

Modeling and Analysis of Frame Aggregation in Unsaturated WLANs with Finite Buffer Stations[†]

Srikant Kuppa and G.R. Dattatreya
Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75083
Email: {ksrikant, datta}@utdallas.edu

Abstract—Frame aggregation is one of the several enhancements proposed by IEEE 802.11 Task Group n to improve channel utilization. In frame aggregation, more than one data frame is encapsulated to form an aggregate, and once an aggregate is formed, a station contends to access the medium to transmit the entire aggregate. We refer to the number of data frames encapsulated within an aggregated frame as aggregate size. We claim that a static assignment of aggregate size leads to the following performance trade-off: a small value might be insufficient to mitigate the transmission overheads, thereby nullifying the whole purpose of frame aggregation; whereas, a large value might affect the quality of service experienced by higher layers due to the extra wait time to build an aggregate. In this paper, we characterize this trade-off by studying the impact of aggregate size on metrics like frame latency and channel utilization. To estimate these metrics, we model the transmission queue of an 802.11n station as a bulk service queuing system. We study the impact of aggregate size over a wide range of operating conditions covering several traffic arrival rates from higher layers, service distribution and data frame sizes. Apart from validating the existence of above-mentioned performance trade-off, our results indicate that the choice of aggregate size not only depends on the traffic arrival rate, but also (more interestingly) on data frame sizes. This calls for a dynamic assignment of aggregate size.

Keywords — Bulk service, Frame aggregation, IEEE 802.11n, Minimum batch size rule.

I. INTRODUCTION

One of the several enhancements proposed by IEEE 802.11 Task Group n ([1], [2]) to improve channel utilization is *frame aggregation*. In order to mitigate transmission overheads, namely, backoffs prior to accessing the shared medium, physical layer preamble and inter-frame spacing, more than one data frame (or fragment) is encapsulated within an *aggregated frame*. The encapsulated frames within the aggregated frame are destined to the same station. The number of frames that can be aggregated (referred to as *aggregate size* in this paper) is limited only by the receive buffer capacity at the destination [9]. Prior to the transmission of data frames, stations are required to exchange short control frames such as Initiator Aggregation Control (IAC) and Responder Aggregation Control (RAC) [1]. These control frames are enhancements to the legacy Request-To-Send (RTS) and Clear-To-Send (CTS) frames [3].

A. Relevant Work

In [10] and [12], several Medium Access Control (MAC) layer enhancements such as packing, concatenation and multiple frame transmissions were introduced to overcome the

transmission overheads listed above. However, these enhancements are conceptually different from frame aggregation proposed by IEEE 802.11 Task Group n . Whereas frame aggregation deals with encapsulation of multiple MAC protocol data units (MPDUs, commonly known as MAC data frames) into a single physical protocol data unit (PPDU), the enhancements in [10] and [12] consider transmission of multiple MPDUs as multiple PPDUs. Due to this, transmission overheads due to physical layer preamble and inter-frame spacing are still incurred. An analytical model to compute network capacity with bi-directional frame aggregation was proposed in [5]. It was reported that aggregation significantly improves channel utilization under saturated conditions (i.e. when all stations have data to transmit at all times). In [9], Abraham et al. presented an overview of MAC and physical layer enhancements to improve channel utilization. Using simulation experiments, they demonstrated that the proposed enhancements significantly improve application layer throughput.

B. Motivation

To the best of our knowledge, there is no work in the literature that provides insights on the impact of aggregate size on channel utilization in IEEE 802.11n networks. In particular, it remains unclear if:

- (i) frame aggregation *always* improves channel utilization irrespective of network load conditions.
- (ii) any arbitrary choice of aggregate size will improve channel utilization.

Moreover, a static assignment of aggregate size may lead to the following performance trade-off. If the aggregate size is small, then the transmission overheads resulting from physical layer preamble and header, inter-frame spacing and contentions, may play a dominant role in lowering channel utilization. On the other hand, if the aggregate size is large, then the amount of time spent by certain frames in the transmission queue until an aggregate is formed might affect the quality of service experienced by higher layers. For this reason, it is important to identify the criteria for selecting the aggregate size and characterize the impact of chosen aggregate size on channel utilization.

C. Our contributions

Major contributions of our work in this paper are:

- Analytically model the transmission queue at the MAC layer of an 802.11n wireless station as a bulk service queuing system assuming Poisson packet arrival from upper layers.

[†] Research for this paper was conducted while the first author was with Intel Corp., USA during Summer 2005.

