

RESEARCH

Open Access

# Adaptive cognitive media delivery over composite wireless networks

Tim Farnham

## Abstract

Over-the-top (OTT) content-on-demand (CoD) media delivery should ideally adapt to the available resources in an opportunistic manner. The dynamic nature of the Internet traffic and wireless local area networking technologies, which are typical within the home, must be considered in order to efficiently use resources without the need and limitations associated with centralised or fixed allocation of resources. It is also undesirable for devices to continuously monitor the available channels, especially if they are battery powered. Therefore, cooperation between devices and modelling of the dynamic adaptive traffic and terminal behaviour is necessary in order that the most suitable resource sharing strategies are employed. This article examines the exploitation of cognitive resource management for delivery of OTT CoD within unmanaged wireless environments. Channel and traffic models are derived based on the Markov modulated Poisson process and this knowledge is used to derive optimal resource sharing policies. Results from simulation and experimental implementation are presented.

**Keywords:** cognitive radio, dynamic resource management, streaming, multimedia

## 1. Introduction

The main motivation for applying cognitive resource management to over-the-top (OTT) content-on-demand (CoD) adaptive media delivery is to improve the resource utilisation efficiency, through opportunistic behaviour, without the need and restrictions associated with statically configured or reserved resources for individual users. The problem that can occur when OTT CoD is delivered within an unmanaged wireless environment is that unfairness can occur (i.e. one user receiving much higher performance than another). Also, there is great potential for under-utilisation of the available resources due to inappropriate reaction to dynamic transient events. This is a particular problem associated with adaptive CoD delivery, which continuously adapts itself to the observed performance, especially when there are several radio resource options available that can dynamically be selected. Previous research in the field of cognitive radio (CR) resource management has considered opportunistic resource sharing (such as within [1-4]). However, application of these techniques to adaptive OTT CoD media delivery introduces different problems related to the adaptive nature of the application

traffic and associated fairness considerations, outlined in [5], which will interact with the dynamic channel state estimation and modelling approach. We therefore focus on the evaluation of a CR resource management approach applied to wireless unmanaged OTT CoD services.

Standardisation activities associated with CR solutions, for composite networks, focus on the architecture, information models and policies necessary to deploy distributed decision making in a flexible manner. These standards are key enablers of the vision to facilitate advanced radio resource management using a common information model (as introduced in [6]). For instance, the IEEE 1900.4 (2009) standard specifications (see [7,8]) provide the system and functional architectures and the information model (including policies) necessary to split cognitive decision-making processes between network and terminal entities. The standard allows for policies and context information, governing decision making, to be distributed to client terminal devices to assist the decision of how to exploit various access options, within the constraints imposed by the policies. Within the framework of this standard, the exemplary steps involved in a typical distributed radio resource utilisation optimisation use-case are collect context

Correspondence: tim.farnham@toshiba-trel.com  
Toshiba Research Europe Ltd., 32 Queen Square, Bristol BS1 4ND, UK

information, generate radio resource selection policies and perform reconfiguration on the terminal side (within the constraints of these policies). The context corresponds to either the terminal side (such as the observed channel and link measurements, etc.) or the radio access network side (such as cell coverage area and associated measurements). Within previous research (such as [9,10]), the useful context measurements are packet delay, packet loss rate, signal-to-noise ratio and channel activity/load. The policies are then derived to specify the conditions placed on the radio resource selection process. For instance, instructing the terminals to use certain radio access networks or channel configurations only when the specified conditions match.

In this article, we first examine the use of this type of distributed decision making (enabled by the IEEE 1900.4 framework) to improve the performance and efficiency of CoD media delivery within a wireless context and then consider derivation (refinement) of policies to improve the decision-making strategies to provide better overall performance. In order to quantify the benefit and optimality of the policies, which are discussed in Section 4, it is necessary to consider the traffic loading and media delivery performance goals that are introduced in Sections 2 and 3. Performance is evaluated by both simulation models (Section 5) and experimental test-bed implementation (Section 6) to consider a real application deployment scenario. Final conclusions are drawn on the merits of this approach to CoD media delivery over composite wireless networks.

## 2. Channel model

### 2.1. Dynamic channel model

Understanding the dynamic nature of channel load (such as from the Internet traffic), using a representation of channel state, is an important way of determining optimal resource utilisation strategies in unmanaged scenarios. We utilise a channel model based on the Markov modulated Poisson process (MMPP) approach (described in [11]) to assist the decision-making process. In this model, the mean channel rate switches between two or more values (e.g.  $\lambda_{x, 1}$  and  $\lambda_{x, 2}$ ), with certain probabilities ( $p_{1-2}$  and  $p_{2-1}$ ). In this manner when the overall time period of interest is large (compared with the transmission time) and so  $p_{2-1} < 1$  and  $p_{1-2} < 1$ , the time spent in each state is proportional to these probabilities and the overall rate becomes;  $(p_{2-1} \lambda_{x, 1} + p_{1-2} \lambda_{x, 2}) / (p_{1-2} + p_{2-1})$ . This general type of model is therefore applicable for composite network scenarios with Internet traffic.

The above analysis indicates the goal to use channels that are likely to exhibit the highest rate and will remain in the high rate state for sufficient time for transfer of media chunks (bursts) in the target time  $NT$ . The

consequence of wrongly estimating the channel state depends on the relative difference between mean rates (i.e.  $\lambda_{x, 1} - \lambda_{x, 2}$ ) and the dynamics (probability) of state transition.

### 2.2. Channel selection

The above analysis assumes knowledge of the channel rates and rate states in order to optimally select channels. However, continuous measurement (such as channel probing and activity monitoring) is not desirable due to the implied need to use wideband and inefficient spectrum sensing devices or to continuously switch and probe different channels. Therefore, it is desirable to restrict the monitoring time to reasonable periodic intervals and use generic channel activity/load context distribution (i.e. such as using IEEE1900.4 context data see [8]). Within the IEEE 1900.4 framework, the observed channel measurement class (managed object) supports the ability for terminals to monitor the observed channel activity/load in a generic manner. In a similar way the link measurement managed object permits link related measurements (such as link signal level, error rate and latency etc.). These context measurements can be distributed (i.e. shared) via the network reconfiguration manager with other terminals. This is abstracted using a generic data model to facilitate the distribution of context data in the most appropriate and timely manner. For instance, context can be sent only when certain value thresholds are crossed or using a particular periodic sampling (i.e. averaging). Using such techniques, the amount of context data distribution is reduced and is also made more meaningful and useful for use within the radio resource usage optimisation policies. This also prevents the terminals from having to continuously monitor all the available channels all of the time, which would incur excessive complexity and power consumption. In addition to the passive channel observation/measurement, it is also possible to use the active measurements (such as observing transmission latency) to determine if the channel rate state has changed during active transmission, without incurring additional channel monitoring/measurement or distribution overhead, for instance, as part of the criteria to trigger channel switching. Therefore, we make a specific interpretation of the definition for observed channel measurement context, for the purposes of this study, as shown in Table 1, which contains a passive channel context measurement (that is activity/load) and is distributed using IEEE 1900.4 approach, and an active channel context measurement that is computed locally and not distributed. We intentionally omit other typical channel and link context data related to signal strength and error rate as we focus on stationary scenarios with no terminal mobility. However, these would be

