

Chapter 4

Case Study: Spectrum Management in CRN Framework

Channel selection in the QoS provisioning framework (as discussed in Sect. 3.2), is done through channel selection and management unit (CSMU) along with the help of QoS control unit (QCU), policy domain and database as shown highlighted in Fig. 4.1. The CSMU module receives the requirement of QoS from QCU, and through the assistance of database and policy domain it selects channel. The selected channel identifiers are then passed to the MAC protocol unit (MPU).

4.1 Spectrum Usage Behavior

As discussed in Sect. 1.6, channel selection technique depends upon various spectrum characteristics including channel availability, SINR, bandwidth, path loss and primary user behavior. The primary user behavior in a cognitive operation based open spectrum system can be classified as: *deterministic* and *stochastic* usage behavior.

4.1.1 Deterministic Usage Behavior

There are some frequency channels which inherently follow a specific pattern of their usage over time because of the nature of the operating application [1]. This is verified by the spectrum utilization studies done by various researchers across the world, that have shown that the spectrum usage by incumbents follow a deterministic time pattern over space as discussed in Sect. 1.3 and work done in [2–18]. Spectrum occupancy measurement of Singapore is done in [19]. From this measurement and analysis, a table is drawn and shown in Table 4.1 for some channels whose observed spectral opportunity shows a deterministic time pattern.

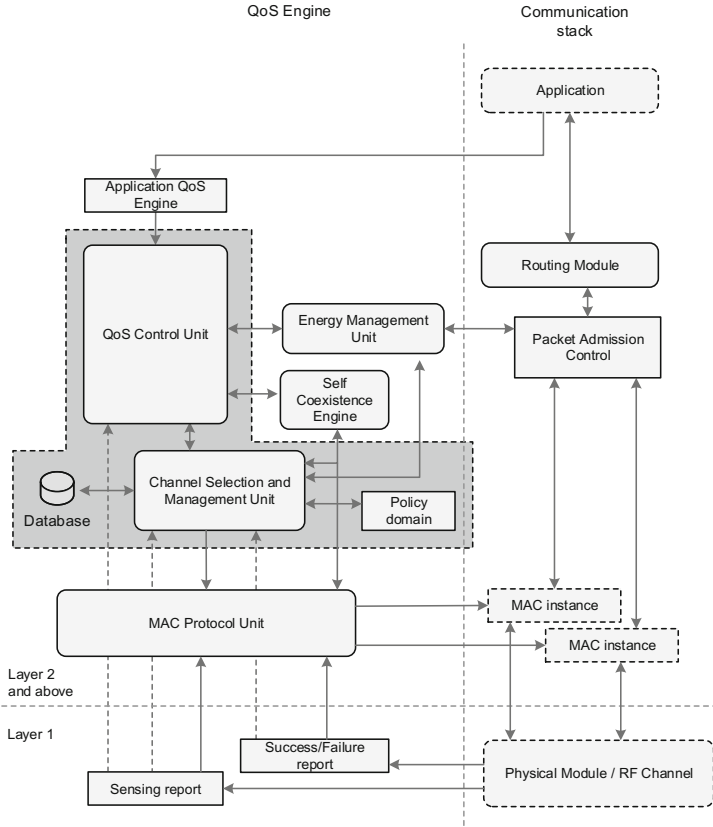


Fig. 4.1 Functional modules responsible for channel selection in the QoS provisioning framework for cognitive radio network

From Table 4.1 it can be observed that there are certain frequency bands which are always free during certain duration of time. Local TV channels like Suria (ch12) and Central (ch24) are closed from 12 am to 9 am and 12 am to 6 am respectively. Therefore these channels are well suited for cognitive radio communication during the mentioned time periods.

4.1.2 Stochastic Usage Behavior

If the spectrum usage pattern by a primary user does not follow a deterministic pattern, it is modelled with the help of statistical learning techniques to find the hidden pattern and to obtain the usage distribution. Several spectrum decision techniques have been proposed in various literatures (discussed previously) and are discussed in Sect. 1.6.

Table 4.1 Available frequency band over time in Singapore

Frequency (MHz)	Type	Free time (H)
80–88	PMR	18–12
183–186	Ch6	12–10
207–209	Ch9	18–16
212–214	Ch10	18–16
218–222	Ch11	11–06
225–230	Ch12	00–09
490–500	Ch24	00–06
614–620	Ch39	00–15
720–740	Channel 56–58	08–00
740–780	Channel 59–64	All Time
1880–1910	DECT and TDD	02–17
2270–2300	Fixed/Mobile	19–07

These works are used to model the spectrum characteristics with the assumption of stochastic usage behavior of incumbent usage pattern on the channel. Generally these schemes utilize the local and statistical observation to make a spectrum decision.

4.2 Reconfigurable Channel Selection

Recently, Federal Communication Corporation (FCC) has mandated the use of database along with spectrum sensing to observe the spectral opportunity at a given place over a given period of time [20]. The database assisted spectrum sensing provide an accurate measurement of spectrum characteristics. Since the spatial and temporal characteristics of spectral opportunities vary rapidly, an up-to-date incumbent operation database is needed for protected broadcast operation in a given location. Since providing an up-to-date information about primary service operation is exhausting in nature, another paradigm for information collection need to be determined. The channels can be classified into two types depending upon their usage: *Deterministic time pattern channels* and *Stochastic channels*. Therefore the system comprises of two step reconfigurable channel selection scheme. During the first step, a channel with deterministic time pattern is selected for usage. If no such channel exists, then stochastic channels are selected based on the requirements. The selection of stochastic channels can be assisted with the help of database.

Figure 4.2 shows the flowchart followed by the CSMU module of the discussed QoS provisioning framework along with the three main classes of channel selection: *Deterministic channel selection*, *Database assisted selection on the go* and *Selection on the go*. Initially channel selection starts with deterministic channel selection scheme, but if the observed spectral opportunities do not verify with that of database,

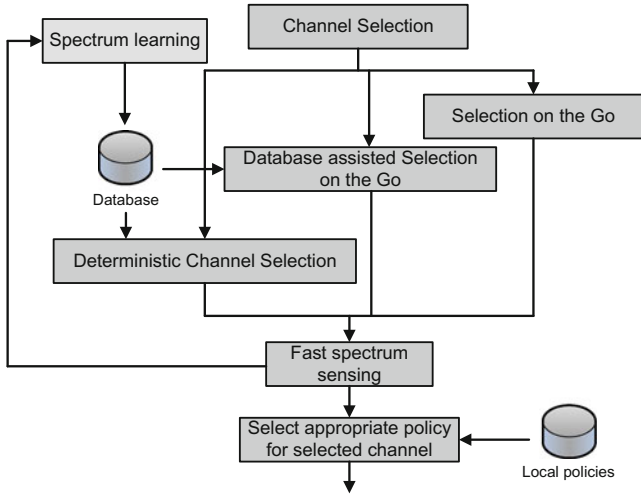


Fig. 4.2 Channel selection flow

then database assisted selection on the go method is used. In this method, statistically learning of spectral opportunities for stochastic channel is used along with database. If utilization of database in this methodology results in high error, then statistical learning is used without usage of databases which is called as independent selection on the go approach. Once a channel is selected, it is fast scanned to check out any incumbent activity, and if it is vacant then appropriate policies are selected for the channel. If the channel is not vacant, then deterministic spectrum learning module updates the database. An error parameter is utilized to realize the reconfiguration of channel selection schemes.

4.3 Channel Selection in Deterministic Environment

As discussed earlier, some channels have inherent characteristic of deterministic time pattern of utilization. These channels are allocated ahead of time for secondary open spectrum system so that better spectrum decisions can be made. A deterministic channel selection scheme for channel allocation ahead of time is discussed in [21]. Some of the spectrum decision criteria discussed for QoS provisioning are: *minimal channel switching* and *maximal throughput* of the system [22]. For the purpose of same, three different channel selection techniques to support minimal channel switching, maximal throughput and a value in between are discussed.

