21st ICCRTS: C2 in a Complex Connected Battlespace

AGGREGATED VALUE ANALYSIS OF THE DELIVERED INFORMATION TO COMMAND AND CONTROL SYSTEMS DECISION MAKERS

Topic 4: Socio-technical Networks in Complex Connected Battlespace

Marcelo José Camilo – Major
Brazilian Army Systems Development Center
camilomj26@gmail.com

Andersonn Kohl – Colonel
Brazilian Army Systems Development Center
kohl@cds.eb.mil.br

David Fernandes Cruz Moura – Major
Defense Communications Research & Innovation Center
Brazilian Army Technological Center
Universidade Estácio de Sá
dfcmoura@gmail.com

Leonardo Henrique Moreira – Captain
Brazilian Army Systems Development Center
moreira@cds.eb.mil.br

Centro de Desenvolvimento de Sistemas (Systems Development Center)
Quartel-General do Exército, Bloco G, 2º andar, Ala Norte
Setor Militar Urbano – Brasília-DF-Brasil - 70630-901
phone: 55-61-3415-5767

Centro Tecnológico do Exército (Brazilian Army Technological Center)
Núcleo de Inovação e Pesquisa em Comunicações Aplicadas à Defesa
Avenida das Américas, 28705 – D13
Guaratiba – Rio de Janeiro – RJ – Brazil – 23020-470
phone: +55-21-2410-6254

Universidade Estácio de Sá (Estácio de Sá University)

AGGREGATED VALUE ANALYSIS OF THE DELIVERED INFORMATION TO COMMAND AND CONTROL SYSTEMS DECISION MAKERS

Abstract

In this paper we aim to analyze the value of the information that is delivered to the decision maker in Command and Control systems. We consider a packet of information as a discrete unit of information. We suppose that the system has a queue in which all packets of information are stored before being sent to the commander. Afterwards, we define functions that assign values over time to the packets of information. Therefore, we analyze two traditional strategies to deliver the packets of information to the decision maker. The first one is to deliver the packets of information in the chronological order that they arrive in the system. The second one is to deliver the packets of information in the descending order of their values. So, we have proposed a new strategy, named SMLPI (Strategy of Minimal Loss for Processed Information). This strategy aims to increase the aggregated value of the packets of information to the decision maker. We defined LVQ_j (Lost of Value to the Queue) of a packet of information j as being the sum of the loss of value in all other packets of information that remain in the queue when we choose the packet of information j to be processed. The SMLPI strategy sends the packet that has the smallest LVQ_j to the decision maker. Simulations show that the average of the aggregated value to the decision maker using the proposed strategy is higher than the evaluated value using the aforementioned traditional strategies. Another important observation is that the most the decision maker is overloaded, the most the proposed strategy is advantageous.

Keywords: Value of information, Command and Control System, decision maker’s Overload.

1. Introduction

C2 (Command and Control) is the exercise of authority and direction by a properly designed individual over assigned resources in the accomplishment of a common goal [1]. According to [2], functions that are associated with the C2 issues of a given military mission are: a) establishing intent (the goal or objective); b) determining roles, responsibilities, and relationships; c) establishing rules and constraints (schedules, etc.); d) monitoring and assessing the situation and progress; e) inspiring, motivating, and engendering trust; f) training and education; and g) provisioning. It is useful to add to the former list the following function: h) risk management.

The objective of most typical military operations is to ensure personnel safety, namely by protecting the life of people. In these operations, decisions must be made in a short period of time, otherwise they will be ineffective or even useless. The use of IT (Information Technology) in such scenario is very important, since because crucial information shall be delivered in a shorter timeframe and with great reliability. The IT military systems that help the execution of C2 functions are known as C2 Systems.

The authors of [3] mention that while it was diagnosed that data is being collected at rates never seen before - thanks to a wide variety of high resolution and transmission rate sensors - it became clear that the tools and algorithms for a satisfactory analysis of extensive data are widely missing. This challenge is greater in military C2 applications involving complex connected battlespace, because the collected data in this environment is unstructured and heterogeneous. Besides that, sensors have different levels of coverage and standardization. Furthermore, it is worthy to mention that the connections used in tactical networks are intermittent, insecure, and have low capacity.

Another point to consider is the fact that a large amount of data in a C2 system may overload it. This means that the system will not be able to transmit, store, or process all data properly. Extensive data may also overload the decision maker. If the commander sees himself in this type of situation, he will not be able to analyze all information that is available in order to make his decision at the moment of performing this task. Therefore, the decision maker will ignore all or part of the information - which will lead to a limited use of the system and undermine the quality of the decision.

The writers of [3] mention that current systems and processes of the US Department of Defense (DoD) for management and analysis of tactical information cannot be effectively scaled to meet the challenge of an increase in the amount of data. In parallel, the authors also state that available commercial tools, algorithms, and techniques cannot be directly applied to the needs of the DoD's applications.