**Table 1 Context measurement attribute definitions**

Attribute	Definition	Units
Passive context Activity/load	Total observed transmitted bytes over a specified time window (from all transmitters using the channel) divided by the time window duration	Mbps
Active context Latency	The average one-way delay for transferring a packet within a media “chunk” over the corresponding channel after the chunk is presented for transmission  Alternatively, the average time taken for the delivery of all packets within a complete media “chunk” over the corresponding channel	ms

applicable in other deployment scenarios and have been subject of other studies (such as in [10]).

The above rationale for efficient channel monitoring assumes that the periodic channel context can be combined to derive better resource selection decision policies. The goal of the optimal channel selection is therefore to avoid channel congestion (i.e. a low rate state) by aiming to always select the channel(s) with the least load/activity and also the lowest latency. To achieve this, the active latency performance observation considers the media “chunk” delivery (rather than passive observation or active probing). If the observed latency is greater than a certain threshold (*tHigh*) then it is an indication that the channel is in a low rate state and another channel should be used. Similarly, if the observed latency is below a certain threshold (*tLow*) it is indicative that the channel is in a high rate state and should be used more (i.e. it is underutilised). In a similar manner, if the channel activity context value is above a certain threshold (*chacHigh*) it also indicates that the channel is in a low rate state and likewise if the activity is below a threshold (*chacLow*) then the channel is in a high rate state.

To solve the problem of finding the optimal channel selection strategy, based on the combination of both passive and active measurement threshold criteria, we also need to consider the adaptive nature of the OTT CoD application traffic, which makes it harder to determine optimal thresholds.

### 3. Application traffic

Media services delivered using CoD have a special property, which is the ability to retrieve the required content chunks a certain time ahead of the need for playback of the content at the terminal. Typically, users are prepared to wait for an initial period of time during the initialisation of a content stream, although this is only in the order of seconds and ideally is on a sub-second scale.

Adaptive streaming approaches are most applicable for OTT CoD delivery in dynamic channels, as they adapt to the available channel mean rate for chunk delivery (i.e.  $\lambda_x$ ), so that the application can strive to achieve the highest quality level (QL)/rate ( $n$ ) supported by the

channel, and hence the best possible quality. The time-scale for the adaptation is normally per media chunk, and is in the order of seconds (e.g. 2 s), so as to avoid reacting to very dynamic transient effects. Therefore, channel selection policies that use both proactive and reactive selection for channels are desirable. Exploiting channel knowledge for adaptive streaming delivery services requires a means to measure adaptive streaming performance, which is discussed next.

#### 3.1. Adaptive streaming

Adaptive streaming-based CoD assumes that content is encoded into several QLs that correspond to different average rates ( $n$ ), with higher rate equating to better quality. The measure of performance that we use is based on the QLs of the successfully delivered content chunks. The typical adaptive behaviour is for an initial estimate of the maximum and minimum QL to be determined during the content initialisation phase. The client requests the manifest file for the content item, which includes the available QLs, and also estimates the channel rate and screen resolution to determine the appropriate bounds. Then the client starts with the worst quality (or often a mid-range quality) and requests content chunks gradually increasing quality until either the maximum bound is reached or the channel rate is reached.

In order to take into account the level of user satisfaction obtained when watching adaptive streamed video we define each QL to have an incremental dissatisfaction multiplier ( $i$ ) corresponding to the QL index. In this way, the higher the QL index (implying lower quality) the greater is the user dissatisfaction. The resulting expression for user dissatisfaction (1) is derived based on a mirror representation of the standard mean opinion scores (MOS) that have been measured for adaptive streaming applications by subjective testing. For instance, the standard five level MOS equates to the  $d_u$  by the expression  $d_u = c(5 - \text{MOS})$ , where  $c$  is a constant that depends on the number of encoded QLs of the media source using typical MPEG4 adaptive streaming content. Therefore, the overall level of dissatisfaction observed by user ( $u$ ) is not a linear relationship with

QL, but instead is given by the  $d_u$  expression that is defined in (1) and implies that, for instance, observing a QL index of 5 for 10% of the time (and 0 for the rest) results in a dissatisfaction of 1.5, which is actually perceived to be much worse than if a QL index of 1 was observed 100% of the time (i.e. resulting in a dissatisfaction of 1).

$$d_u = \sum_{i=1}^N i \cdot P_{u,i}(QL \geq i) \quad (1)$$

where  $P_{u,i}(QL \geq i)$  is the proportion of the time (or chunks) that the observed QL index for user  $u$  is greater or equal to the  $i$ th index.

In order to provide a combined dissatisfaction level for all users we take each  $d_u$  and weight it with the corresponding user privilege level  $W_u$  before summation to arrive at a combined overall dissatisfaction ( $D$ ), for all users, as defined in (2). The privilege level is a way to take into account that some users may be more important and should have a lower dissatisfaction level than other users.

$$D = \sum_{u=1}^M d_u \cdot W_u \quad (2)$$

The aim is now to minimise the overall user dissatisfaction ( $D$ ), which is observed. In order to achieve this aim it is necessary to carefully consider the timescales over which estimates are made. For instance, as in all adaptive systems that vary dynamically, taking a period of time that is too short will result in variable and inaccurate predictions of  $P_{u,i}(QL \geq i)$ , that may lead to incorrect decisions being taken.

#### 4. Resource management policies

In this section, we consider the impact of the CoD rate adaptation policies that govern the behaviour of the application as well as channel selection. For this we must first consider what criteria or constraints the policies utilise and how they are formed.

##### 4.1. General form

The aim of policy-based approaches for resource management (such as within IEEE 1900.4 [8]) is to decouple the policy derivation and evaluation process from the policy enforcement point. In this manner, it becomes possible to devolve decision-making functions from one logical entity (server) to another (client terminals).

The policy rules considered are a subset of the general Event-Condition-Action (ECA) form. It is simplified by the fact that all policy rules are evaluated in a priority order on occurrence of a corresponding event (causing an attribute update). The conditions within a policy rule

are formed from simple threshold criteria corresponding to different device specific attributes, and actions are only of three possible types as shown below:

IF { <condition> } <logical> {<condition>} THEN <action>

where <condition> is attribute, operator and threshold criteria and <action> is either EXCLUDE/MUSTUSE/MAYUSE and <logical> is OR/AND.

