A Study of Timing Constraints and SAS Overload of SAS-CBSD Protocol in the CBRS Band

Anirudha Sahoo, Naceur El Ouni* and Vineet Shenoy*
National Institute of Standards and Technology
Email: anirudha.sahoo@nist.gov, naceuramine@gmail.com, vineet.r.shenoy@rutgers.edu

***Abstract*—In the Citizens Broadband Radio Service (CBRS) band, the Federal Communications Commission (FCC) has set stringent timing constraints for the lower tier users to vacate the channel on which an incumbent shipborne radar appears. The standards body formulating various specifications for the CBRS operation has taken these timing constraints into consideration in the Spectrum Access System (SAS) – CBRS Device (CBSD) protocol. A transmitting CBSD continually heartbeats with its SAS. When required, the SAS sends commands to vacate a channel through these heartbeat messages. In this paper, we study the impact of the heartbeat interval on the CBRS system in terms of meeting the FCC timing constraints. We also study how the heartbeat interval can overload a SAS and how it can be used to determine the number of CBSDs a SAS can serve without causing unnecessary suspension of CBSD transmissions. We show the tradeoff between using a short heartbeat interval to meet the timing constraint early and the number of CBSDs that can be served by a SAS without causing unnecessary suspension of CBSD transmissions.**

I. Introduction

The Federal Communications Commission (FCC) recently published rules for commercial operators to use the 3.5 GHz band, also termed as Citizens Broadband Radio Service (CBRS) band, on a priority based sharing [1]. A Spectrum Access System (SAS) manages the use of spectrum in the CBRS band. A CBRS device (CBSD) has to get authorization from its managing SAS to use the spectrum and must vacate the spectrum when instructed by its SAS to do so. The communication protocol between a SAS and a CBSD has been standardized by the Spectrum Sharing Committee (SSC) of the Wireless Innovation Forum (WInnForum), commonly referred to as the *SAS-CBSD* protocol [2]. The CBRS band operationally is a three tiered system. The incumbents operate in tier-1 with the highest priority. The commercial operators may operate in tier-2 with medium priority or in tier-3 with lowest priority. One of the incumbents in the CBRS band is the shipborne Navy radars. When an incumbent Navy radar appears within the harmful interference range of deployed CBSDs, a SAS has to carefully identify the CBSDs which should be instructed to vacate the channel (in which the radar is operating) such that the interference to the radar receiver falls below a given threshold. The presence of Navy radars is detected by a set of sensors, known as Environment Sensing Capability (ESC) sensors, usually deployed along the coast of the US. The FCC rules have stringent timing requirements for the SAS and the CBSDs to protect incumbent radars from harmful interference. Once a SAS is notified of the presence of a Navy radar by an ESC, the SAS must ensure that the CBSDs, which may cause harmful interference to the radar, have vacated the channel within 300 s [1]. Once a CBSD has been instructed by the SAS to vacate the channel, the CBSD must do so within 60 s [1].

The SAS-CBSD protocol specification has been carefully designed to ensure that the timing requirements set forth by the FCC rules can be met [2]. A CBSD requests authorization to transmit on a channel (frequency range) by sending a grant request to the SAS. Each active grant in the SAS-CBSD protocol has a heartbeat mechanism through which the CBSD knows that the SAS is alive and vice-versa. A CBSD has to send a Heartbeat Request message to its managing SAS periodically for each of its active grants. How frequently a CBSD should send a Heartbeat Request message is decided by the SAS by setting the *heartbeatInterval* parameter. An actively transmitting Grant has a timer called *transmitExpireTime* timer. If and when this timer expires, the CBSD will have 60 s to turn off its transmission. Hence, to meet the end to end timing requirement of 300 s, the value of transmitExpireTime timer should not be more than 240 s for actively transmitting grants. The transmitExpireTime timer thus guards against violation of timing constraints in case communication between SAS and CBSD fails. A SAS can also change various protocol parameter values through the heartbeat messages to control the system timings. For example, the transmitExpireTime timer value and heartbeatInterval value can be changed through the hearbeat messages. All the request messages originate from the CBSDs and the SAS reponds to them with the corresponding response messages. Thus, when a SAS gets the notification from an ESC that an incumbent has appeared on a channel, the SAS has to wait for the next Heartbeat Request message to instruct the CBSD to vacate the channel via the corresponding Heartbeat Response message. There is a tradeoff between heartbeatInterval and processing load on the SAS. When the heartbeatInterval is small, the SAS can instruct the CBSD to vacate a channel sooner, but the processing load on the SAS is higher since the heartbeat rate is higher. On the other hand, when the heartbeatInterval is large, the processing load on the SAS is lower, but the SAS has to wait longer, on the average, to intsruct a CBSD to vacate a channel. The latter configuration can push the time a SAS takes to ensure that a CBSD vacates the channel close to the time limit of 300 s.

