.556, 0.556, 0.1%)	(0.556, 0.556, −0.1%)	(0.555, 0.553, 0.4%)	(0.555, 0.553, 0.4%)	(0.555, 0.555, 0.0%)
0.4	(0.664, 0.663, 0.2%)	(0.666, 0.665, 0.1%)	(0.667, 0.667, 0.0%)	(0.667, 0.667, 0.0%)	(0.667, 0.667, 0.0%)
0.5	(0.809, 0.807, 0.3%)	(0.821, 0.819, 0.2%)	(0.831, 0.827, 0.5%)	(0.831, 0.827, 0.5%)	(0.834, 0.835, −0.2%)
0.6	(1.019, 1.016, 0.3%)	(1.047, 1.052, −0.4%)	(1.070, 1.068, 0.2%)	(1.070, 1.068, 0.2%)	(1.100, 1.100, 0.1%)
0.7	(1.331, 1.322, 0.7%)	(1.399, 1.410, −0.8%)	(1.477, 1.482, −0.4%)	(1.477, 1.482, −0.4%)	(1.602, 1.597, 0.3%)
0.8	(1.841, 1.817, 1.3%)	(2.023, 2.057, −1.6%)	(2.313, 2.299, 0.6%)	(2.313, 2.299, 0.6%)	(2.866, 2.876, −0.4%)

4.2.2. The case of unequal capacities

Let us now consider the case where the access networks have different capacities, i.e., $C_1 \neq C_2$. To this end, we consider the case $C_1 : C_2 = 1 : 0.17$. The results of the simulations experiments are outlined in Tables 6–8, for $\rho_0 = 0.1$ and a number of combinations ρ_1 and ρ_2. Tables 6 and 7 show the results for the WJSQ and the CC policies, benchmarked against the full MDP policy (similar to the results in Tables 3 and 4 for the equal capacity case). Table 8 shows a comparison between the CC policy and the Bayesian policy (similar to Table 5 in Section 4.2.1). Recall that the parameter values in this setting are obtained according to the parameterization discussed in [6].

The results in Table 6 show that the WJSQ policy is highly inefficient when the network capacities are strongly asymmetric, with error even up to over 200%. The results reveal that the WJSQ performs particularly bad in low-load scenarios. However, the results in Table 7 show that the CC policy remains to be highly efficient, even when the networks are strongly

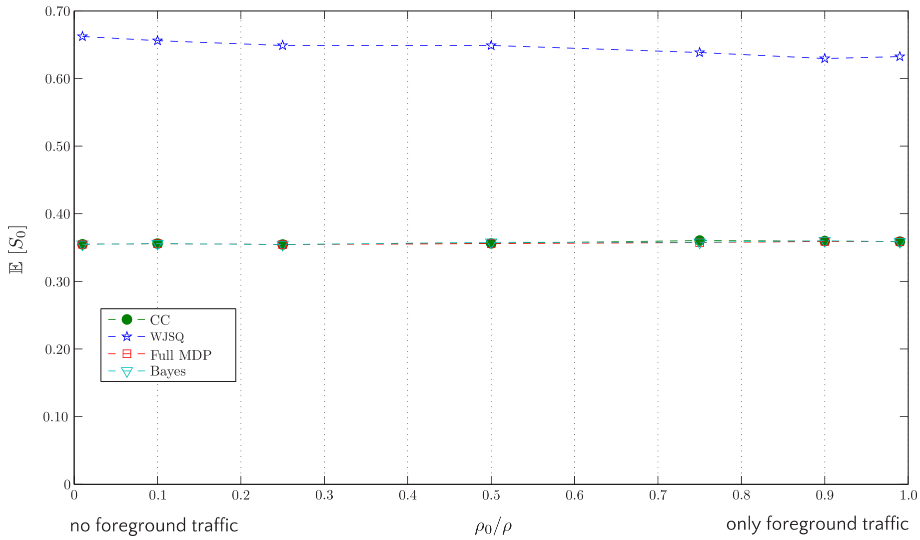

Fig. 3. Comparison of $\mathbb{E}[S_0|\text{CC}]$, $\mathbb{E}[S_0|\text{full MDP}]$, $\mathbb{E}[S_0|\text{WJSQ}]$ and $\mathbb{E}[S_0|\text{Bayes}]$ in OPNET for $\hat{\rho} = 0.65$.

