On the Design of Access Network Packet Consolidation Protocol of CDMA 1xEV-DO Systems

Venkatesh Ramaswamy, Pradeepkumar Mani, and Olufemi Adeyemi

Airvana Network Solutions,
19 Alpha Road, Chelmsford, MA 01824
{vramaswamy,pmani,fadeyemi}@airvana.com

Abstract. We seek to formulate an optimal packet consolidation policy to transmit messages in the forward link control channel of CDMA 1xEV-DO air-interface standard such that the policy consumes minimum number of physical layer transmission slots. The consolidation policy is constrained by a given set of rate and delay constraints. We present an analysis of the mean message delay in the system and formulate the process as an generalized optimization problem to determine the optimal value for a variety of system-wide objectives, subject to various rate and delay constraints. We verify our analytical results through simulations, and present numerical examples to illustrate the design principles.

Key words: 1xEV-DO, Control Channel, Packet Consolidation Protocol, Performance Analysis.

1 Introduction

The Control Channel (CC) in a 1xEV-DO [1, 2] network is the signaling plane in the network that carries signaling messages from the Access Network (AN) to the Access Terminal (AT). The messages are transmitted to the AT by packing them in Synchronous capsules or Asynchronous capsules. The synchronous capsules are scheduled for transmission at fixed instances in time (once every 256 slots, 1 slot = 1.667 msec), where as the asynchronous capsules are scheduled as and when needed [4, 5].

The synchronous capsule can hold up to eight 1xEV-DO medium access control (MAC) packets (each of length 1024 bits) if the synchronous capsule CC rate is chosen to be 76.8 Kb/s or up to four MAC packets if the synchronous capsule CC rate is 38.4 Kb/s. The asynchronous capsule is exactly one MAC packet in length regardless of the CC rate; but the length of each MAC packet varies based on the packet transmission format chosen from the following [1]:

[1] interpret the tuple $< X, Y, Z >$ as X = length of the physical layer packet in bits, Y = number of physical layer slots used for transmission, Z = length of preamble in chips; the transmission rate offered by a particular format is simply X/Y expressed in bits/slot or $3X/5Y$ bits/sec.

QSHINE 2010, November 17-19, Houston, United States

DOI 10.4108/eai.17-11-2010.2245433

– For synchronous capsule, the transmission format is $< 1024, 16, 1024 >$ for 38.4 Kb/s or $< 1024, 8, 512 >$ for 76.8 Kb/s
– For the asynchronous capsule, one of the following transmission formats can be used: $< 128, 4, 1024 >$, $< 256, 4, 1024 >$, $< 512, 4, 1024 >$, $< 1024, 16, 1024 >$, or $< 1024, 8, 1024 >$. All packet formats can be used when the synchronous capsule CC rate is 76.8 Kb/s, while only 1, 2, and 4 can be used for a synchronous capsule CC rate of 38.4 Kb/s.

It must be noted that for the transmission format chosen, if there are not enough CC bits to fill up an entire MAC packet, then the remaining bits are padded as zeros. These padding bits constitute wasted transmission effort. For ease of exposition, for the remainder of this paper, we will only use a synchronous capsule CC rate of 76.8 Kb/s.

Each of the message types has a priority value assigned to it, which can be used to control its resource access. For example, the priority values can be used to determine the order in which the messages are packed into the synchronous or asynchronous capsules. While all message types can be packed into the synchronous capsule, some messages are "synchronous-only" and cannot be packed in an asynchronous capsule. Some message types that are common to both synchronous and asynchronous capsules are the *Traffic Channel Assignment* message, and the *Access Channel Acknowledgement* message while *Page* message, and *Sector Parameters* message constitute some of the "synchronous-only" or "exclusive" messages.

The synchronous capsule is periodically scheduled, and hence provides very little scope for implementation of a scheduling/consolidation policy to achieve some objective while satisfying certain constraints. On the other hand, the asynchronous capsule can be formed and scheduled in a flexible manner, and is a very valuable tool in enforcing a particular consolidation policy.

We are particularly interested in a consolidation policy that minimizes the number of slots consumed by the CC capsules for a given arrival process to the queues. We seek to minimize the number of slots consumed because the MAC packets carrying control channel capsules are Time-Division Multiplexed (TDM) with the MAC packets carrying user data from the Forward Traffic Channel (FTC). While the CC capsules use a transmission format such that a maximum rate of 76.8 Kb/s is supported, the FTC packets use transmission formats that support up to 3.1 Mb/s. As with traditional networking systems, the control plane (CC) data has higher priority in resource access (physical layer slots) when compared to data plane (FTC) data. Hence, sufficient care must be exercised when scheduling packets because an erroneous scheduling policy could adversely impact the FTC traffic.

