Obstacles to Predictive Logistics

A method for sustainment in a communication-challenged environment

by LtCol Roy L. Miner

>LtCol Miner is a Logistics Officer currently serving with Operations Analysis Directorate within Combat Development and Integration. He recently completed an assignment at Marine Corps Warfighting Lab's Wargaming Division where he served as the government lead for logistics-related wargames. A version of this article originally appeared in the December 2020 issue of the **Logistics Innovation Office Newsletter**, *a monthly publication discussing innovation efforts focused on enhancing the Marine Corps logistics community.*

One of the problems confronting logisticians today is the execution of predictive logistics and sustainment in a communications-denied environment. How does the logistician ensure supported units receive sufficient sustainment to conduct persistent operations when the supported unit cannot communicate its requirements? Would it be a push system for sustainment? A pull system? Or some type of hybrid system? Additionally, how does the logistician provide sustainment while managing the physical signatures and patterns of life resupply actions of the supporting and supported units to frustrate adversary sensing and targeting activities? Does the future operating environment diminish the relevancy of employing scheduled resupply run concepts? This article discusses a hybrid push-pull model that bases the provision of sustainment on probability using pre-established cache site locations known to both the supported and supporting units in a communication-denied or degraded environment, effectively demonstrating enhancements to sustainment success despite operating in an environment of uncertainty.

The following scenario illustrates how to construct and analyze a simple hybrid push-pull model. First, assume the supported and supporting units are unable to communicate with each other. Second, assume there are three pre-positioned cache sites full of the required sustainment materiel for a supported unit in a hypothetical area of operation. Third, assume that the supported unit can only make one trip to any one of the three cache sites in a given time interval to retrieve the sustainment material. Fourth, assume that the supporting unit can only make one trip to any one of the cache sites in the same given time interval for the purpose of replenishing the cache site. If the cache is empty, the supporting unit replenishes the cache. If it is full, the supporting unit does not replenish the cache. Fifth, for simplicity, assume the supported unit attempts to retrieve the sustainment material before the supporting unit attempts to replenish the cache site. The completion of one retrieval and one replenishment attempt is considered one turn. Figure 1 depicts an example of one turn. Here a yellow-colored cache indicates a cache full of sustainment material in a hypothetical operating environment. The full turn sequence displayed in Figure 1 is initial sustainment conditions, retrieval attempt by the supported unit, replenishment attempt by the supporting unit, and resulting sustainment conditions to start the next turn.

In the Figure 1 example, the supported unit randomly chooses cache

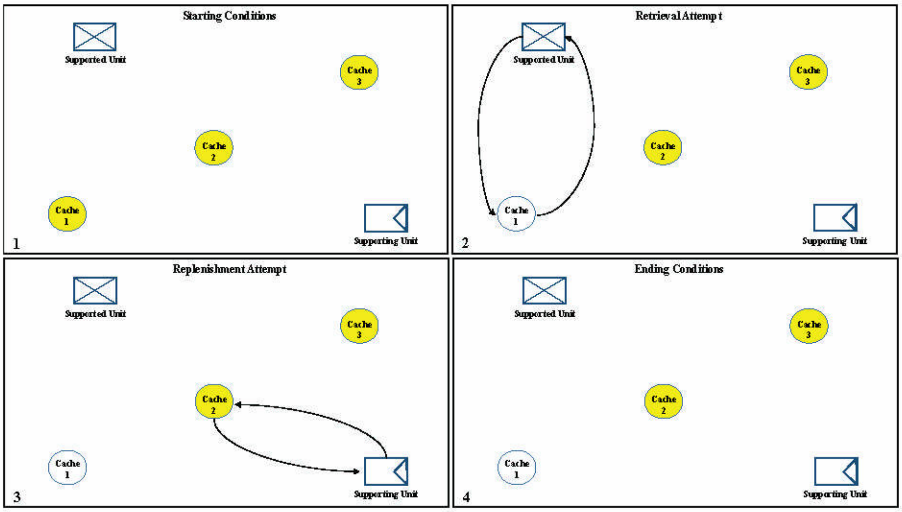

Figure 1. Sustainment Turn example. *(Figure provided by author.)*

1 to attempt to retrieve sustainment material and is successful. The supporting unit randomly chooses cache 2 to replenish; however, it is already full of sustainment. This leaves two caches full of material before the supported unit makes its next attempt to retrieve sustainment material.