The meaning of the action EXCLUDE is that the objects (such as channels and links) matching the condition criteria must not be selected. The action MUSTUSE implies that the matching objects must be used in preference to objects (i.e. channels or links) that are not matching the policy rule criteria. The additional action MAYUSE is the default action when neither of the EXCLUDE or MUSTUSE condition applies to them and so there is no constraint on whether or not the associated object is used. However, it can also be used by a high priority rule to specify that a low importance be placed on certain options. Each policy rule within a policy set (ordered list of rules) is then evaluated in a priority order with the first matching criteria taking precedence over subsequent rules. In this manner, the evaluation of the policies results in an unambiguous association of the objects (i.e. channels or links) with the action EXCLUDE, MUSTUSE or MAYUSE. The policy does not specify how the final selection is performed but objects tagged with EXCLUDE cannot be selected under any circumstance and objects associated with the MUSTUSE action take precedence over the objects tagged with the default MAYUSE action.

##### 4.2. Reactive policies

We consider reactive policies to be those that trigger on changes in observed active context performance measurements. For instance, the average latency of the packets delivered can form the basis for one reactive threshold. Two latency thresholds are assumed to be useful for OTT CoD adaptive streaming, which are a high latency threshold,  $t_{High}$ , and a low latency threshold,  $t_{Low}$ . The reactive policies that can be derived to trigger a channel reselection based on the observed latency, such as

IF {channel.latency( $u$ ) > $t_{High}$ } THEN EXCLUDE  
 IF {channel.latency( $u$ ) < $t_{Low}$ } THEN MUSTUSE

where  $u$  is the user (or device) identifier, which means that different users can be assigned different selection policies.

##### 4.3. Proactive policies

Proactive policies relate to monitoring passive context criteria (i.e. observed channel monitoring) to provide a



prediction about likely performance. For instance, the observed channel activity thresholds (*chacLow* and *chacHigh*) can be used to predict the channel state without active transmission or probing. However, the passive context is not necessarily exactly correlated with active context measurements due to the time delay between measurements and also the fact that actual channel rate cannot be measured by passive means alone (i.e. incomplete information). Therefore, the benefit of using passive context is that it has wider applicability for other terminals in the vicinity and can provide a certain prediction about likely performance without the need for any transmissions or probing on alternative channels (which would also incur extra delay). The basic form of the proactive resource management policies is given by

```
IF {channel.activityLevel(u) <chacLow} THEN
MUSTUSE
IF {channel.activityLevel(u) >chacHigh} THEN
EXCLUDE
```

The challenge for both proactive and reactive channel selection policies is to derive a set of policies which give optimal sharing of resources for adaptive CoD media delivery, which is itself changing rate in response to the measured performance and is considered next.

#### 4.4. Rate adaptation policies

Most adaptive streaming applications utilise the measurement of end-to-end latency, in order to calculate available throughput and adapt the rate ( $n$ ) or QL of the media being delivered in a reactive manner. Therefore, the reactive policies that govern the selection of channels should ideally be aligned to the application rate adaptation policies to avoid mismatch and oscillation. However, generally, this is not possible as most adaptive video streaming algorithms are not accessible and do not indicate their adaptation criteria. We consider whether knowledge and control of this behaviour is beneficial by comparing a simulation model, exploiting the same rate adaptation and channel switching criteria, with real measurements using an adaptive streaming application.

#### 5. Simulation model

In order to determine the performance of the above policies, and provide a basis for determining how much benefit is obtained with accurate knowledge of the adaptive behaviour and latency performance, we have developed a simulation environment that permits the evaluation of the policies to trigger channel selection as well as rate adaptation. The same reactive policies are used to accomplish both the channel selection and the rate adaptation processes. Hence, when a reactive policy

such as

```
IF {channel.latency(u) <tLow} THEN MUSTUSE
```

is evaluated, and the condition matches, the rate ( $n$ ) of the application is increased (by an amount *inc*) to attempt to approach as closely as possible to the policy constraint. When no match is obtained then it is assumed that the ideal target latency threshold is obtained or exceeded and the rate ( $n$ ) is subsequently reduced (by an amount *dec*). Therefore, there is always some dynamic perturbation around what is considered as the ideal operating rate.

#### 5.1. Configuration

The simulation model consists of two identical IEEE 802.11 WiFi channels and two or three users (denoted as  $u_1, u_2, u_3$ ). The users are in close proximity and are equidistant from the access point (AP), which is used for OTT CoD delivery. The CoD traffic is modelled using different Poisson arrival rates (with mean  $n_x$ ) corresponding to the different QLs ( $x$ ). The policy evaluation is performed at the equivalent of 2-s intervals, in simulation time, in a priority order such that the first rule (within a set) that has a matching condition is considered to be triggered. In this manner, we avoid any policy rule conflicts. The policy rule set used for simulations is shown in Table 2 and indicates the fact that both reactive and proactive policies can be combined within a single rule if they have the same priority (i.e. hierarchical level) and action. The meaning of rules within this set is, firstly, that the policy constraints that EXCLUDE certain channels from being considered are evaluated first (i.e. these are the channels that must not be selected) and are tagged correspondingly. The rule for this evaluates both the active context (latency) and passive measurements (activity) to determine whether they should be excluded (based on the thresholds). Next, the channels that exhibit a low active context (i.e. latency) are considered to be suitable and hence have a MUSTUSE action and take precedence over other channels. It is considered that this active context (low latency) is a very reliable measure of the current performance and hence has higher priority than the proactive and passive context rule (low activity). Then if there are still no matches with these first two rules, channels that are matched with low passive context (i.e. activity) should take precedence next. Finally, the remainder of the channels (i.e. those that have no matching policy criteria) can be considered to have the default action of MAYUSE, and can be selected if necessary. In the second rule, the adaptive application target one-way latency and proactive channel switching time is *tLow* which is the same for all users (i.e. assuming identical traffic). In

**Table 2 Policy rule set**

IF {channel.latency(1) > <i>tHigh</i> <sub>1</sub> } OR {channel.activityLevel(1) > <i>chacHigh</i> <sub>1</sub> } OR {channel.latency(2) > <i>tHigh</i> <sub>2</sub> } OR {channel.activityLevel(2) > <i>chacHigh</i> <sub>2</sub> } THEN EXCLUDE
IF {channel.latency(1) < <i>tLow</i> } OR {channel.latency(2) < <i>tLow</i> } OR {channel.latency(3) < <i>tLow</i> } THEN MUSTUSE
IF {channel.activityLevel(1) < <i>chacLow</i> <sub>1</sub> } OR {channel.activityLevel(2) < <i>chacLow</i> <sub>2</sub> } THEN MUSTUSE

the simulations we consider the impact of the variables, *tHigh*<sub>1</sub>, *tHigh*<sub>2</sub>, *chacLow*<sub>1</sub> and *chacLow*<sub>2</sub>.