In the absence of any delay constraints for a particular message type, the optimal scheduling strategy is one that schedules transmission of an asynchronous capsule after accumulating enough bits to fill up the packet format that offers the maximum CC rate. In the presence of delay constraints, it may not be acceptable to delay transmission of certain messages until enough bits are accumulated. Frequent and unnecessary scheduling of the asynchronous capsules to meet delay constraints can lead to significant reduction in available transmission slots for FTC traffic, thus impacting overall network data throughput. Hence, we seek a policy that chooses the optimal frequency of forming the asynchronous capsules based on the arrival process and given delay constraints.

This paper has four sections. In Section 2, we describe our model, and formulate the problem of designing an optimal packet consolidation policy as a generalized optimization problem. We also provide computational formulas needed to solve the problem. We dedicate Section 3 to validating the correctness of our model through simulations, and presenting numerical results obtained from our model. In Section 4, we draw conclusions and provide pointers for future work.

2 Analysis

In this section, we model the control channel packet consolidation protocol at the Access Network. Figure 1 depicts the CC capsule creation process. For ease of representation, the exclusive queues are consolidated and shown as one queue (colored red). There are k common queues (colored blue) with the arrival to the ith queue modeled as a Poisson process with parameter λ_i. The queue i has a higher priority than queue j if $i < j$. The aggregate arrival to the exclusive queues are also modeled as a Poisson process with rate λ_E. Synchronous capsules are formed at the end of each control channel cycle. Asynchronous capsules are formed every τ seconds termed as the *asynchronous capsule cycle*. Note that τ is a design parameter. Synchronous capsules are formed by first packing messages from the exclusive queues, and then depending on the availability of space in the capsule, messages from the asynchronous capsules are added. The queues are drained always in the order of their priority—higher priority queues are drained before draining the lower priority queues. Asynchronous capsules are formed independent of the synchronous capsule, and are formed every τ seconds from messages in the common queue. Again the higher priority queues are drained before draining the lower priority queues. The maximum size of a synchronous capsule is denoted by μ_E messages, whereas that of an asynchronous capsule is given by μ_C messages. The offset between the beginning of a CC cycle and the first asynchronous capsule in a CC cycle is denoted by x. The maximum expected waiting time of the ith common queue message is denoted by the $d^i_{\max}$. Let n be the asynchronous capsule formation frequency, given by T/τ. That is, there will be an average of T/τ asynchronous capsules in a CC cycle. The number of physical layer slots used for transmitting the asynchronous capsules in a CC cycle is denoted by U and can be represented simply by $nS(\mu_C)$, where $S(\mu_C)$ is the number of slots used by a packet of size μ_C. We refer to U as the forward link *control channel utilization.*

We consider the forward link control channel utilization as the performance parameter in designing the system. Our objective is to find the optimal way of creating the control channel packets that minimizes the forward link control channel utilization under the constraint that expected delay of each type of message is within a maximum acceptable expected delay for that message. In

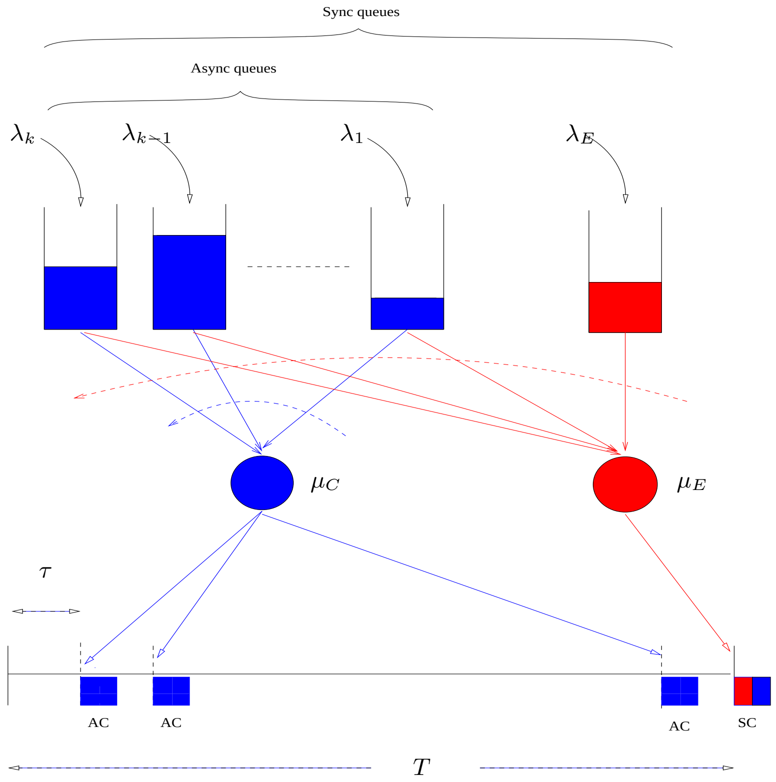

Fig. 1. Illustration of the control channel packet consolidation procedure at the access network.

