# Streaming versus Batch Processing of Sensor Data in a Hazardous Weather Detection System

Mark Sims, Jim Kurose, and Victor Lesser
Computer Science Department
University of Massachusetts
Amherst, MA 01003
Email: {msims,kurose,lesser}@cs.umass.edu

***Abstract*— We present several discrete-time Markov queuing models to compare the performance of batch versus streaming processing of sensor data in a weather detection and monitoring system architecture. The first model assumes independent arrivals and illustrates the average case behavior of the system. The remaining models assume correlated arrivals and demonstrate how different scan strategies across multiple elevations impact system performance. We also show how the models are useful in dimensioning the computational resources of the system given workload arrival characteristics. We apply the models to several hypothetical scenarios with varying weather feature arrival and processing rates. We also evaluate our models using processing runtime data obtained by running two NEXRAD algorithms on weather data from six airports. Our results show that for this particular application and reasonable system utilizations, delaying processing start time to perform batch processing does not adversely affect system performance or require significant, additional computational resources. More generally, our models show how system performance is dependent on the scan strategy and the burstiness of the sensor data.**

## I. Introduction

In this paper, we use discrete time Markov queuing models to compare the performance of batch versus streaming processing of sensor data in a weather detection and monitoring system. In particular, we model and analyze the feature detection subsystem of the Meteorological Command and Control (MC&C) component of the proposed architecture for the Collaborative Adaptive Sensing of the Atmosphere (CASA) weather prediction system [1], [2]. In the proposed MC&C, weather data from sensors are input to feature detection algorithms during a fixed window of time. During that window, the detection algorithms must process the data and deposit any detected features into a repository. These features are then used by the system to redirect the sensors to scan in new regions, and the process then repeats. We compare two approaches to processing the data from sensors. In the first, the detection algorithms process data in a streaming fashion by continuously processing the data during the time window. In the second, the algorithms use batch processing in which they wait some amount of time before beginning to process the accumulated data. Once they have processed all accumulated data, the algorithms switch to the streaming method for the remaining time. Our goal in undertaking this study is to determine the extent to which batch processing of “sit and spin” radars (in which radars are not retasked in real-time based on detected meteorological features) is still appropriate in the case that radars are dynamically retasked in real-time based on detected features.

We present several Markov chain models of the above system. Since feature processing occurs over a finite window, we perform a transient analysis. In the first model, we represent the feature detection system as a Geo/Geo/1 discrete-time queue. Features (which require processing to be detected) arrive independently during a slot according to a geometric distribution. Service completions within a slot are modeled with a Bernoulli random variable. This results in a one-dimensional Markov chain that we use to compute the work completion probability—the probability that at most one unprocessed feature will remain in the system at the end of the window (i.e., the probability that the work that has arrived during the interval is processed by the end of the interval). This allows us to find the latest time to begin batch processing such that the work completion probability is greater than or equal to some target probability. The model is also used to determine the processing power needed to ensure that the work completion probability is above the target given known feature arrival characteristics and a desired batch processing start time.

The next two models are extensions of the first, but no longer assume independent feature arrivals. In slightly different ways, each uses a two-state source Markov process to model correlated meteorological feature arrivals. The motivation for this model is that features of weather phenomena arrive in clusters that are dependent on the elevation at which radar scans take place. In one model, the rate at which features arrive is dependent on the state of the source process. In the other, it is the processing time that is dependent on the state of the source process.

To do the analysis for all models, we explore several hypothetical scenarios as well as utilize empirical runtimes derived from two NEXRAD [3] algorithms run on data collected by several NEXRAD radars. Our results show that for this particular application and reasonable system utilizations, delaying processing start time to perform batch processing does not adversely affect system performance or require significant, additional computational resources. More generally, our models show how system performance is dependent on the radar scan strategy and the burstiness of the sensor data.

The remainder of the paper is structured as follows. Section II discusses the simple model in which features arrive independently of one another. Section III discusses the extended models with correlated feature arrivals. We conclude in Section IV.

## II. Independent Arrivals

In this section, we model the MC&C feature detection system as a Geo/Geo/1 queue with finite capacity. This is a discrete time model in which time is divided into equally sized intervals or slots. Meteorological features arrive at the queue during each slot according to a geometric distribution and service begins on a feature only at the beginning of a slot [4]. Thus, the earliest that service can begin on a feature is in the slot immediately after the one in which it arrived. At most one feature can be processed during a slot and service completions are modeled by a Bernoulli random variable. The resulting model of this system is a one-dimensional Markov chain.

We use discrete-time queues, as opposed to continuous-time queues, as the basis of our models because exact, time-dependent solutions for continuous time queuing systems are computationally complex and intractable for all but the simplest models [5], requiring approximate numerical approaches instead [6]. Our discrete-time model lends itself to a simpler approach by which we can approximate continuous time queuing solutions if the slot size used is small enough. Another decision we made was to focus on the transient behavior of the queue over a finite time window as we are interested in the time-dependent behavior of feature detection, especially at the end of the time window rather than its steady state behavior.

TABLE I

Parameters for the model with independent arrivals.

| | |
|---|---|
| $N$ | System capacity |
| $s$ | Number of slots per second |
| $[0, \tau]$ | Processing interval |
| $t = s\tau$ | Number of slots in $[0, \tau]$ |
| $\tau_0$ | Processing start time (seconds) |
| $t_0 = s\tau_0$ | Processing starting slot |
| $m = t - t_0$ | Processing duration in slots |
| $a(n) = (1-p)p^n$ | Feature arrival distribution per slot |
| $\lambda = \frac{sp}{1-p}$ | Mean feature arrival rate |
| $\sigma$ | Prob. service will complete in current slot |
| $u = 1 - \sigma$ | Prob. service will require one or more slots |
| $\mu = s\sigma$ | Mean service completion rate |
| $\rho = \frac{\lambda}{\mu}$ | System utilization |
| $\xi$ | Prob. of at most one feature at slot $t$ |
| $\beta(i)$ | Prob. of $i$ arrivals by slot $t_0$ |

While there has been considerable research pertaining to transient and nonstationary analysis of queues (for example [6]–[11]), to our knowledge this work is the first to use transient analysis to compare streaming versus batch processing of sensor data and to determine the amount of computational resources needed to achieve satisfactory system performance in each case.

### A. Definitions

Table I summarizes the parameters of our first model. $N$ is maximum number of features able to be in the system at one time, including the feature being processed. We assume that features are lost if they arrive when the system is at capacity. (Note that we can choose $N$ large enough to make the probability of this event exceedingly small.) The parameter $s$ gives the number of slots per second, which allows us to control the granularity of the model. The interval $[0, \tau]$ gives the window of time during which weather data arrive from sensors and feature processing must occur. $t = s\tau$ is the number of slots over the interval $[0, \tau]$. $\tau_0$ and $t_0 = s\tau_0$ are the time in seconds and number of slots at which processing begins. Obviously, $0 \leq \tau_0 \leq \tau$ and $0 \leq t_0 \leq t$. Processing lasts $m = t - t_0$ slots after $t_0$.

In our first model, features arrive according to a geometric distribution with parameter $p$:

$$a(n) = (1-p)p^n \qquad n = 0, 1, 2, \ldots \tag{1}$$

where $n$ is the number of arrivals in a slot. Given this distribution, the mean number of feature arrivals per slot is given by

$$E[n] = \sum_{n=0}^{\infty} n(1-p)p^n = \frac{p}{1-p}$$

and the mean number of features per second $\lambda$ is given by $\frac{sp}{1-p}$.