In order to avoid the overload mentioned above, data in C2 system should be properly handled. From the C2 system’s perspective, data fusion and data mining are well known techniques used for dealing with this issue. Data fusion is the process of integrating multiple data and knowledge, turning the same real world object into a consistent, accurate, and useful representation. Data mining is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. On the other hand, from the decision maker’s perspective, this problem brings us a great challenge. Therefore, new approaches for the management and analysis of information that is delivered to the decision maker are necessary.

If the system has a lot of information and the decision maker cannot process all of them, the system has to send only a piece of this information to the commander. The problem that arises from such scenario is how to define which piece of information will end up being more advantageous to be sent at first.

In this paper we aim to analyze the value of the information that is delivered to the decision maker in C2 systems. In order to enable this work, we consider a packet of information as a discrete unit of information, like a text message, image, or video [4]. We suppose that the system has a queue in which all packets of information are stored before being sent to the commander. Then we define functions that assign values over time to the packets of information. We choose the decrease linear and step functions for this issue. Therefore, we analyze the two traditional strategies to deliver the packets of information to the decision maker. The first one is to deliver them in the chronological order that they arrive in the system. The second one is to deliver them in the descending order of their values.

Because of the weakness of the cited strategies, we have proposed a new one, named SMLPI (Strategy of Minimal Loss for Processed Information). The proposed strategy aims to increase the aggregated value to the decision maker. We defined LVQ_J (Lost of Value to the Queue) of a packet of information J as the sum of the loss of value in all packets of information that remain in the queue when we choose the packet of information J to be processed. The SMLPI strategy sends to the decision maker the packet of information that has the smallest LVQ_J.

Simulations show that the average of the aggregated value to the decision maker using the proposed strategy is higher than this value using the Strategy of Chronological and than the Strategy of Delivering the Packets of Information in the Descending Order of their Values. Another important observation is that the most the decision maker is overloaded, the most the proposed strategy is advantageous.

The remainder of this work is structured as follows: in Section II, we model the problem. In Section III, we analyze and describe the strategy that delivers the packets of information in the chronological order that they arrive in the system, the strategy that delivers them in descending order of their values, and the proposed strategy. In Section IV, we perform simulations to compare the aggregated value to the decision maker using the three cited strategies. Finally, in Section V, we present the conclusions.

II. Modeling of the Problem

Aiming at visualizing the information that is delivered to the decision maker in C2 systems, we use the Dynamic Model of Situated Cognition. This model is described as follows:

A. Dynamic Model of Situated Cognition

The Dynamic Model of Situated Cognition (DMSC) [5], shown in Figure 1, was first introduced in 2003 and was quickly accepted by the DoD community as a descriptive model for processing and making the data and information flow on a battlefield [4]. The model is described using a military C2 system structure.