- Provide explicit expressions for state probabilities at departure, arrival and random epochs. These state probabilities provide a framework to derive several performance metrics like mean queue length, mean waiting time, frame latency, utilization, blocking probability, etc.
- Provide a framework to derive optimal aggregate size for a given arrival and service distribution.
- Show that the choice of an aggregate size not only depends on the offered traffic within a station, but also (more interestingly) on data frame sizes.
- Deduce that there is no unique aggregate size that maximizes utilization under all operating conditions. Thus, there is need for a dynamic assignment of aggregate size based on traffic arrival rate and data frame size.

The rest of this paper is organized as follows. In Section II, we describe the system model, list the assumptions and present the formulation of the transmission queue at the MAC layer of an 802.11n station as a bulk service queuing system. In Sections III and IV, we present the derivation of steady state probabilities at departure and arbitrary epochs, respectively. In Section V, we analyze the impact of various aggregate sizes on various performance metrics such as frame latency and channel utilization. Finally, Section VI concludes this paper.

II. SYSTEM MODEL AND ASSUMPTIONS

We model the transmission queue at the MAC layer of an 802.11n station as a bulk service queuing system with no vacations and minimum batch size rule. We now summarize more specific details of the queuing system considered. Throughout this paper, the terms *frame* and *job* in 802.11n and queuing systems literature, respectively, are used interchangeably.

The queuing system under study consists of a finite buffer (with capacity equal to N) and a batch server. We assume that jobs arrive to the system according to a Poisson process with rate λ . The batch server services the jobs in batches of size equal to K . Let $f_{S_b}(t)$ and $L_{S_b}(s)$ be the probability density function and Laplace-Stieltjes transform of the service time S_b of a typical batch. The mean service time for a batch, $E[S_b]$, can be derived as follows: $E[S_b] = -\frac{d}{ds}L_{S_b}(s)|_{s=0}$. When a service period ends and there are less than K jobs waiting in the queue, the server remains idle until K jobs have accumulated. This is referred to as servicing with *minimum batch size* rule. Due to the finite buffer capacity of size N , the maximum number of jobs allowed in the system at any time is equal to $(N + K)$. If a job on arrival finds N jobs in the queue, then it gets dropped (i.e. blocked) without being served. Based on the above description of the service model, the batch server is either idle or busy at any instant of time. Let I be the random variable denoting the length of idle period. Further, let $E[I]$ be the mean length of the idle period. As it can be seen, the described model can be represented as $M/G^{[K]}/1/N$ queuing system with no vacations.

III. STATE DISTRIBUTION AT DEPARTURE EPOCHS

A. Terms and definitions

Let a_m be the random variable corresponding to the number of arrivals to the queuing system during the *service time* of m^{th} batch of jobs. Let ξ_m denote the number of jobs in the queue immediately after the completion of service of the m^{th} batch. Then, the following relation holds:

$$\xi_{m+1} = \begin{cases} \min(a_{m+1}, N), & \text{if } \xi_m < K \\ \min(\xi_m - K + a_{m+1}, N), & \text{if } \xi_m \geq K \end{cases}$$

Let $\pi_j^{(D)}(m)$, $j = 0 \dots N$ be defined as follows:

$$\pi_j^{(D)}(m) = Prob.\{\xi_m = j\}$$

In this section, we shall determine the limiting probabilities

$$\pi_j^{(D)} = \lim_{m \rightarrow \infty} \pi_j^{(D)}(m) = \lim_{m \rightarrow \infty} Prob.\{\xi_m = j\}, \forall j = 0 \dots N$$

In the above equation, $\pi_j^{(D)}$ is the equilibrium state probability corresponding to the number of pending jobs in the queue at a departure instant (i.e. at the instant of service completion of a batch of jobs). Here, superscript D stands for departure time instants to signify the departure of jobs from the system. It can be easily proved using Ergodicity theorem ([7], [11]) that the distribution $\{\pi_j^{(D)}\}$ exists and is independent of the initial state of the system.

Let the state transition probabilities between two consecutive departure instants, say m and $m+1$, be defined as follows:

$$p_{ij} = Prob.\{\xi_{m+1} = j | \xi_m = i\}, 0 \leq i, j \leq N$$

Let A_j be defined as the probability of j arrivals *during the service time of a batch of jobs*. Since $f_{S_b}(t)$ is the probability density function of the service time of a batch of jobs, we have:

$$A_j = \int_{t=0}^{\infty} \frac{(\lambda t)^j}{j!} e^{-\lambda t} f_{S_b}(t) dt \quad (1)$$

B. Derivation of State Transition Probabilities

We shall now describe the derivation of state transition probabilities. For the ease of illustration, we categorize the state transitions into the following four classes:

Class I: Transitions from state $0 \leq i < K$ to state $0 \leq j < N$

The service of the $(m+1)^{st}$ batch of jobs begins only after K jobs have accumulated in the queue. Immediately after the accumulation of K jobs, they are dequeued and scheduled for service. Accordingly, the state transition from state $0 \leq i < K$ to state $0 \leq j < N$ depends only on the number of arrivals (exactly equal to j) that take place during the service time of the $(m+1)^{st}$ batch of jobs. This takes place with a probability equal to A_j .

