Software Engineering Institute

Using the Vickrey-Clarke-Groves Auction Mechanism for Enhanced Bandwidth Allocation in Tactical Data Networks

Mark Klein
Daniel Plakosh
Kurt Wallnau

January 2008

TECHNICAL REPORT
CMU/SEI-2008-TR-004
ESC-TR-2008-004

Product Line Systems Program

Carnegie Mellon

This report was prepared for the

SEI Administrative Agent
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

List of Figures

Acknowledgements

Thanks to David Parkes, Peter Welch, Rick Kazman, and Gabriel Moreno for their suggestions about our work and this report. Also, thanks to Rajeep Dash and Alex Rogers, whose original work on auction mechanisms for sensor data fusion and recent discussions with the authors of this report provided a basis for the work reported here.

Abstract

A mechanism is an institution such as an auction, voting protocol, or a market that defines the rules for how humans are allowed to interact, and governs the procedure for how collective decisions are made. Computational mechanisms arise where computational agents work on behalf of humans. This report describes an investigation of the potential for using computational mechanisms to improve the quality of a combat group's common operating picture, in a setting where network bandwidth is scarce. Technical details are provided about a robust emulation of a tactical data network (based loosely on the Navy LINK-11) that was developed for the study. The report also outlines the basic principles of mechanism design, as well as the features of the Vickrey-Clarke-Groves (VCG) auction mechanism implemented for the study. The report describes how the VCG mechanism was used to allocate network bandwidth for sensor data fusion. Empirical results of the investigation are presented, and ideas for further exploration are offered. The overall conclusion of the study is that computational mechanism design is a promising alternative to traditional systems approaches to resource allocation in systems that are highly dynamic, involve many actors engaged in varying activities, and have varying—and possibly competing—goals.

1 Introduction

Systems make decisions. Control systems sense state and decide on control actions to keep key state parameters within a control envelope. Program-trading systems monitor financial markets and decide when to buy and when to sell. Both use information obtained from the parts of the system to make these decisions. In many cases, a decision maker can obtain the necessary information from the parts and make optimal decisions accordingly. However, this scheme can break down as systems get bigger. Two dimensions of scale are sufficient to demonstrate this point:

- The system is developed by and/or serves a growing *number of human* users; and human users have their *own incentives*. For example, in market-trading systems and frequently encountered peer-to-peer systems, computational agents act on behalf of humans. In this setting, the users must have an incentive to provide truthful information (e.g., how much a user values an item that is for sale) to the decision maker. Without this incentive, we can depend on users to hide or misrepresent this information, if it is in their interest to do so, even if this deception comes at the expense of the system as a whole.
- The system is increasingly distributed and performs a growing *number and diversity* of tasks. For example, *ad hoc* sensor networks and network-centric combat systems will support a (possibly open-ended) number and variety of human tasks and computational agents. In these settings, it is impractical to assume that a decision maker can be constructed that knows enough about each of these tasks to impose an efficient solution. By analogy, one can think of the economic distortions (e.g., supply, price, forecasting) introduced by centralized command economies. Such distortions become more prominent and severe as economies grow and become more diversified.

As systems scale up in these dimensions, interaction protocols are needed that are resistant to strategic manipulation by selfish users and that efficiently aggregate information from the parts of a system to enable effective global decision making. Computational mechanism design is the discipline of designing such interaction protocols.

We investigate the application of computational mechanism design to systems of interest to the U.S. Department of Defense (DoD), with particular emphasis on using computational mechanisms in highly dynamic, resource-constrained, performance-critical systems. To provide the investigation with clear and realistic scale dimensions, we developed an application framework that emulates a tactical data network. We then investigated the use of one class of mechanism, the Vickrey-Clarke-Groves (VCG) auction, to efficiently allocate bandwidth on the tactical network to improve the quality of a common operating picture (COP).

1.1 COMPUTATIONAL MECHANISM DESIGN

A *mechanism* is an institution such as an auction, a voting protocol, or a market that defines the rules or protocols for how individuals are allowed to interact and governs the procedure for how collective decisions are made. *Mechanism design* is the subdiscipline of game theory and economics concerned with designing such institutions so that they achieve prescribed and desirable global outcomes. *Computational mechanism design*[1] addresses situations where individuals are computational agents working on behalf of human agents.

Mechanism design has a deep research tradition in game theory, where it is sometimes known as implementation theory, and in microeconomics, where it is sometimes known as institution design. There are many examples of the practical use of mechanism design to achieve large-scale social objectives. McMillan offers a good discussion of the importance of getting the details of mechanism design right (in the U.S. public radio spectrum auction) and illustrates the consequences of mechanism defects (in the New Zealand radio spectrum auction) [McMillan 1994].

Computational mechanism design has a more recent history of practical application. One substantial and well-documented use of computational mechanisms falls under the general heading of e-commerce. For example, it has been reported that more than 98 percent of Google's $6.14 billion revenue (as of 2006) is achieved through the use of an explicitly designed auction mechanism for allocating advertising space on Web pages returned from keyword searches [Edelman 2007]. Another substantial application area in e-commerce is in supply chain optimization [Staib 2001, Chen 2005, Sandholm 2006].