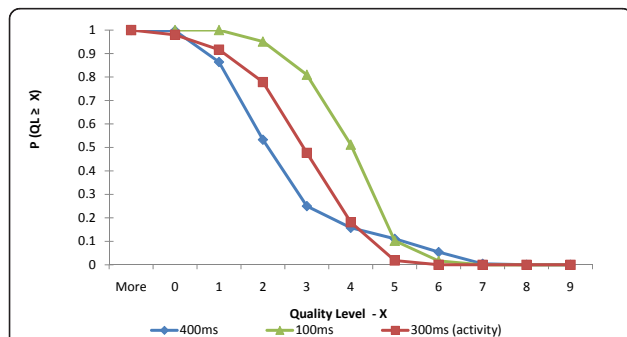
The simulation model consists of OTT CoD media delivery traffic over the composite (multi-channel) WiFi network. Therefore, the performance bottleneck is caused by the WiFi channels, which are rate limited. The important measurements taken within the model are the average frame delivery latency and total channel activity at 2-s intervals.

**5.2. Results**

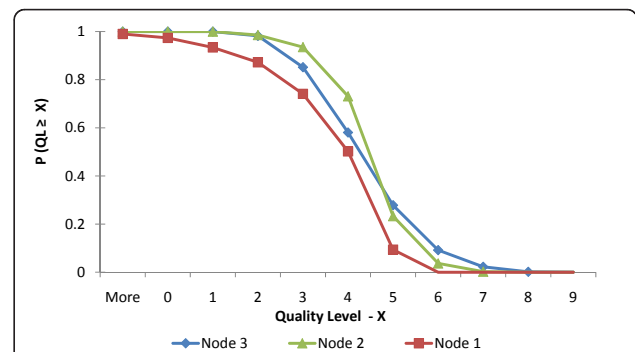
Here, we present the results of simulations to consider the effect of changing the policy variables shown in Table 2 (apart from *chacHigh*, which is fixed at 10). This is important from the point of view of determining optimal policy sets that combine both proactive and reactive policy condition criteria. Firstly, we consider the effect of only the proactive activity-related policy thresholds on performance. In this case, the *tHigh*<sub>1</sub> threshold for user 1 is varied from 100 to 400 ms to determine the effect on the user dissatisfaction, and user 2 remains on the same channel (i.e. the *tHigh*<sub>2</sub> level is set such that it never triggers a policy action). The results in Figure 1 show the distribution of the observed QLs from which the user dissatisfaction is derived, as defined in (1), and hence the benefit (in terms of observed QL) of setting the threshold for the best and worst observed cases of 400 and 100 ms, respectively. This indicates that the high active context threshold (*tHigh*) has a negative impact on performance in the two-node case without other policy constraints. In contrast, when the

proactive activity threshold policy is introduced (with *chacLow*<sub>1</sub> = 1), the optimal latency threshold *tHigh*<sub>1</sub> reduces to 300 ms (with the 400 ms threshold providing worst performance than when at 300 ms). However, in this case it is possible to see that although less media chunks observe very poor quality (i.e. greater than 5 QL index), there are proportionately fewer chunks delivered with the higher QLs (i.e. 2-4 QL index). This implies that the introduction of the proactive policy constraint is good at avoiding the cases in which channel rate is very low, but conversely it is unnecessarily limiting the use of channels when a higher rate could be obtained.

The next consideration is when both of the users (*u*<sub>1</sub> and *u*<sub>2</sub>), corresponding to nodes 1 and 2, respectively, have active policy thresholds (*tHigh*<sub>1</sub>, *tHigh*<sub>2</sub>, *chacLow*<sub>1</sub> and *chacLow*<sub>2</sub>) in the presence of a third user (*u*<sub>3</sub>). In this case, we consider that a set of ten random policy thresholds for *tHigh*<sub>1</sub>, *tHigh*<sub>2</sub>, *chacLow*<sub>1</sub> and *chacLow*<sub>2</sub> are generated and used for the same period of time in an iterative sequence (note that we do not place policy constraints on user 3 in this case and so this user remains on the same channel). The resulting performance obtained is shown in Figure 2 and shows the performance in terms of observed QL distribution, from which the user dissatisfaction is defined using (1), of each of the three nodes (1, 2 and 3) separately. The results indicate that there is both a high degree of unfairness between users (each corresponding to a different node), and that the observed performance is worse than the two-node case. The next step is the selection of a single set of policies that exhibit the best



**Figure 1** Distribution of simulated media delivery performance with different latency thresholds (*tHigh*)–two-node case.

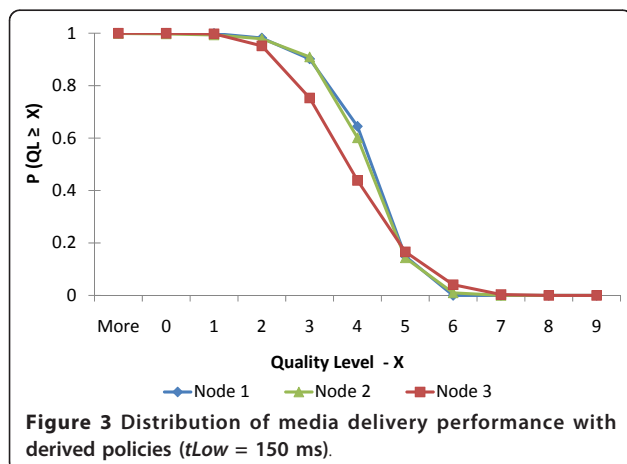


**Figure 2** Distribution of media delivery performance for random policies (*tLow* = 150 ms).

overall performance characteristics. The process used for the selection is based on fuzzy-*c*-means clustering of the performance results and corresponding threshold vectors for each user to attempt to match the best performance with the corresponding policy thresholds. The reason for using clustering (rather than interpolation or curve fitting) is that we assume that in a real scenario we do not have any prior knowledge of the relationship between policy thresholds and the corresponding user dissatisfaction performance. Therefore, it is not possible to take the simplifying step of assuming a direct linear or other type of relationship. Clustering is also able to consider many dimensions within the vectors representing the policy thresholds and in this way the complex non-linear relationship between different policy thresholds is captured. The results with only the best policy rule set, corresponding to the cluster centre which exhibits the lowest user dissatisfaction, are then selected and the results shown in Figure 3.