Class II: Transitions from state $0 \leq i < K$ to state $j = N$

Once again, the transition from state $0 \leq i < K$ to state $j = N$ is dependent only on the number of arrivals (greater than or equal to N) that take place during the service time of the $(m+1)^{st}$ batch of jobs. This takes place with a probability equal to $\sum_{n=N}^{\infty} A_n$ due to finite buffer capacity.

Class III: Transitions from state $i \geq K$ to state $0 \leq j < N$

Note that there is no idle period between the departure instant of m^{th} batch and the beginning of service of the $(m+1)^{st}$ batch. Immediately after the $(m+1)^{st}$ batch begins to get serviced, there are $\xi_m - K$ pending jobs. Thus, a transition to state j ($\geq \xi_m - K$) will occur provided there are $j - (\xi_m - K)$ arrivals during the service time of $(m+1)^{st}$ batch. Thus, a transition from state $i \geq K$ to state $i - K \leq j < N$ takes place with probability equal to $A_{j-(i-K)}$.

Class IV: Transitions from state $i \geq K$ to state $j = N$

Once again, there are $\xi_m - K$ jobs remaining in the queue when the $(m+1)^{st}$ batch begins to get serviced. Due to finite

buffer capacity, jobs will be blocked if the number of arrivals during the service of $(m+1)^{st}$ batch exceeds $N - (\xi_m - K)$. Thus, a transition from state $i \geq K$ to state $j = N$ takes place with probability equal to $\sum_{n=N-(i-K)}^{\infty} A_n$.

The following equation summarizes the state transition probabilities:

$$p_{ij} = \begin{cases} \begin{cases} A_j, & \text{if } 0 \leq j \leq N-1 \\ \sum_{n=N}^{\infty} A_n, & \text{if } j = N \end{cases} & ; 0 \leq i < K \\ 0, & \text{if } 0 \leq j < i - K \\ \begin{cases} A_{j-(i-K)}, & \text{if } i - K \leq j \leq N-1 \\ \sum_{n=N-(i-K)}^{\infty} A_n, & \text{if } j = N \end{cases} & ; K \leq i \leq N \end{cases} \quad (2)$$

Using these transition probabilities, the equilibrium state probabilities $\pi_j^{(D)}$, $j = 0 \dots N$ can be obtained by solving the balance equations along with the normalization condition. These equations are as follows:

$$\pi_j^{(D)} = \sum_{i=0}^{\min(j+K, N)} \pi_i^{(D)} p_{ij}, \quad \forall j = 0 \dots N \quad (3)$$

$$\sum_{j=0}^N \pi_j^{(D)} = 1 \quad (4)$$

IV. STATE DISTRIBUTION AT ARBITRARY EPOCHS

In [8], Chaudhry and Gupta derive state probabilities at arbitrary epochs by employing a bi-dimensional Markov process to model "residual service times" in queuing systems with general service distributions by maintaining *supplementary variables*. Here, we provide a simpler and more straightforward approach based on renewal theory to derive state probabilities at random epochs.

A. Terms and definitions

Let $\pi_j^{(A)}$ and π_j , where $j = 0 \dots N$, be the limiting state probabilities corresponding to the number of pending jobs in the queue at an arrival and arbitrary time instant, respectively. As before, the superscript A signifies arrival time instant of a job. Due to the memoryless property of time for next arrival, the arrival of a job is independent of any other event in the queuing system. Hence, it follows that $\pi_j^{(A)} = \pi_j$, $\forall j = 0 \dots N$ [6]. Recall that our service model assumes no server vacations. Hence, the server is either idle or busy at any instant of time. Also, note that if the server is found idle, then it must be true that there are less than K pending jobs in the queue. Accordingly, let $\pi_{j, idle}$, $0 \leq j < K$ be the limiting probability that the server is found idle at an arbitrary time instant and there are exactly j pending jobs in the queue. Likewise, let $\pi_{j, busy}$, $0 \leq j \leq N$ be the limiting probability that the server is found busy at an arbitrary time instant and there are exactly j pending jobs in the queue. Note that

$$\pi_j = \begin{cases} \pi_{j, idle} + \pi_{j, busy}, & 0 \leq j < K \\ \pi_{j, busy}, & K \leq j \leq N \end{cases} \quad (5)$$

Finally, let P_{idle} and P_{busy} be the equilibrium probability of finding the server idle and busy at any arbitrary instant of time, respectively. In the remainder of this section, we shall determine expressions for limiting probabilities $\pi_{j, idle}$ and $\pi_{j, busy}$.