1xEV-DO, control channel is time division multiplexed with the traffic channel in the forward link. If the control channel transmission occupied more slots to transmit, it means that there will be less number of slots available for user data traffic—resulting in a low forward link user throughput. If the asynchronous capsules are formed frequently, then they are likely to have more padding bits. This means that some of the forward link slots are used to transmit padding bits, wasting air-link resources. On the other hand, if asynchronous capsules are formed less frequently, then some of the messages in the common queues might miss their delay deadline, and will have to be re-transmitted. Therefore, the rate at which asynchronous capsule are created must be such that forward link CC utilization is minimized while meeting the deadline of the messages.

The optimum policy is the solution of the following optimization problem, Problem P1.

Problem P1:

Minimize $U = n\ S(\mu_C)$,

subject to

$$(128-\mu_C)(256-\mu_C)(512-\mu_C)(1024-\mu_C)=0, \tag{1}$$
$$(S(\mu_\mathrm{C})-4)(S(\mu_\mathrm{C})-8)=0, \tag{2}$$

$$S(\mu_\mathrm{C}) \geq \frac{8\mu_c}{1024}, \tag{3}$$
$$S(\mu_\mathrm{C}) \geq 4, \tag{4}$$
$$d^i_{\mathrm{avg}}(n) \leq d^i_{\mathrm{max}} \qquad i = 1 \cdots k, \tag{5}$$
$$n \geq \left\lceil \frac{(\lambda_\mathrm{E}+\lambda_\mathrm{C})T-\mu_\mathrm{E}}{\mu_\mathrm{C}} \right\rceil, \tag{6}$$
$$n \geq 1. \tag{7}$$

The size of the asynchronous capsule can only be 128, 256, 512 or 1024 messages. Constraint (1) enforces this. For asynchronous capsule size of 128, 256 or 512 messages, four physical layer transmission slots are required, whereas a capsule size of 1024 messages requires 8 physical layer transmission slots. Constraints (2), (3) and (4) capture these requirements. The average delay experienced by messages in each queue, denoted by $d^i_{\mathrm{avg}}(n)$ for the ith queue must be less than a maximum average delay for the ith queue, d^i_{max}. This constraint is given in (5). Constraint (6) guarantees system stability—the average number of arrivals in a CC cycle must be less than the maximum number of messages that can be drained in a CC cycle, and finally constraint (7) enforces that there be at least one asynchronous capsule formed in a CC cycle.

In order to solve the above optimization problem, we need to derive a computational formula for obtaining the expected delay suffered by an arbitrary message in each common queue. We consider the ith queue, and refer an arbitrary message arriving at the queue as tagged message. Let the arrival time of this tagged message be t sec after the beginning of the last CC cycle. The idea exploited to compute the delay experienced by the tagged message is simple—the waiting time of the tagged message is equal to the time it takes to drain all the high priority message that are already in the system. Note that this may also include messages that arrived after t. For stability, we assume that in the beginning of a CC cycle, the average number of messages held over from previous CC cycles is zero. For the purpose of analysis, we also assume that a message arriving in a control channel cycle will be removed from the queue using one of the asynchronous capsules or using the synchronous capsule before the end of the cycle. Note that when the load in control channel is high, messages may only be removed in the subsequent control channel cycles. This assumption is not very realistic, and can make our delay analysis less accurate at high operating loads.

We compute the average delay experienced by a tagged message for the case when the tagged customer arrives after the first asynchronous capsule is created in a CC ($t > x$) and for the case when it arrives before the first asynchronous capsule is created ($t \leq x$).