Let $\sigma$ be the probability that service on the feature being processed will complete in the current slot, and $u = 1 - \sigma$ be the probability that service on the feature will require at least one more slot. Since service completions are governed by a Bernoulli random variable, the mean number of completions per second is given by $\mu = s\sigma$. Given $\lambda$ and $\mu$, let $\rho = \frac{\lambda}{\mu}$ be the system utilization.

Finally, our performance metric $\xi$ is the work completion probability—the probability that at most one unprocessed feature will remain in the system at the end of the processing window at slot $t$. $\beta(i)$ is the probability that $i$ features have accumulated in the system by the time processing begins at slot $t_0$. We discuss $\xi$ and $\beta(i)$ in more detail below.

We use a one-dimensional Markov chain to model the system. The state space is the set $\{i = 0, 1, 2, \ldots, N\}$ where $i$ is the number of features (including the one in service) in the system. With the definitions in Table I, it is relatively straightforward to determine the following one-step transition probabilities of the Markov chain.

$$P_{0j} = a(j), \qquad 0 \le j < N \quad (2)$$

$$P_{ii-1} = \sigma a(0), \qquad 0 < i \le N \quad (3)$$

$$P_{ii+j} = \sigma a(j+1) + u a(j), \quad 0 < i \le i+j < N \quad (4)$$

$$P_{iN} = 1 - \sum_{j=0}^{N-1} P_{ij}, \qquad 0 \le i \le N \quad (5)$$

$$P_{ij} = 0, \qquad \mathit{otherwise} \quad (6)$$

The one-step transition probabilities define the one-step transition probability matrix, $\mathbf{P}$. From $\mathbf{P}$ we can compute the $m$-step transition probability matrix by computing $\mathbf{P}^m$ [12], where $m = t - t_0$ is the number of slots during which processing occurs. $\mathbf{P}^m$ provides the probability of starting in any state $i$ during slot $t_0$ and ending in any other state $j$ when processing stops at slot $t$. To compute the work completion probability $\xi$, we must condition on the starting state at $t_0$. This is the number of features that have accumulated in the system from slot zero to slot $t_0$ when processing starts. To do this, we must first find the distribution of the number of features that have arrived by slot $t_0$.

Let $X(t_0)$ be the number of features that have accumulated in the system before processing begins at slot $t_0$ and $\beta(i)$ be the probability that $X(t_0) = i$. $\beta(i)$ has the following distribution:

$$\beta(i) = \begin{cases} \binom{i+t_0-1}{t_0-1}(1-p)^{t_0}p^i & \text{for } 0 \le i < N, \\ 1 - \sum_{k=0}^{N-1}\beta(k)\} & \text{for } i = N, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

To see that this is the case, consider the distribution of arrivals within a single slot $a(n) = (1-p)p^n$ for $n = 0, 1, 2, \ldots$. The parameter $p$ can be seen as the probability that a first feature will arrive in a slot and $1-p$ as the probability that no more features will arrive in a slot. In order for there to be $i$ arrivals in $t_0$ slots, $i$ features must arrive and features must stop arriving $t_0$ times. Thus, we can think of there being $i + t_0$ flips of a coin necessary to have $i$ arrivals. Since the last time that features stop arriving must occur on the final flip of the coin there are a total of $\binom{i+t_0-1}{t_0-1}$ ways in which the $i$ features can arrive. This accounts for the case when $0 \le i < N$. The second case follows since the system has finite capacity, and the third since there can only be a positive number of arrivals.

Given the distribution of the number of features that have accumulated by time $t_0$ and the $m$-step transition probability matrix, $\mathbf{P}^m$, we are in a position to compute the work completion probability, $\xi$. Let $\xi_0(m)$ and $\xi_1(m)$ be the probabilities of having zero and one features in the system respectively when processing stops at slot $t$. We determine these by conditioning on the number of features in the system when processing starts at $t_0$.

$$\xi_0(m) = \sum_{i=0}^{N-1} \beta(i) P_{i0}^m + \big[1 - \sum_{i=0}^{N-1} \beta(i)\big] P_{N0}^m$$

$$\xi_1(m) = \sum_{i=0}^{N-1} \beta(i) P_{i1}^m + \big[1 - \sum_{i=0}^{N-1} \beta(i)\big] P_{N1}^m$$

where $P_{ij}^m$ is an entry in $\mathbf{P}^m$ and $1 - \sum_{i=0}^{N-1} \beta(i)$ is the probability of being in state $N$ at $t_0$. Therefore,

$$\xi = \xi_0(m) + \xi_1(m)$$

gives the work completion probability $\xi$.

### B. Numerical Calculations

All computations in this and subsequent sections were performed using Matlab.

*1) Latest Processing Start Time:* Our first performance metric of interest is the workload completion probability $\xi$. Our goal is to find the latest time at which processing can start, $\tau_0^{max}$, such that $\xi$ is greater than some target probability given $\lambda$ and $\mu$. Since actual values for $\lambda$ and $\mu$ are not yet available for the CASA project, we take two approaches to exploring the model. We first compare $\tau_0^{max}$ across several hypothetical scenarios. We then use the runtime data that we have collected on NEXRAD detection algorithms to evaluate the model. In this and subsequent sections we assume that $\tau = 30$ seconds.

Tables II and III summarize the results for two values of $\mu$ and two target probabilities. In all cases $s$ was held constant at ten slots per second, and the system capacity was held constant at 20 features. In the scenario shown in Table II, $\mu$ was held constant at 1 feature processed per second, and $\lambda$ increased such that $\frac{\lambda}{\mu}$ had the values shown. For the target probability of 0.9 in this scenario, the largest possible value of $\lambda$ such that the target could be reached at all was 0.3. For a target of 0.95, the largest value was 0.2. In the second scenario $\mu$ was held constant at 4 features per second.

It is also instructive to view how $\xi$ changes as we increase the processing start time. Figures 1 and 2 plot $\xi$ versus processing start time for some of the values of $\lambda$ and $\mu$ in the two scenarios represented by Tables II and III. The graphs in Figures 1 and 2 are flat initially, and as the ratio of $\lambda$ to $\mu$ increases, the duration over which each graph is flat decreases. This indicates that for the relatively small values of $\frac{\lambda}{\mu}$ discussed here, there is little difference between streaming processing (when $t_0 = 0$) and batch processing even if batch processing starts as much as half way through the processing window.

Another implication is that the initial state of the system does not significantly affect the outcome if the processing start time is early. In other words, although we are analyzing the system over a finite period of time, if $\frac{\lambda}{\mu}$ is small and the processing start time is not too large, the system approaches its steady state distribution. We have verified this by comparing the value of $\xi$ when $\tau_0 = 0$ to the analogous value in a continuous time M/M/1/N queue with the same $\lambda$, $\mu$ and $N$. As the number of slots per second increases in the Geo/Geo/1 queue, the value of $\xi$ approaches the closed form solution of the continuous case.

Tables II and III and Figures 1 and 2 illustrate the effect that $\lambda$ and $\mu$ have on the transient behavior of the system. They show that for a given value of $\frac{\lambda}{\mu}$, the graph is flat for longer and $\tau_0^{max}$ is higher if the actual values of $\lambda$ and $\mu$ are higher.