B. Expression for $\pi_{j, idle}$

Probability of finding the batch server idle at an arbitrary time instant (i.e. P_{idle}) is the fraction of time the server stays idle in the time interval between two successive embedded points (i.e. departure instants). Without loss of generality, consider any two successive departure instants, say m and $m+1$. Time interval between embedded points m and $m+1$ consists of an idle period (if $\xi_m < K$) and a service period. From the definition of $E[I]$ and $E[S_b]$, we get:

$$P_{idle} = \frac{E[I]}{E[I] + E[S_b]} \quad (6)$$

The server begins to serve the $(m+1)^{st}$ batch of jobs only when there are $\geq K$ jobs pending in the queue. Suppose, $\xi_m = i$ ($< K$). On average, it takes $\frac{1}{\lambda}$ time units for a job to arrive. Hence, the idle period before the server begins to serve the $(m+1)^{st}$ batch is equal to $(K-i)\frac{1}{\lambda}$. Thus, average idle period between the departure instants m and $m+1$ is:

$$E[I](m) = \sum_{i=0}^{K-1} (K-i) \frac{1}{\lambda} \pi_i^{(D)}(m) + \sum_{i=K}^N 0 \times \pi_i^{(D)}(m)$$

where $\pi_i^{(D)}(m) = Prob.\{\xi_m = i\}$. In the above, expression $E[I](m)$ is the mean idle period observed immediately after the departure of m^{th} batch. The second term signifies that there is no idle period between departure instants m and $m+1$, if $\xi_m \geq K$. Taking limits as $m \rightarrow \infty$ on both sides, we get

$$\begin{aligned} E[I] &= \lim_{m \rightarrow \infty} E[I](m) = \lim_{m \rightarrow \infty} \sum_{i=0}^{K-1} (K-i) \frac{1}{\lambda} \pi_i^{(D)}(m) \\ &= \sum_{i=0}^{K-1} (K-i) \frac{1}{\lambda} \pi_i^{(D)} \end{aligned} \quad (7)$$

Probability of finding the server idle and exactly j ($0 \leq j < K$) pending jobs in the queue at an arbitrary instant of time (i.e. $\pi_{j, idle}$) is simply the product of P_{idle} and the fraction of idle period the queue has exactly j jobs. Recall that if the server is found to be idle, then there are less than K jobs in the queue. Once again, without loss of generality, consider two successive departure instants, m and $m+1$. Since $j < K$, it is possible to find exactly j jobs in the queue (during the idle period between departure instant m and $m+1$) for a duration of $\frac{1}{\lambda}$ time units (which is the mean inter-arrival duration) provided $0 \leq \xi_m \leq j$. Thus,

$$\pi_{j, idle}(m) = P_{idle} \times \frac{\sum_{i=0}^j \pi_i^{(D)}(m) \frac{1}{\lambda}}{E[I](m)}$$

where $\pi_{j, idle}(m)$ is the probability of finding the server to be idle and exactly j pending jobs in the queue immediately after the departure event of m^{th} batch. Taking limits as $m \rightarrow \infty$ on both sides, we get:

$$\begin{aligned} \pi_{j, idle} &= \lim_{m \rightarrow \infty} \pi_{j, idle}(m) \\ &= \lim_{m \rightarrow \infty} P_{idle} \times \frac{\sum_{i=0}^j \pi_i^{(D)}(m) \frac{1}{\lambda}}{E[I](m)} \\ &= P_{idle} \times \frac{\sum_{i=0}^j \pi_i^{(D)} \frac{1}{\lambda}}{E[I]} \\ &= \frac{\sum_{i=0}^j \pi_i^{(D)}}{\lambda E[S_b] + \sum_{i=0}^{K-1} (K-i) \pi_i^{(D)}}, \quad 0 \leq j < K \end{aligned} \quad (8)$$

C. Expression for $\pi_{j,busy}$

Like P_{idle} , the probability of finding the batch server busy at an arbitrary time instant (i.e. P_{busy}) is the fraction of time the server stays busy in the time interval between two successive embedded points. Thus,

$$P_{busy} = \frac{E[S_b]}{E[I] + E[S_b]} \quad (9)$$

The probability of finding the server busy and exactly j ($0 \leq j < N$) pending jobs in the queue at an arbitrary instant of time (i.e. $\pi_{j,busy}$) can be expressed as the product of P_{busy} and the fraction of busy period the queue has exactly j jobs. Let $E[T_{j,busy}]$ be the average time interval during the busy period when the queue has exactly j jobs. Thus,

$$\pi_{j,busy} = P_{busy} \times \frac{E[T_{j,busy}]}{E[S_b]} = \frac{E[T_{j,busy}]}{E[I] + E[S_b]} \quad (10)$$