2.1 Case I: $t > x$

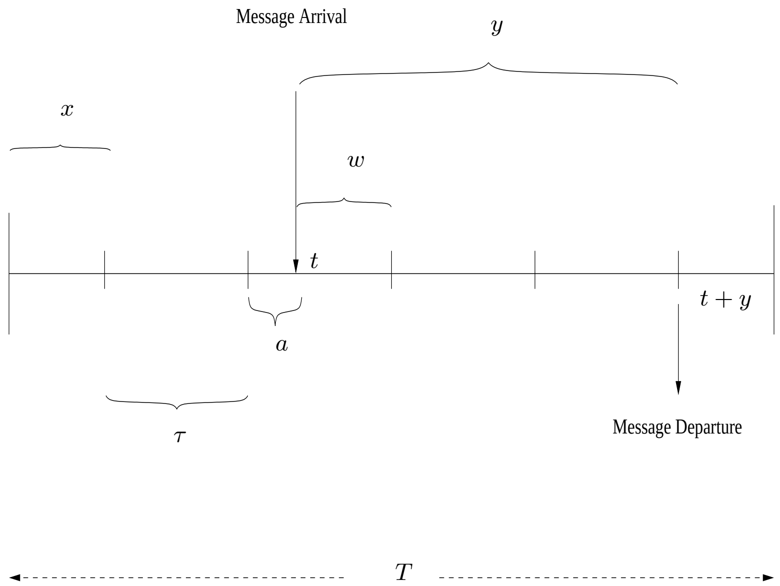

Fig. 2. Diagram illustrating various components of message delay when $t > x$.

The components needed to compute the expected delay are depicted in Figure 2. We start by computing the expected number of messages with higher priority that arrived before the tagged message that are in the queue by time t. This can be determined by computing the number of messages left in the system during three intervals which include:

- at time x, there will be an average of $\sum_{j=1}^{i} \lambda_j x$ messages with higher priority than the tagged message, and right after x, there will be $\left[\sum_{j=1}^{i} \lambda_j x - \mu_C\right]^+$ messages left in the system [2]. Let this be B_1;
- between x and $t-a$, there would be $\lfloor n(t-x)/T \rfloor$ asynchronous capsule cycles, and each of them would leave $\left[\sum_{j=1}^{i} \lambda_j T/n - \mu_C\right]^+$ messages, totaling to $\lfloor n(t-x)/T \rfloor \left[\sum_{j=1}^{i} \lambda_j T/n - \mu_C\right]^+$ messages; Denote this by B_2;

[2] The notation $[x]^+$ denotes $\max\{0, x\}$.

– during the time period a, there would be an average of $\sum_{j=1}^{i} \lambda_j a$ arrivals, represented by B_3.

The total number of messages that arrived before the tagged message that needs to be removed before removing the tagged message is then obtained by adding the number of messages during each of the above three intervals. Now let y be the delay experienced by the tagged message. Then $y \sum_{j=1}^{i-1} \lambda_j$ messages arrived, on an average, when the tagged message was waiting in the queue. The expected number of messages that needs to be removed after t and before removing the tagged messages is then $B_1 + B_2 + B_3 + y \sum_{j=1}^{i-1} \lambda_j$, which requires $\lfloor (B_1 + B_2 + B_3 + y \sum_{j=1}^{i-1} \lambda_j)/\mu_C \rfloor$ asynchronous capsule cycles after the next asynchronous capsule. After t, the next asynchronous capsule cycle happens after w seconds (see Figure 2) given by $(\lfloor n(t-x)/T \rfloor + 1)\, T/n - (t-x)$. Therefore, it would require $w + T/n \lfloor (B_1 + B_2 + B_3 + y \sum_{j=1}^{i-1} \lambda_j)/\mu_C \rfloor$ sec after t to remove all the messages that need to be removed before removing the tagged message. Therefore the expected delay experienced by a tagged message arriving to queue i at time t, represented by $d^i_{\{t>x\}}(n,t,x)$ is the solution to the following equation

$$d^i_{\{t>x\}}(n,t,x) = \Bigg\{ y : y - w + \frac{T}{b} \left\lfloor \frac{B_1 + B_2 + B_3 + y \sum_{j=1}^{i-1} \lambda_j}{\mu_C} \right\rfloor = 0 \Bigg\}.$$

Note that the above equation can yield a set of values for $d^i_{\{t>x\}}(n,t,x)$—we only need to consider the minimum positive solution. Substituting all the values in the above equation, we get (8). Because of our assumption that messages will be removed before the beginning of the next CC cycle, we can write the expected delay as $\min\,(T-t), d^i_{\{t>x\}}(n,t,x)$. We now turn our attention to the computation of the delay when $t \le x$.