Channel selection techniques for deterministic environment employ two phases for its operation. The first phase involves learning of the channel spectral opportunity for 24 h and then based on the data obtained, the system may select a channel

satisfying the given constraints. These constraints could be minimal channel switching for reducing the packet delay and losses, high throughput or a value in between. Users can also impose their own constraints to obtain the set of desired channels. The second phase involves usage of these channels and recording the errors in channel spectral opportunities. The errors are then used to check whether the system should continue with the current approach or change to the selection on the go approach.

4.3.1 System Model

The system comprises of N cognitive end-users coordinated with the help of a central controller *Cognitive Access Point* (CAP). Each user is equipped with a single half-duplex transceiver, capable of performing channel sensing. The system utilizes C non-overlapping heterogeneous primary bands each having bandwidth of B_c where $c \in C'$ and $C' = \{1, \dots, C\}$. These C channels are selected from a set of TV and FM bands and are modeled as two state Markov chain as discussed in Sect. 3.4.1. A common control channel is utilized by CAP to coordinate with its users in the network; for which any ISM band or a channel with high a availability can be selected. An efficient spectrum sensing scheme from the literature is used at the physical layer for spectrum sensing; and it is assumed that the result obtained by it are correct. Therefore each CR user keeps track of the list of the available channels. This information is passed to the CAP on common control channel.

Each CR user node uses linked list as a database for capturing these spectral opportunities. There is a linked list assigned for each channel, therefore there are in total C linked lists maintained by each node. The nodes of each linked list contain the information of spectral opportunity represented by two fields: the *start time* and the *end time* (in time-units) during which the channel is free. A typical linked list is shown in Fig. 4.3.

The fields have the following meanings:

- S_t : Start time from which the channel seems to be unoccupied by the primary user.
 - E_t : End time at which the channel ceases to be unoccupied by the primary user.
- The subscript t represents the node ID within a linked list and is given by $t \in \{1, \dots, T\}$ and T is the last node ID in the linked list. The value of T may differ for each linked list.

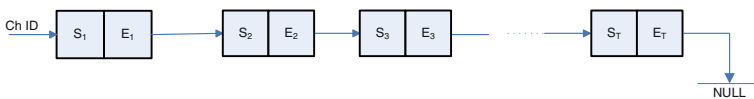


Fig. 4.3 Linked list

4.3.2 Deterministic Learning with Spectrum Selection and Usage

The system samples all C channels for 24 h with sampling interval of x minutes and sampling time of z seconds (where $z \ll x * 60$) i.e. the channel will be sensed every x minutes for z seconds. Thus the total time required for sampling C channel is $z * C/60$ min (where $z * C/60 < x$). The value of x is chosen such that the system would not sample channels too frequently nor miss the important transitions on the channels. The values of S_t and E_t in Fig. 4.3 represent the time-units mapped to the real world time with $(S_t, E_t) \in \{0, \dots, 1439\}/\text{sampling interval}$; whereas 1 time-unit duration = sampling interval (x minutes). The terms time-unit and time-slot are used interchangeably. If x is small, then the number of nodes in the linked list may increase, and in the worst case scenario, the maximum number of nodes in the linked list equals $((24 * 60)/x)/2$ when the channel alternates between free and occupied states in consecutive samplings. To shorten the length of a linked list, a threshold parameter Ω is utilized. If the length of the spectral opportunity is greater than Ω (i.e. $\forall t\{E_t - S_t \geq \Omega\}$), then this spectral opportunity is inserted in the linked list. For example if $\Omega = 20$, then if one channel is sensed free for 19 consecutive time-units while other channel is sensed free for 22 consecutive time-slots, the availability information of the former channel will not be entered in the linked list while that of the latter channel will be entered.

At startup, all linked lists are empty and point to address NULL. With each sampling interval and satisfaction of the threshold parameters, the nodes are updated or inserted in the linked list. The time-unit starts from 0 (00:00 h) and increases by 1 with each sampling interval or every x minutes. An example of the linked list of Channel 2 after 24 h using $x=10$ min is shown in Fig. 4.4. It can be observed that the Channel 2 is unoccupied during the time periods 00:00–11:40 h, 16:40–20:10 h and 23:00–24:00 (00:00); and is occupied during the time periods 11:40–16:40 h and 20:10–23:00 h.

The sampling continues for 24 h, at the end of which the system would have obtained spectral opportunities for all channels. This information is passed to CAP, which consolidates the information received from all the nodes to obtain the global system spectral opportunity. This consolidation is required so that every node in the system agrees on the availability of the channel. A simple method to obtain the global channel opportunity is to convert all the linked lists into matrices and compute the intersection on them, thus reducing the computation complexity.

The linked lists received by CAP from each node are described as $LL_{i,j}$ where $i \in C'$, $j \in N'$ and $N' = \{1, \dots, N\}$. Let $\eta = (24 * 60/x) - 1$ and $\eta' = \{0, \dots, \eta\}$. Let the matrix for the linked list received from node j be $\text{MAT}(j)$, where $j \in N'$. The

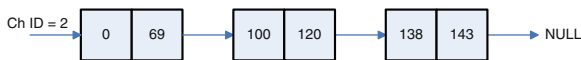


Fig. 4.4 Channel usage pattern of Channel 2 for one day (24 h)

row and column of the matrix represent channels and time-units respectively. The conversion from linked lists to this matrix can be done as follows. In the following conversion index j is dropped.

$$\mathbf{MAT}_{i,k} = \begin{cases} 1, & \exists t[k \in \{S_t \dots E_t\}] \forall i \in C', \forall k \in \eta', t \in T' \\ 0, & \text{otherwise} \end{cases} \quad (4.1)$$

The predicate logic $\exists t[k \in \{S_t, \dots, E_t\}]$ or $\exists t[S_t \leq k \leq E_t]$, evaluates to true whenever the time-unit value falls within the node field S_t and E_t values. Therefore the elements of the matrix are either 0 or 1 representing a PU is present or not respectively on a channel corresponding to the given time-slot for the node j . Overall, N such matrices are obtained. To consolidate all the global information, the CAP performs AND operation on all the corresponding locations of the matrices to obtain a channel availability matrix satisfying the channel availability of all the nodes in the system, i.e.

$$\mathbf{M}_{i,k} = \bigcap_{j=1}^{j=N} \mathbf{MAT}_{i,k}(j), \forall i \in C' \text{ and } \forall k \in \eta' \quad (4.2)$$

The channel availability matrix \mathbf{M} agrees with all the CR nodes present in the system with respect to the availability of the channel. An example matrix is shown in Fig. 4.5. It is assumed that at any given time-slot (time-unit) at least one channel is available for communication. Hence

$$\forall k \exists i \{\mathbf{M}_{i,k} = 1\}, \text{ where } k \in \eta', i \in C' \quad (4.3)$$

Different users have different requirements for the channel selection. The matrix \mathbf{M} can be processed to obtain the desired channel or set of channels satisfying the users constraints. If a single channel is required at a given time-slot, then the solution is represented with a vector. The vector index is represented by the time-slot while its content represents the channel ID. Therefore this vector guides the system to choose a given channel at a given time-slot. A typical vector could be initialized as:

Fig. 4.5 A typical example of matrix showing availability of channels in CR network

		Time Units (k)								
		0	1	2	3	4	5	6	...	η
Channel ID (i)	1	0	0	0	0	1	1	0	...	0
	2	1	1	1	1	0	0	0	...	1
	3	0	1	1	1	0	0	0	...	1
	4	0	0	0	0	1	1	1	...	0
	⋮									
C	0	1	1	1	1	0	0	...	1	

$$\mathbf{V}_k = \{i : \forall k \exists i [M_{i,k} = 1]\}, \text{ where } \forall i \in C' \text{ and } \forall k \in \eta' \quad (4.4)$$

Multiple vectors may satisfy this equation and hence multiple solutions may exist if no constraints are applied. The above vector simply chooses a channel which is available during a given time-slot and thus represents the solution space for all possible vectors. A user may apply single or multiple constraints on the system to impose restriction on channel selection. That is, the solution vector would be an element from the solution space set obtained in Eq.4.4. Three different strategies of channel selection are discussed so as to meet the constraints of: *minimal channel switching*, *maximal throughput* and a *value in between*.