Once again, consider two successive embedded points m and $m+1$. Then, for $0 \leq j < N$,

$$E[T_{j,busy}](m) = Prob.\{\xi_{m+1} > j | \xi_m \leq \min(K+j, N)\} \times \frac{1}{\lambda}$$

where $E[T_{j,busy}](m)$ is the average time interval during the busy period following m^{th} departure instant when the queue has exactly j pending jobs. Note that the mean inter-arrival time is $\frac{1}{\lambda}$ time units. The inequalities $\xi_m \leq \min(K+j, N)$ and $\xi_{m+1} > j$ guarantee that the queue has exactly j jobs for a duration of $\frac{1}{\lambda}$ time units during the busy period (between the departure instants m and $m+1$). Upon further simplification, we get

$$E[T_{j,busy}](m) = \frac{1}{\lambda} \sum_{i=0}^{\min(K+j, N)} \left(\pi_i^{(D)}(m) \sum_{k=j+1}^N p_{ik} \right)$$

Taking limits as $m \rightarrow \infty$ on both sides, we get

$$\begin{aligned} E[T_{j,busy}] &= \frac{1}{\lambda} \sum_{i=0}^{\min(K+j, N)} \pi_i^{(D)} \sum_{k=j+1}^N p_{ik} \\ &= \frac{1}{\lambda} \sum_{i=0}^{\min(K+j, N)} \pi_i^{(D)} \left(1 - \sum_{k=0}^j p_{ik} \right), \because \sum_{k=0}^N p_{ik} = 1 \\ &= \frac{1}{\lambda} \left(\sum_{i=0}^{\min(K+j, N)} \pi_i^{(D)} - \sum_{i=0}^{\min(K+j, N)} \pi_i^{(D)} \sum_{k=0}^j p_{ik} \right) \\ &= \frac{1}{\lambda} \left(\sum_{i=0}^{\min(K+j, N)} \pi_i^{(D)} - \sum_{k=0}^j \pi_k^{(D)} \right), \text{ Using (3)} \\ &= \frac{1}{\lambda} \sum_{i=j+1}^{\min(K+j, N)} \pi_i^{(D)} \end{aligned}$$

Hence,

$$\begin{aligned} \pi_{j,busy} &= \frac{\frac{1}{\lambda} \sum_{i=j+1}^{\min(K+j, N)} \pi_i^{(D)}}{E[S_b] + \frac{1}{\lambda} \sum_{i=0}^{K-1} (K-i) \pi_i^{(D)}} \\ &= \frac{\sum_{i=j+1}^{\min(K+j, N)} \pi_i^{(D)}}{\lambda E[S_b] + \sum_{i=0}^{K-1} (K-i) \pi_i^{(D)}}, \quad 0 \leq j < N \end{aligned}$$

Finally,

$$\pi_{N,busy} = 1 - \left(\sum_{j=0}^{K-1} \pi_{j,idle} + \sum_{j=0}^{N-1} \pi_{j,busy} \right) \quad (11)$$

V. PERFORMANCE EVALUATION

A. Service Time Distribution

The service time distribution ($f_{S_b}(t)$) required to compute the performance metrics in the considered $M/G^{[K]}/1/N$ queuing system corresponds to the probability distribution of the MAC layer latency¹. The latter distribution is obtained by accurately modeling the exponential backoff process of DCF scheme in presence of frame aggregation. To the best of our knowledge, there is no work in the literature that determines the distribution of MAC layer latency in a network with stations having finite queue size, operating under unsaturated load conditions and employing frame aggregation. The focus of our work is evaluating the effectiveness of frame aggregation and not characterizing the MAC layer service time. Hence, we estimate *approximate* service time distribution using results obtained from experiments. The approximated distribution not only characterizes the MAC layer service time distribution, but also provides a convenient methodology to derive several performance metrics which are otherwise non-trivial to compute from simulation experiments.