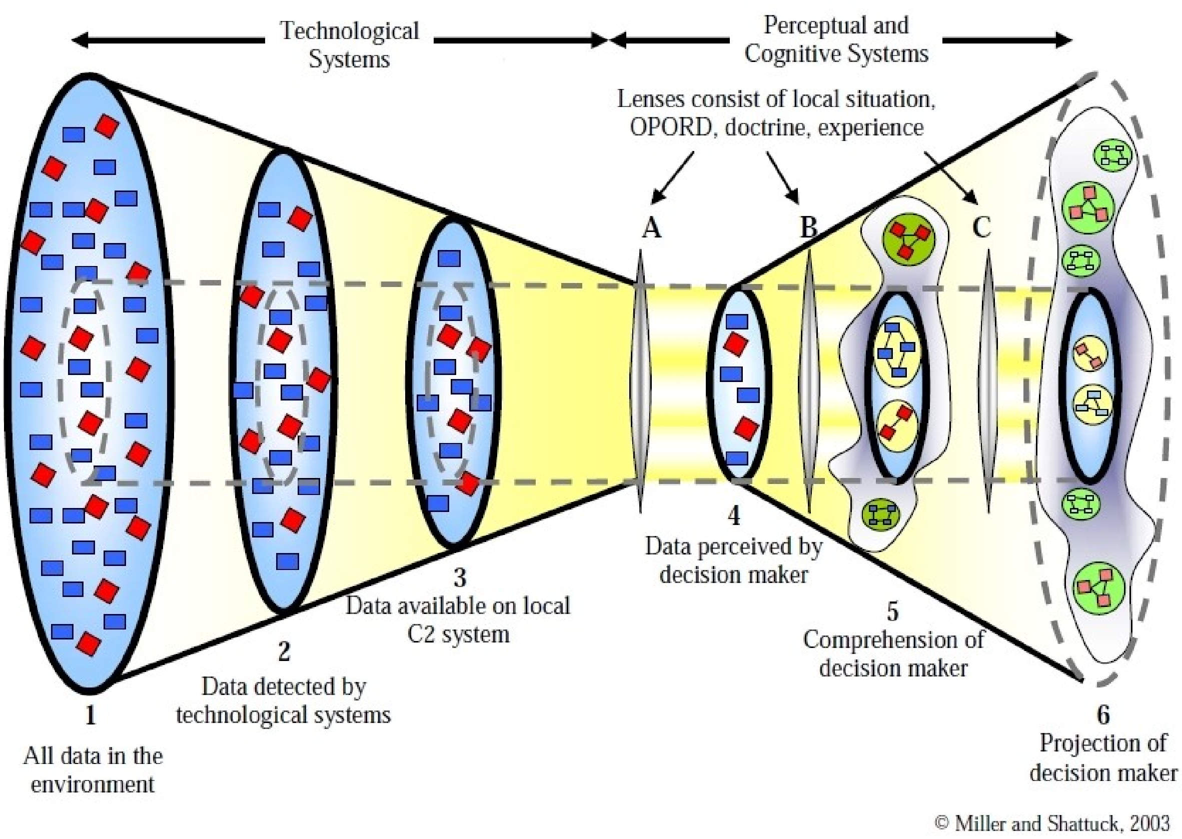

Figure 1 – Dynamic Model of Situated Cognition [5]

The model shows the relationship between the battlefield, the technological systems, and the perceptual and cognitive human process, providing an effective framework to see the flow of data and information in a complex system. Although it is not an analytical model, DMSC provides a way to see the flow of data and information that is accessible not only for analysts but also for decision makers.

The model has six ellipses. The Ellipse 1 consists of all data that exists in the operating environmental. The Ellipsis 2 and 3 represent, respectively, the data collected by the sensors and those shown to the decision maker in C2 systems. These ellipses represent the technological portion of the model. Ellipsis 4, 5, and 6 are, respectively, the perception, understanding, and the projection of the decision maker. The last three ellipses target the cognitive part of the model.

B. Working Model

Since the objective of this work is to model the information delivered to the decision maker, we chose to focus on data that is raised by the sensors, passes through the C2 system, and is delivered to the commander. It means that we have focused on the Ellipsis 2, 3, and 4 of the DMSC and on the data and information that flows throughout those ellipses.

There are three elements in our working model as it is shown in Figure 2:

- One queue where packages of information are stored before being processed by the decision maker;
- One set of sensors - all of them generate packages of information to the decision maker;
- One decision maker who gets the packets of information stored in the queue and processes them (one after another).

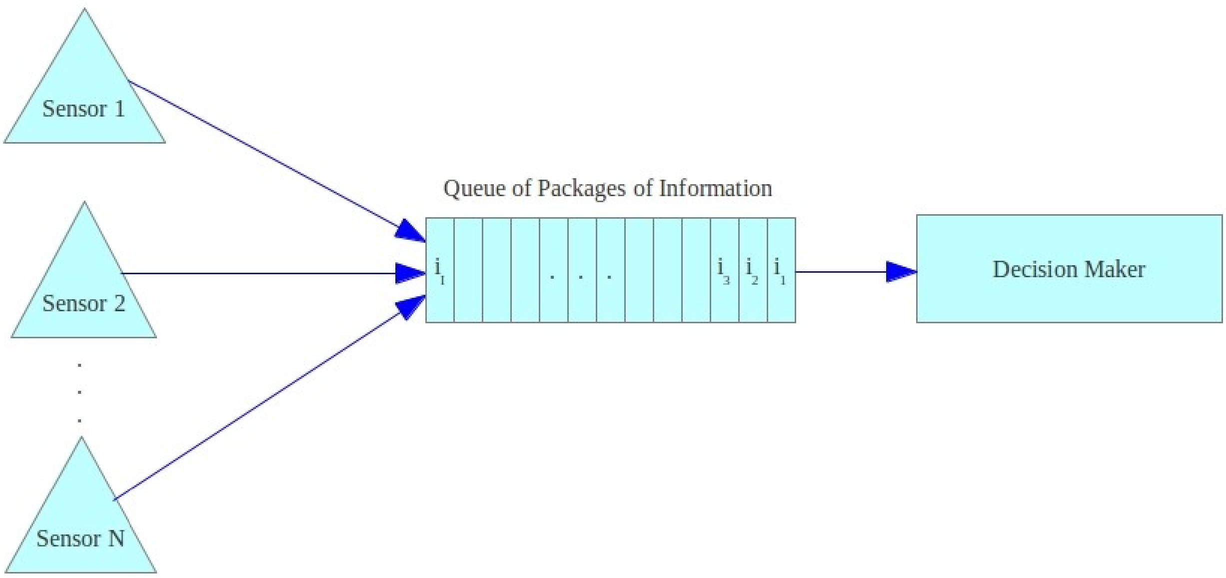

Fig. 2. Working Model

We consider only one decision maker based on the tenet of “Unity of Command” by Sun Tzu [6] and we have also taken into consideration the Brazilian Army’s culture. However, the issue of having more than one commander could be taken into consideration when adopting the proposed ideas of this paper as long as some appropriate adjustments are made.