A SAS also needs to have adequate provisioning of processing resources so that it can handle CBSD request messages in

*The author was at National Institute of Standards and Technology when this work was done.

a timely manner. Since Heartbeat Request messages are the most frequent messages a SAS receives, their load dominates in determining the processing resource requirement of a SAS. If a Heartbeat Request message is dropped or sufficiently delayed due to lack of processing resources at a SAS, then the transmitExpireTime timer at the CBSD would expire. This would force the CBSD to suspend its transmission. This obviously is an unnecessary timeout caused by poor provisioning of processing resources in the SAS which leads to inefficient use of the spectrum.

Various efforts towards spectrum sharing in the US are summarized in [3]. The article also presents an example SAS architecture that facilitates tiered services in the CBRS band. The FCC rules on the CBRS operation were published in [1]. Based on the FCC rules, the Spectrum Sharing Committee (SSC) of Wireless Innovation Forum (WInnForum) has developed requirements for commercial operation in the CBRS band [4]. The WInnForum SSC has also developed the specification for the SAS-CBSD protocol [2]. Besides these, there has not been any work on the SAS-CBSD protocol that is publicly available.

To the best of our knowledge, there is no study on the SAS-CBSD protocol that is available in the public domain. Hence, the motivation behind this paper is two fold. First, we want to study the impact of heartbeatInterval on the performance of the CBRS system in terms of meeting the end to end timing constraint. Second, we want to study how message overload on a SAS impacts the performance of the CBRS system in terms of unnecessary timeouts of the transmitExpireTime timer which leads to suspension of CBSD transmission. To achieve these goals we have developed a basic simulator of the SAS-CBSD protocol. The protocol is implemented using the Omnet++ discrete event simulator package [5]. Our experiments were run for a CBRS system having very high number of CBSDs (up to 50 000) in the system. So, we believe the scale of simulation is similar to what we may see in practice. Our results show that a moderate mean hearbeatInterval of 150 s provides a good balance between handling the processing load and meeting the timing constraint. For a given SAS message service rate, our study also provides approximately how many CBSDs the SAS can serve without causing unnecessary timeouts (which would cause CBSDs to stop their transmissions). We present an approximate method of calculating the number of CBSDs a SAS can manage for a given request service rate, without causing unnecessary transmitExpireTime timeouts. This model can be used, for example, by SAS providers to decide when a new SAS process should be spawned to serve increasing number of CBSDs.

II. Simulation of SAS-CBSD Protocol

A. Brief Description of the SAS-CBSD Protocol

The SAS-CBSD protocol has two state machines: *Registration State Machine* and *Grant State Machine*. A CBSD starts out in the Unregistered state. When it registers with its SAS, it goes through the Registration State Machine as shown in Figure 1. If the registration is successful it transitions into Registered state. Once the CBSD is in Registered state, it is allowed to ask for spectrum grants. Spectrum grant is the process by which a CBSD asks the SAS for authorization to transmit in a particular frequency range at a particular transmission power and follows the Grant state machine shown

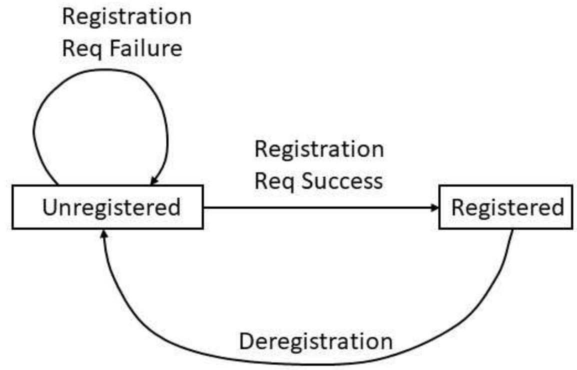

Fig. 1. Registration State Machine of CBRS (adapted from [2])