4.3.3 Minimal Channel Switch Requirement

Channel switching is costly for a radio device especially in CR devices [23]. Real-time applications suffer from packet losses, jitters and delays induced by frequent channel switching in CRN. Therefore a comparison of different channel switching techniques indirectly corresponds to a comparison of different packet delays, losses and jitters induced by channel switching in the network. Minimal channel switching refers to the selection of channels such that the system would stay on that channel for a long period of time without hopping onto other channels. The problem can be formulated as to obtain a solution vector \mathbf{VMIN} such that when the system follows the solution vector, the overall switching is minimized for the entire duration. Therefore the problem is to find \mathbf{VMIN} which has the least total number of channel switching as compared to other vectors available, i.e.,

$$\sum_{k=0}^{\eta-1} Switching(\mathbf{VMIN}_k) \leq \sum_{k=0}^{\eta-1} Switching(\mathbf{V}_k) \quad (4.5)$$

where $\forall V \in \{\text{set of all possible vectors } \mathbf{V} \text{ obtained from Eq.4.4}\}$ and $Switching()$ is a binary function defined as:

$$Switching(X_i) = \begin{cases} 1, & \text{if } X_i \neq X_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (4.6)$$

where X is channel vector and i is its time index.

A solution to this problem can be obtained by using a greedy approach. This approach selects the channel which has the maximum period of continuous vacancy over a time interval. Algorithm 4.1 gives the implementation detail of the greedy approach. The algorithm first initializes the vector to value *zero* (line 1). If the solution vector contains *zero* at any time-slot, it means there are no channels available during that time-slot. The outer loop is used to index the time-slots (lines 3–18), the middle loop is used to check out all the channels (lines 4–14) and the inner most loop

is used to check out the length of the continuous vacancy (lines 7–11). The variable array *trail* stores the length of continuous vacancy over all channels. At the end, another loop is used to check which channel has the longest spectral opportunity and is stored in **VMIN** (lines 15–17).

Lemma 4.1 *If on average there are θ channels available per time-unit, then the total number of vectors \mathbf{V} available is $\theta^{\eta+1}$. The average is obtained over space or location.*

Algorithm 4.1: Algorithm for minimal channel switching using greedy approach. The *max()* function returns the maximum value among all elements of a vector and *argmax()* function returns the argument for which the function value is maximum.

```

1 VMINk ← {0}, ∀k ∈ η';
2 traili ← {0}, ∀i ∈ C';
3 for k = 0 to η - 1 do
4   for i = 1 to C do
5     traili ← 0;
6     if Mi,k = 1 then
7       for d = k + 1 to η do
8         if Mi,d = 0 then
9           break-for;
10        end
11       end
12       traili ← d - k
13     end
14   end
15   for k = k to k + max(trail) do
16     VMINk = argmaxj ∈ C'(trail);
17   end
18 end

```

Proof The total number of vectors \mathbf{V} equals to the total number of combinations through which the system can find at least one channel available in each time-slot. Since on average there are θ channels available at each time-slot, the total number of combinations available from time-unit = 0 to time-unit = η , equals $\theta \cdot \theta \cdot \dots \cdot \eta + 1$ times i.e. $\theta^{\eta+1}$. Thus, the search space is exponential.

Lemma 4.2 *The algorithm based on the greedy approach of selecting the longest consecutive free time-slots provides an optimal solution to the minimal channel switching problem. In other words there exist no other vector \mathbf{U} that has smaller number of channel switching than that obtained by the Algorithm 4.1. The proof is given in Appendix A.*

4.3.4 Maximum Throughput Requirement

Some bandwidth intensive applications like peer-to-peer (P2P) file sharing require high throughput across 24 h. Normally, these applications are unconstrained from the perspective of channel selection and the only requirement is high bandwidth. Therefore, an un-constrained channel selection technique with the aim of maximizing throughput across 24 h is provided. Maximum throughput tends to provide channel selection strategy in such a way that the system will select those channels which provide highest bandwidth. Therefore the solution vector to this problem should provide the list of channels which would provide maximum bandwidth over a given time-slot. The maximum throughput vector \mathbf{VTH} can be calculated as shown in Algorithm 4.2.

The algorithm first sorts the IDs of the channel in such a way that the lowest index has the highest bandwidth. The mapping is done internally from the older index to the newly assigned index, and is not shown here. The algorithm then selects the channel which has maximum bandwidth over a given time-slot and enter its ID into the vector \mathbf{VTH} .

Algorithm 4.2: Algorithm for obtaining maximum throughput vector

```

1 Rearrange the rows representing the channels in the matrix M such that lowest index has the
  highest bandwidth i.e.  $B_i > B_{i+1}, \forall i \in C'$ ;
2  $\mathbf{VTH}_k \leftarrow \{0\}, \forall k \in \eta'$ ;
3 for  $k = 0$  to  $\eta$  do
4   for  $i = 1$  to  $C$  do
5     if  $M_{i,k} = 1$  then
6        $\mathbf{VTH}_k \leftarrow i$ ;
7       break-for;
8     end
9   end
10 end
```

4.3.5 Intermediate Solution to Provide High Throughput Along with Minimal Channel Switch Requirement

The previous two approaches provide either the minimum number of channel switching or the highest throughput. Sometimes, however it is required to have a solution in between i.e. to provide a vector consisting of channel IDs such that if the system follows the list, then it would provide an intermediate solution which lies in-between the two approaches. Throughput loss caused by channel switching is not considered, as switching to a channel having a high bandwidth can easily compensate for the loss. Therefore the aim of this channel selection mechanism is to find a solution

vector such that it provides bounded channel switching (for bounded packet loss, jitter and delay induced by channel switching) along with the guaranteed throughput. This problem lies in the domain of NP-complete problems, and called as the High Throughput Minimal Switching (HTMS) problem.

Complexity Analysis of HTMS Problem

The optimization problem of HTMS is converted into decision problem to define its complexity. Every optimization problem can be converted to a decision problem [24], and the decision problem corresponding to HTMS problem is defined as follows:

Definition 4.1 *Given a channel availability matrix \mathbf{M} , and numbers ω and δ , does there exist a vector \mathbf{V} which satisfies all the constraints of the HTMS problem along with the condition that total number of switching $< \delta$ and bandwidth $> \omega$?*

Lemma 4.3 *HTMS decision problem is in NP. The proof is given in Appendix B.*

Lemma 4.4 *HTMS decision problem is NP-Hard. The proof is given in Appendix C.*

Theorem 4.1 *HTMS decision problem is NP-complete problem.*

Proof From Lemmas 4.3 and 4.4, it can be concluded that HTMS problem is NP-complete problem.

Heuristic Algorithm for HTMS problem

Since HTMS problem is NP-complete problem, a simple heuristic approach for its sub-optimal solution is provided in Algorithm 4.3. The algorithm first selects the minimal channel switching vector and then tries to satisfy the goal of guaranteed bandwidth. This is done by selecting the channels which were previously not selected and have higher bandwidth than that of the currently selected channel. The allowed switching and guaranteed bandwidth is represented by δ and ω respectively.