Specifically, the fuzzy-*c*-means clustering algorithm takes the data vector sets (corresponding to the policy constraint thresholds and observed user dissatisfaction performance) and partitions them into *c* clusters, such that the objective function (sum of the Euclidean distance between each vector and its hypothetical cluster centre) is minimised. The clustering algorithm is an iterative process for minimising the objective function involving selecting an initial random cluster membership matrix, where the sum of utilities is unity. Then the initial cluster centres  $v_i$  are computed followed by the calculation of the objective function to re-compute a new membership matrix. The iterations continue until either the maximum iterations (*R*) is reached or the maximum difference between membership values, from the previous iteration, is less than a threshold amount ( $\alpha$ ).

We choose to utilise only two cluster centres (*c*) as we are approximately modelling the policies that identify

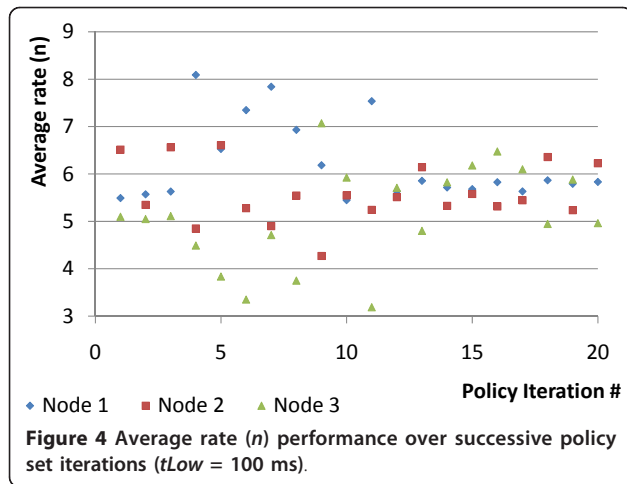


and result in the selection of the two main channel states of interest (i.e. high and low rate). However, clearly this approach can be extended to more than two channels and channel states easily (by increasing *c*). The resulting performance with random and the best derived policy rule set (corresponding to the best cluster centre) are shown in Figures 2 and 3, respectively (and the overall user dissatisfaction obtained over the entire sequence, for individual users and in total, are summarised in Table 3). These results indicate the selection of policy rule set derived from the observed performance (i.e. by clustering) results in policy rule sets with a higher degree of fairness between users 1 and 2 and at the same time the overall performance (*D*) has improved by 6%. The unfairness in the actual channel rate obtained by each user with both random channel selection policy rule sets and the derived rules are illustrated in Figure 4. This result shows that over the sequence of applying the random policies (i.e. iteration 1-11), the unfairness is highly variable (channel rates ranging from 3 to 8). With the derived policy iterations obtained using the clustering process (i.e. iteration 12-20), the fairness is reasonably good and constant (i.e. channel rate only varies from 5 to 6.5). For the case of *tLow* = 200 ms, the finally derived policy thresholds (i.e. corresponding to the best cluster centre vectors over ten iterations) are  $tHigh_1 = 250$ ,  $tHigh_2 = 530$ ,  $chacLow_1 = 6.0$  and  $chacLow_2 = 2.2$ , which implies user 1 exhibits more proactive and reactive opportunistic behaviour than user 2 while user 3 remains static on the same channel.

Observations made while adjusting *tLow* indicate that the optimal user satisfaction (i.e. the minimum total user dissatisfaction for all users) occurs at a *tLow* value of 200 ms (as shown in Figure 5). With lower values than this the overall performance (*D*) is worse as the application cannot take advantage of opportunities to increase the rate (*n*). For higher values, above 200 ms, the application becomes too opportunistic and adapts inappropriately (i.e. when opportunities are only transient). The observed channel activity (i.e. passive context) on the channel is shown in Figure 6 and illustrates the effect that the active context (latency) threshold (*tLow*) has on these passive context measurements. For instance, increasing the *tLow* threshold from 150 to 200 ms results in a corresponding increase in the channel activity measured. More specifically, it increases the

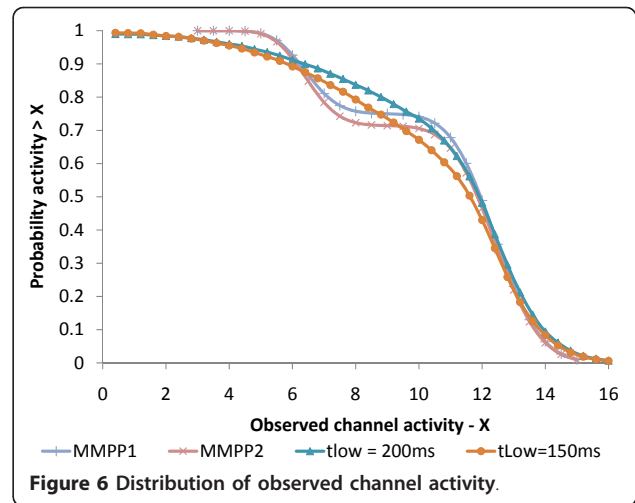
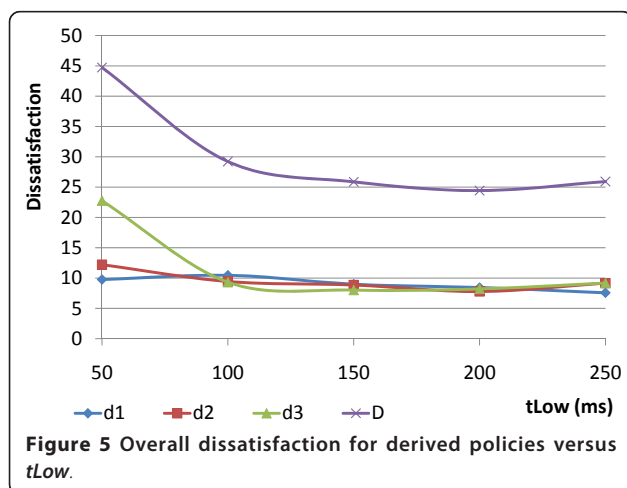
**Table 3** Overall dissatisfaction (per user *d* and total *D*) with equal weighting—for random and derived policies (*tLow* = 150 ms)

Policy	$d_1$	$d_2$	$d_3$	Total ( <i>D</i> )
Random	7.367104	10.09998	9.950516	27.4176
Derived	8.9951	8.85843	8.014565	25.8681



amount of time that each user (node) spends in a channel high rate state (as opposed to observing low rate). The actual increase can be estimated by approximately curve fitting the observed activity using the two-state MMPP model. When  $t_{Low}$  is 200 ms the parameters (corresponding to MMPP1) are  $p_{2-1} = 3.p_{1-2}$  and  $\lambda_1 = 12.5$  &  $\lambda_2 = 6.5$ , and when  $t_{Low}$  is 150 ms the transition probabilities (corresponding to MMPP2) are  $p_{2-1} = (2.5).p_{1-2}$ . Therefore, there is a significant increase (i.e. from 2.5 to 3) in the probability of being able to detect and exploit the channel high rate states. However, in this particular case, the difference between high and low rate states is relatively small. This implies that the average overall activity for both channel states is  $(\lambda_2 + 3\lambda_1)/4 = 11$  for  $t_{Low} = 200$  ms and is  $(\lambda_2 + (2.5).\lambda_1)/3.5 = 10.8$  for  $t_{Low} = 150$  ms, which is also a marginal improvement in overall average observed channel activity for  $t_{Low} = 200$  ms.