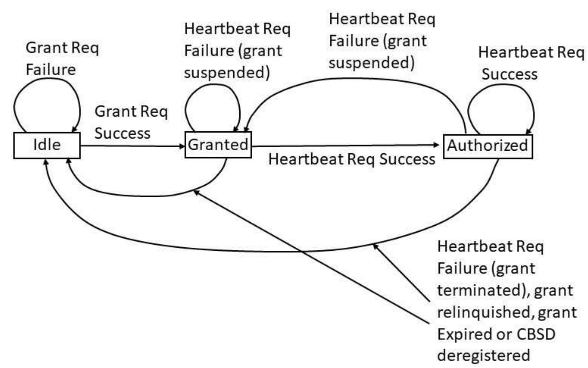

Fig. 2. Grant State Machine of CBRS (adapted from [2])

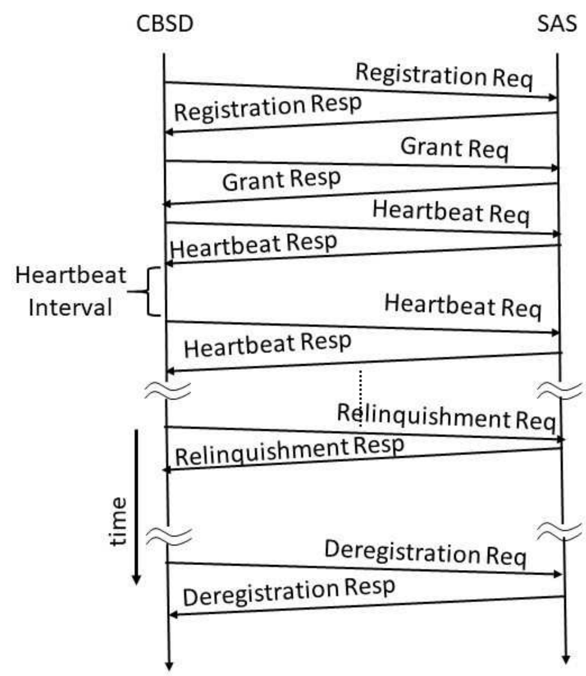

Fig. 3. A Typical Message Sequence Diagram of the SAS-CBSD Protocol

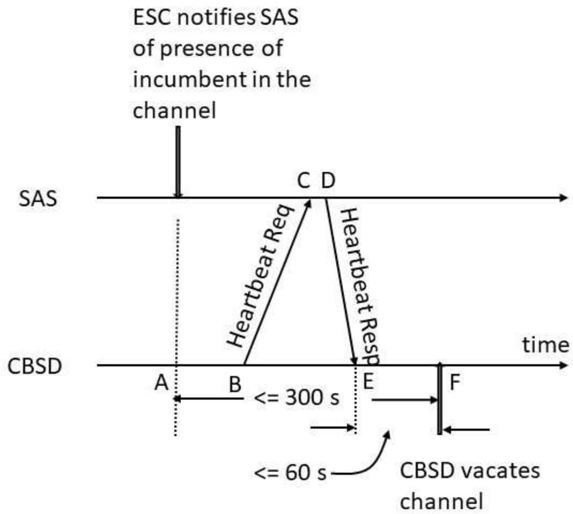

Fig. 4. Timing Constraint of CBRS

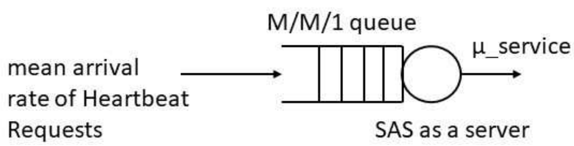

Fig. 5. SAS Message Processing Model

in Figure 2. A CBSD sends a Grant Request to its SAS while in the Idle state. The SAS checks if the grant does not cause harmful interference to other users (as per the Part 96 rules [1]), and if not, the CBSD is granted persmission and it goes into Granted state. If the Grant Request fails, the CBSD stays in the Idle state. A CBSD is not allowed to transmit while in the Granted state. It starts a heartbeat process while in the Granted state by sending a Heartbeat Request message. If the Heartbeat Request is successful, then the CBSD receives a successful Heartbeat Response from the SAS and moves to Authorized state, at which time it is allowed to transmit. A CBSD has to continue sending Heartbeat Requests periodically while in the Authorized state and can continue to transmit as long as it receives successful Heartbeat Responses. The SAS can ask the CBSD to stop transmission by indicating failure in the Heartbeat Response, in which case the CBSD moves to the Granted state from the Authorized state. A SAS may want to do this, for example, if an incumbent appears on the channel used by the CBSD and may experience harmful interference. A CBSD keeps heartbeating in the Granted state to wait for a successful Heartbeat Response from the SAS to move to the Authorized state and resume its transmission. When the CBSD does not want to transmit on that channel any more, it sends a Relinquishment Request to the SAS. The SAS deallocates the resources from the CBSD and sends a Relinquishment Response message to the CBSD. When a CBSD does not wish to participate in the CBRS band any more, it deregisters from its SAS and goes into the Unregistered state as shown in Figure 1. Figure 3 shows a typical successful message sequence diagram of the SAS-CBSD protocol.

Since spectrum is shared in the CBRS band on a priority basis, a SAS may ask a CBSD to vacate its occupied channel to prevent harmful interference to a higher priority incumbent through the Heartbeat Response message. Hence, periodic heartbeat is essential to protect higher priority incumbents. How frequently heartbeats should be sent is decided by the heartbeatInterval parameter which is set in the Heartbeat Response message by the SAS. Since it is possible that Heartbeat Request or Heartbeat Response messages may be lost, there is a transmitExpireTime