First, we consider $t \in \mathbb{R}_+$, the system time, and $t=0$ as the initial time of the system. So, we have taken into account the following conditions:

1. The number of packages of information in the queue is $I, I \in \mathbb{N}_+$.
2. The system has $C, C \in \mathbb{N}_+$ types of packet of information.
3. Each $C_c, C_c \in \mathbb{N}_+$ types of packet of information has an expected time $TPC_c, TPC_c \in \mathbb{R}_+^*$ in which each packet of information of this type is processed.
4. The number of packets of information of the type C_c is $IC_c, IC_c \in \mathbb{N}_+$. Therefore, we have $\sum_{C_c=1}^{C} IC_c = I$.
5. We can assign, to each packet of information I_i, a value function $v_i, v_i \in \mathbb{R}_+$ which describes how the value of packets of information varies over time.

We have studied the values of information packets as follows:

C. Value of Packages of Information

Based on [4], we defined a packet of information as a discrete unit that could come in various forms such as: text message, image, video, and so on. Examples of packets of information include:

1. 2 (two) minutes of video, received from a high altitude drone that shows a supposed group of soldiery going to a meeting place.
2. 30 (thirty) seconds of voice transmission that contains a report of a situation, which reports the location and the activity of an enemy antiaircraft defense battery.
3. 3 (three) megabyte satellite image of a 2 (two) by 2 (two) kilometer region, in which there is a city of interest.

The types of value of information included in our model are:

1. Decreasing Linear, in which:

$$v_i(t) = b_i - a_i * t, \; if \; t \leq \frac{b_i}{a_i}, v_i(t) = 0, otherwise$$

2. Step, in which:

$$v_i(t) = c_i, if \; t \leq t_i, v_i(t) = 0, otherwise$$

In the Decreasing Linear function the parameters b_i and a_i, define, respectively, the initial value and the rate of change (decrease) of the value of the packet of information over time. In the Step function the parameters c_i and t_i, define, respectively, the “fixed” value of packet of information and its expiration time.

The assignment of values to these parameters for each package of information belongs to the C2 applications domain. In this paper, we are concerned with the change of values of packets of information over time. Other aspects of the values of packets of information, such as reliability, quality, geographical location, and so on, are outside the scope of our analysis and they should be encapsulated in the mentioned parameters.

As an example of an information packet that has a Decreasing Linear value function, we consider the information of a car containing the commander of an enemy force in a certain position. At the moment when the information is detected, the value of the packet of information is higher (its initial value) due the fact that there is only one position to find the enemy. As time passes by, its value decreases because the number of positions where the vehicle may be found increases over time due to displacement. This value decrease happens because the cost that would be spent in people and equipment to make a possible seizure increases.

The information about a radioactive cloud that is estimated to reach a troop within 30 minutes is an example of a packet of information which value function is Step. If the packet of information is delivered to the troop within the 30-minute time frame or less, the anti radiation masks can be distributed and no problem will occur. If the packet of information is delivered to the troop in more than 30 minutes, then it will have no value since the threat will have already caused some serious damage.

We also chose these types of value functions based on [4]. Other types of value functions can be used in this analysis. The C2 systems that we work with do not (yet) attribute value (therefore they do not attribute value function) to packets of information. We consider the attributions of value function to the packets of information as one of the many challenges of XXI century C2 systems.

D. Aggregated Value to the Decision Maker

Another important point in this work is the Aggregated Value to the decision maker. Therefore, we consider this topic in this section. One function of a C2 system is to deliver packets of information about an operating environment that aggregated value to the decision maker. Based on this requirement, we define the following:

1. $vi, vi \in \mathbb{R}_+$ as the total aggregated value of the packets of information provided by the C2 system if all of them have already been processed as soon as they arrive in the queue. So, we have:

$$vi = \sum_{i=1}^{I} TPC_i.$$

Where C_i is the type of the packet of information i.

2. $t_{end}, t_{end} \in \mathbb{R}_+$ as the time in which the decision maker finishes processing all the packets of information in the queue.

3. $v(t), v(t) \in \mathbb{R}_+$ as the aggregated value to the decision maker, in time *t*, due to the fact he has processed packets of information. $v(0) = 0$ because the decision maker has not yet processed any packet of information and therefore he has not aggregated any value using the system. Whenever the decision maker is idle, he picks up a packet of information *i* in the queue and process it. Then he spends TPC_j, where *j* is the type of the packet of information *i* that will be processed, units of time in busy state due to the processing. After finishing the processing, the decision maker aggregates the value of this packet of information at the current time, that is:
$v(t+TPC_{j)}) = v(t) + v_i(t+TPC_{j)})$.