The simulation results have shown that there are merits in deriving common channel selection and adaptation



policies to achieve both fairness and high user satisfaction and channel utilisation in composite network scenarios. There appears to be an optimal latency threshold ( $t_{Low}$ ) of between 150 and 250 ms, which provides the best performance, when policies are evaluated on 2-s intervals.

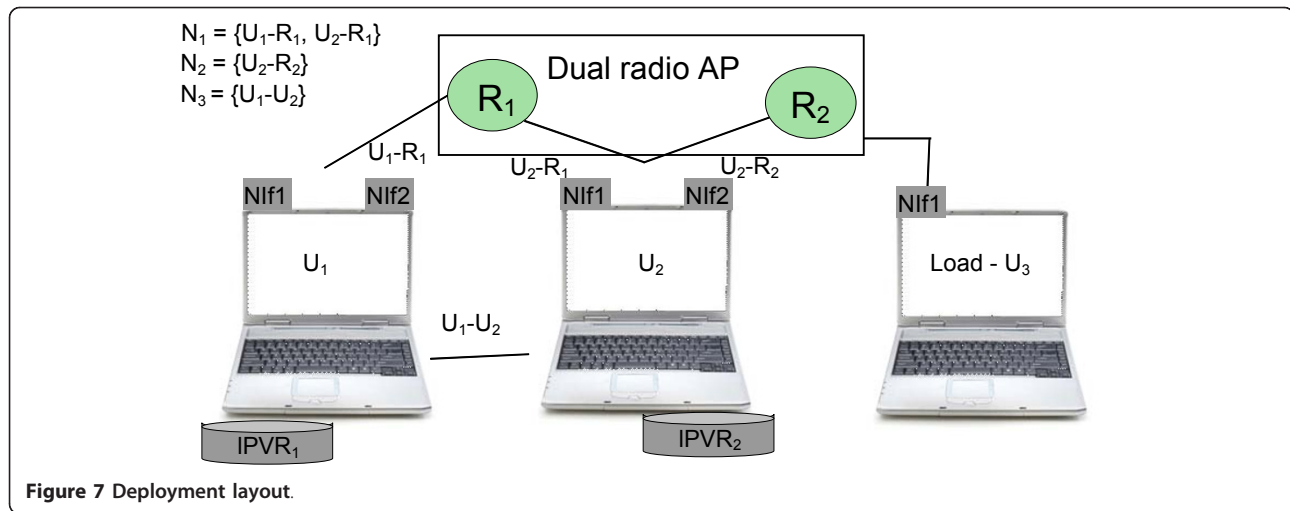
## 6. Experimental measurements

The simulation model examines optimal policy thresholds with homogeneous traffic channels (i.e. three identical adaptive streaming sessions and two identical channels) and continuous periodic context distribution updates (at 2-s intervals). In reality, it is undesirable for continuous periodic context updates and also the assumption regarding the per-channel/session one-way latency, as the basis for performing both rate adaptation and channel selections, is unrealistic. This is because most off-the-shelf APs and wireless devices do not measure or report this level of link and channel-related performance information. In addition, the assumption that the application rate adaptation algorithm is accessible or that resetting the QL to a default (i.e. QL5) may not be possible in practice. An experimental test-bed deployment is used to consider the impact and limitations associated with already deployed CoD solutions. We use an open source demonstration platform (which is available for download from [http://ict-aragorn.eu/fileadmin/user\\_upload/downloads/IPTVDemo.zip](http://ict-aragorn.eu/fileadmin/user_upload/downloads/IPTVDemo.zip)), that uses the popular and flexible HTTP adaptive streaming approach.

### 6.1. Deployment layout

In order to determine performance in a more realistic, but simplified, wireless media deployment scenario, we consider two adaptive streaming CoD users ( $U_1$  and  $U_2$ ) with two client devices, each supporting a local personal video recorder (IPVR) function. The AP contains two





radios ( $R_1$  and  $R_2$ ) supporting 802.11a and 802.11b/g and operating in the 5 and 2.4 GHz ISM bands denoted by channels 1 and 2, respectively, as shown in Figure 7.

The AP that we use is a Hewlett Packard Pro Curve MSM 325 dual radio AP and is configured to limit the maximum rate of the radios to 6 and 11 Mbps, respectively. The client of user 2 ( $U_2$ ) has the capability to support three networks simultaneously,  $N_1$ ,  $N_2$  and  $N_3$ .  $N_2$  and  $N_3$  operate on the same channel (channel 2).  $N_1$  is supported by a separate interface operating on channel 1. The load device performs network loading, via the AP, using the netperf Internet (TCP & UDP) benchmark software, to  $U_{1\&2}$ . The load is intentionally bursty in nature, with an overall 50% duty cycle to cause dynamic channel state changes and thoroughly test the triggering of the policy conditions. Therefore,  $U_1$  has two main options for retrieving media content streams, which are firstly from IPVR<sub>2</sub> via  $N_1$  (or  $N_2$ ), secondly from IPVR<sub>2</sub> direct via  $N_3$ . A similar set of options is available to  $U_2$ , firstly IPVR<sub>1</sub> direct over  $N_3$ , secondly IPVR<sub>1</sub> via  $N_1$  or  $N_2$ . The video player utilised in this case is the Silverlight-based smooth streaming player (available from <http://smoothhd.com>) and the video encoding levels (corresponding to the different QL index) are shown in Table 4.