timer that runs at the CBSD. The value of this timer is decided by the SAS and is carried in a Heartbeat Response message. If Heartbeat Request or Response is lost, then the transmitExpireTime timer will eventually expire at which point the CBSD has to turn off its transmission. There are stringent timing requirements set forth by the FCC for the lower priority user to vacate the channel when a higher priority incumbent radar appears in the same channel. As per the Part 96 Rules [1], a SAS should make sure that when an incumbent appears on a channel, the CBSDs that may cause harmful interference to the incumbent should stop transmitting within $300\,\mathrm{s}$ from the time the SAS is notified about the presence of the incumbent. The FCC rules also specifies that a CBSD has up to $60\,\mathrm{s}$ to turn off its transmission from the time its managing SAS directs it to do so. Thus, the maximum value of transmitExpireTime timer when a CBSD is in the Authorized state is $240\,\mathrm{s}$. Hence, the heartbeatInterval should be less than $240\,\mathrm{s}$ when a CBSD is transmitting in the Authorized state to prevent the CBSD from unnecessarily shutting down its transmitter due to expiry of transmitExpireTime timer. These timing constraints are depicted in Figure 4. A SAS gets notification from the ESC that an incumbent has appeared on the channel at time A. The SAS has to wait for the next Heartbeat Request message from the CBSD to inform it to stop its transmission. The CBSD sends a Heartbeat Request at time B. The SAS sends a Heartbeat Response which carries a command from the SAS to the CBSD to stop transmission. Once the CBSD receives this command at time E it should stop transmission within $60\,\mathrm{s}$. As mentioned earlier, the time the SAS is notified of the presence of an incumbent to the time the CBSD vacates the channel, should not be more than $300\,\mathrm{s}$, i.e., duration between A and F should be less than or equal to $300\,\mathrm{s}$. From the figure it is clear that the heartbeatInterval parameter plays an important role in deciding how soon or late the CBSD will be able to vacate the channel. Note that the worst case happens when SAS gets ESC notification right after it sent out a Heartbeat Response message. In this case, the SAS would get to notify the CBSD to vacate its channel when it gets the next Heartbeat Request that would arrive after heartbeatInterval. Hence, large heartbeatInterval would lead to a SAS taking longer to ensure that a CBSD vacate the channel. In this study, we evaluate performance of CBRS system in terms of time taken to vacate a channel after a SAS is notified of presence of an incumbent radar in the channel.