4. $lv(t), lv(t) \in \mathbb{R}_+$ as the loss of aggregated value to the decision maker in the time t due to the fact that all the packets of information have not been processed at the time they arrived in the queue. So we have:
$lv(t_{end}) = vi - v(t_{end})$

E. Definition of the Problem

As the value of packets of information in the queue can vary over time, the goal is, given a queue with I packets, to determine an order to deliver the packets of information to the decision maker aiming to maximize the aggregated value at the end of the processing. One example of delivering order is the chronological order in which the packets of information are delivered in the order they arrived in the queue - that is, the order is $(i_1, i_2, i_3, ..., i_I)$.

If $I = 0$ or $I = 1$, the problem is trivial, because $v(t_{end}) = 0$ and $t_{end} = 0$, in the first case, and $v(t_{end}) = v_1(TPC_1)$ and $t_{end} = TPC_1$, in the second one. If $I > 1$, we have the following: the number of orders in which the packets of information can be delivered to the decision maker is $I!$ (the factorial of I). Excluding computational limitations, it would be enough to test all $I!$ possible orders and verify which one generates the biggest aggregated value to the decision maker.

However, as mentioned in [7], as the factorial function grows faster than even the exponential function, it is not feasible to list all the possible orders and verify which one generates the highest aggregated value to the decision maker, unless we take I as a small value. This problem is further deepened in military missions, since decisions must be made in a short period of time otherwise they will be ineffective or even useless. We need a strategy to maximize (or increase as much as possible) the aggregated value to the decision maker without testing all possible orders to deliver the packets of information.

F. Example Situation for Work

As an example situation for work, we consider a system with three types of packets of information, that is, $C = 3$. The processing times expected for each type of packet of information is: $TPC_1 = 4$, $TPC_2 = 5$ e $TPC_3 = 10$. In this context, we have a queue with 4 packets of information, i_1, i_2, i_3, and i_4, which types and value functions are listed in Table I.

Packet of information	***Type***	***Value Function***
i_1	*1*	$v_1(t) = 300$, if $t \leq 10$; $v_1(t) = 0$, otherwise
i_2	*2*	$v_2(t) = 500 - 10t$, if $t \leq 50$; $v_2(t) = 0$, otherwise
i_3	*1*	$v_3(t) = 500$, if $t \leq 5$; $v_3(t) = 0$, otherwise
i_4	*3*	$v_4(t) = 100 - 2t$, if $t \leq 50$; $v_4(t) = 0$, otherwise

Table I
Types and Value Functions of Packets of Information in Example Situation for Work

We define the values in the Table I just to explain the cited and proposed strategies. There is no relationship between these values and the C2 systems that we work with. Other values may be used without any loss of generality.

If the decision maker could start processing all the packets of information at $t = 0$, the aggregated value would be: $vi = v_1(4) + v_2(5) + v_3(4) + v_4(10) = 300 + 450 + 500 + 80 = 1{,}330$. In this case $t_{end} = 10$, it is the time in which the decision maker finishes processing the packet of information that has the bigger TPC_i, i_4. We conclude that the aggregated value in using the system will be at most 1,330.

We have studied the value of the aggregated value to the decision maker given three strategies for sorting the packets of information to be delivered to the decision maker: a) the Strategy of Chronological Order; b) the strategy of delivering the packets of information in the descending order of their values, and; c) the Strategy of Minimal Loss for Processed Information, proposed in this work.

In order to guide our study, in Table II we show all *4! = 24* orders to deliver the packets of information that is possible in our example situation, the values of the packets of information when they are processed, and in the last column the aggregated value of the order.

Order	***Values of Packets of Information***	***Aggregated Value***
(i1, i2, i3, i4)	*v1(4) + v2(9) + v3(13) + v4(23)*	*764*
(i1, i2, i4, i3)	*v1(4) + v2(9) + v4(19) + v3(23)*	*772*
(i1, i3, i2, i4)	*v1(4) + v3(8) + v2(13) + v4(23)*	*724*
(i1, i3, i4, i2)	*v1(4) + v3(8) + v4(18) + v2(23)*	*634*
(i1, i4, i2, i3)	*v1(4) + v4(14) + v2(19) + v3(23)*	*682*
(i1, i4, i3, i2)	*v1(4) + v4(14) + v3(18) + v2(23)*	*642*
(i2, i1, i3, i4)	*v2(5) + v1(9) + v3(13) + v4(23)*	*804*
(i2, i1, i4, i3)	*v2(5) + v1(9) + v4(19) + v3(23)*	*812*
(i2, i3, i1, i4)	*v2(5) + v3(9) + v1(13) + v4(23)*	*504*
(i2, i3, i4, i1)	*v2(5) + v3(9) + v4(19) + v1(23)*	*512*
(i2, i4, i1, i3)	*v2(5) + v4(15) + v1(19) + v3(23)*	*520*
(i2, i4, i3, i1)	*v2(5) + v4(15) + v3(19) + v1(23)*	*520*
(i3, i1, i2, i4)	***v3(4) + v1(8) + v2(13) + v4(23)***	***1,224***
(i3, i1, i4, i2)	*v3(4) + v1(8) + v4(18) + v2(23)*	*1,134*
(i3, i2, i1, i4)	*v3(4) + v2(9) + v1(13) + v4(23)*	*964*
(i3, i2, i4, i1)	*v3(4) + v2(9) + v4(19) + v1(23)*	*972*
(i3, i4, i1, i2)	*v3(4) + v4(14) + v1(18) + v2(23)*	*842*
(i3, i4, i2, i1)	*v3(4) + v4(14) + v2(19) + v1(23)*	*882*
(i4, i1, i2, i3)	*v4(10) + v1(14) + v2(19) + v3(23)*	*390*
(i4, i1, i3, i2)	*v4(10) + v1(14) + v3(18) + v2(23)*	*350*
(i4, i2, i1, i3)	v4(10) + v2(15) + v1(19) + v3(23)	430
(i4, i2, i3, i1)	v4(10) + v2(15) + v3(19) + v1(23)	430
(i4, i3, i1, i2)	v4(10) + v3(14) + v1(18) + v2(23)	350
(i4, i3, i2, i1)	v4(10) + v3(14) + v2(19) + v1(23)	390