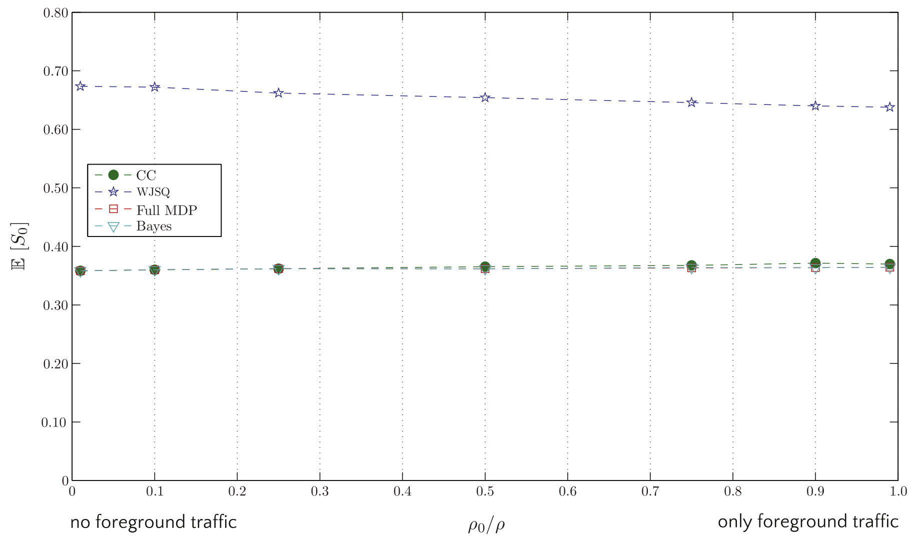

Fig. 4. Comparison of $\mathbb{E}[S_0|\text{CC}]$, $\mathbb{E}[S_0|\text{full MDP}]$, $\mathbb{E}[S|\text{WJSQ}]$ and $\mathbb{E}[S_0|\text{Bayes}]$ in OPNET for $\rho = 0.75$.

asymmetric, with a worst-case error less than 4%. Table 8 shows again that the CC policy performs comparably well to the Bayesian policy, and that both policies are highly accurate.

4.2.3. Varying the asymmetry in foreground versus background load

Finally, we check the efficiency of the splitting policies where we vary the fraction of the foreground compared to the total load. Similar to Section 4.2.2, we assume that the ratios of the network capacities are $C_1 : C_2 = 1 : 0.17$. Figs. 3–5 show the expected value of the transfer time of an arbitrary foreground flow (i.e., $E[S_0]$) as a function of the ratio $\beta = \rho_0/\rho$, where the overall load ρ is kept fixed, for each of the routing policies CC, WJSQ, full MDP and Bayes. Figs. 3–5 show the results for $\rho = 0.65$, $\rho = 0.70$ and $\rho = 0.80$, respectively.

The results in Figs. 3–5 show again that in all cases the WJSQ policy is strongly outperformed by the other policies. Moreover, we observe that the CC policy, which is based on partial information only, is extremely close to the full MDP solution, which is based on full state information. Also, we observe our simplistic index-based CC rule performs comparably well to the more complicated Bayesian policy. We re-emphasize that the importance of this observation for practical engineering purposes.