As mentioned earlier, after receiving the first successful Heartbeat Response, a CBSD goes into the Authorized state and has to continue to heartbeat periodically while it is transmitting in the Authorized state. When there are thousands of authorized grants managed by a SAS, these periodic heartbeats exert significant processing load on the SAS. If processing power of a SAS is not adequately provisioned, then the SAS may not be able to provide Heartbeat Responses in a timely manner. This could lead to transmitExpireTime timer to expire, resulting in unnecessary suspension of CBSDs which do not get Heartbeat Responses in time. Using the SAS-CBSD simulator we evaluate the limits of a SAS in terms of number of CBSDs it can serve successfully.

B. SAS-CBSD Protocol Simulator

1) Message Processing Model: We use the M/M/1 queueing model to represent the message processing service of a SAS provided to various messages sent from its CBSDs (Figure 5). While this model keeps the analysis simple, it gives a fairly good insight into the timing contstraints and SAS overload aspects of the protocol. Among all the messages, the Heartbeat Request message is the most frequent message arriving at the SAS, whereas other messages are relatively infrequent. Hence, the arrival to the M/M/1 queue is approximated by considering only the Heartbeat Request messages. The service rate of the queue is the rate at which the SAS can process a message ($\mu\_service$) and the mean arrival rate (λ_msg_arr) is approximated by the taking the ratio of number of existing grants in the system to the mean heartbeatInterval time. In the simulation, when a message arrives, we check the utilization ($\frac{\lambda\_msg\_arr}{\mu\_service}$) of the queue at that instant, and if it is greater than or equal to 1, then we drop the message. This indicates that the SAS is not provisioned with adequate processing resources and hence the message is dropped.

The above M/M/1 queueing model can be used to calculate the maximum number of CBSDs that a SAS can manage such that there is no unnecessary shut down of CBSDs. Let λ_msg_arr be the mean arrival rate of messages to the SAS, $\mu\_service$ be the mean service rate of the SAS, N_CBSD be the number of CBSDs, G be the maximum number of grants a CBSD is allowed to have and HBI_mean be the mean heartbeatInterval. We consider only Hearbeat Request messages to compute the processing load on a SAS, since other messages arrive relatively infrequently. Hence, the mean message arrival rate at the SAS is given by

$$\lambda\_msg\_arr = \frac{N\_CBSD \times G}{HBI\_mean} \tag{1}$$

To prevent loss of messages at the SAS because the SAS cannot handle the rate at which messages are arriving, the M/M/1 queue should be stable, i.e., utilization of the queue should be less than 1. Hence,

$$\frac{\lambda\_msg\_arr}{\mu\_service} < 1 \tag{2}$$

Using (1) in (2) we get

$$N\_CBSD < \frac{HBI\_mean \cdot \mu\_service}{G} \tag{3}$$

Thus, (3) provides a method to calculate the maximum number of CBSDs a SAS can serve so that it can process messages in a timely manner.

2) Modeling Detection of Incumbent Radar: We have a simple mechanism for modeling detection of incumbent shipborne radar. The incumbent radar appears according to an exponential distribution with mean arrival rate $\lambda\_incumbent$ (see Table I for these parameters and their values) and the channel on which it appears is randomly chosen. When a shipborne radar is detected, we randomly choose $Grants\_affected\_by\_incumbent$ percentage of total grants which are operating in the same channel as the incumbent and put those grants in suspension, i.e., they go into Granted state. We understand that this is far from the real operation of a SAS. In a real operation, a SAS has to compute path loss from each CBSD in the neighborhood of the shipborne radar to the radar receiver and compute the aggregate interference at the radar receiver. If the aggregate interference is more than the specified Interference to Noise (I/N) threshold of $-6\,\mathrm{dB}$ [4, Requirement R2-IPM-01] then the SAS has to identify which CBSDs should be turned off to bring the aggregate interference down below the threshold. The exact requirement for this operation is specified in [4, Requirement R2-SGN-24]. Even though our shipborne radar protection model does not resemble the real operation, it is adequate to provide essential insight into the time sensitive aspects of vacating a channel by a CBSD.