Table II
Delivering Orders of Packets of Information and Their Aggregated Values

In the Table II, we can see that the best delivering order of the packages of information is (i_3, i_1, i_2, i_4) (its line is underlined in the table) that has $v(t_{end}) = 1{,}224$ and $lv(t_{end}) = 106$. The loss is $\frac{106}{1{,}330} = 0.08\, vi = 8\%\, of\, vi$.

III. Description and Analysis of Traditional and Proposed Strategies

In this section, we describe and analyze the two traditional strategies to deliver the packets of information to the decision maker using the example situation for work. The first one is to deliver the packets of information in the chronological order that they arrive in the system. The second one is to deliver the packages of information in the descending order of their value. So, we have described and studied the proposed one, named SMLPI (Strategy of Minimal Loss for Processed Information).

A. Strategy of Delivering the Packets of Information in Chronological Order

Our experience in the use of Command and Control Systems has shown us that most current C2 systems deliver packets of information to the decision maker in chronological order that they arrive at the system, that is, they deliver the packet of information i_1, then i_2, and so on, until i_I. They use the order $(i_1, i_2, ..., i_I)$ to deal with our problem.

Considering the queue of our example situation, the aggregated value to the decision maker using this strategy is: $v(t_{end}) = v_1(4) + v_2(9) + v_3(13) + v_4(23) = 300 + 410 + 0 + 54 = 764$. In this case, $lv(t_{end}) = 1{,}330 - 764 = 566$ and $t_{end} = 23$. This means that about *42.56%* of the packets of information value was lost. This result is also only *46.24%* of the aggregated value obtained with the best order to deliver the packets of information.

It is important to note in this example that the system delivers the packet of information i_3 to the decision maker, at $t = 9$, and it took *4* minutes to process it, although i_3 did not aggregate any more value because it has exceeded its expiration time. A simple improvement to this strategy would be to discard all packets of information whose values are 0. After applying this improvement to our example situation, we have:

$v(t_{end}) = v_1(4) + v_2(9) + v_4(19) = 300 + 410 + 62 = 772$, *and* $l(t_{end}) = 1{,}330 - 772 = 558$, *and* $t_{end} = 19$.

This means that about $\frac{558}{1{,}330} = 0.4195 = 41.95\%$ of the value of information was lost.

In this case, $t_{end} = 19$, hence there was also a decrease in the occupation time of the decision maker that would be possibly converted into increase in aggregated value if there was more packets of information to be processed.

B. Strategy of Delivering the Packets of Information in the Descending Order of their Values

Again, based on our experience in the use of command and control systems, we have noted that some C2 systems deliver packets of information to the decision maker in descending order of their values, that is, every time that the decision maker is idle, the system computes which packet of information has the biggest value and delivers this to the decision maker.

Considering again the queue of our example situation, in $t = 0$ the values of the packets of information are: $v_1(0) = 300$, $v_2(0) = 500$, $v_3(0) = 500$, and $v_4(0) = 100$. So the system sends i_2 to the decision maker (it can send i_2 or i_3 but it choose i_2). So the decision maker spends 5 minutes to process the i_2. In $t = 5$ the values of the packets of information are: $v_1(5) = 300$, $v_3(5) = 500$, and $v_4(5) = 90$. Based on this, the system sends i_3 to the decision maker. Afterwards the decision maker spends 4 minutes to process this packet. Then, in $t = 9$ the values of the packets of information are: $v_1(9) = 300$ and $v_4(9) = 82$. So the system sends i_1 to the decision maker. Additionally, the decision maker spends 4 minutes to process this packet. Finally, in $t = 13$, as there is only one packet in the queue, the system sends $i4$ to the decision maker.