The algorithm first obtains the minimal channel switching vector \mathbf{VMIN} from Algorithm 4.1. It then updates the vector with the channel IDs which have higher bandwidth than the currently selected channels in \mathbf{VMIN} (lines 5–12). The *while* loop iterates until all channels have been checked (lines 14–16) or the bandwidth desired is obtained (line 3) and hence guarantees the convergence.

4.3.6 Complexity Analysis of Algorithms

The worst case scenarios for all algorithms are observed. Algorithm 4.1 has one parent *for* loop (lines 3–18) with two child *for* loops (lines 4–14 and lines 15–17). The running length of the parent loop is η and that of child loops are C and α , whereas the parameter α is variable and its value depends on the parameter η . The child loop (lines 4–14) with running length C is nested with another child loop (lines 7–11) with

Algorithm 4.3: Algorithm for intermediate channel selection for HTMS problem. The function $band_sum()$ returns the overall bandwidth of the system.

```

1 Obtain VMIN from Algorithm 4.1
2  $i = 1$ ;
3 while  $\omega \leq band\_sum(\mathbf{VMIN})$  do
4    $flag = 0$ ;
5   for  $k = 0$  to  $\eta$  do
6     if  $M_{i,k} = 1$  &  $\mathbf{VMIN}_k > i$  then
7        $\mathbf{VMIN}_k = i$  ;
8     end
9     if  $\omega \leq band\_sum(\mathbf{VMIN})$  then
10       $flag=1$ ; break-for;
11    end
12  end
13   $i = i + 1$ ;
14  if  $i = C$  ||  $flag = 1$  then
15    break-while;
16  end
17 end

```

running length α . The outermost parent loop (line 3) is complemented with the help of innermost for loop (line 7) and the other child for loop (line 15). Thus the running length of outermost loop is complemented. The complexity of $max()$ function and $argmax()$ function for current application is assumed to be $O(n)$, where n is the length of the vectors for these functions. Thus the complexity obtained of Algorithm 4.1 is $O(\eta(C + n))$ or $O(\eta^2)$.

Algorithm 4.2 has two nested *for* loops with running length of η and C . Thus the complexity of the algorithm is $O(\eta C)$ or $O(\eta^2)$.

Algorithm 4.3 obtains the vector **VMIN** using Algorithm 4.1 and then updates the vector to satisfy the bandwidth guarantee. The outer *while* loop has running length of C and the inner *for* loop has running length of $\eta + 1$. Thus the complexity of this algorithm equals, the complexity of Algorithm 4.1 $+O(C(\eta + 1))$ or $O(\eta^2)$.

4.3.7 Spectrum Usage

When the system has obtained the spectral opportunities for 24 h and channel selection vectors (following the user constraints), the selected channel can be utilized for communication purposes. Spectrum usage consists of normal communication between nodes in CRN using the channels provided by the vectors obtained with different channel selection algorithms. An error counter $\xi_i, \forall i \in C'$ is associated with each channel. Each time before a packet is transmitted, fast sensing [25] is performed on the channel to detect the presence of any PU. If a PU is present on the channel, then that information is updated in the global channel availability matrix **M** and the error counter associated with that channel is increased by 1. If the error counter reaches

the threshold Ψ , then that channel (row) information is removed from the channel availability matrix \mathbf{M} . If the total error on all channels crosses the threshold χ , then the system switches onto Selection on the go approach. The parameters Ψ and χ are user selectable and depends upon the channel environment.

4.3.8 Performance Analysis

Since deterministic time pattern of the channels are being utilized, this model is initialized for the first 24 h of the system, after that the recorded state is used for the next 24 h. The given scheme is evaluated using discrete event simulation in Matlab which simulates the network up to frame level. The channel availability parameters (discussed in Sect. 3.4.1) α and β are set to 0.1, so as to model the channels whose availability do not change abruptly. A low value of α and β means that the system is likely to stay either in ON state or in OFF state most of the time without state transition. The bandwidth (data rate) of the channel is modelled as to be linear with the number of channels used, i.e. the minimum bandwidth of a channel is given a value of 2 Megabit per minute (Mbps) and the maximum bandwidth of the channel is $2 * C$, where C is the maximum number of channels used for the simulation. Therefore the bandwidth is distributed linearly from Channel 1 to Channel C as 2 to $2 * C$ Mbps. The error counter $\xi_i, \forall i \in C'$ is set to 10.

The discussed techniques are compared against three other channel selection techniques: stochastic channel selection (SCS) [26], automatic channel selection (ACS) [27], and minimum variance-based spectrum decision single selection (MVSD-SS) [23]. SCS [26] utilizes learning automata and aims at reducing the number of channel switching by selecting a channel which maximizes the probability of successful transmission. ACS [27] aims at selecting a channel that provides guaranteed bandwidth and fairness to the system with the help of measured primary user activity on the channel. MVSD-SS [23] aims at selecting a channel that increases the bandwidth utilization of the system. For a fair comparison, some of the constraints have been removed which were utilized in the literature. The given channel selection techniques utilize the infrastructure version of the media access scheme discussed in Chap. 5. Each data point in the graph is an average of 100 runs while the unit of bandwidth and sampling interval are Mbps and minute respectively.

The algorithms is evaluated on the basis of following performances:

1. Overall Bandwidth of the system
2. Overall Channel switching required.

Overall bandwidth of the system refers to the bandwidth obtained by the system in 24 h. At some point the bandwidth obtained might be less and sometime might be the high. The bandwidths are consolidated by summing them up for 24 h. Overall channel switching required refers to the total switching needed for the 24 h operation of the system.

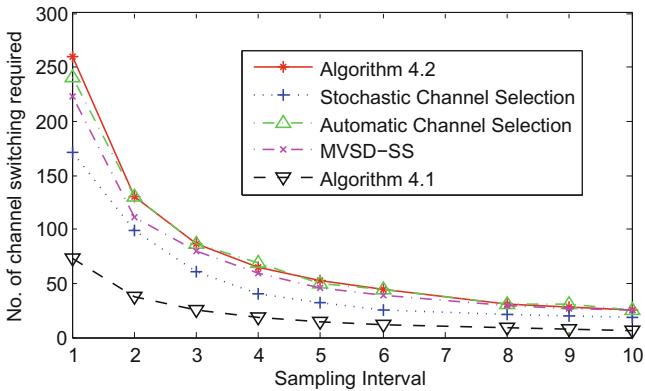


Fig. 4.6 Number of channel switching required versus sampling interval (x minutes)

Figure 4.6 shows sampling interval versus number of channel switching required for overall duration of 24 h with $C = 10$. It can be seen from the graph that Algorithm 4.1 provides minimal channel switching which is better than SCS technique. Although SCS technique aims at minimizing the channel switching, it lacks the knowledge of spectral opportunity ahead of time and thus suffers from degraded performance. All other techniques do not consider channel switching in their decision making process and therefore suffers a large amount of channel switching. Figure 4.6 also implies that Algorithm 4.2 is not good for minimal channel switching as it has the worst performance.