TABLE II

VALUES OF $\tau_0^{max}$ FOR TWO TARGET PROBABILITIES IN THE SCENARIO WHERE $\mu$ IS HELD CONSTANT AT 1 FEATURE PER SECOND AND $\lambda$ IS INCREASED. $\tau_0^{max}$ IS THE LATEST TIME THAT PROCESSING CAN START WITHIN THE PROCESSING WINDOW $[0, \tau]$ SUCH THAT THE WORK COMPLETION PROBABILITY IS AT LEAST THE TARGET.

| $\tau = 30$ sec | | | Target Prob. = 0.9 | Target Prob. = 0.95 |
|---|---|---|---|---|
| $\lambda$ | $\mu$ | $\frac{\lambda}{\mu}$ | $\tau_0^{max}$ (seconds) | $\tau_0^{max}$ (seconds) |
| 0.1 | 1 | 0.1 | 24 | 23 |
| 0.125 | 1 | 0.125 | 23 | 21 |
| 0.15 | 1 | 0.15 | 22 | 20 |
| 0.175 | 1 | 0.175 | 21 | 18 |
| 0.2 | 1 | 0.2 | 19 | 16 |
| 0.225 | 1 | 0.225 | 18 | |
| 0.25 | 1 | 0.25 | 16 | |
| 0.275 | 1 | 0.275 | 14 | |
| 0.3 | 1 | 0.30 | 12 | |

TABLE III

VALUES OF $\tau_0^{max}$ FOR TWO TARGET PROBABILITIES IN THE SCENARIO WHERE $\mu$ IS HELD CONSTANT AT 4 FEATURE PER SECOND AND $\lambda$ IS INCREASED. $\tau_0^{max}$ IS THE LATEST TIME THAT PROCESSING CAN START WITHIN THE PROCESSING WINDOW $[0, \tau]$ SUCH THAT THE WORK COMPLETION PROBABILITY IS AT LEAST THE TARGET.

| $\tau = 30$ sec | | | Target Prob. = 0.9 | Target Prob. = 0.95 |
|---|---|---|---|---|
| $\lambda$ | $\mu$ | $\frac{\lambda}{\mu}$ | $\tau_0^{max}$ (seconds) | $\tau_0^{max}$ (seconds) |
| 0.4 | 4 | 0.1 | 25 | 25 |
| 0.5 | 4 | 0.125 | 24 | 24 |
| 0.6 | 4 | 0.15 | 23 | 23 |
| 0.7 | 4 | 0.175 | 23 | 22 |
| 0.8 | 4 | 0.2 | 22 | 21 |
| 0.9 | 4 | 0.225 | 21 | |
| 1.0 | 4 | 0.25 | 21 | |
| 1.1 | 4 | 0.275 | 20 | |
| 1.2 | 4 | 0.30 | 18 | |

*2) Independent Arrivals and NEXRAD Data:* The above examples explored the independent arrival model for several hypothetical scenarios. We next use empirical data from two NEXRAD algorithms to evaluate the model. Doing so requires a slight modification. As described, the model assumes that fine-grained features arrive individually to be processed. The NEXRAD data we have, however, are at the granularity of entire elevation scans. Each elevation scan contains a number of

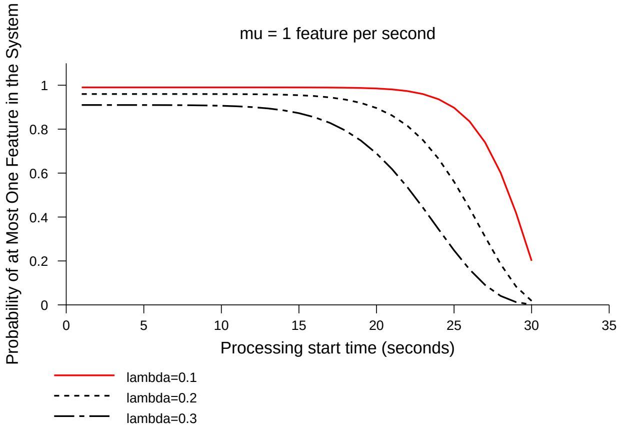

Fig. 1. Probability of at most one feature in the system at the end of the 30 second processing window vs. processing start time for three values of $\frac{\lambda}{\mu}$ where $\mu$ is held constant at 1 feature processed per second.

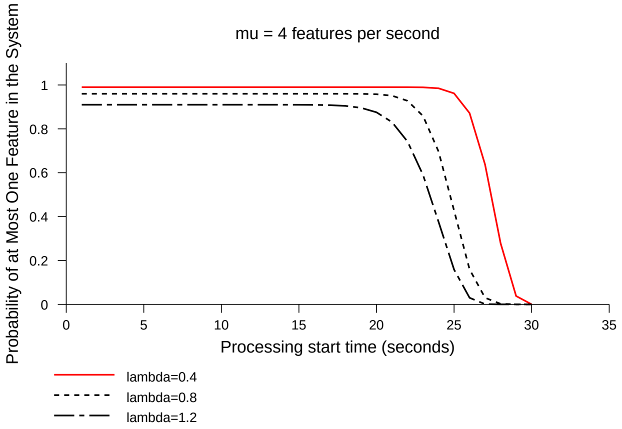

Fig. 2. Probability of at most one feature in the system at the end of the 30 second processing window vs. processing start time for three values of $\frac{\lambda}{\mu}$ where $\mu$ is held constant at 4 features processed per second.

features, and, as such, the processing per elevation varies according to the number of features in the scan. A scan with many features will take longer to process than a scan of a clear sky, which has few features. As such, we modify the model to view each “arrival” as bringing a group of features. $\lambda$ then is the rate at which elevation scans arrive, and $\mu$ is the mean processing rate of scans.

The two algorithms whose runtimes we have benchmarked are llsd (Linear Least Squares Derivative) and w2merger which are used in the Warning Decision Support System - Integrated Information (WDSS-II) [13]. The first algorithm calculates wind shear; the second algorithm merges data in overlapped scanned regions and performs radial-to-Cartesian coordinate transformation. The raw data was collected from single radars at six airports. To process the data and obtain runtime information, both algorithms were run using a 3.25 gHz Xeon processor, using this raw input data.

Given the data, we can determine the parameters of the independent arrival model. To determine the mean service completion rate $\mu$, we use the runtime data. The mean runtime for llsd was 0.193 seconds per elevation and 0.272 seconds for w2merger. The mean total processing rate $\mu$ is thus 2.151 elevation scans per second. (Note that in the actual CASA implementation a suite of detection algorithms will be run on incoming data, all of which will contribute to the total runtime.) As in the above subsection, we assume that the processing times are geometrically distributed.

To determine the arrival rate $\lambda$ of elevation scans, we assume that on average three elevation scans arrive from a single radar during the thirty second interval. This gives a mean arrival rate of $\lambda = 0.1$ elevations per second. We model the number of arrivals per slot as being geometrically distributed with this mean arrival rate. The geometric variability in the arrival process models the fact that radar dwell times can be variable; the networked transfer of radar data to a remote processing site adds further variability to the interarrival times. Because in the CASA project data will arrive from multiple radars, we also run the model using several larger values of $\lambda$ to simulate scenarios with a greater number of radars.

Table IV and Figure 3 summarize the results of applying the above parameters to the model. Again the processing window is assumed to be 30 seconds. Table IV shows how the latest processing start time changes for $\mu = 2.15$ elevation scans processed per second as we increase $\lambda$ for two target probabilities. Figure 3 plots the work completion probability $\xi$ versus processing start time. Again, for the values shown, there is little difference between streaming processing ($t_0 = 0$) and batch processing as long as the delay is not more than half the 30-second processing window.