What will the success rate of the supported unit be in retrieving sustainment material from the cache if it simply guesses what cache to retrieve the required sustainment from and the supporting unit also guesses what cache to replenish? To answer this question, we can employ a technique known to operations research analysts as a Markov Chain. A Markov Chain is a stochastic model that represents a sequence of probable events where the probability of the next event occurring only depends on the current conditions or state (a *chain* of events). Events that occur before the current conditions will have no effect or influence on the probability of what will occur next in the sequence of events. Table 1 depicts the scenario described above after one turn represented as a Markov Chain.

	1	2	
1	55.6%	44.4%	0.0%
2	22.2%	66.7%	11.1%
3	0.0%	66.7%	33.3%

Table 1. Cache random retrieval and replenish (after one turn).

The numbers to the left of the table indicate the possible states before the execution actions of the supported and supporting units in a given turn. That is the possible number of pre-positioned caches that are filled with the required sustainment material before the supported unit randomly chooses a cache to attempt to retrieve sustainment material. Using the example in Figure 1, the appropriate state to reference in Table 1 based on the starting conditions of having all three cache sites full of sustainment will be the third row.

The numbers along the top of Table 1 indicate the possible states after the execution actions of the supported and supporting units at the conclusion of a given turn. That is the number of prepositioned caches holding sustainment material after the supporting unit attempts to replenish one of the three caches. Using the example in Figure 1, the appropriate state to reference in Table 1 will be the second column indicating two caches remain full at the end of the turn.

The percentages within the table cells represent the transition probabilities of the system during a single time interval encompassing one resupply attempt and one retrieval attempt. Each percentage represents the probability that the system will transition from containing the number of filled caches given on the left of the table to the number of filled caches given at the top of the table after the completed actions of one turn.

Thus, Table 1 demonstrates the possible future outcomes of caches filled with sustainment material as indicated by the column headers. The supported unit chose to retrieve sustainment material from cache 1 and left cache 1 empty. When the supporting unit randomly chooses to replenish one of the three cache locations there is a 66.7 percent chance they will choose a location already containing sustainment (cache 2 or cache 3), which results in only two caches being full the next time the supported unit attempts to retrieve sustainment (highlighted in red in Table 1). There is a 33.3 percent chance the supporting unit chooses to replenish the location from which the supported unit retrieved sustainment material (cache 1). This would leave three caches full the next opportunity the supported unit attempts to retrieve sustainment material (highlighted in blue in Table 1). Additionally, given our assumptions outlined at the beginning, no matter what cache the supporting unit chooses to replenish, there is a 0.0 percent chance there will only be one cache filled with sustainment during the next opportunity the supported unit attempts to retrieve sustainment material (highlighted in green in Table 1).

We can use this same Markov chain to determine the probabilities of how many caches will have sustainment after two attempts by the respective units simply by multiplying the matrix above with itself one time. Table 2 below shows the results of that matrix multiplication and demonstrates the probabilities for the number of caches we can expect to be filled with sustainment two time intervals from now (after two turns) given the supported and supporting units are randomly attempting to retrieve and replenish the caches. For example, given we started with three caches full in our example above, we would reference row three of Table 2 and observe there is a 14.8 percent chance there will be only one cache with sustainment after the units make two attempts at retrieval and replenishment (represented by the third row's first column in Table 2). There is a 66.7 percent chance there will be two caches filled with sustainment (represented by the third row's second column in Table 2) and an 18.5 percent chance there will be three caches filled with sustainment

What will the success rate of the supported unit be in retrieving sustainment material ... if it simply guesses what cache to retrieve the required sustainment from ... ?

	1	2	
1	40.7%	54.3%	4.9%
2	27.2%	61.7%	11.1%
3	14.8%	66.7%	18.5%

Table 2. Cache random retrieval and replenish (after two turns).

(represented by the third row's third column in Table 2) after the units make two attempts at retrieval and replenishment.

As mentioned, the Markov Chain is a memoryless model. Previous events have no bearing on the probabilities of future outcomes. Going back to the example of Figure 1, if we wanted to know the probabilities for full caches at the beginning of the next turn (that is the beginning of the second turn) after our first turn yielded the condition of only having two caches full, we simply look at the second row of Table 1 to determine the probabilities of the resulting caches for the next immediate step.

If we were to conduct this matrix multiplication over and over again, we can find the probability of the future number of expected filled caches after any number of attempts to retrieve and resupply sustainment by the supported and supporting units. In fact, after multiple matrix multiplications, we find the transition probabilities within the matrix converge to steady states. This means at any given current state for the number of caches filled, we can calculate the expected probability that there will be one, two, or three caches filled after a number of attempts at retrieval and replenishment by the supported and supporting units. Table 3 shows the steady states for our current cache sustainment system.