2.2 Case II: $t \le x$

Figure 3 portrays this scenario. The average number of messages that are of higher priority than the tagged message remaining after time x, denoted by B_4, is simply $\left[\sum_{j=1}^{i-1} \lambda_j x + \lambda_i t - \mu_c\right]^+$. Let y be the average delay experienced by the tagged message, then the number of messages that are of higher priority than the tagged message arriving after x that needs to be removed before servicing the tagged message is $(t + y - x) \sum_{j=1}^{i-1} \lambda_j$. Let this be B_5. It is obvious that after x, we need $\lfloor (B_4 + B_5)/\mu_c \rfloor$ asynchronous capsule cycles to remove all the messages with higher priority than the tagged message. Therefore the average total delay will be $x - t + \lfloor (B_4 + B_5)/\mu_c \rfloor T/n$, where $x - t$ represents the time to next asynchronous capsule after t. The expected delay for this case denoted by $d^i_{\{t\le x\}}(n,t,x)$ can be computed as a solution to (9). As in the previous case, we select the minimum positive solution of the equation. Combining the delay for both the cases and assuming that messages are not carried over to the next control channel cycle, we can represent the delay the tagged message as given in (10).

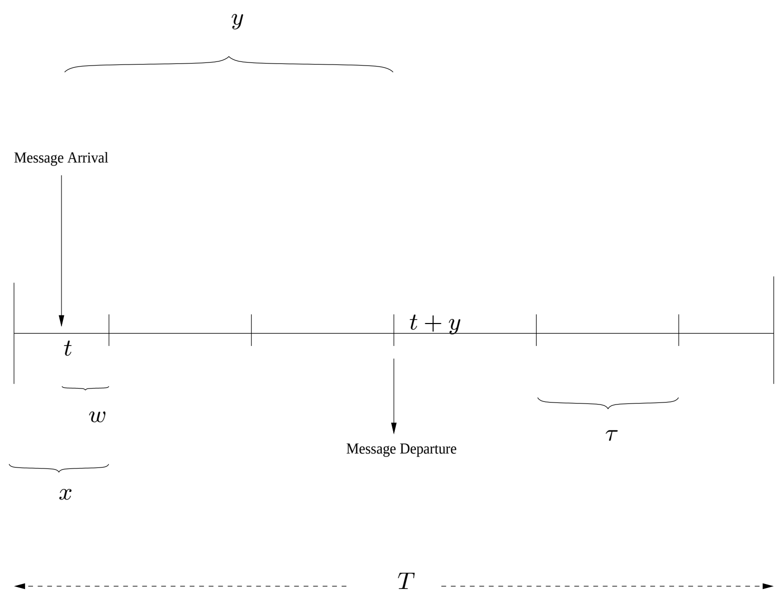

Fig. 3. Diagram illustrating various components of message delay when $t \leq x$.

We assumed in our model that all messages arriving during a CC cycle will be removed by the end of that cycle. This assumption is not always true, especially when the CC load is high, but will let us work on a CC cycle in isolation. This makes the computation of delay much easier. We shall show later by simulation that errors introduced by this assumption is negligible for normal operating parameters.

$$d^i_{\{t>x\}}(n,t,x) = \min_{y\geq 0}\Bigg\{ y : y - \left(\left\lfloor \frac{n(t-x)}{T} \right\rfloor + 1 \right) \frac{T}{n} - (t-x)$$
$$+ \left\lfloor \frac{\left[\sum_{j=1}^{i}\lambda_j x - \mu_c\right]^+ + \left[\sum_{j=1}^{i}\lambda_j \frac{T}{n} - \mu_c\right]^+ \lfloor \frac{n(t-x)}{T} \rfloor + \left[t - \left(x + \lfloor\frac{n(t-x)}{T}\rfloor \frac{T}{n}\right)\right] \sum_{j=1}^{i}\lambda_j + y\sum_{j=1}^{i-1}\lambda_j}{\mu_c} \right\rfloor \frac{T}{n} = 0 \Bigg\}. \quad (8)$$

$$d^i_{\{t\leq x\}}(n,t,x) = \min_{y\geq 0}\left\{ y : y - \left((x-t) + \left\lfloor \frac{\left[\sum_{j=1}^{i-1}\lambda_j x + \lambda_i t - \mu_c\right]^+ + (t+y-x)\sum_{j=1}^{i-1}\lambda_j}{\mu_c} \right\rfloor \frac{T}{n} \right) = 0 \right\}. \quad (9)$$

$$d^i(n,t,x) = \begin{cases} \min\left\{(T-t), d^i_{\{t>x\}}(n,t,x)\right\}, & t > x \\ \min\left\{(T-t), d^i_{\{t\leq x\}}(n,t,x)\right\}, & t \leq x \end{cases} \quad (10)$$