*3) Dimensioning:* The above model can also determine the processing power needed to ensure that the work completion probability is above some threshold for a known arrival rate $\lambda$ and batch processing start time $\tau_0$. Given such values, we can use the model to find a value of the service completion rate (i.e., processing power) $\mu = s\sigma$ that allows the system to complete its work that arrived during the 30 second interval with high probability. Figure 4 plots the value of $\mu$ versus

TABLE IV

VALUES OF $\tau_0^{max}$ FOR TWO TARGET PROBABILITIES IN THE SCENARIO WHERE $\mu$ IS HELD CONSTANT AT 2.15 FEATURES PER SECOND AS DERIVED FROM THE EMPIRICAL NEXRAD DATA. $\lambda$ IS INCREASED. $\tau_0^{max}$ IS THE LATEST TIME THAT PROCESSING CAN START WITHIN THE PROCESSING WINDOW $[0, \tau]$ SUCH THAT THE WORK COMPLETION PROBABILITY IS AT LEAST THE TARGET.

| $\tau = 30$ sec | | | Target Prob. = 0.9 | Target Prob. = 0.95 |
|---|---|---|---|---|
| $\lambda$ | $\mu$ | $\frac{\lambda}{\mu}$ | $\tau_0^{max}$ (seconds) | $\tau_0^{max}$ (seconds) |
| 0.1 | 2.15 | 0.047 | 27 | 27 |
| 0.2 | 2.15 | 0.093 | 25 | 24 |
| 0.3 | 2.15 | 0.140 | 23 | 22 |
| 0.4 | 2.15 | 0.186 | 21 | 19 |
| 0.5 | 2.15 | 0.233 | 19 | |
| 0.6 | 2.15 | 0.279 | 16 | |

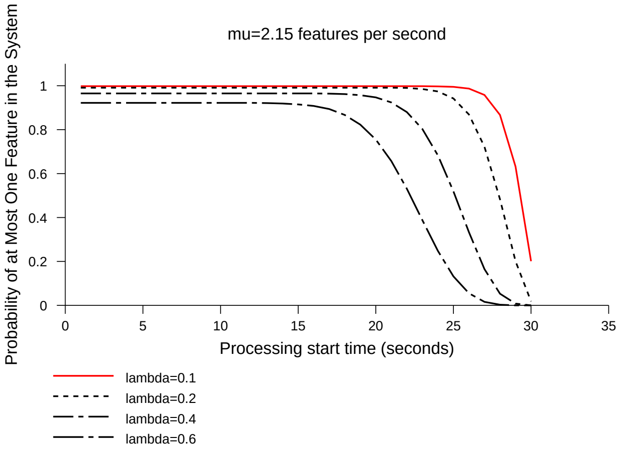

Fig. 3. Probability of at most one feature in the system at the end of the processing window vs. processing start time for four values of $\lambda$ with $\mu$ held constant at 2.15 elevations processed per second.

$\tau_0$ for $\lambda = 0.1$ and $\lambda = 0.3$ features per second given a target probability of 0.95. We again assume a processing window of 30 seconds and 10 slots per second.

Figure 4 demonstrates that, for the values of $\lambda$ shown, delaying the processing start time by as much as half of the processing window to do batch processing does not cause a significant increase in the computational resources required to keep the work completion probability above the target probability. For example, for $\lambda = 0.3$ feature arrivals per second, a service completion rate $\mu$ of 1.5 features processed per second is required when $\tau_0 = 0$ (streaming mode) to keep the work completion probability above the target. As $\tau_0$ increases for batch processing, the $\mu$ required remains constant until $\tau_0 = 18$ when $\mu$ must increase to 2 features per second in order to maintain the target work completion probability. As we increase $\tau_0$ further, $\mu$ increases rapidly, approaching $\infty$ as $\tau_0$ approaches $\tau$.

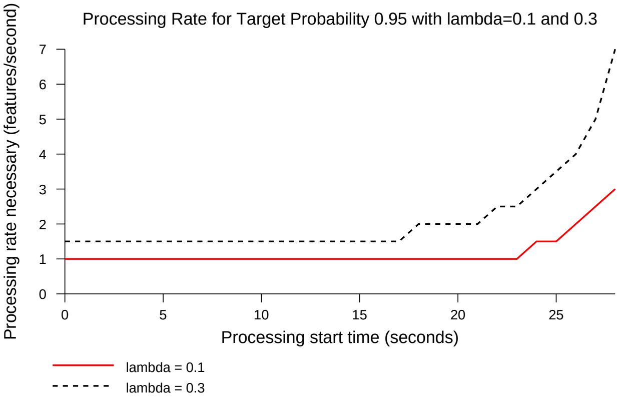

Fig. 4. Feature processing rate necessary for $\lambda = 0.1$ and 0.3 features arriving per second with a target probability of 0.95.

As before, we see the effect that $\lambda$ and $\mu$ have on the transient behavior. The graph is flat and then curves upward approaching $\infty$ as $\tau_0$ approaches $\tau$, but the upward trend begins earlier for the higher value of $\lambda$.

## III. CORRELATED MODELS

In this section, we explore two models in which meteorological feature arrivals are correlated. Both are extensions of the independent arrival model, but no longer assume that arrivals are independent since in the case of weather phenomena, features arrive in clusters, with the number of features depending on the elevation at which the radars are scanning. We model meteorological feature correlation with a two-state Markov process. In the first correlated model below this source process controls the parameter of the geometric arrival distribution. When the process is in state 0, the distribution has parameter $p_0$; when it is in state 1, the distribution has parameter $p_1$. This allows us to model the fact that features arrive in bursts by setting one parameter high and the other low. In the second correlated model, we assume a constant rate of feature arrivals, and the two-state source Markov process controls the probability that service will complete within the slot. In both cases, by viewing the states of the source process as representing either low or high elevation scans, we can model different radar scan strategies with different processing times.

The two correlated models are really two sides of the same coin and can both be used to model the same

TABLE V

DEFINITIONS FOR THE CORRELATED ARRIVAL MODEL.

| | |
|---|---|
| $q_{kl}$ | Trans. probs of source Markov process. |
| $P_{ik,jl}$ | Trans. probabilities of the 2-dim chain |
| $a_k(n) = (1 - p_k)p_k^n$ | Arrival dist. for source process state $k$. |
| $\pi_0 = \frac{q_{10}}{(1+q_{10}-q_{00})}$ | Limiting prob. of source process state 0. |
| $\pi_1 = 1 - \pi_0$ | Limiting prob. of source process state 1. |
| $\lambda = \pi_0\lambda_0 + \pi_1\lambda_1$ | Average feature arrival rate. |
| $\lambda_k = \frac{sp_k}{1-p_k}$ | Mean arrival rate for source process state $k$. |
| $\beta_k(i)$ | Pr. $i$ arrivals by $t_0$, source process state $k$. |
| $\delta(i)$ | Prob. $i$ accumulated features by slot $t_0$. |

phenomenon. The difference is a matter of perspective. Similar to the description in Section II-B.1, the first correlated model applies to a situation in which feature arrivals are fine-grained and each of the individual features is processed in turn. The second applies to situations in which a single arrival brings several features to be processed. This is similar to our empirical data for the processing times of NEXRAD elevation scans, as discussed in the previous section. Each elevation scan contains a variable number of weather features on which the processing time depends.

### A. Correlated Arrivals

In this subsection we develop and analyze the model in which fine-grained features arrive during a slot at rates determined by a two-state source Markov process.