### 6.2. Policy criteria

In existing adaptive streaming CoD delivery solutions, it is generally not possible, and unnecessary, to measure one-way latency and to perform a continuous periodic

policy evaluation. Typically, chunk delivery time is used to compute the available throughput (rate) at the terminal client to use as the basis for adjusting the rate. It can be assumed that the one-way latency approximates to half the chunk delivery latency. There are different ways to measure the chunk delivery latency; one is to simply measure the time taken from a HTTP media chunk request being made (i.e. a HTTP get request) till the complete response corresponding to the media chunk is returned. It is also possible to measure the time to first byte of the response message in order to eliminate the HTTP fetching time from this latency calculation. The latter approach is beneficial when considering only the link/channel performance as the variation in fetching time can be incorrectly interpreted as a channel rate change. However, in the current test-bed implementation, the simple HTTP request/response latency measurement approach is used with a timeout threshold (i.e. in a corresponding way to the previous *tHigh* threshold). This provides an ability to react quickly to degrading channels (i.e. excluding channels that move from high to low rate state). However, a single static timeout threshold is not optimal; as it depends on the probability of achieving a better performance on an alternative channel, which we have seen has a degree of uncertainty even with passive context measurements. Therefore, during the initiation of a streaming session, in which pre-fetching is performed, all channels are estimated by selecting media chunks using all channels on a load sharing basis (i.e. channels with a corresponding policy action of MAYUSE or MUSTUSE are used simultaneously). Then an optimal waiting time is calculated to compare with the most recently measured channel latencies, based on a trade-off between the potential benefit of using an alternative channel versus the channel switching efficiency (i.e. overhead). The optimal

**Table 4** Average encoded bitrates (kbps) and corresponding QL

0	1	2	3	4	5	6	7	8	9
6000	4176	2907	2023	1408	980	682	475	330	230

waiting time, derived within [12], depends on the target delivery latency ( $NT$ ) and the timeout probability and is approximated in (3).

$$t_{n,i}^* = [-\ln(P_{n,i}(T))]^{-1/2} \cdot [P_{n,i}(T)]^{-N/4} \quad (3)$$

where  $P_{n,i}(T)$  is the estimated probability of successfully completing delivery of the media data unit  $n$  in discrete time interval  $T$  on channel  $i$ ,  $NT$  is the target chunk delivery latency and  $t_{n,i}^*$  is the optimal waiting threshold (timeout in terms of intervals  $T$ ) for receipt of the media chunk before triggering a switch to channel  $i$ .

The determination of optimal waiting time assumes that  $P_{n,i}(T)$  can be estimated by recent delivery latency measurements on channel  $i$  or by approximating to the observed performance on the current channel (assuming similar average performance). Therefore, it may not reflect the actual performance that will be obtained. For instance, if the state of channel  $i$  changes then the estimate is no longer accurate and the waiting time will not be optimal. Therefore, we consider the longer term evaluation of the average performance to compute  $P_{n,i}(T)$  and also low values of  $N$  (i.e.  $N < 6$ ) to provide the greatest tolerance to errors in the estimation of  $P_{n,i}(T)$ . Consequently,  $t_{n,i}^*$  is limited in the range 2 to 3 for a wide range of  $P_{n,i}(T)$ . Also, the initial pre-fetch interval is important to obtain a first approximation of  $P_{n,i}(T)$ , after which there could be a significantly higher probability of having an incorrect estimation for reactive triggering of alternative channel selection. Consequently, it is anticipated that this type reactive trigger is better for reacting to the rapidly degrading current channel, and may not be able to determine and react appropriately to potential opportunities (i.e. better channels), which requires proactive policies.

### 6.3. Policy derivation

In the same manner as for the simulation model, the fuzzy- $c$ -means algorithm clusters the corresponding vectors comprising the policy thresholds. But this time only for the proactive observed channel activity thresholds ( $chacHigh_w$ ,  $chacLow_u$ ) and resulting dissatisfaction levels for each user ( $d_u$ ). The clustering process again attempts to form two cluster centres that best fit the measured performance data. In this way, the resulting cluster centres indicate the thresholds that are most likely to provide either “high” or “low” rates for each user (corresponding to  $\lambda_{x,1}$  and  $\lambda_{x,2}$ , respectively). The “high rate” cluster (i.e. the one giving the lowest overall dissatisfaction  $D$ ) is then used as the basis to set new (default) policy thresholds ( $chacLow$  and  $chacHigh$ ) and the whole process repeated. In this manner, it is assumed that any cluster policies extending the low rate state are revoked and not re-used whereas policies

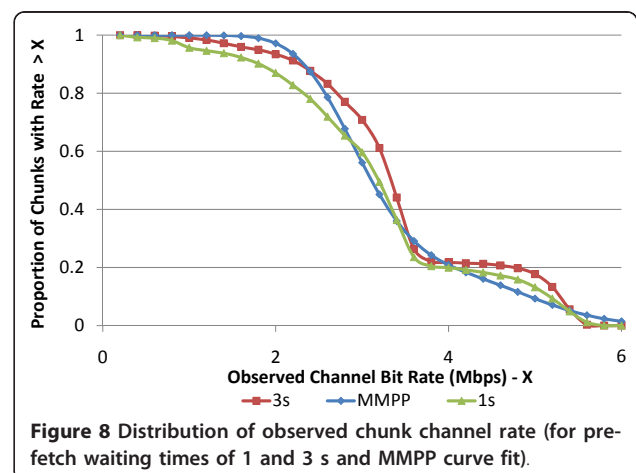
encouraging high rate states will be reinforced and provide more suitable resource utilisation.

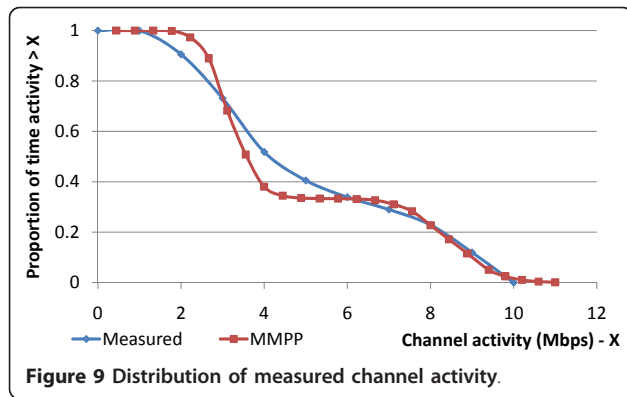
### 6.4. Results

The overall distributions of the measured channel rates, observed during the experiments, are shown in Figure 8. The figure also contains the best fit two-state MMPP for the combined channel rate distributions, characterised by  $p_{1-2} = 4.p_{2-1}$  and  $\lambda_1 = 5$  Mbps &  $\lambda_2 = 3$  Mbps. This indicates that it is four times more likely, with the combination of both channels, for the selected channel to exhibit low rate state. Interestingly, if too little pre-fetching is performed ( $< 3$  s) there is a corresponding reduction in the observed channel bit rate due to an increase in the error resulting from incorrect channel state estimation and consequently inappropriate channel selections being taken.

The corresponding channel activity distribution, considering all traffic (including the load), is shown in Figure 9, with MMPP parameters given by  $p_{1-2} = 2.p_{2-1}$  and  $\lambda_1 = 8.5$  &  $\lambda_2 = 3.2$ . Therefore, the overall average channel rate, based on the curve fit, is  $(\lambda_1 + 2.\lambda_2)/3 = 4.97$ . The curve fit in this case is not as closely matched with the measured channel activity as the simulations. Also, the overall channel utilisation achieved is lower due to the less predictable behaviour and dynamic loading, which makes the opportunity exploitation harder.