The message arrivals to all the queues are assumed to be Poisson. If there is at least one arrival in a CC cycle, then because of the stationary and independent increment properties of the Poisson process, we know that the arrival time of that message will be uniformly distributed in the CC cycle [3]. Therefore the arrival time, t of the tagged message is uniformly distributed in $(0,T)$. The offset between the CC cycle and the asynchronous capsule cycle is also uniformly distributed in $(0,\tau)$. Therefore, averaging over t and offset x gives the expected delay as

$$d^i_{\text{avg}}(n) = \frac{1}{T\tau}\int_0^{\tau}\int_0^{T} d^i(n,t,x)\ dt\ dx. \quad (11)$$

3 Numerical Examples

In this section, we check the reasonableness of the model using simulations, and present a limited set of numerical results obtained from the model. For validation, we consider a system where there are two common queues and one exclusive queue. We assume the following parameters for simulation. The arrival process to highest priority common queue is Poisson with rate $\lambda_1 = 0.3$ messages/slot, and that of the lower priority queue is Poisson with rate $\lambda_2 = 0.1$ messages/slot. The arrivals to the exclusive queue is also assumed to be Poisson with rate $\lambda_E = 0.1$ messages/slot. The synchronous and asynchronous capsule sizes are assumed to be 128 messages and 64 messages, respectively. All the messages, arriving to both common and exclusive queues are assumed to be of equal length. Figure 4, shows the results from simulation and analysis. We can see that the model matches with the simulation for the high priority queue. However for the low priority queue, the model matches well with the simulation when the asynchronous capsule cycle (τ) is less than 64 slots (≈106 ms). The difference between the model and the simulation is expected due to our assumption that all the messages arriving in a CC cycle departs by the end of the cycle. In reality, there will be cycles when messages are carried over to the next or subsequent cycles, and when τ is large, the carried over messages will be delayed in multiples of τ resulting in a much higher delay than the case when message can leave the same CC cycle it arrived. Therefore, at higher loads and larger asynchronous capsule cycles, the model predicts a lower delay than the actual delay. In most practical systems, the average delay of messages of interest is well within 100 ms, and therefore our approximate model works well within the operating region.

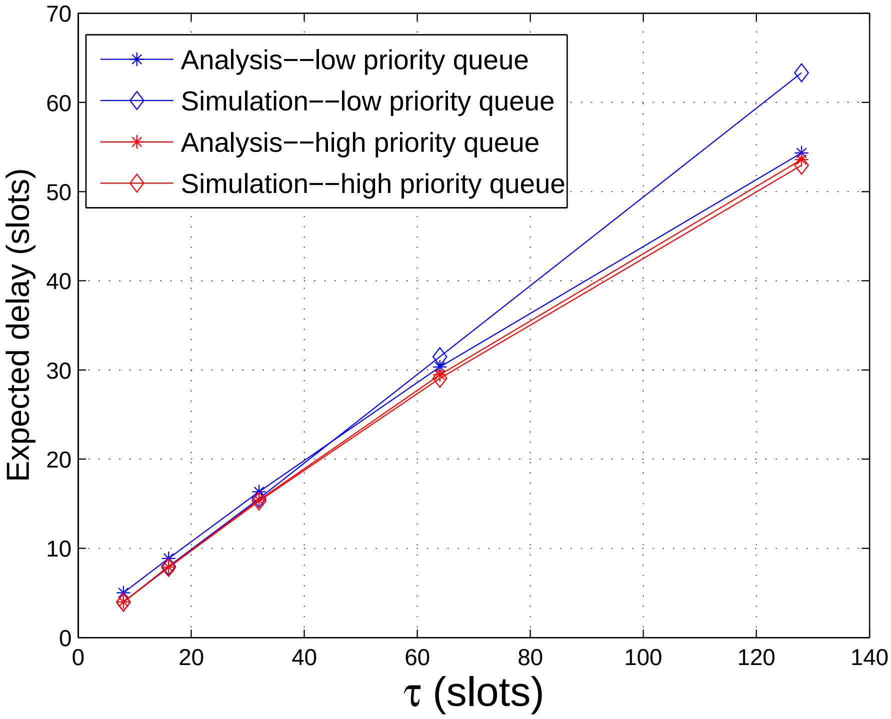

Fig. 4. Delay experienced by high priority and low priority messages for different values of τ—analysis and simulation. The following parameters are used: $\lambda_1 = 0.3$ msgs/slot (high priority), $\lambda_2 = 0.1$ msgs/slot (low priority), $\lambda_E = 0.1$ msgs/slot, $\mu_E = 128$ msgs, and $\mu_C = 64$ msgs.