*1) Definitions:* We use a 2-dimensional Markov chain to analyze this model. The state is defined by the set $\{i, k\}$ where $i = 0, 1, 2, \ldots N$ is the number of features in the system, and $k = 0, 1$ is the state of source arrival process. Table V summarizes the definitions where they differ from the first model. $q_{kl}$ for $k, l = 0, 1$ represents the transition probabilities of the source process. $P_{ik,jl}$ for $i, j = 0, 1, 2, \ldots N$ and $k, l = 0, 1$ represent the transition probabilities of the 2-dimensional chain.

Features arrive with distribution $a_k(n) = (1 - p_k)p_k^n$ when the source process is in state $k$. $\lambda_k$ is the mean feature arrival rate for the source process state $k$, and $\lambda = \pi_0\lambda_0 + \pi_1\lambda_1$ is the average feature arrival rate where $\pi_0$ and $\pi_1$ are the limiting probabilities of the source process. Finally, let $\delta(i)$ be the probability that $i$ features have arrived by slot $t_0$. This is analogous to $\beta(i)$ in the first model. $\beta_k(i)$ is discussed below.

Given the above information, we define the one-step transition probabilities to be

$$P_{0k,il} = q_{kl}a_k(i) \qquad 0 \leq i < N \quad k, l = 0, 1 \tag{8}$$

$$P_{ik,i-1l} = q_{kl}\sigma a_k(0) \qquad 0 < i \leq N \quad k, l = 0, 1 \tag{9}$$

$$P_{ik,i+jl} = q_{kl}[\sigma a_k(j+1) + ua_k(j)] \qquad 0 < i \leq i+j < N \qquad k, l = 0, 1 \tag{10}$$

$$P_{ik,Nl} = q_{kl} - \sum_{j=0}^{N-1} P_{ik,jl} \quad 0 \leq i \leq N \quad k, l = 0, 1 \tag{11}$$

$$P_{ik,jl} = 0 \qquad\qquad otherwise. \tag{12}$$

These equations are the analogues of Equations 2 through 6 of the independent arrival model and follow from similar reasoning. The 2-dimensional chain will move to another state depending on the feature arrival distribution for the source arrival process state $k$, the service completion probability, and the source process's transition probabilities.

The one-step transition probabilities define the one-step transition probability matrix. Whereas in the independent arrival model we had an $(N+1) \times (N+1)$ matrix, here the matrix is $2(N+1) \times 2(N+1)$. We index the matrix from 0 to $2N+1$ where an index $r$ gives the state of the two-dimensional Markov chain as $\{r \mod N+1, \lfloor \frac{r}{N+1} \rfloor\}$. In other words $\mathbf{P}(r, s)$ represents the probability of moving from a state with $r \mod N+1$ features when the source process is in state $\lfloor \frac{r}{N+1} \rfloor$ to a state with $s \mod N+1$ features when the source process is in state $\lfloor \frac{s}{N+1} \rfloor$.

Again, our goal is to compute the work completion probability $\xi$, the probability that at most one unprocessed feature will remain in the system at time $\tau$ (slot $t$) given a processing start time of $\tau_0$ (slot $t_0$). We first compute the $m$-step transition probability matrix $\mathbf{P}^m$ where $m = t - t_0$. From $\mathbf{P}^m$, we can compute the probability $Q(i, j)$ of going from a state with $i$ features in the system at slot $t_0$ to a state with $j$ features in the system at slot $t$ by conditioning on the state of the source Markov process. We assume that the probability that the source process is initially in state $k$ at $t_0$ is equal to its limiting probability $\pi_k$. We, therefore, define $Q(i, j)$ to be

$$\begin{aligned} Q(i,j) = \pi_0\big[&\mathbf{P}^m(i,j) + \mathbf{P}^m(i, N+1+j)\big] \\ &+ \pi_1\big[\mathbf{P}^m(i+N+1, j) \\ &+ \mathbf{P}^m(i+N+1, N+1+j)\big] \end{aligned} \tag{13}$$

where $\pi_0 = \frac{q_{10}}{(1+q_{10}-q_{00})}$ and $\pi_1 = 1 - \pi_0$ which can be easily verified since for a two state chain $\pi_j = \sum_{i=0}^{1} \pi_i q_{ij}$ for $j = 0, 1$ and $\pi_0 + \pi_1 = 1$ [12].

Given $Q(i, j)$, we compute the work completion probability by conditioning on the number of features in the system at slot $t_0$. To do this, we need the distribution $\delta(i)$ of the number of features in the system at slot $t_0$ which depends on the initial state of the source process. Again, assuming that the probability that the source process is in initial state $k$ is equal to $\pi_k$, we define $\delta(i)$ to be

$$\delta(i) = \begin{cases} \pi_0\beta_0(i) + \pi_1\beta_1(i) & i < N \\ 1 - \sum_{i=0}^{N-1}(\pi_0\beta_0(i) + \pi_1\beta_1(i)) & i = N \end{cases} \tag{14}$$

where $\beta_k(i)$ is the probability that $i$ features have arrived by slot $t_0$ given that the source process is in state $k$. $\beta_k(i)$ has the negative binomial distribution (as in the case for $\beta(i)$ in Equation 7) with parameter $p_k$.

Finally, the work completion probability $\xi$ is

$$\xi = \xi_0(m) + \xi_1(m) \tag{15}$$

where

$$\xi_0(m) = \sum_{i=0}^{N-1} \delta(i)Q(i, 0) + (1 - \sum_{i=0}^{N-1} \delta(i))Q(N, 0)$$
$$\xi_1(m) = \sum_{i=0}^{N-1} \delta(i)Q(i, 1) + (1 - \sum_{i=0}^{N-1} \delta(i))Q(N, 1)$$

*2) Numerical Calculations:* We analyze the correlated arrival model as we did the independent arrival model. To examine the relationship between the feature arrival rate and the feature service rate, we must consider not only the average arrival rate, but also the burstiness of the arrivals. In this subsection we examine four scenarios. In two of the four, the source arrival process spends 80% of the time in state zero with transition probabilities $q_{00} = 0.8$, $q_{01} = 0.2$, $q_{11} = 0.2$, and $q_{10} = 0.8$. In the other two scenarios the source process spends 20% of the time in state zero with $q_{00} = 0.2$, $q_{01} = 0.8$, $q_{11} = 0.8$, and $q_{10} = 0.2$.

Table VI compares $\tau_0^{max}$ when the arrival process spends 80% of the time in state zero to the case where it spends 20% of the time in state 0. In both cases the target probability is 0.9. $\mu$ is held constant at 1 feature processed per second, and $\frac{\lambda}{\mu}$ varies. $\lambda_0$, which is always less than $\lambda_1$, is fixed at 0.05 features per second. $\lambda_1$ varies accordingly to maintain the value of $\frac{\lambda}{\mu}$ given $\lambda_0$. Thus, the expected system utilization over the processing window is the same in both scenarios.