Figure 4.7 shows the bandwidth obtained versus sampling interval with $C = 10$. Results show that the bandwidth of the system is independent of the sampling interval used. It is also noted that Algorithm 4.2 provides the highest bandwidth among all

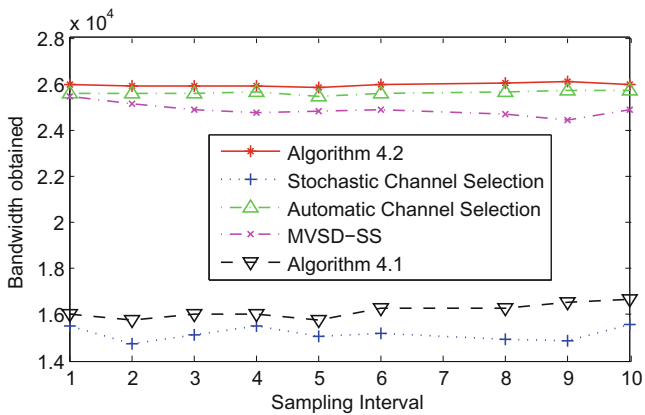


Fig. 4.7 Bandwidth versus sampling interval (x minutes)

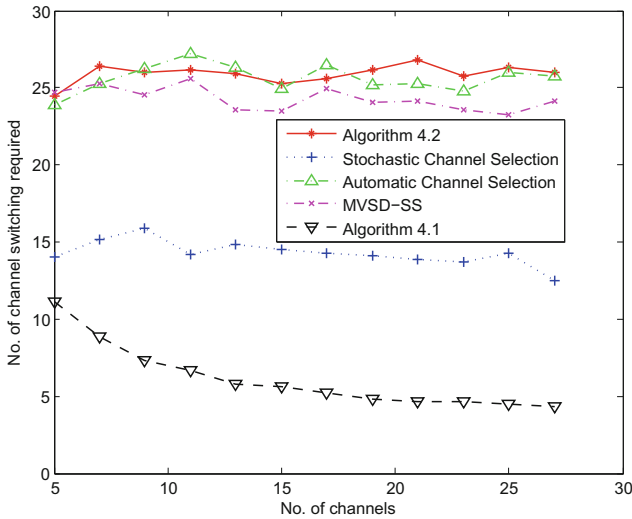


Fig. 4.8 Number of channel switching required versus number of channels utilized in the system

other techniques. This is the maximum unconstrained bandwidth that can be obtained by the system. Despite the removal of constraints from ACS and MVSD-SS to select high bandwidth channels for a fair comparison, these techniques do not have the knowledge of spectral opportunities ahead of time and suffer from performance loss. SCS aims at reducing the number of channel switching and does not account for throughput. Therefore its performance is near to that of Algorithm 4.1.

Figure 4.8 shows the number of channel switching required versus number of channels in the system with sampling interval = 10 min. It might be noted that the number of channel switching required in Algorithm 4.2 and all other techniques except Algorithm 4.1 is insensitive to the number of channels used in the system. The number of channel switching decreases with an increase in the number of channels in the system for Algorithm 4.1. With an increase in number of channels in the system, the number of combinations of vector \mathbf{V} increases in the channel availability matrix and hence the switching decreases.

Figure 4.9 shows bandwidth obtained versus number of channels used in the system with sampling interval = 10. Note that Algorithm 4.2 obtains the highest bandwidth while other techniques (ACS and MVSD-SS) perform marginally poorer. The difference becomes significant when the number of channels used in the system becomes large. The performance of the Algorithm 4.1 and SCS technique is similar. The bandwidth of each scheme increases linearly because the bandwidth of each channel also increases linearly with the number of channels in the system (bandwidth is distributed linearly from channel 1 to channel C as 2 to $2 * C$ Mbps).

Figure 4.10a, b show the performance for Algorithm 4.3 with guaranteed throughput (bandwidth) and maximum allowable switching. For the purpose of generalization, the rule of thumb “more than half” is used and assumed that the number of

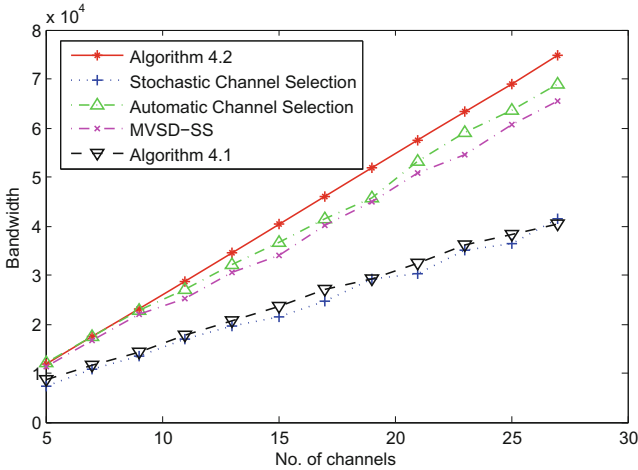


Fig. 4.9 Bandwidth versus number of channels

switching allowed per slot = 0.1 and total bandwidth available per slot = 60% of the maximum bandwidth of the channel. The parameters α and β are set to 0.1.

Figure 4.10a shows number of channel switching allowed versus number of channels in the system with sampling interval = 10. The figure shows that the algorithm obtains channel switching lower than the allowed value. In addition the graph shows that there is no particular relation between the number of channels used in the system with the number of channel switching required.

Figure 4.10b shows bandwidth obtained in the system versus number of channels used in the system with sampling interval = 10. Note that Algorithm 4.3 satisfies the bandwidth guarantee required by the system. The bandwidth of each scheme increases linearly because the bandwidth of each channel also increases linearly with the number of channels in the system.

Figure 4.10c shows number of channel switching allowed versus sampling interval for an overall duration of 24 h with $C = 10$. Algorithm 4.3 provides channel switching lower than that of the allowed value. The allowed switching (δ) decreases with the increase in sampling interval value and so is the value obtained by Algorithm 4.3.

Figure 4.10d shows bandwidth obtained versus sampling interval with $C = 10$. Note that the algorithm guarantees the minimum bandwidth to the system. Here the bandwidth obtained from Algorithm 4.3 increases in an up down fashion. The randomness is due to the random channel availability and the increase in overall bandwidth is due to the selection of channels having high bandwidth and high switching. Figure 4.10c shows a decrease in the difference between the switching obtained by Algorithm 4.3 and the allowed switching, and is accounted for increase in bandwidth as shown in Fig. 4.10d.

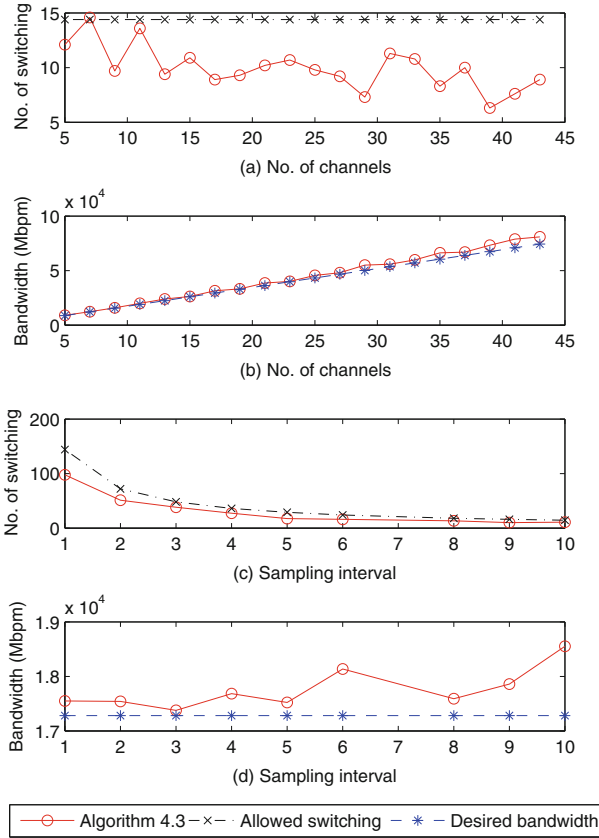


Fig. 4.10 Performance analysis of Algorithm 4.3 with respect to desired bandwidth and allowed switching

4.3.9 Ad-hoc Mode Operation

The channel selection techniques discussed in the previous section assumed the system model of Single Network in Infrastructure Mode. The discussed channel selection techniques can be modified to apply for networks operating in ad-hoc mode. To do so, additional protocol is required to achieve the goal of global channel availability matrix which is same for all the CR users in the ad-hoc network. If every CR user has same channel availability matrix, then the channel selection algorithm with desired constraints operating over all CR users will result in common channel selection, which can be utilized by an ad-hoc media access scheme for communication purposes.