	1	2	
1	30.0%	60.0%	10.0%
2	30.0%	60.0%	10.0%
3	30.0%	60.0%	10.0%

Table 3. Cache random retrieval and replenish steady-states.

In this instance, where the units are randomly selecting which cache to retrieve or replenish, the Markov Chain steady-state probabilities converge within +/- two percent of the steady-state probabilities shown in Table 3 after four iterations of retrieval and resupply attempts by the supported and supporting units. This means after four turns or more, no matter how many caches are filled (the initial state), we can expect only one cache will be full of sustainment material 30 percent of the time, two caches will be full 60 percent of the time and all three caches will be full 10 percent of the time. Thus, the expected value of full caches at a given time is 1.8 (the math for this is simply [.30*1] + [.60*2] + [.10*3] = 1.8) and the steady state probability of success for the supported unit in retrieving sustainment, in general, is 60 percent (1.8 divided by 3). This assumes there are no other interfering variables such as actions from an adversary or weather which an analyst could introduce into the model's complexity to achieve a new expected probability for successful resupply. This example demonstrates a fundamental method to gain insight toward successful sustainment if employed in a degraded or denied communication environment between the supported and supporting unit.

In choosing the number of caches to employ, the logistician faces a tradeoff between the success rate of sustainment attempts and the variability in the pattern of life ...

In choosing the number of caches to employ, the logistician faces a tradeoff between the success rate of sustainment attempts and the variability in the pattern of life achieved by the supporting and supported units. Using the scenario above, a one-cache system would improve the steady-state probability of success for the supported unit retrieving sustainment material to 100 percent. However, there would be no variance in the pattern of life for the supported and supporting unit in retrieving and replenishing sustainment. A two-cache system has a 66.7 percent steady-state success probability for retrieving sustainment. In this instance, the system balances a decrease in the success probability with a small increase in the pattern of life variability. A four-cache system drops the probability of success to approximately 57 percent with a convergence to steady-state probabilities occurring after seven iterations. Despite the drop in the probability of success, there is an increase in the supported and supporting units' ability to vary their pattern of life when attempting to retrieve and replenish sustainment in the caches. Finally, as the number of caches employed in the system increases, the steady-state probabilities converge in a manner that suggests a 50 percent success rate for the supported unit in retrieving sustainment material from a cache. The number of iterations required to achieve steady state probabilities also increases and we can calculate the expected number of iterations before achieving a steady state regardless of whether the system started with all cache sites full, some full, or even all cache sites empty.

Obviously, the Markov Chains and the associated probabilities outlined in this article are specific to this scenario. The probability of success would change if, for example: the supporting unit makes two attempts to replenish for every one attempt of the supported unit to retrieve; if there were three supported units and only two supporting units involved in the sustainment system; or if there was a chance the supported or supporting unit did not make an attempt to retrieve or resupply sustainment from a cache site for any reason (which would mandate the introduction of a possible event where no cache sites are full which would be reflected as a zero to the left of the table and a zero across the top part of the table representing our Markov chain). Additionally, the probability of success increases when supporting and supported units can communicate their intentions. Nonetheless, if resupply and replenishment were completely random due to communications denial, we have shown that the success rate for the supported unit retrieving sustainment (once achieving a steady state for the system) in the above scenario will range between 50 percent and 100 percent depending on the number of caches employed. How quickly the system converges to these steady-state probabilities depends on several factors. In our single supported unit and single supporting unit example, the primary factors are the number of possible cache sites and the number of caches filled at the beginning. However, whatever the system variables are, we can model the system with a Markov Chain once we determine the transition probabilities.

With the employment of Markov Chains in planning for a communications denied or degraded environment, logisticians and operating units can use the resulting probabilities to determine how much material using units should carry to minimize the risk of depleting on-hand sustainment, how many caches to employ to vary the pattern of life and minimize detection, how many attempts should be made to retrieve and replenish, and how much sustainment to place at caches as well as other logistics planning considerations. Markov Chains are not just limited to retrieval and resupply models. They can be applied to maintenance models to determine expected future readiness, manpower models to determine expected retention rates, and detection practices to determine rates of success and failure in finding a target, as well as other applications. In short, Markov Chains can enhance the understanding of logistics conditions in the planning and execution of operations in a variety of ways. In the scenario described in this article, we demonstrated a fundamental method of circumventing the difficulties of sustaining operations in an environment where the supported and supporting units are unable to communicate with each other.

Markov Chains can enhance the understanding of logistics conditions in the planning and execution of operations in a variety of ways.