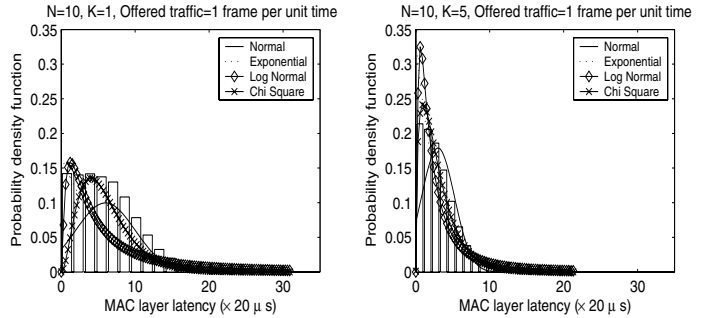


Fig. 1. Identifying approximate service time distribution

Fig. 1 shows the probability density function for the MAC layer latency observed at an 802.11n station. The histograms represent simulation results derived from experiments using the setup described in the next subsection. We used `dfittool` included in the Statistics toolbox of Matlab R13 [4] to obtain approximate continuous time distributions. Fig. 1 depicts several well-known continuous distributions like Normal, Exponential, Log-Normal and Chi Square. From the figure, we observe that Chi Square distribution with degrees of freedom equal to 6 for $K=1$ and 3 for $K=5$ provides a good approximation for MAC layer service time.

B. Simulation Setup

We considered a one-hop ad hoc network wherein all the stations were kept stationary and within range of each other. All the stations communicated with a single dummy station (i.e. the access point). The traffic generation rate was varied over a wide range of values covering saturated and unsaturated

¹MAC layer latency is defined as the duration from the time instant at which backoff procedure is initiated for an aggregate to the instant at which the aggregate is successfully transmitted.

network conditions. Since aggregated frames are transmitted only after a successful handshake of short control frames between the source and destination (like IAC and RAC in [1]), we considered only RTS/CTS mechanism of the legacy IEEE 802.11 Distributed Coordination Function (DCF). Default 802.11b MAC and Direct Sequence Spread Spectrum PHY parameters [3] were used for all the simulations. Following changes were made to the implementation of the *wlan-mac process model* in Opnet to simulate frame aggregation:

- 1) Exit condition from Idle state was modified so that transition to Defer or Backoff takes place only when the number of data frames in HigherLayerDataQueue (HLDQ) \geq aggregate size, say K .
- 2) Immediately after receiving a CTS frame, an aggregate frame of length equal to the sum of the lengths of K frames from the head-of-line in HLDQ was generated. This ensures proper computation of the time required to transmit the aggregated frame in Opnet. Note that all the K data frames in HLDQ are destined to the same station (i.e. access point).
- 3) After generating the aggregate frame, all the K data frames from the head-of-line in HLDQ were dequeued. Since we consider one-hop ad hoc network and employ RTS/CTS mechanism, collisions involve only RTS frames. So, if CTS is received by a station, it is guaranteed that ensuing data transmission will succeed. This justifies discarding K data frames from HLDQ.
- 4) Exit condition from Frame End state was modified such that after the receipt of acknowledgment frame, transition to Defer or Backoff takes place only if number of pending frames in HLDQ $\geq K$. If the number of pending frames $< K$, a state transition takes place to Idle state.

All the simulation experiments were run for 500 seconds of simulated time.

C. Discussion of Results

In the context of an 802.11n network, a departure instant corresponds to the completion of a successful transmission of a frame aggregate. Time between two successive departure instants is composed of:

- (i) Wait time to build an aggregate (if number of frames in the transmission queue following a successful transmission is less than the aggregate size). Recall that $E[I]$, as defined in Section II, gives the average wait time spent on building an aggregate.
- (ii) Time spent to win the channel contention via the exponential backoff process. This constitutes the backoff overheads. A station is said to be a winner if a CTS frame is successfully received in response to the transmission of an RTS frame.
- (iii) Time spent on inter-frame spacing, transmitting the physical header, preamble and frame aggregate. This time shall be collectively referred to as transmission overheads.

Recall that $E[S_b]$, as defined in Section II, denotes the average time spent on backoff and transmission overheads. Note that once an aggregate is formed, the time spent on backoffs to win the channel contention is independent of the choice of aggregate size.

A small aggregate size not only decreases $E[I]$, but also reduces the time spent on physical transmission of the aggregate. However, it may not be able to mitigate backoff and

transmission overheads, thereby degrading channel utilization. On the other hand, a large aggregate size can mitigate both transmission and backoff overheads. However, the extra time spent on accumulating an aggregate may not only degrade utilization, but also affect the quality of service experienced by higher layers. In order to characterize this performance trade-off, we consider the following two metrics.