4.4 Channel Selection in Stochastic Environment

If the error counter associated with the channel crosses a threshold (ξ), then that channel is classified as stochastic channel. If the total error counter for all channels crosses a given threshold (Ψ), then the system switches onto Selection on the go approach. The system may utilize any of the stochastic channel selection technique available in the literature. Most of these channel selection technique are based on statistical learning, which introduces complexity to the system. Here a stochastic channel selection technique is discussed which map packets to channels on the basis of a tabular mapping [28]. This tabular mapping reduces the complexity of the system as well as facilitates fast channel selection.

A generalization is made regarding the user capability of multi-tasking and running multiple different applications, and hence it can be assumed that a user can generate traffic with multiple priorities. Therefore a user is not classified as real time or non-real time, instead data packets are classified as real time or non-real time. Here, a spectrum decision scheme for cognitive radio network in stochastic channel environment is discussed which divides the data traffic into four priorities on the basis of the QoS requirements and channels into eight categories on the basis of their bandwidth, channel availability and primary user (PU) fluctuation (discussed later).

4.4.1 System Model

The system comprises of N open spectrum CR users and utilizes a common control channel which may be selected from one of the ISM bands or a channel with high availability. Each of the C channels utilized in the system has a bandwidth of B_i , $\forall i \in \{1, \dots, C\}$. Each channel is associated with an ID and counter for its identification and load balancing purposes (discussed later) respectively. Therefore a total of C counters is maintained by each node in the network. Each CR user is in communication range with all other CR users and the transmission range of a PU is more than that of the CR user.

As discussed in Chap. 3, the user traffic is classified into four different priorities as shown in Table 3.1. It is assumed that the application layer is capable of tagging (marking) the packets associated with a particular traffic. The channel availability is modelled as two-state Markov chain as discussed in Sect. 3.4.1. It is assumed that each CR user statistically learns the channel transition probabilities from their previous sensing. Therefore each CR user knows the channel transition probabilities for all C channels.

4.4.2 Communication Segment

A communication segment (t, r) is defined as a pair of transmitting and receiving CR users for a session. A communication segment operates over a single channel and does not spawn over multiple channels during a session. Also a communication segment can carry only a single class (type/priority) of data and therefore has the priority depending upon the type of data traffic it is carrying. This section will use the word ‘connection’ interchangeably with the word ‘communication-segment’.

4.4.3 Spectrum Decision and Mapping of Packets

The spectrum decision involves selection of channels by CR users depending upon the data packet needed to be sent. All C channels are classified into eight types on the basis of bandwidth, channel availability and PU fluctuation. The calculation of channel availability and fluctuation is discussed later. Table 4.2 shows the tagging of these channels. The term ‘High’ and ‘Low’ are subjective and may vary depending upon application and licensed channels. Voice packets need those channels which have higher availability and low fluctuation. An example would be if $\alpha = 0.9$ and $\beta = 0.9$; the channel is although available with probability of 0.5 but it alternates at each time slot, resulting in jittering at each step. Therefore PU fluctuation need to be taken care of by using fluctuation parameter. The fluctuation parameter δ is defined as:

$$\delta_i = \frac{\alpha_i + \beta_i}{2} \quad \forall i \in C' \quad (4.7)$$

The fluctuation parameter $\delta \in [0, 1]$ tells how much the PUs’ presence fluctuate (varies) on the given channel. A higher value of δ represents high fluctuation while

Table 4.2 Tagging and classification of primary licensed channels

Bandwidth	Channel availability	Primary user fluctuation	Tag
High	High	High	0
High	High	Low	1
High	Low	High	2
High	Low	Low	3
Low	High	High	4
Low	High	Low	5
Low	Low	High	6
Low	Low	Low	7

Table 4.3 Mapping of data packets to channel requirements

Access category	Bandwidth	Channel availability	Primary user fluctuation
Voice	High/Low	High	Low
Video	High	High	High/Low
Best effort	Low	High/Low	Low
	High	High/Low	High/low
Background	Low	High/Low	High/Low

a low value of δ represents low fluctuation. Fluctuation parameter only represent the PU fluctuation and does not represents the actual availability of the channel.

Therefore high availability of the channel and minimal fluctuation of the PU on that channel is desired for the system supporting real time traffic like voice. A quick analysis of Eqs. 3.2 and 4.7 indicates that these criteria can be met with low value of β and equal value of α and β . The low value of β decreases the fluctuation of the PU on the channel while an equal value of α and β increases the channel availability.

Once the quantity ‘High’ and ‘Low’ are associated with bandwidth, availability of all the channels along with the PUs fluctuation on it, the CSMU can define mapping function for mapping packets to channels. The mapping function is defined in Table 4.3. The table shows that the voice packets can choose channels with low bandwidth, as the bandwidth of voice packet is small. The guarantee is made in terms of channel availability and PU fluctuation on the channel. Similarly, video packets can choose a channel with high PU fluctuation as videos can be buffered if needed. Background traffic can only choose channels with low bandwidth, whereas Best Effort data considerably overlaps with other data priorities. The reason is that if video or voice traffic is not present then best-effort traffic can use the channel. Each traffic starts with selecting low bandwidth channel in its class i.e. a voice and best-effort traffic will first try to choose a low bandwidth channel and if not available then move to next higher bandwidth channel.

4.4.4 Load Balancing

Since there are eight categories of channel as shown in Table 4.2, the numbers of channel in a single category is represented by $\gamma_i, \forall i \in C'$. Therefore $\sum_{i=0}^7 \gamma_i = C$. It may be possible that some category may not have any channels in them. Let $\xi_k, \forall k \in \{0..7\}$ represent the set of channels IDs in a given category. The purpose of the load balancing system is to uniformly distribute the communication segments within the channels of a given category. Algorithm 4.4 provides the method of load balancing to the system while doing the spectrum decision for a communication segment. The index for ξ is dropped because the algorithm represents the load balancing within a

category. The variable array **cntr** maintains the list of counter for all the channels, and it is used to count the activities on a channel. The counter value for each channel is set to zero during the initialization of the system. The sensing report unit of the framework updates the value of the counter based on the external activity (PU or any other CR user) whereas the counter variable is also updated inside the algorithm block to count for the internal activity. The higher the value of the counter for a channel, the higher the activities it has got on it. The goal of the algorithm is to select a channel within a given category of channels such that it has minimal activity on it. If a channel with minimal number of activity is selected and used, a report is generated with the help of success/failure unit of the framework (lines 6–8). If the success reported is negative, then that channel is removed from the category for the current session and minimal activity channel is selected from the rest of the channels (lines 9–12). If all channels are removed then the algorithm starts from the beginning. The counter is reset to 0 once it reaches a threshold. The threshold can vary according to the application.