3) Implementation Details: We have implemented the SAS-CBSD protocol using OMNET++ discrete event simulation software [5]. The core parts of the implementation are the two state machines of the protocol for which we have used the Finite State Machine (FSM) support provided in the Omnet++ simulator [6] and the models presented in the previous sections. The Grant State Machine of a CBSD cannot start unless the Registration State Machine of the CBSD is in the Registered state. Hence the Grant State Machine is implemented as a nested state machine which is supported by Omnet++ [6, Section 4.10.1]. Grant Requests are sent from each CBSD according to a Poisson process, i.e., the inter-arrival of two consecutive Grant Requests is exponentially distributed. The lifetime of a grant is also exponentially distributed. Each CBSD is allowed to have up to 7 grants. Total number of channels in the system is set to 10. When a CBSD sends a Grant Request, it randomly chooses one of the 10 channels for the grant. The parameters used in our experiments and their values are listed in Table I.

III. Simulations and Results

In this section we describe our experiments and present the results. We primarily ran two types of experiments. The first type is designed to show the effect of overloading the SAS in terms of message handling. In this experiment the focus was only on the message handling of the SAS and hence simulation of presence of incumbent radar was not enabled. The second type of experiment was carried out to study how quickly CBSDs vacate the channel when an incumbent radar appears on the channel to satisfy the timing constraint set forth by the FCC. Table I lists the important parameters and their values used in our experiments.

A. Unnecessary Timeout

In this set of experiments, the mean message service rate of the SAS is fixed (as per M/M/1 queue service rate) while the

Parameter	Description	Distribution	value
λ_{grant}	mean interarrival rate of Grant Requests	exponential	$300\,\text{s}^{-1}$
$grant_{life}$	mean lifetime of Grant	exponential	900 s
λ_{incumb}	mean interarrival rate of incumbent	exponential	$(1/180)\,\text{s}^{-1}$
$incumb_{life}$	mean lifetime of presence of incumbent	exponential	300 s
$heartbeatInterval$	parameters of heartbeat interval	uniform	[70, 110] s [120. 180] s [200, 240] s
$\mu_{service}$	mean SAS service rate	exponential	40, 60 s^{-1}
T_{sim}	simulation time	–	86 400 s (1 day)
$HB\_success\_rate$	percentage of hearbeat success	–	100 %
$Grant\_success\_rate$	percentage of grant success	–	95 %
$Grant\_suspend\_rate$	percentage of grant put into suspension when incumbent appears	–	100 %
$Grants\_affected\_by\_incumbent$	percentage of existing grant affected when incumbent appears	–	90 %
G	maximum num of grants per CBSD	–	7
$MAX\_CHANNELS$	maximum number of channels	–	10

TABLE I
PARAMETERS USED IN OUR SIMULATION

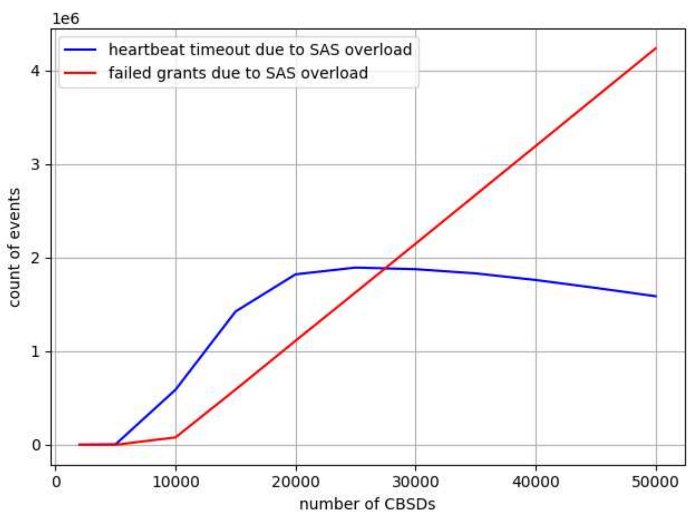

Fig. 6. Unnecessary Timeout vs Num of CBSDs (service rate=40 rps)