1) *Mean waiting time in queue*: In equilibrium state, average number of data frames yet to be transmitted in the transmission queue at the MAC layer is given by:

$$E[N_w] = \sum_{j=0}^N j \times \pi_j$$

Since not all arriving jobs are admitted due to finite buffer capacity, the effective arrival rate to the transmission queue is given by $\lambda_{eff} = \lambda \times (1 - \pi_N)$. Mean waiting time is defined as the average time spent by an admitted frame in the transmission queue. Using Little's Law, we get:

$$E[T_w] = \frac{\text{Mean frames in queue}}{\text{Effective arrival rate}} = \frac{E[N_w]}{\lambda_{eff}} \quad (12)$$

2) *Utilization*: We define utilization as the percentage of time spent on transmission of *useful data bits* between two successive departure instants. It can be computed as follows:

$$\mathcal{U} = \frac{K \times \text{Transmission time for one data frame}}{E[I] + E[S_b]}$$

We used `lsqnonlin` function included in the Optimization toolbox of Matlab R13 [4] to obtain steady state probabilities at arbitrary epochs and the performance metrics defined above. Figs. 2(a) and 2(b) plot results for the performance metrics as a function of the offered traffic from higher layers (λ), which is measured as *frames per unit time* where *unit time* = `dot11aSlotTime` [3]. We considered four different data frame sizes, namely, 128, 256, 512 and 1024 bytes. Length of the MAC transmission queue (N) was set to 10 independent of the frame size. That is, with 128 byte frame size, the buffer capacity was 1280 bytes. Several possible values of K ($1 < K \leq N$) were considered. From figs. 2(a) and 2(b), we observe that:

- There is an optimal value of λ for a given value of K , say λ_K^{OPT} , for which the mean wait time is minimum and utilization is maximum. For $\lambda < \lambda_K^{OPT}$, the higher (lower) wait time (utilization) is largely due to the time spent in the transmission queue accumulating an aggregate. When $\lambda > \lambda_K^{OPT}$, the time spent by a frame waiting for the transmission of frames ahead of it in the queue becomes a dominating factor (due to increased contentions for the shared medium under high loads). This once again results in higher (lower) waiting time (utilization). Also, higher the value of K , greater is the value of λ_K^{OPT} . This calls for increasing aggregate size with increasing λ .
- Under low rates of offered traffic, the choice of aggregate size depends on the data frame size. For example, when $\lambda = 0.5$ frames/unit time and frame size equal to 128 bytes, utilization drops by 44% from 0.36 with $K = 1$ to 0.20 with $K = 10$. And, the difference in magnitude of mean wait time with $K = 1$ and $K = 10$ is in the order of *ms*. Clearly, such an increase in waiting time is not desirable as it may result in timer expiries at higher layers, affect quality of service experienced by real-time applications, etc. When $\lambda = 0.5$ frames/unit time and