We can see that this strategy chose (i_2, i_3, i_1, i_4) as the order to deliver the packets of information to the decision maker. The aggregated value to the decision maker using this strategy is: $v(t_{end}) = v_2(5) + v_3(9) + v_1(13) + v_4(23) = 450 + 0 + 0 + 54 = 504$. In this case, $lv(t_{end}) = 1{,}330 - 504 = 826$ and $t_{end} = 23$. This means that about 62.11% of the packets of information value were lost. This result is hugely smaller than the aggregated value obtained with the best order to deliver the packets of information.

C. Strategy of Minimal Loss for Processed Information (Proposed Strategy)

The motivation to our strategy is the concept of loss in the aggregated value to the decision maker. Therefore, we have analyzed what happens to the values of the packets of information that remain in the queue when we choose a packet of information to be processed.

We have defined lv_j as the loss of value of the packets of information that remains in the queue when we have chosen the packet of information i_j to be processed. Therefore, we have:

$$lv_j = \sum_{i=1, i\neq j}^{I} (v_i(t+TP_{Ci}) - v_i(t + TP_{Ci} + TP_{Cj})).$$

Where C_i and C_j are, respectively, the types of the packets of information i and j.

The SMLPI (Strategy of Minimal Loss for Processed Information) chooses for processing the packet of information j that has the lowest lv_j. After this packet of information is processed, it is removed from the queue. This process is repeated until there is only one packet of information in the queue. So this last one is processed.

Using our example situation once more, in step 1 of SMLPI, we have calculated the lv_j for each packet of information that is in the queue. So we have:

$lv_1 = v_2(5)-v_2(9) + v_3(4)-v_3(8) + v_4(10)-v_4(14) = 548,$
$lv_2 = v_1(4)-v_1(9) + v_3(4)-v_3(9) + v_4(10)-v_4(15) = 510,$
$lv_3 = v_1(4)-v_1(8) + v_2(5)-v_2(9) + v_4(10)-v_4(14) = 48$, *and*
$lv_4 = v_1(4)-v_1(14) + v_2(5)-v_2(15) + v_3(4)-v_3(14) = 900.$

So, it is easy to verify that the first packet of information that will be delivered is i_3.

After we deliver i_3 and we remove it from the queue, in step 2, we calculate the lv_j for each packet of information that still remains in the queue. Note that now $t = 4$. So we have:

$lv_1 = v_2(9)-v_2(13) + v_4(14)-v_4(18) = 48,$
$lv_2 = v_1(8)-v_1(13) + v_4(14)-v_4(19) = 310$, *and*

$lv_4 = v_1(8) - v_1(18) + v_2(9) - v_2(19) = 400.$
Again, it is easy to see that the second packet of information that will be delivered is i_1.
After we deliver i_1 and we remove it from the queue, in step 3, we calculate the lv_j for each packet of information that still remain in the queue. Note that now $t = 8$. So we have:

$lv_2 = v_4(18) - v_4(23) = 10$, *and*
$lv_4 = v_2(13) - v_2(23) = 100.$

Finally, we have noted that third packet of information that will be delivered is i_2. Consequently, the last packet of information that will be delivered is i_4. In this example, the order obtained by using the SMLPI, that is (i_3, i_1, i_2, i_4), is the best one, as we can see in Table II.

However, there is no guarantee that this strategy will always provide the best order. Although the loss in aggregated value to the decision maker is minimized in every step, the steps are independent among them, which does not guarantee the minimization of total loss.

The computation of lv_j in each step of the SMLPI is $O(I)$. As we have to calculate it $(I-1)$ times to find the desired order, the computational time of the proposed strategy is $O(I^2)$, which accordingly with [7] makes it feasible in a computational point of view.

IV. Simulations

Aiming to compare the aggregated value to the decision maker using the Strategy of Chronological Order, the Strategy of Delivering the Packets of Information in the Descending Order of their Values, and the SMLPI, we have run the simulations described below. In the simulations, we did not calculate the optimal orders since this calculation is impractical for large values of I.

A. Parameters of Simulations

1. We have carried out simulations for 5 (five) values of I (the number of packets of information in the queue), namely *20*, *30*, *40*, *50*, and *60*.

2. The system has two types of information, so $C = 2$. The value function of the first type is Linear Descending and the value function of the last one is Step. The simulation chooses the type of each packet of information with 50% of chance for each one. The expected time for these types of information, namely the time that is needed to process each packet of information of this type, is equal to 10 for each one. In other words $TPC_1 = 10$ and $TPC_2 = 10$.

3. If the type of the packet of information is equal to *1*, that is, the type that its value function is the Decreasing Linear, the simulation chooses:
the value of b in the set {100, 150, 200, 250, 300, 350, 400, 450, 500, 600}, with
10% of chance for each one of 10 values
the value of a in the set {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, with 10% of chance for each one of 10 values.