In conclusion, the experimental results demonstrate that the CC-method using partial information strongly outperforms the WJSQ policy, and even leads to close-to-optimal performance that can be obtained using the full-state information MDP. Moreover, the CC method performs equally well when compared to the Bayesian policy, which has a number of drawbacks:

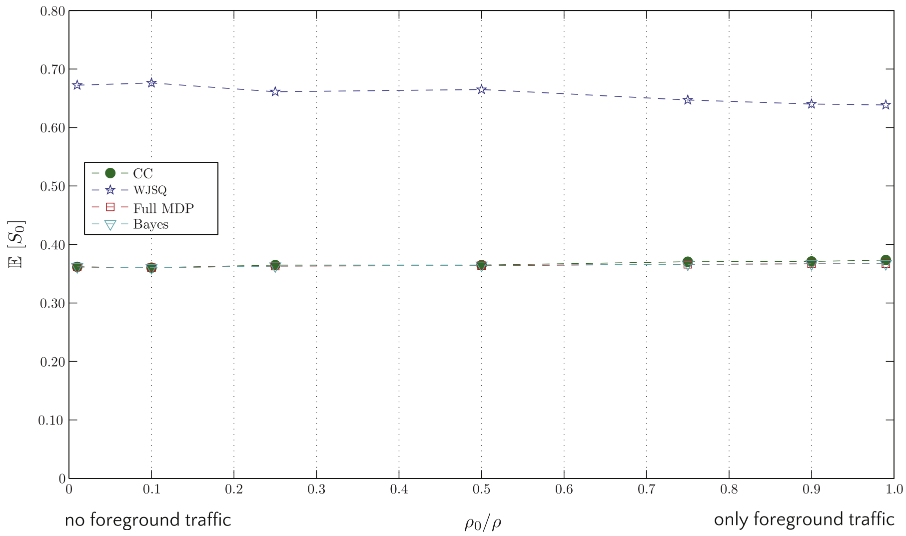

Fig. 5. Comparison of $\mathbb{E}[S_0|\text{CC}]$, $\mathbb{E}[S_0|\text{full MDP}]$, $\mathbb{E}[S_0|\text{WJSQ}]$ and $\mathbb{E}[S_0|\text{Bayes}]$ in OPNET for $\rho = 0.8$.

(1) it is inherently complicated, which limits its practical usefulness, (2) it is not scalable in the number of access networks N, because it needs the full-state information MDP solution, which suffers from the curse of dimensionality. Typically, this will limit the applicability of the Bayesian methods due to memory constraints. These observations lead to the conclusion that the CC index rule has a considerable advantage over the Bayesian approach with respect to its practical usefulness and engineering.

Acknowledgments

This work has been carried out in the context of the IOP GenCom project Service Optimization and Quality (SeQual), supported by the Dutch Ministry of Economic Affairs, Agriculture and Innovation via its agency Agentschap NL. The contribution from Gerard Hoekstra has been carried out in the context of the RRR (*R*ealisation of *R*eliable and Secure *R*esidential Sensor Platforms) project of the Dutch program *IOP Generieke Communicatie*, number IGC1020, supported by the Subsidieregeling Sterktes in Innovatie.