TABLE VI

Comparison of $\tau_0^{max}$ when the source process spends 80% of the time in state 0 to $\tau_0^{max}$ when it spends 20% of the time in state 0 for several values of $\lambda$. $\mu$ is held constant at 1 feature processed per second and $\lambda_0$ at 0.05 features per second. The target probability is 0.9

| | 80% State 0 | 20% State 0 |
|---|---|---|
| $\frac{\lambda}{\mu}$ | $\tau_0^{max}$ | $\tau_0^{max}$ |
| 0.1 | 23 | 24 |
| 0.125 | 20 | 23 |
| 0.15 | 18 | 22 |
| 0.175 | 16 | 20 |
| 0.2 | 14 | 19 |
| 0.225 | 12 | 17 |
| 0.25 | 10 | 16 |
| 0.275 | 7 | 14 |
| 0.3 | | 11 |

TABLE VII

Comparison of $\tau_0^{max}$ when the source process spends 80% of the time in state 0 to $\tau_0^{max}$ when it spends 20% of the time in state 0 for several values of $\lambda$. $\mu$ is held constant at 4 feature processed per second and $\lambda_0$ at 0.2 features per second. The target probability is 0.9

| | 80 % State 0 | 20 % State 0 |
|---|---|---|
| $\frac{\lambda}{\mu}$ | $\tau_0^{max}$ | $\tau_0^{max}$ |
| 0.1 | 24 | 25 |
| 0.125 | 24 | 24 |
| 0.15 | 24 | 23 |
| 0.175 | 23 | 22 |
| 0.2 | 23 | 22 |
| 0.225 | 22 | 21 |
| 0.25 | 19 | 20 |
| 0.275 | | 20 |
| 0.3 | | 18 |

Table VII shows the same information as Table VI but for $\mu = 4$ features processed per second and $\lambda_0 = 0.2$ features per second. Figures 5 and 6 plot the work completion probability $\xi$ versus processing start time for the examples in Table VI. Figures 7 and 8 do the same for the examples in Table VII.

The above tables and figures show that the system can tolerate later processing start times for batch processing when the source process spends more time in the state with the higher feature arrival rate. This follows our intuition since in order to maintain the average utilization with fixed $\lambda_0$, $\lambda_1$ must be higher when the system spends

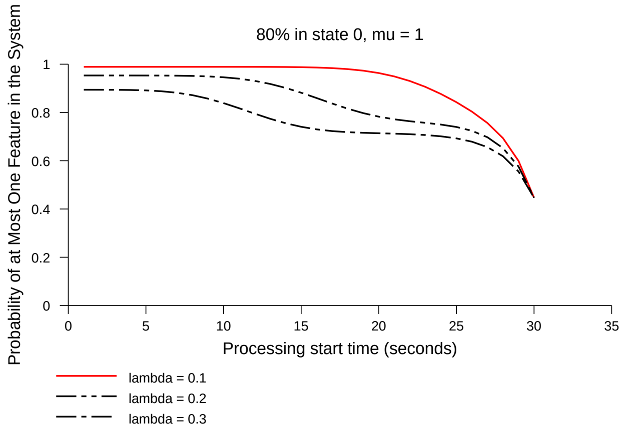

Fig. 5. Probability of at most one feature in the system at the end of the processing window vs. processing start time for three values of $\lambda$ for the example in Table VI with $\mu$ held at 1 feature processed per second and the source process in state 0 80% of the time.

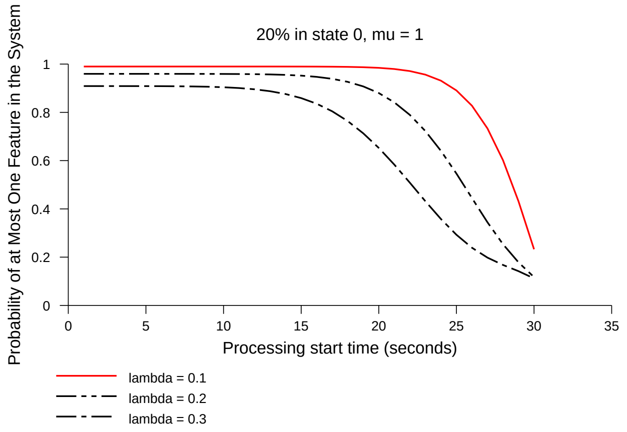

Fig. 6. Probability of at most one feature in the system at the end of the processing window vs. processing start time for three values of $\lambda$ for the example in Table VI with $\mu$ held at 1 feature processed per second and the source process in state 0 20% of the time.

most of its time in state 0. The figures also show that $\xi$ drops much more gradually when the source process spends more time in the state with the lower feature arrival rates. The reason is that when the source process spends more time in the state with lower feature arrivals, it is less likely that a burst of features will arrive near the end of the processing window. These differences emphasize the importance of being able to characterize the arrival process accurately.

As with the independent arrival model, we can use the correlated arrival model to determine the processing power needed to maintain $\xi$ above a threshold, given

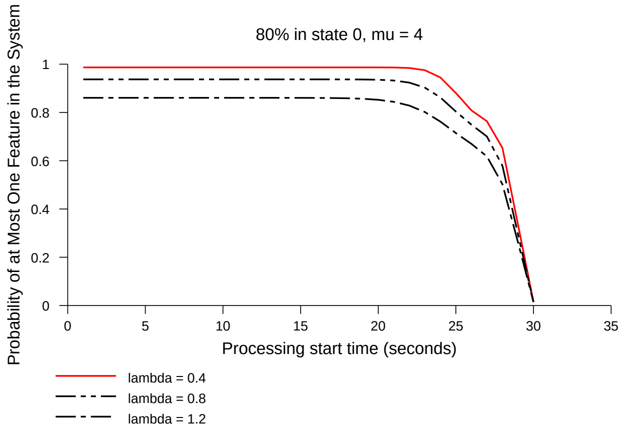

Fig. 7. Probability of at most one feature in the system at the end of the processing window vs. processing start time for three values of $\lambda$ for the example in Table VII with $\mu$ held at 4 features processed per second and the source process in state 0 80% of the time.

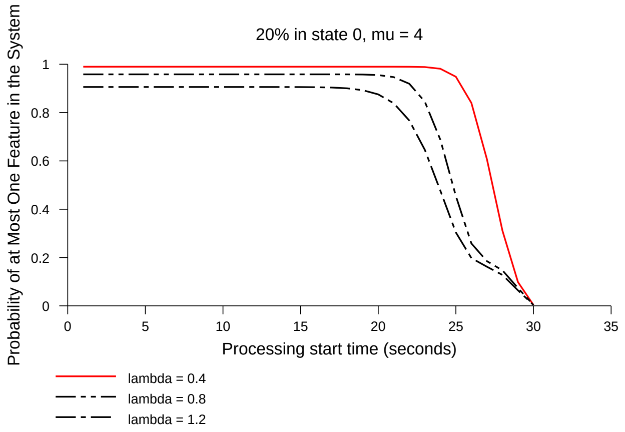

Fig. 8. Probability of at most one feature in the system at the end of the processing window vs. processing start time for three values of $\lambda$ for the example in Table VII with $\mu$ held at 4 features processed per second and the source process in state 0 20% of the time.

the known characteristics of the system. Figure 9 plots for target probabilities 0.9 and 0.95 the value of $\mu$ versus processing start time in the case when the source process spends 80% of the time in state 0, given an average feature arrival rate of $\lambda = 0.4$ features per second. $\lambda_0 = 0.2$ and $\lambda_1 = 1.2$ features per second. Again, we see that the processing power for batch processing is the same as that for streaming as long as batch processing begins within the first half of the processing window.

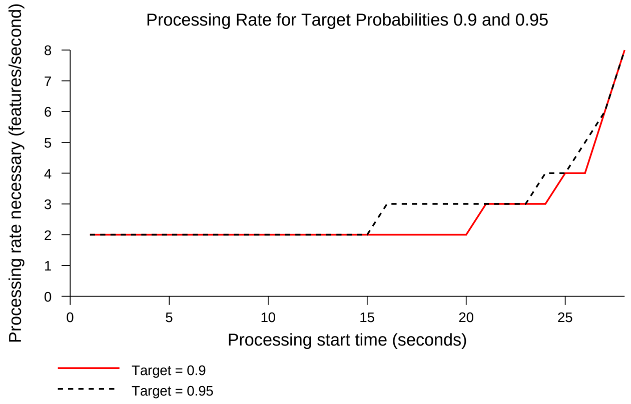

Fig. 9. Feature processing rate necessary for target probabilities of 0.9 and 0.95 given that of $\lambda = 0.4$, $\lambda_0 = 0.2$, $\lambda_1 = 1.2$, $q_{00} = 0.8$, $q_{01} = 0.2$, $q_{11} = 0.2$, and $q_{10} = 0.8$.