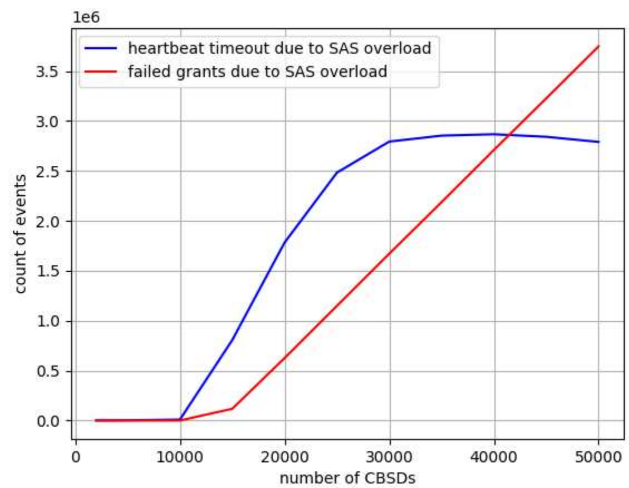

Fig. 7. Unnecessary Timeout vs Num of CBSDs (service rate=60 rps)

number of CBSDs is progressively increased. As the number of CBSDs increases, the number of grants also increases, which in turn increases the number of heartbeat messages to be handled by the SAS. Thus, the message handling load on the SAS increases. At some point the SAS fails to keep up with the rate at which the Heartbeat Request messages arrive. This leads to some Heartbeat Request as well as Grant Request messages being dropped. When a Heartbeat Request message is dropped, the CBSD does not get the corresponding Heartbeat Response message. Hence, the transmitExpireTime timer in the corresponding Grant *unnecessarily* times out. This would force the CBSD to stop its transmission. Clearly this situation arises due to poor provisioning of processing power of the SAS and is not desirable by the commercial operators.

Figure 6 presents the unnecessary heartbeat timeout and failed grants (due to SAS overload) vs number of CBSDs when the mean SAS service rate is 40 requests per second (rps) and the heartbeatIntervals of grants are uniformly distributed between 200 s and 240 s. For a given number of CBSDs the simulation is run for $86\,400$ s (1 day). The heartbeat timeout remains low until about 5000 CBSDs after which it rises rapidly. But beyond $20\,000$ CBSDs, it tapers off. The failed grant count starts to take off rapidly around $10\,000$ CBSDs and continues to increase. Since grants fail, there are less number of grants in the system. Hence, when the high number of Grant Requests fail, the number of heartbeat failures does not increase and more or less remains flat.

Figure 7 depicts the same performance metrics but with mean SAS service rate of 60 rps. The SAS can handle more CBSDs compared to the 40 rps case before heartbeats start to time out. For this case the heartbeat fail count takes off at around $10\,000$ CBSDs which is higher than the corresponding point when service rate is 40 rps. However, once the heartbeat fail count rises, it rises more rapidly compared to 40 rps case.

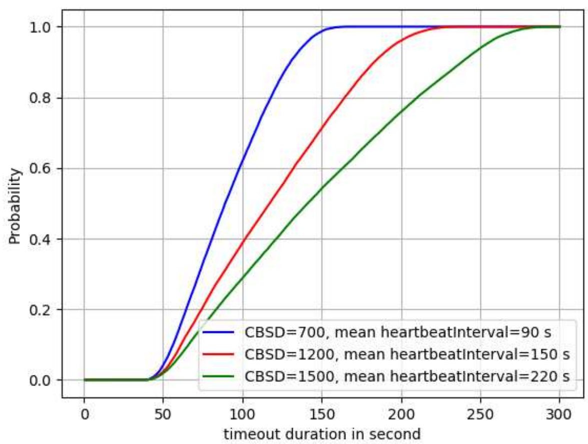

Fig. 8. CDF of Duration of CBSDs Vacating a Channel (service rate=60 rps)

The failed grant count remains zero until about $10\,000$ CBSDs after which it takes off. The grant fail count happens to be lower than that for the 40 rps case for a given number of CBSDs. Hence, there are more active grants in the system than 40 rps case which implies that there are more heartbeat requests. This leads to a higher heartbeat fail count compared to the 40 rps system.