References

[1] D. Cox, Fundamental limitations on the data rate in wireless systems, IEEE Communications Magazine 46 (12) (2008) 16–17.
[2] FCC, Report of the spectrum efficiency working group, Technical report, Federal Communications Commission Spectrum Policy Task Force, 2002.
[3] D. Sarkar, P.D. Amer, R. Stewart, Guest editorial: concurrent multipath transport, Computer Communications 30 (17) (2007) 3215–3217.
[4] Multipath tcp (mptcp) charter, April 2011. http://datatracker.ietf.org/wg/mptcp/charter/.
[5] S.C. Borst, O.J. Boxma, N. Hegde, Sojourn times in finite-capacity processor-sharing queues, in: Proceedings NGI 2005 Conference, 2005.
[6] G.J. Hoekstra, R.D. van der Mei, Effective load for flow-level performance modelling of file transfers in wireless LANs, Computer Communications 33 (16) (2010) 1972–1981.
[7] Y. Wu, C. Williamson, J. Luo, On processor sharing and its applications to cellular data network provisioning, Performance Evaluation 64 (9–12) (2007) 892–908.
[8] J.V.L. Beckers, I. Hendrawan, R.E. Kooij, R.D. van der Mei, Generalized processor sharing models for internet access lines, in: Proceedings of IFIP Conference on Performance Modelling and Evaluation of ATM and IP networks, Budapest, 2001, pp. 101–112.
[9] J. Duncanson, Inverse multiplexing, IEEE Communications Magazine 32 (4) (1994) 34–41.
[10] IEEE Standard 802.11n, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer specifications Enhancements for Higher Throughput, October 2009.
[11] R. Chandra, P. Bahl, P. Bahl, Multinet: connecting to multiple IEEE 802.11 networks using a single wireless card, in: Proceedings IEEE INFOCOM 2004, 2004.
[12] G.P. Koudouris, R. Agüero, E. Alexandri, J. Choque, K. Dimou, H.R. Karimi, H. Lederer, J. Sachs, R. Sigle, Generic link layer functionality for multi-radio access networks, in: Proceedings 14th IST Mobile and Wireless Communications Summit, 2005.
[13] P. Rodriguez, A. Kirpal, E. Biersack, Parallel-access for mirror sites in the internet, in: Proceedings IEEE INFOCOM 2000, 2000, pp. 864–873.
[14] C. Gkantsidis, M. Ammar, E. Zegura, On the effect of large-scale deployment of parallel downloading, in: Proceedings of the Third IEEE Workshop on Internet Applications, WIAPP'03, IEEE Computer Society, Washington, DC, USA, 2003, p. 79.
[15] Y. Hasegawa, I. Yamaguchi, T. Hama, H. Shimonishi, T. Murase, Deployable multipath communication scheme with sufficient performance data distribution method, Computer Communications 30 (17) (2007) 3285–3292.
[16] G.J. Hoekstra, R.D. van der Mei, J.W. Bosman, Efficient traffic splitting in parallel TCP-based wireless networks: modelling and experimental evaluation, in: Proceedings 25th International Teletraffic Congress, ITC 2013, Shanghai, PR China, 2013.
[17] G.J. Hoekstra, F.J.M. Panken, Increasing throughput of data applications on heterogeneous wireless access networks, in: Proceedings 12th IEEE Symposium on Communication and Vehicular Technology in the Benelux, 2005.
[18] H.Y. Hsieh, R. Sivakumar, A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts, Wireless Networks 11 (1) (2005) 99–114.
[19] R. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, 1961.