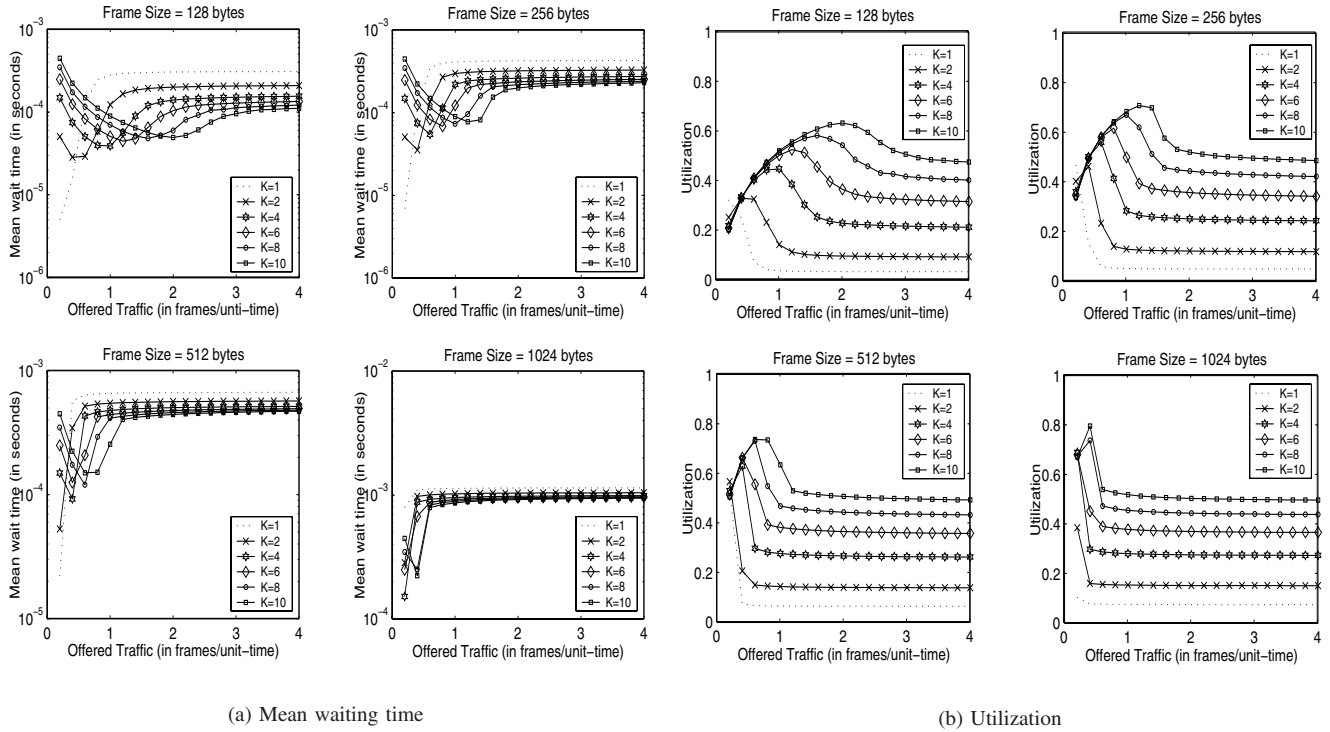


Fig. 2. Illustrating the impact of aggregate size

frame size equal to 1024 bytes, utilization increases by 700% from 0.10 with $K = 1$ to 0.70 with $K = 10$. At the same time, the magnitudes of mean wait time with $K = 1$ and $K = 10$ are comparable. Thus, an aggregate size equal to 10 can be used with 1024 byte frame size even under low loads.

- Under high loads, the choice of aggregate size is rather obvious. It is desirable to select as high an aggregate size as possible consistent with the inferences made in [5].

These observations suggest that there is no optimal aggregate size under all operating conditions. This motivates the need for a dynamic assignment of aggregate size based on traffic arrival rate and data frame size.

VI. CONCLUSIONS

Proposals for IEEE 802.11 Task Group n include data frame aggregation as one of the enhancements to improve channel utilization. In this paper, we characterized the impact of aggregate size on mean waiting time in the transmission queue at a station and utilization. We modeled the transmission queue at the MAC layer of an 802.11n station as a bulk service queuing system to estimate these metrics. The proposed model used realistic assumptions like finite transmission queue size and unsaturated network load conditions. The model can be used as an analytical framework to determine optimal frame aggregate size given a traffic arrival rate and MAC layer service rate. We studied the impact of aggregate size on channel utilization over a wide range of operating conditions and reported that there is no unique aggregate size that maximizes utilization under all scenarios. This motivates the need for a dynamic assignment of aggregate size.

ACKNOWLEDGMENTS

We would like to thank Changwen Liu of Intel Corp. for helping us with simulation of IEEE 802.11n in Opnet simulator tool, participating in several valuable discussions and helping us to improve the overall quality of this paper.

REFERENCES

- [1] IEEE P802.11 Wireless LANs: TGN Sync proposal technical specification, IEEE 802.11-04/0889r44.
- [2] IEEE P802.11 Wireless LANs: WWiSE proposal for high throughput extension to the 802.11 standard, IEEE 802.11-05/0149r1.
- [3] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical (PHY) Layer Specifications, ANSI/IEEE Std 802.11, 1999 Edition.
- [4] Matlab 7.0.1 The Language of Technical Computing, Webpage - <http://www.mathworks.com/products/matlab/>.
- [5] Changwen Liu and Adrian P. Stephens. An analytical model for infrastructure WLAN capacity with bidirectional frame aggregation. In *Proceedings of Wireless Conference on Networking and Communications*, pages 113–119, March 2005.
- [6] F.G. Foster and A.G.A.D. Perera. Queues with batch departures II. In *The Annals of Mathematical Statistics*, pages 1147–1156, September 1964.
- [7] F.G. Foster and K.M. Nyunt. Queues with batch departures I. In *The Annals of Mathematical Statistics*, pages 1324–1332, December 1961.
- [8] M.L. Chaudhry and U.C. Gupta. Modeling and analysis of $M/G^{a,b}/1/N$ queue-A simple alternative approach. In *Queueing Systems*, pages 95–100, March 1999.
- [9] S. Abraham, A. Meylan and S. Nanda. 802.11n MAC design and system performance. In *Proceedings of International Conference on Communications*, pages 2957–2961, May 2005.
- [10] V. Vitsas, P. Chatzimisios, A. C. Boucouvalas, P. Raptis, K. Paparrizos and D. Kleftouris. Enhancing performance of the IEEE 802.11 Distributed Coordination Function via packet bursting. In *Globecom Workshops*, pages 245–252, December 2004.
- [11] W. Feller. An introduction to probability theory and its applications. In *Wiley, Vol. I, 3rd Edition*, 1968.
- [12] Yang Xiao. Packing mechanisms for the IEEE 802.11n wireless LANs. In *Proceedings of Globecom*, pages 3275–3279, December 2004.