The resulting performance observed (see Figure 10) indicates that the benefit of using the policies, derived (i.e. learnt) over successive iterations, is around a factor of 2 compared with having no policies (i.e. statically configured channels). Although this seems a relatively small gain, in terms of real user dissatisfaction this improvement is significant as it can directly correspond to a change from a MOS level of poor to good (as we have intentionally derived user dissatisfaction from the MOS). Also, there is significant advantage in the fairness of the derived optimal

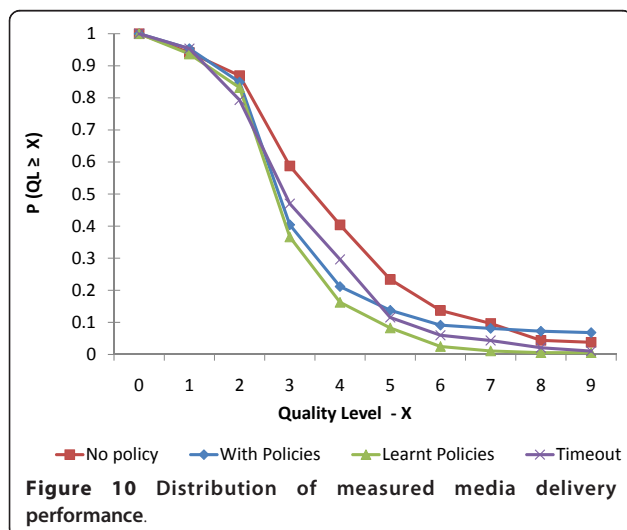




policies compared with the default policies or relying on a simple timeout mechanism. Comparison of the experimental and the simulation results also confirms that policy derivation can be performed without complete knowledge of the adaptive characteristics of the application, as similar optimal thresholds were obtained in the same number of policy iterations. However, the overall channel exploitation is lower in the real test-bed system as the reaction time to the dynamic variations takes longer, which impacts the overall performance. Therefore, slower convergence of the derived policies and lower channel utilisation are the main drawbacks of not knowing or being able to control how the application adapts its rate. Otherwise, the unsupervised learning process shows benefits in both cases and has the benefit of not requiring prior knowledge of the channel performance characteristics and the complex relationship between policy thresholds and the resulting user dissatisfaction levels.

## 7. Conclusions

This article has explored policy-based resource management in cognitive adaptive CoD media delivery over



composite wireless networks. This type of application usage scenario has special characteristics in that it can adapt itself to cope with the existence of dynamic performance observed across multiple wireless channels or networks. However, this also complicates the job of determining the optimal radio resource selection strategies (i.e. channel selection policies). The optimum strategy for utilisation of the different radio resources (channels) is therefore dependent on the ability to determine the opportunities and derive policy rules that improve the overall performance (user satisfaction). This article proposed and evaluated suitable policy condition criteria, and policy rule sets, to achieve the best overall resource sharing strategies, without the need to continuously actively probe or monitor the wireless channels. Instead, use was made of generic IEEE1900.4 observed channel context data (load/activity) and reaction to locally obtained (latency) performance. The results show that there are significant benefits of employing cognitive adaptive approaches to CoD media delivery in unmanaged composite network scenarios.

Further study will determine whether the dynamic channel model characteristics can be easily and beneficially extended, with additional states to capture the dynamic nature of adaptive traffic and radio channels on different timescales and with other channels and traffic types. It is also attractive to consider other policy derivation algorithms to determine the trade-off between time taken to accurately capture traffic/channel behaviour and the opportunity exploitation potential. Finally, we have not evaluated the replacement of the built-in adaptive algorithms within the Smooth streaming media player. This can potentially provide the benefit of faster convergence to optimal policies and also better overall performance. However, there are clearly implications on the system stability and the need to ensure fairness between different OTT CoD users is not compromised in larger deployment scenarios.

### Acknowledgements

This study was supported in part by the European Union (ARAGORN project-ICT 216856); see [www.ict-aragorn.eu](http://www.ict-aragorn.eu).

### Competing interests

The authors declare that they have no competing interests.

Received: 27 April 2011 Accepted: 31 October 2011

Published: 31 October 2011

### References

1. T Farnham, M Sooriyabandara, C Efthymiou, Terminal centric optimisation for the wireless local area, in *IEEE PIMRC*, pp. 1-5 (2007)
2. T Farnham, M Sooriyabandara, C Efthymiou, Enhancing multimedia streaming over existing wireless LAN technology using the unified link layer API. *Int J Netw Manag.* **17**, 331-346 (2007). doi:10.1002/nem.667
3. F Berggren, A Bria, L Badia, I Karla, R Litjens, P Magnusson, F Meago, Haitao Tang, R Veronesi, Multi-radio resource management for ambient networks, in *IEEE PIMRC*, pp. 942-946 (2005)

4. P Bahl, P Bahl, R Chandra, MultiNet, connecting to multiple IEEE 802.11 networks using a single wireless card, Microsoft Technical Report MSR-TR-2003-46, (August 2003)
5. S Akhshabi, A Begen, C Dovrolis, An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP, in *ACM Multimedia Systems Conference* (2011)
6. K Nolte, M Sooriyabandara, M Stamatelatos, T Farnham, C Polychronopoulos, O Holland, A Attar, C Fortuna, Facilitating advanced radio resource management using a common information model, in *The 1st Workshop of COST Action IC0902 Cognitive Radio and Networking for Cooperative Coexistence of Heterogeneous Wireless Networks*, Bologna, Italy, (November 23-25, 2010)
7. S Buljore, H Harada, S Filin, P Houze, K Tsagkaris, V Ivanov, K Nolte, T Farnham, O Holland, Architecture and enablers for optimised radio resource usage in heterogeneous wireless access networks: the IEEE p1900.4 working group. *IEEE Commun.* **47**(1), 122–129 (2009)
8. IEEE 1900.4, Standard for "Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks". (2009)
9. T Farnham, M Sooriyabandara, Cognitive resource management for wireless LAN within the home, in *ICT Mobile Summit* (2009)
10. EH Ong, JY Khan, Distributed radio resource usage optimization of WLANs based on IEEE 1900.4 architecture, in *2nd IFIP IEEE Wireless Days (WD), 2009* (December 2009)
11. D Heyman, D Lucantoni, Modeling multiple IP traffic streams with rate limits. *IEEE/ACM Trans Netw.* **11**(6), 948–958 (2003). doi:10.1109/TNET.2003.820252
12. T Farnham, IPTV delivery using cognitive radio principles, in *IEEE DySPAN*, pp. 618–622 (2011)

doi:10.1186/1687-1499-2011-148

**Cite this article as:** Farnham: Adaptive cognitive media delivery over composite wireless networks. *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:148.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---