4.4.5 Performance Analysis

The terms ‘High’ and ‘Low’ for bandwidth of a channel, channel availability and PU fluctuation on a given channel are defined in order to obtain the performance analysis. Any channel having bandwidth greater than or equal to average bandwidth is called as ‘High’ bandwidth or otherwise ‘Low’ bandwidth. The average bandwidth is calculated using Eq. 4.8.

$$avg_band = \frac{\sum_{i=1}^C B_i}{C} \quad (4.8)$$

Algorithm 4.4: Algorithm for load balancing in a given category of channels

```

1 Initialize :  $\psi = \xi$ ;
2 Label :
3 if  $\psi = \{\}$  then
4   |  $\psi = \xi$ ;
5 end
6  $ch\_id = argmin_{i \in \psi} (cntr_i)$ ;
7  $cntr_i = cntr_i + 1$ ;
8  $report = success\_report(ch\_id)$ ;
9 if  $report = 0$  then
10  |  $\psi = \psi - \{ch\_id\}$ ;
11  | goto Label;
12 end
```

Therefore,

$$Bandwidth_i = \begin{cases} High, & \text{If } Bandwidth_i \geq avg_band \\ Low, & \text{otherwise} \end{cases} \quad (4.9)$$

Channel availability and PU fluctuation is similarly assigned as ‘High’ and ‘Low’ as follows.

$$avg_ch_av = \frac{\sum_{i=1}^C \Pi_i}{C} \quad (4.10)$$

$$ChannelAvailability_i = \begin{cases} High, & \text{If } \Pi_i \geq avg_ch_av \\ Low, & \text{otherwise} \end{cases} \quad (4.11)$$

$$avg_pu_fl = \frac{\sum_{i=1}^C \delta_i}{C} \quad (4.12)$$

$$PuFluctuation_i = \begin{cases} High, & \text{If } \delta_i \geq avg_pu_fl \\ Low, & \text{otherwise} \end{cases} \quad (4.13)$$

The discussed channel selection scheme is implemented on top of the MAC protocol discussed in Chap. 5. The discussed scheme is evaluated using discrete event simulation in Matlab which simulates the network up to frame level. The performance of the discussed framework is evaluated in terms of connection fail ratio, number of connections failed and number of channels used. Connection fail ratio is calculated by

$$\text{Connection fail ratio} = \frac{\text{number of failed connections}}{\text{number of connections initiated}}$$

The parameters α and β are assumed to be uniformly distributed over the range [0, 1] i.e. the transition probabilities of the channels are uniformly distributed over the range [0, 1]. The bandwidth of the channel is modelled to be linear with the number of channels used, i.e. the minimum bandwidth of a channel is given a value of 50 Kbps and the maximum bandwidth of the channel is $50 * C$ Kbps, where C is the maximum number of channels used for the simulation. Therefore the bandwidth is distributed linearly from channel 1 to channel C as 50 to $50 * C$ Kbps.

Figure 4.11 shows the graph of connection fail ratio versus the total number of channels used the system. The number of voice communication segment, video communication segment, best-effort communication segment and background traffic communication segment used are 2, 5, 30 and 13 respectively giving the percentage ratio of 4, 10, 60 and 26 % for voice, video, best-effort and background traffic communication segments respectively. The number of channels is varied from 10 to 40. The connection fail ratio decreases with increase in number of channels for all types of communication segments. Since the number of voice traffic is small and the number of channels is adequate to satisfy the requirement, the connection fail ratio for

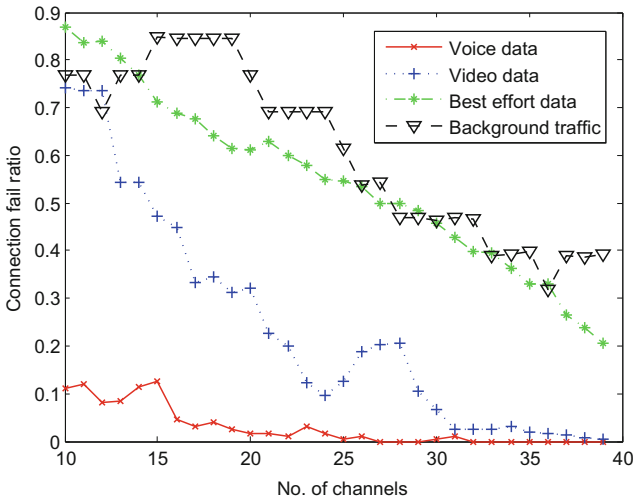


Fig. 4.11 Connection fail ratio versus number of channels

voice traffic is smallest. Background traffic shows higher blockage at start because most of the channels are used by the best-effort traffic (since there is overlapping of the channels for these two traffic types in mapping Table 4.3).

Figure 4.12 shows the graph of number of failed connections versus total number of communication segments for $C = 50$. The total number of communication segments comprises 4, 10, 60 and 26% of voice, video, best-effort and background traffic communication segments respectively. From the graph it is clear that the num-

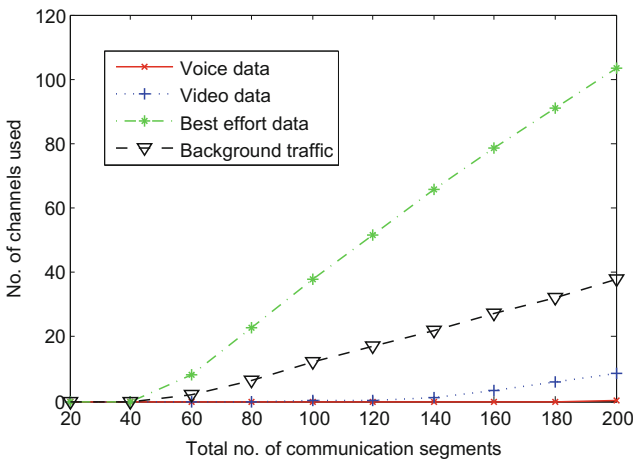


Fig. 4.12 Number of connections failed versus total number of communication segments

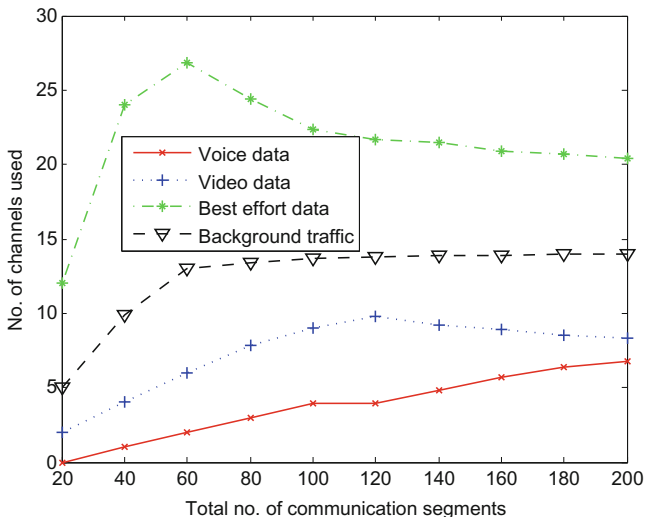


Fig. 4.13 Number of channels used versus total number of communication segments

ber of connections will fail as the number of connections (communication segments) increases. A steep increase can be seen when total number of communication segments equals 60, which can be explained by the fact that channels become busy after this point leading to failure in new connections. There is steep increase in the best-effort segment losses because of the large share of the total segments and some of the channels mapped to this traffic type is used by higher order traffic too (Table 4.3).

Figure 4.13 shows the load balancing in the system by taking in to account the number of channels used by a traffic type with respect to total number of communication segments. The total number of communication segments comprises 4, 10, 60 and 26% of voice, video, best-effort and background traffic communication segments respectively. The total number of channels used is 50. It is clear from the graph that the number of channels used for all traffic types increases till total number of communication segments equals 60. After which there are no channels to fulfil the upcoming requests. As the number of communication segment increases, the number of voice and video data communication segments also increase leading to more use of the channels mapped by voice and video data. Thus there is decrease in the channel use by the web traffic data which were overlapped with the voice and video data (Table 4.3).