4. If the type of packet of information is equal to 2, that is, the type that its value function is Step, the simulation chooses:
the value of c in the set {100, 150, 200, 250, 300, 350, 400, 450, 500, 600} again, with 10% of chance for each one of 10 values.
the value of t in the set shown in the Table III. The probability of each value of t is also shown in the Table III.

Value of t	***Probability of the Value***
10	40%
50	30%
100	20%
200	10%

Table III
Values of *t and* their Probabilities

Again, we define these values just to perform the simulations. There is no relationship between these values and C2 systems that we work with. Other values may be used without loss of generality.

The pseudo code of the algorithm that runs the simulations is described as follows:

Algorithm to Comparison of the Aggregated Value to the decision maker Using the Strategy of Chronological Order, the Strategy of Delivering the Packets of Information in the Descending Order of their Values, and the SMLPI

```
queue size = I

sum_vi = 0
sum_vc =0
sum_vv =0
sum_vp =0

For k = 1 to 100,000 do
        For l = 1 to I do
                Choose the type of packet of information i_l ,
                If C_l = 1
                        Choose the value of b for Decreasing Linear
                        Choose the value of a for Decreasing Linear
                Else
                        Choose the value of c for Step
                        Choose the value of t for Step
                End of If
        End of For

        vi =  aggregated value of the packets of information if all of them had been processed as
                soon as arrived at the queue
        vc = aggregated values to the decision maker using the Strategy of Chronological Order
        vv = aggregated values to the decision maker using the  Strategy of Delivering the Packets
                of Information in the Descending Order of their Values
        vp = aggregated values to the decision maker using the SMLPI

        sum_vi = sum_vi + vi
        sum_vc = sum_vc + vc
```

sum_vv = sum_vv + vv
sum_vp = sum_vp + vp
End of For

average_vi = sum_vi / 100,000
average_vc = sum_vc / 100,000
average_vv = sum_vv / 100,000
average_vp = sum_vp / 100,000

The results of simulations are shown in Table IV, where we show the average of the aggregated values of the packets of information if all of them had been processed as soon as they arrived in the queue, and these values using the Strategy of Chronological Order, the Strategy of Delivering the Packets of Information in the Descending Order of their Values, and the SMLPI for the values of I that we have studied.

Value of I	***vi***	***Aggregated Values of the Packets of Information Using***		
		Strategy of Chronological Order	***Strategy of Delivering the Packets of Information in the Descending Order of their Values***	***SMLPI***
10	1,313	266	562	620
20	2,630	267	749	830
30	3,950	266	853	932
40	5,261	266	918	993
50	6,573	266	967	1,039
60	7,884	266	1,000	1,074

Table IV
Comparison of the Average of the Aggregated Value to the Decision Maker Using the Strategy of Chronological Order, Strategy of Delivering the Packets of Information in the Descending Order of Their Values, and the SMLPI

We can see in the results shown in the Table IV that the average of the aggregated value to the decision maker using the SMLPI is better than this value using the Strategy of Chronological Order and than the Strategy of Delivering the Packets of Information in the Descending Order of their Values. Another important observation is that the most the decision maker is overloaded, the most the proposed strategy is advantageous.

V. Conclusion

In this work, we propose a strategy, called Strategy of Minimal Loss for Processed Information, for delivering the packets of information to the decision maker in order to increase its aggregated value in using C2 systems. Simulations show that the average of the aggregated value to the decision maker using the proposed strategy is higher than this value using the Strategy of Chronological and than the Strategy of Delivering the Packets of Information in the Descending Order of their Values. Another important observation is that the most the decision maker is overloaded, the most the proposed strategy is advantageous.

References

[1] Neville Stanton, Christopher Baber, Don Harris. *Modelling Command and Control: Event Analysis of Systemic Teamwork*. Ashgate Publishing, Ltd, 2008.
[2] D. Albert and Hayes, R. E. Command and Control, Department of Defense, 2006.
[3] O. Savas and Y. Sagduyu and J. Deng and J. Li, Tactical big data analytics: challenges, use cases, and solutions, ACM SIGMETRICS Performance Evaluation Review, v. 41, pp. 86–89, 2014.
[4] D. D. Phillips, Mathematical Modeling and Optimal Control of Battlefield Information Flow, Naval Postgraduate School, 2008.
[5] N.L. Miller and L.G. Shattuck, A process model of situated cognition in military command and control, Proceedings of the Command and Control Research and Technology Symposium, 2004.
[6] N. Maquiavel and S. Tzu and W. C. H. Maquiavel and J. S. Cardozo and P. Holzchuh, A arte da guerra, Editora Evora, 2008.
[7] T. H. Cormen and C. E. Leiserson and R. L. Rivest and C Stein, Introduction to algorithms, 3rd ed. Cambridge: MIT Press, 2001.