B. Time to Vacate a Channel

In this experiment, we enabled presence of incumbent radar as per the parameters specified in Table I. When an incumbent radar appears, the SAS randomly chooses $90\,\%$ of grants which are in the same channel as the radar and instructs those CBSDs to suspend the grants via Heartbeat Response message. We assume that the SAS sets transmitExpiryTime timer to zero when it commands the CBSD to suspend the grant in the Heartbeat Response message. Hence, the CBSD has up to 60 s to vacate the channel. Figure 8 shows the Cumulative Distribution Function (CDF) of the CBSDs vacating the channel for different mean heartbeatInterval. In this experiment we assume that the SAS is adequately provisioned with processing power such that there is no Heartbeat Request or Grant Request message loss. To achieve this, for a given mean heartbeatInterval, we calculate the number of CBSDs (using (3)) such that the SAS does not drop any message due to overloading of message arrival. For example, when the heartbeatInterval is uniformly distributed between $(200,\, 240)$ s, the mean hearbeatInterval is 220 s. If a SAS has a service rate of 60 rps, then using (3), the SAS can handle up to 1885 CBSDs. So, we fix the number of CBSDs at 1500. From Figure 8, it can be noticed that when the mean heartbeatInterval increases, more and more grants vacate the channel closer to the 300 s limit. In fact, for a mean heartbeatInterval of 220 s some grants vacate channel very close to 300 s. Since there is randomness involved in the process, it is not advisable to have CBSDs vacating the channel so close to the deadline although it allows more CBSDs to be served for a given SAS service rate. Hence, a heartbeatInterval of 220 s may not be a good choice. On the contrary, if heartbeatInterval is set to a low value of 90 s, then CBSDs can vacate the channel much ahead of the deadline. However, shorter heartbeatInterval exerts heavy load on the SAS which means a smaller number of CBSDs (only 700 CBSDs when mean heartbeatInterval is 90 s) can be served by the SAS for a given service rate. Setting heartbeatInterval to around 150 s seems to be a good choice to achieve a reasonable balance between the number of CBSDs a SAS can manage and the time taken for CBSDs to vacate a channel. Note that the SAS service rate does not have much effect on the CDF of time taken by CBSDs to vacate a channel when SAS is operating close to its full capacity (i.e., close to M/M/1 server utilization of 1). This is true in our study, since the number of CBSDs a SAS manages is set close to the maximum possible value as per (3).

IV. Conclusion and Future Work

We have developed a basic SAS-CBSD protocol simulator to study the impact of heartbeatInterval on the CBRS system in terms of meeting the end to end timing constraint set forth by the FCC rules. We also use the simulator to study how message overload on a SAS leads to unnecessary timeout of transmitExpireTime timer which leads to suspension of CBSD transmission thereby reducing spectrum utilization. Through our experimental results we have shown the tradeoff between time taken to meet the end to end timing constraint and the number CBSDs that can be served by a SAS without causing unnecessary supension of CBSD transmission. With a lower heartbeatInterval the end to end timing constraint of vacating a channel can be met earlier, however, the number of CBSDs a SAS can serve will be lower. Based on our results, setting heartbeatInterval to around 150 s may strike a good balance between the tradeoffs.

Implementing a more practical model to represent the method of vacating a channel when an incumbent shipborne radar appears on the channel can be an useful future work. This would require implementing a propagation model and the so called *move-list* algorithm (which identifies the CBSDs that should vacate the channel when a incumbent shipborne radar appears on the channel) defined in the standards [4].

References

[1] "Citizens broadband radio service," Title 47 Code of Federal Regulations Part 96, 2019.

[2] "Signaling protocols and procedures for citizens broadband radio service (cbrs): Spectrum access system (sas) - citizens broadband radio service device (cbsd) interface technical specification," Wireless Innovation Forum Document WINNF-TS-0016, Version V1.2.4, 2019. [Online]. Available: https://winnf.memberclicks.net/assets/CBRS/WINNF-TS-0016.pdf

[3] M. M. Sohul, M. Yao, T. Yang, and J. H. Reed, "Spectrum access system for the citizen broadband radio service," *IEEE Communications Magazine*, vol. 53, no. 7, pp. 18–25, 2015.

[4] "Requirements for commercial operation in the U.S. 3550–3700 MHz citizens broadband radio service band," Wireless Innovation Forum Document WINNF-TS-0112, Version V1.8.0, 2019. [Online]. Available: https://winnf.memberclicks.net/assets/CBRS/WINNF-TS-0112.pdf

[5] OMNET++ Discrete Event Simulator. [Online]. Available: https://www.omnetpp.org/

[6] OMNET++ Simulation Manual. [Online]. Available: https://www.omnetpp.org/doc/omnetpp/manual/