### B. *Correlated Processing Times*

We next consider the model in which groups of features arrive within a time slot with an identical probability from slot to slot, but the processing rate of those groups is dependent on a two-state source arrival process. We again utilize empirical runtime data for the llsd and w2merger NEXRAD algorithms to evaluate this model.

*1) Definitions:* As in Subsection III-A, a two-dimensional Markov chain is the basis of this model. Its state and transition probabilities as well as those of the source process are as defined in Subsection III-A. Groups of features (elevation scans) arrive within a slot according to a geometric distribution with parameter $p$ as in Equation 1 of Subsection II-A. The arrival rate $\lambda$ is also as defined in Section II-A and is the arrival rate of elevation scans.

Let $\sigma_k$ be that probability that service on an elevation scan will complete in the current slot given that the source process is in state $k$. Let $u_k = 1 - \sigma_k$ be the probability that service on an elevation scan will require at least one more slot given that the source process is in state $k$. The remaining definitions are the same as those in Tables I and V.

Given the above, the one-step transition probabilities are the same as Equations 8 through 12 except that $\sigma$ and $u$ are replaced by $\sigma_k$ and $u_k$ respectively and $a_k(n)$ is replaced by $a(n)$. To compute the work completion probability, $\xi$, we compute the $m$-step transition probability matrix and condition on the number of elevation scans accumulated in the system at slot $t_0$. Because the arrival distribution of scans is independent of the source process, this distribution is simply $\beta(i)$ from the model in Section II defined in Equation 7. The expression for $Q(i, j)$, the probability of going from an initial state that has $i$ scans in the system at slot $t_0$ to a state that has $j$ features in the system at slot $t$, is the same as that in Equation 13. The work completion probability, $\xi$, is, therefore:

$$\xi = \xi_0(m) + \xi_1(m) \tag{16}$$

where

$$\xi_0(m) = \sum_{i=0}^{N-1} \beta(i)Q(i,0) + (1 - \sum_{i=0}^{N-1} \beta(i))Q(N,0)$$

$$\xi_1(m) = \sum_{i=0}^{N-1} \beta(i)Q(i,1) + (1 - \sum_{i=0}^{N-1} \beta(i))Q(N,1)$$

*2) Numerical Calculations:* As noted earlier, our empirical data indicates that scans of low elevations take more time on average to process than scans of higher elevations. Thus, for this model state 0 of the source Markov process represents scanning being done on low elevations, and state 1 represents scanning at higher elevations. By manipulating the transition probabilities of the source process, we can model different scan strategies as discussed below.

The runtime information we have is composed of data from scans at thirty-one different elevations. We consider a low elevation to be in one of the first fifteen elevations and a high scan to be at one of the second sixteen. The mean combined runtime for the llsd and w2merger algorithms on low scans is 0.551 seconds for a processing rate of $\mu_{low} = 1.814$ scans per second. The mean combined runtime for high scans is 0.376 seconds for a processing rate of $\mu_{high} = 2.662$ scans per second.

As with the other models, we explore the relationship of $\lambda$ and $\mu$ to the maximum time at which batch processing can start while still meeting a target work completion probability. We also explore how different scan strategies affect this measure. Table VIII compares $\tau_0^{max}$ for three sets of transition probabilities for the source process, which model three scan strategies. In the first, the source process spends equal proportions of time in state 0 and state1 with $q_{00} = q_{01} = q_{10} = q_{11} = 0.5$. In the second, the source process spends 80% of the time in state 0 (low elevation scans), and in the third, the source process spends 90% of the time in state 0. In all three cases, $\mu_{low} = 1.814$ and $\mu_{high} = 2.662$, and the target probability is 0.9. As in the previous models, we assume geometric scan arrivals due to variability

TABLE VIII

VALUES OF $\tau_0^{max}$ FOR THREE DIFFERENT SCAN STRATEGIES. IN ALL CASES $\mu_{low}$ IS HELD CONSTANT AT 1.814 ELEVATIONS SCANS PROCESSED PER SECOND, $\mu_{high}$ IS HELD AT 2.662 SCANS PER SECOND, AND $\lambda$ IS INCREASED. IN THE FIRST STRATEGY, LOW AND HIGH SCANS ARRIVE IN EQUAL PROPORTIONS. IN THE SECOND, 80% OF THE SCANS ARE LOW, AND IN THE THIRD, 90% ARE LOW. THE TARGET PROBABILITY WAS 0.9.

|  | 50/50 | 80/20 | 90/10 |
|---|---|---|---|
| $\lambda$ | $\tau_0^{max}$ (seconds) | $\tau_0^{max}$ | $\tau_0^{max}$ |
| 0.1 | 27 | 27 | 27 |
| 0.2 | 25 | 25 | 25 |
| 0.3 | 23 | 23 | 22 |
| 0.4 | 22 | 20 | 20 |
| 0.5 | 19 | 18 | 17 |
| 0.6 | 17 | 14 | 8 |

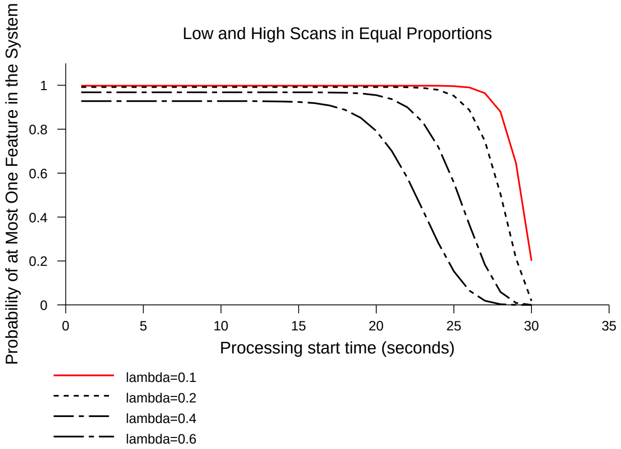

Fig. 10. Probability of at most one feature in the system at the end of the processing window vs. processing start time for four values of $\lambda$ where $\mu_{low}$ is held constant at 1.814 elevation scans processed per second and $\mu_{high}$ is held at 2.662. The source process spends equal proportions of time in each state.

in network delay and radar dwell times and start with $\lambda = 0.1$ elevation scans per second representing data from a single radar. We increase $\lambda$ to model greater numbers of radars.

Figures 10, 11, and 12 plot $\xi$ versus processing start time for each of the strategies above for several values of $\lambda$. Figure 13 compares the results from each of the strategies when $\lambda = 0.6$ scans arriving per second. These results demonstrate that the particular scan strategy impacts the latest time at which batch processing can begin and still maintain the same work completion probability as streaming processing. The figures show that for a given value of $\lambda$ the knee of the curve shifts to the left as the scan strategy spends more time in the state representing low elevations. For example, for $\lambda = 0.6$ with a batch processing start time of 20 seconds into the 30-second window, $\xi$ equals 0.79, 0.67, and 0.63 for the three strategies. Thus, as we increase $\lambda$ and the amount of time the radars scan at low elevations, the earlier we must start batch processing. It is, therefore, important to be able to characterize the expected range of scan strategies that the CASA implementation will use in order to model it effectively.