In the next set of results, we assume that only the highest priority common queue has the delay constraint. The arrival rate of the highest priority common queue is assumed as $\lambda_1 = 0.4$ messages per CC slot. The aggregate arrival rate for the exclusive queue is fixed to be $\lambda_E = 0.1$ msgs/slot, and the synchronous capsules are assumed to be of size 128 messages. Figure 5 plots delay as a function of τ for different values of asynchronous capsule size. We use (11) to compute the expected delay for different values of μ_C. From the plots, we can see that increasing the asynchronous capsule size gives diminishing returns on the expected delay of the highest priority messages.

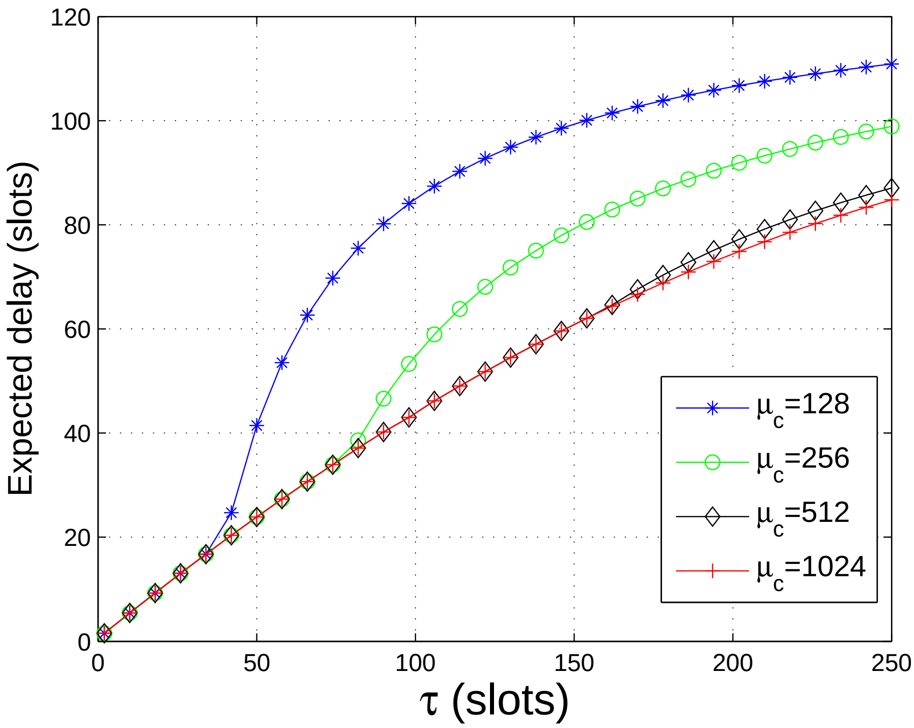

Fig. 5. Delay experienced by high priority messages as a function of τ for all allowed asynchronous capsule sizes. The arrival rate of high priority messages is assumed to be 0.4 msgs/slot, and that of exclusive queue is assumed to be 0.1 msgs/slot. $\mu_E = 128$ msgs.

Now we wish to find the optimum asynchronous capsule formation policy satisfying the delay constraints of the highest priority queue and other constrains given in Problem P1. In Figure 6, we plot forward link control channel slot utilization versus the delay experienced by the highest priority messages for different values of asynchronous capsule sizes. The optimum policy is then the selection of asynchronous capsule size that minimizes the slot utilization for each value of maximum acceptable expected delay of the highest priority messages.

Once the capsule size is selected, we can select τ that satisfies the delay constraint for the selected μ_C using Figure 5.