4.5 Discussion

The performance of a network depends upon the bandwidth and quality of the wireless channel in which it operates. A channel can be characterized on the basis of its availability; and on the basis of observation, channels may have deterministic/non-deterministic usage pattern. Channels like TV and FM bands can be utilized for the purpose of modelling deterministic behavior of licensed users. If channels do not have any deterministic usage pattern, then stochastic channel selection techniques (Selection on the go approach) are utilized.

In this chapter we have discussed a reconfigurable channel selection technique which initially starts with deterministic channel selection techniques for channels with deterministic usage behavior. If channels do not follow a deterministic behavior, the increase in error counter forces the system to utilize another discussed channel selection technique for stochastic environment. Three different algorithms have been discussed in deterministic channel environment targeting different aspects. These algorithms solve the problem of minimal channel switching and high throughput. Channel switching is reduced to mitigate the effects of packet losses, delays and jitter caused by channel switching. One of the discussed algorithm provides an intermediate result between minimal channel switching and maximum throughput, solving the high throughput minimal channel switching problem. In stochastic channel environment, the given technique divides channel into into eight different categories based on bandwidth, channel availability and PU fluctuation parameter. A mapping is provided from the data traffic priority to the channel type for the purpose of channel selection, along with a load balancing algorithm to balance the traffic on different channels.

Performance analysis of channel selection techniques in deterministic environment is done with three other channel selection techniques. Simulation result shows that the performance of the discussed algorithms based on the given system model approach provides better performance of the overall system. In addition the given model can provide a trade-off between minimal channel switching and maximum bandwidth. Similarly, in stochastic environment, the performance analysis shows that the channel selection can be utilized for obtaining different priorities with effective load distribution. Although this chapter covered the schemes for spectrum selection in deterministic and stochastic channel environment, it misses the important parameter of energy consumption.

References

1. Mishra, V., Tong, L. C., & Syin, C. (2013). Deterministic time pattern based channel selection in cognitive radio network. In *The International Conference on Information Networking 2013 (ICOIN)* (pp. 169–174).
2. Erpek, T., Lofquist, M., & Patton, K. (2007). Spectrum occupancy measurements loring commerce centre limestone, Maine September 18–20, 2007. Shared Spectrum Company Report.

3. Erpek, T., Steadman, K., & Jones, D. (2007). Dublin Ireland spectrum occupancy measurements collected on April 16–18, 2007. Shared Spectrum Company Report.
4. McHenry, M. A., McCloskey, D., Roberson, D., & MacDonald, J. T. (2005). Spectrum occupancy measurements Chicago, Illinois November 16–18, 2005. Shared Spectrum Company Report.
5. McHenry, M. A., & Steadman, K. (2005). Spectrum occupancy measurements location 1 of 6: Riverbend park, Great Falls, Virginia April 7, 2004. Shared Spectrum Company Report.
6. McHenry, M. A., & Steadman, K. (2005). Spectrum occupancy measurements location 2 of 6: Tyson's square center, Vienna, Virginia April 9, 2004. Shared Spectrum Company Report.
7. McHenry, M. A., & Chunduri, S. (2005). Spectrum occupancy measurements location 3 of 6: National science foundation building roof April 16, 2004. Shared Spectrum Company Report, Revision 2.
8. McHenry, M. A., & McCloskey, D. (2005). Spectrum occupancy measurements location 4 of 6: Republican national convention, New York City, New York August 30, 2004–September 3, 2004. Shared Spectrum Company Report, Revision 2.
9. McHenry, M. A., & Steadman, K. (2005). Spectrum occupancy measurements location 5 of 6: National radio astronomy observatory (NRAO) Green Bank, West Virginia October 10–11, 2004. Shared Spectrum Company Report, Revision 3.
10. McHenry, M. A., & Steadman, K. (2005). Spectrum occupancy measurements: Shared spectrum building roof, Vienna, Virginia, December 15–16, 2004. Shared Spectrum Company Report.
11. Valenta, V., Marsalek, R., Baudoin, G., Villegas, M., Suarez, M., & Robert, F. (2010). Survey on spectrum utilization in europe: Measurements, analyses and observations. In *Proceedings of the Fifth International Conference on Cognitive Radio Oriented Wireless Networks Communications (CROWNCOM)* (pp. 1–5).
12. Patil, K., Skouby, K., Chandra, A., & Prasad, R. (2011). Spectrum occupancy statistics in the context of cognitive radio. In *14th International Symposium on Wireless Personal Multimedia Communications (WPMC)* (pp. 1–5).
13. Sanders, F. (1998). Broadband spectrum surveys in Denver, CO, San Diego, CA, and Los Angeles, CA: Methodology, analysis, and comparative results. In *IEEE International Symposium on Electromagnetic Compatibility* (Vol. 2, pp. 988–993).
14. Chiang, R., Rowe, G., & Sowerby, K. (2007). A quantitative analysis of spectral occupancy measurements for cognitive radio. In *IEEE 65th Vehicular Technology Conference, 2007. VTC2007-Spring* (pp. 3016–3020).
15. Wellens, M., Wu, J., & Mahonen, P. (2007). Evaluation of spectrum occupancy in indoor and outdoor scenario in the context of cognitive radio. In *2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2007. CROWNCOM* (pp. 420–427).
16. Yin, S., Chen, D., Zhang, Q., Liu, M., & Li, S. (2012). Mining spectrum usage data: A large-scale spectrum measurement study. *IEEE Transactions on Mobile Computing*, 11, 1033–1046.
17. Lopez-Benitez, M., Umbert, A., & Casadevall, F. (2009). Evaluation of spectrum occupancy in spain for cognitive radio applications. In *IEEE 69th Vehicular Technology Conference, 2009. VTC Spring* (pp. 1–5).
18. Bao, V. N. Q., Cuong, L. Q., Phu, L. Q., Thuan, T. D., Quy, N. T., & Trung, L. M. (2011). Vietnam spectrum occupancy measurements and analysis for cognitive radio applications. In *2011 International Conference on Advanced Technologies for Communications (ATC)* (pp. 135–143).
19. Islam, M., Koh, C., Oh, S. W., Qing, X., Lai, Y., Wang, C., et al. (2008). Spectrum survey in Singapore: Occupancy measurements and analyses. In *3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2008. CROWNCOM* (pp. 1–7).
20. Cordeiro, C., Challapali, K., Birru, D., & Sai Shankar, N. (2005). IEEE 802.22: The first worldwide wireless standard based on cognitive radios. In *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN* (pp. 328–337).
21. Mishra, V., Lau, C. T., & Chan, S. (2013). Reconfigurable channel selection technique for cognitive radio network with heterogeneous primary bands. *IEEE Transactions on Vehicular Technology*, 62, 3942–3952.

22. Song, Y., & Xie, J. (2010). Common hopping based proactive spectrum handoff in cognitive radio ad hoc networks. In *IEEE Global Telecommunications Conference (GLOBECOM 2010)* (pp. 1–5).
23. Lee, W.-Y., & Akyldiz, I. (2011). A spectrum decision framework for cognitive radio networks. *IEEE Transactions on Mobile Computing, 10*, 161–174.
24. Garey, M. R., & Johnson, D. S. (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman & Co.
25. Working group on wireless regional area networks, IEEE Std. 802.22, 2011. <http://www.ieee802.org/22/>.
26. Song, Y., Fang, Y., & Zhang, Y. (2007). Stochastic channel selection in cognitive radio networks. In *IEEE Global Telecommunications Conference, 2007. GLOBECOM'07* (pp. 4878–4882).
27. Bennai, M., Sydor, J., & Rahman, M. (2010). Automatic channel selection for cognitive radio systems. In *IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)* (pp. 1831–1835).
28. Mishra, V., Tong, L. C., & Syin, C. (2012). Qos based spectrum decision framework for cognitive radio networks. In *18th IEEE International Conference on Networks (ICON)* (pp. 18–23).