## IV. CONCLUSIONS

In this paper, we have presented several discrete-time queuing models of the feature detection subsystem of the Meteorological Command and Control (MC&C) component of the proposed architecture for the Collaborative Adaptive Sensing of the Atmosphere (CASA) hazardous weather detection system. The initial model that assumes independent arrivals is useful in performing simple average case analyses. It shows the effects that the arrival and processing rates have on the transient behavior of the system and helps dimension the system given known characteristics.

The other models assume correlated arrivals and show that different scan strategies affect when batch processing can start and still maintain the same work completion probability as streaming processing. It is thus important to characterize the data and scan strategies the system will encounter.

These models and the analyses we have done represent initial work. As we obtain data that pertains more closely to the CASA implementation, we can model the system more accurately. We could also extend the correlated models to include more states in the source process in order to model more elevations and a greater variety of scan strategies. We are also working on a new model that incorporates an advantage of batch processing due to the better data compression that it allows. This model will help in tradeoff decisions about when to start processing.

## ACKNOWLEDGMENT

The authors would like to thank Eric Lyons for his help in understanding the weather detection algorithms and for generating the runtime data for the llsd and w2merger algorithms. We would also like to thank David Westbrook for his help in understanding the proposed CASA architecture.

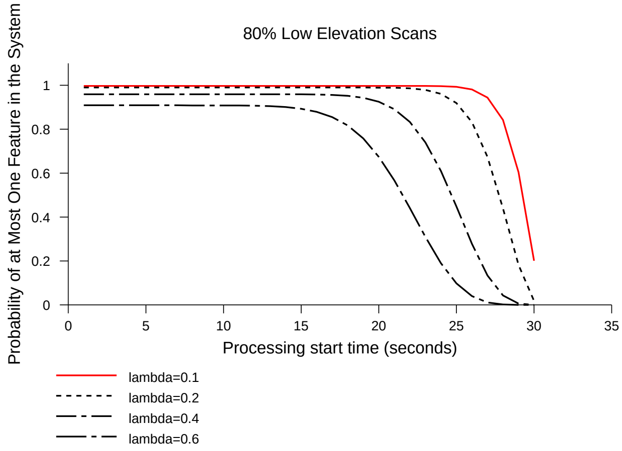

Fig. 11. Probability of at most one feature in the system at the end of the processing window vs. processing start time for four values of $\lambda$ where $\mu_{low}$ is held constant at 1.814 elevation scans processed per second and $\mu_{high}$ is held at 2.662. The source process spends 80% of time in state 0 (low elevations).

## References

[1] M. Zink, J. Brotzge, V. Chandresakar, K. Droegemeier, J. Kurose, D. McLaughlin, B. Philips, M. Preston, P. Rees, S. Sekelsky, and D. Westbrook, "Distributed collaborative adaptive sensing for hazardous weather detection, tracking, and predicting," *Kluwer Lecture Notes in Computer Science (Dynamic Data Driven Application Systems (DDDAS))*, 2004.

[2] M. Zink, D. Westbrook, S. Abdallah, B. Horling, V. Lakamraju, E. Lyons, V. Manfredi, J. Kurose, and K. Hondl, "Meteorological command and control: An end-to-end architecture for a hazardous weather detection sensor network," February 2005.

[3] "National oceanic and atmospheric administration, radar resources," http://www.ncdc.noaa.gov/oa/radar/radarresources.html.

[4] H. Bruneel and B. G. Kim, *Discrete-Time Models for Communication Systems Including ATM*. Boston, Massachusetts: Kluwer Academic Publishers, 1993, ch. 1.

[5] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*. New York: John Wiley & Sons, 1974, ch. 2.

[6] W.-P. Wang, D. Tipper, and S. Banerjee, "A simple approximation for modeling nonstationary queues," in *Proceedings of IEEE INFOCOM '96*, March 1996.

[7] A.M.K.Tarabia, "Transient analysis of m/m/1/n queue - an alternative approach," *Tamkang Journal of Science and Engineering*, vol. 3, no. 4, pp. 263–266, 2000.

[8] A. Chydzinski, "On transient analysis of queues with poisson input stream," in *Proceedings of International Workshop in Applied Probability*, March 2004.

[9] B. Groskinsky, D. Medhi, and D. Tipper, "An investigation of adaptive capacity control schemes in a dynamic traffic environment," *IEICE Transactions on Communication*, vol. E84-B, no. 2, pp. 263–274, February 2001.

[10] M. W. McKinnon and D. Tipper, "A comparison of transient period detection algorithms applied to markovian systems," in *Proceedings of the 1995 Summer Simulation Conference*, July 1995.

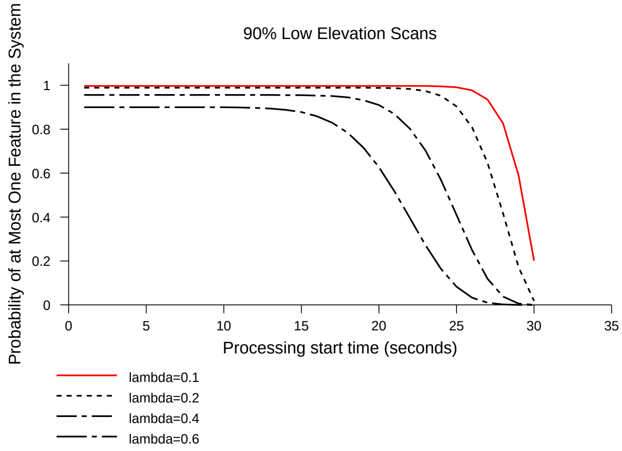

Fig. 12. Probability of at most one feature in the system at the end of the processing window vs. processing start time for four values of $\lambda$ where $\mu_{low}$ is held constant at 1.814 elevation scans processed per second and $\mu_{high}$ is held at 2.662. The source process spends 90% of time in state 0 (low elevations).

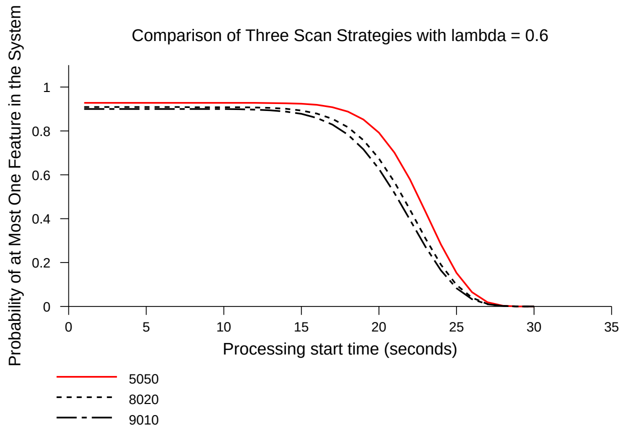

Fig. 13. Comparison of the probability of at most one feature in the system at the end of the processing window vs. processing start time for each of the three scan strategies when $\lambda = 0.6$.

[11] H. Nassar and Y. Fouad, "Analysis of two-class discrete packet queues with homogenous arrivals and prioritized service," *Communications in Information and Systems*, vol. 3, no. 2, pp. 101–118, 2003.

[12] S. M. Ross, *Introduction to Probability Models*, 8th ed. Boston, Massachusetts: Academic Press, 2003, ch. 4.

[13] K. Hondl, "Capabilities and components of the warning decision support system - integrated information (wdss-ii)," in *Proceedings of the American Meteorological Society Annual Meeting*, January 2003.