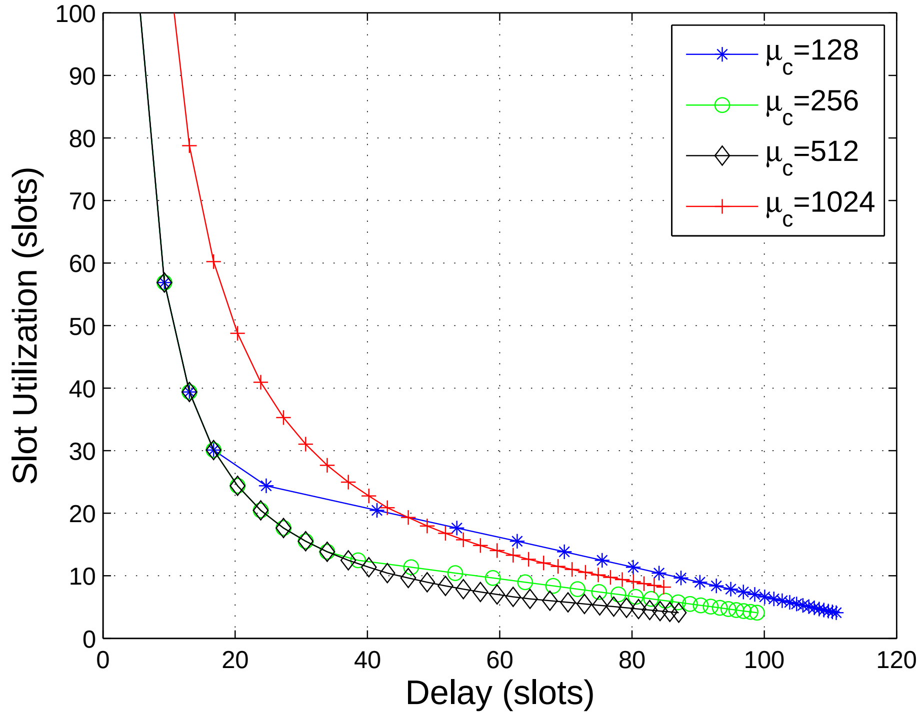

Fig. 6. Forward link physical layer slot utilization as a function of the experienced delay of high priority messages for different values of asynchronous capsule sizes. The parameters used are same as in Figure 5.

In the above example, suppose that we want to find the optimum value of asynchronous capsule size and asynchronous capsule cycle that minimizes the forward link CC utilization, while achieving a delay constraint of 40 slots ($\approx$ 66 ms) for the highest priority message. Then, from Figure 6, we know that asynchronous capsule size of 512 messages achieves the minimum slot utilization, and from Figure 5, we know that to meet the 40 slots delay constraint of the highest priority message using a asynchronous capsule size of 512 messages, we need to form asynchronous capsules approximately every 90 slots.

In certain cases, the access network may not be able to transmit in all the available asynchronous capsule sizes. For example, if the feasible asynchronous capsule sizes are 128 and 512 messages, then depending on the delay requirements, different asynchronous capsule sizes will yield optimum performance. In our example, when maximum expected delay is less than 45 slots, choosing μ_C

as 128 yields the minimum forward link CC utilization. However, for delay requirements for more than 45 slots, it is better to opt for μ_C as 512.

We note in passing that the procedure developed can be used to determine other performance measures. For instance, it is straight forward to determine the FL traffic channel utilization for a given value of asynchronous capsule cycle and asynchronous capsule size.

4 Concluding Remarks

In this work, we studied the optimal design of control channel packet consolidation protocol at the access network in a 1xEV-DO system. The design discussed herein allows for optimal formation of asynchronous capsules such that forward link control channel utilization is minimized while satisfying the delay constraints of all signaling messages. We first described the operation of packet consolidation protocol, and then formulated the problem of designing an optimal policy as a generalized optimization problem. We presented a simple model for computing the average delay of messages needed to solve the optimization problem. We validated our model through limited set of simulation results, and finally presented some numerical examples to illustrate the design principles. Our hope is that the guidelines presented here provides cellular operators useful order-of-magnitude estimates for the parameters of interests.

This work can be extended in a number of interesting ways. The delay model presented uses some simplifying assumptions; a more accurate delay model can be worked out. It appears worthwhile to develop the delay distribution of messages. Currently, we consider the average delay as an optimization constraint. An interesting extension would be to include constraints that guarantee that the probability that the delay exceeds a certain threshold is less than a maximum acceptable value. Finally, comparing the results from the model to that of a real 1xEV-DO network would be a valuable next step.

References

1. 3GPP2 C.S0024-A v1.0, *CDMA 2000 High Data Rate Packet Data Air Interface Specification*, Available from www.3gpp2.org.
2. N. Bhushan, C. Lott, P. Black, R. Attar, Yu-Cheun Jou, M. Fan, D. Ghosh, and J. Au, *CDMA2000 1xEV-DO revision a: a physical layer and MAC layer overview*, IEEE Communications Magazine **44** (2006), no. 2.
3. Robert Gallager, *Discrete stochastic processes*, 1st ed., Springer, October 1995.
4. T. Gopal, *EVDO Rev. A control channel bandwidth analysis for paging*, Proc. of IEEE Wireless Communications and Networking Conference (2007), 3262–3267.
5. V. Ramaswamy and J. Chung, *Performance analysis of the quick idle state protocol of CDMA 1xEV-DO Rev. B systems*, To Appear in Proc. of IEEE Global Telecommunications Conference (2010).