[20] C.H. Papadimitriou, J.N. Tsitsiklis, The complexity of Markov decision processes, Mathematics of Operations Research 12 (3) (1987) 441–450.
[21] A.N. Burnetas, M.N. Katehakis, Optimal adaptive policies for Markov decision processes, Mathematics of Operations Research 22 (1997) 222–255.
[22] S.C. Albright, Structural results for partially observable Markov decision processes, Operations Research 27 (1979) 1041–1053.
[23] K.M. van Hee, Bayesian control of Markov chains, Ph.D. Thesis, Technical University of Eindhoven, 1978.
[24] J.A. Loeve, Markov decision chains with partial information, Ph.D. Thesis, Leiden University, 1995.
[25] M. Littman, A. Cassandra, L. Kaelbling, Learning policies for partially observable environments: schaling up, in: Proceedings of the Twelfth International Conference on Machine Learning, 1995, pp. 362–370.
[26] R. Parr, S. Russell, Approximating optimal policies for partially observable stochastic domains, in: Proceedings of the International Joint Conference on Artificial Intelligence, 1995, pp. 1088–1094.
[27] N.L. Zhang, W. Liu, Region-based approximations for planning in stochastic domains, in: Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence, 1997, pp. 472–480.
[28] M. Hauskrecht, Planning and control in stochastic domains with imperfect information, Ph.D. Thesis, Massachusetts Institute of Technology, 1997.
[29] R.I. Brafman, A heuristic variable grid solution method for POMDPs, in: Proceedings of the Fourteenth National Conference on Artificial Intelligence, 1997, pp. 727–733.
[30] A.R. Cassandra, Exact and approximate algorithms for partially observable markov decision processes, Ph.D. Thesis, Brown University, 1998.
[31] W.S. Lovejoy, A survey of algorithmic methods for partially observed Markov decision processes, Annals of Operations Research 28 (1991) 47–66.
[32] G.E. Monahan, A survey of partially observable Markov decision processes: theory, models, and algorithms, Management Science 28 (1982) 1–16.
[33] C.C. White III, A survey of solution techniques for the partially observed Markov decision process, Annals of Operations Research 32 (1991) 215–230.
[34] P.R. Kumar, A survey of some results in stochastic adaptive control, SIAM Journal on Control and Optimization 23 (1985) 329–380.
[35] S. Bhulai, G.J. Hoekstra, J.W. Bosman, R.D. van der Mei, Dynamic traffic splitting to parallel wireless networks with partial information: a Bayesian approach, Performance Evaluation 69 (1) (2012) 41–52.
[36] OPNET Technologies Inc., OPNET modeler, November 2011. http://www.opnet.com/solutions/network_rd/modeler.html.
[37] M.L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, John Wiley & Sons, 1994.
[38] B. Sengupta, D.L. Jagerman, A conditional response time of the $M/M/1$ processor-sharing queue, AT&T Technical Journal 64 (2) (1985) 409–421.
[39] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, TCP selective acknowledgment options, in: RFC 2018, Internet Engineering Task Force, 1996.

J.W. Bosman (1983) received his M.Sc. (cum laude) degree in Business Mathematics and Informatics in 2009 from the VU University Amsterdam. After his master thesis project at CWI on concurrent-access in wireless networks he started as Ph.D. at CWI in 2009. His research interests include performance and QoS modeling of ICT systems, and queueing theory. In 2010 he co-organized the 72nd European Study Group Mathematics with Industry in Amsterdam.

G.J. Hoekstra received the M.Sc. (cum laude) degree in Electrical Engineering from the University of Twente, the Netherlands and the Ph.D. degree in Applied Mathematics from VU University, Amsterdam. He has been working for Alcatel's Corporate Research Centre in Antwerp, Belgium and from 1999 until 2008 for (Alcatel-) Lucent's Bell-Labs Research department. Currently, he is working as a senior innovation engineer and project manager at the Innovation Research & Technology department in Thales, Huizen, the Netherlands and as a part-time postdoc researcher at CWI.

R.D. van der Mei is the head of the research cluster Probability, Networks and Algorithms (PNA), the leader of the research theme Logistics within CWI, and a (part-time) full professor at the VU University, Amsterdam. Before going to academia, he has been working for over a decade as a consultant and researcher in the ICT industry, working for PTT, KPN, AT&T Bell Labs and TNO ICT. His research interests include performance modeling and scalability analysis of ICT systems, logistics, grid computing, revenue management, military operations research, sensor networks, call centers and queueing theory. He is the co-author of some 120 papers in journals and refereed proceedings.

S. Bhulai (1976) received his M.Sc. degrees in Mathematics and in Business Mathematics and Informatics, both from the VU University Amsterdam, The Netherlands. He carried out his Ph.D. research on "Markov decision processes: the control of high-dimensional systems" at the same university for which he received his Ph.D. degree in 2002. After that he has been a postdoctoral researcher at Lucent Technologies, Bell Laboratories as NWO Talent Stipend fellow. In 2003 he joined the Mathematics department at the VU University Amsterdam, where he is an associate professor in Applied Probability and Operations Research. His primary research interests are in the general area of stochastic modeling and optimization, in particular, the theory and applications of Markov decision processes. His favorite application areas include telecommunication networks and call centers. He is currently involved in the control of time-varying systems, partial information models, dynamic programming value functions, and reinforcement learning.