Our investigation focuses on the comparatively less well-understood use of computational mechanisms to control or direct the behavior of large-scale, decentralized systems and in particular to achieve an efficient allocation of computational resources using economic mechanisms. In this use, computational systems are viewed as virtual economies, with computational elements competing to use scarce computational resources to achieve the elements' individual objectives.

The research literature provides examples of mechanisms being used to allocate processor cycles for scientific computing on the worldwide grid [Chen 2004]; for network routing [Holzman 2003]; for allocating network capacity [Anshelevich 2004, Anderson 2005]; for sensor fusion [Rogers 2006, Dang 2006]; for peer-to-peer systems [Chen 2004]; for task allocation for autonomous robots [Gerkey 2002]; and for electricity markets [Hinz 2003, Federico 2003]. This is not in any sense an exhaustive survey, and the use of market mechanisms to control complex system behavior is receiving considerable attention.[2]

1.2 CONTRIBUTION OF THIS WORK

The viability of techniques such as computational mechanism design can be established only when they are applied to problems of sufficient scale and complexity to expose the practical limitations of the techniques in question. When theory is applied to practice, practice invariably "pushes back." This "push back" often identifies opportunities to advance and refine the underlying theory of a technique that would never have been identified but for the confounding—and

---

1 The term *algorithmic mechanism design* is sometimes used instead.

2 See http://www.marketbasedcontrol.com/, for example.

impossible to predict—effects of real-world problems. Our work builds on earlier work by Dash and others [Rogers 2006, Dang 2006] but adds significantly to the complexity (and, we claim, the resulting fidelity) of the experimental setting. In addition, our work emphasizes the importance of accounting for human incentives in the process of designing computational mechanisms. With this philosophical background, our work has made three key contributions:

1. We developed an application framework[3] that exhibits sufficient scale and dynamic complexity to study the feasibility of computational mechanism design in a practical setting. The framework emulates a combat tactical data network and includes much of what is required to construct a common operating picture from radar sensor data.
2. We designed and implemented a variant of the well-known VCG auction for use in the application framework. The auction is used to efficiently allocate a fixed, but selectable, amount of network bandwidth to permit fusion of additional sensor data to improve the common operating picture. One novel aspect of our auction is that participants are both buyers and sellers of information, and each can obtain value from buying and selling.
3. Finally, and perhaps most importantly, we demonstrated that computational mechanisms can be used to implement distributed, value-based resource allocation schemes in the kind of highly dynamic, resource-constrained, performance-critical systems found in DoD combat systems. Such systems are *non plus ultra* for evaluating the practicality of using computational mechanisms to control the behavior of complex systems.

1.3 STRUCTURE OF THIS REPORT

Section 2 provides a high-level description of forming a common operating picture in tactical data networks and introduces the application framework as a way of making the problem domain concrete. Section 3 provides a brief overview of the key theoretical constructs underlying mechanism design. Section 4 applies these concepts to design an auction mechanism for allocating bandwidth in a tactical data network to provide incremental improvements to the common operating picture. Section 5 provides further details of how the application framework was used to study the behavior of the auction mechanism. Finally, Section 6 summarizes our key findings from this work.

[3] The implementation has been packaged for use by external research collaborators and has already been made available to select researchers at the Naval Postgraduate School and Harvard University.

2 Track Data Fusion and Information Gain

2.1 TRACK DATA FUSION SETTING

Almost all military group operations rely on the platforms (air, sea, and ground) involved in the operation to act as a cohesive force. To act as such a force, the platforms must establish and maintain a common understanding of the tactical situation. This common or shared understanding is achieved through the sharing and exchange of tactical data from sensors on each of the platforms in the group.

Tactical data is often exchanged and shared among the platforms using a standardized radio network, commonly called a tactical data information link (TADIL). TADILs are characterized by their standard message and transmission formats. Golliday's survey of TADILs circa 1985 is outdated in its fine details but still valid in the main features and vocabularies of TADILs today [Golliday 1985]. Our concern is with a subset of TADIL capabilities used to construct a common operating picture.

Major kinds of tactical information exchanged on a TADIL are the positions and movements of the platforms themselves and track data observed by the platforms. Track data is processed radar data which typically represents real objects such as airplanes, helicopters, missiles, ships, boats, land vehicles, and submarines.

This shared tactical data is then used by each platform to create a common operational picture by combing selected information from all the platforms. The accuracy and effectiveness of this shared tactical picture can critically depend on

- eliminating or minimizing sensor alignment errors on each platform, platform navigational position errors, and sensor biases and position errors, through a process commonly known as "data registration" or "gridlock"
- minimizing the display of multiple tracks that actually represent one object, through a process commonly known as "track correlation"[4]
- minimizing data latency by preventing multiple track reports on the link for a single real object through the use of reporting responsibility (R^2) rules. R^2 rules assign a track to the platform that has the best quality data for that track (position, velocity, etc.). The platform that is selected to report the track is said to have R^2 for that track.

R^2 rules minimize data latency by disallowing the redundant reporting of a single object by multiple platforms over the data link. The R^2 approach can be viewed as an extreme minimalist approach to the treatment of network bandwidth. It has the disadvantage of reducing the diversity of the source data to the platforms. At another extreme is an approach that assumes a superabun-

[4] The elimination of track identifier conflicts is beyond the scope of this work.

dance of network bandwidth, where even raw (unprocessed) radar returns are shared on the TADIL. This approach has the disadvantage of requiring communication technologies that are not yet practically available.

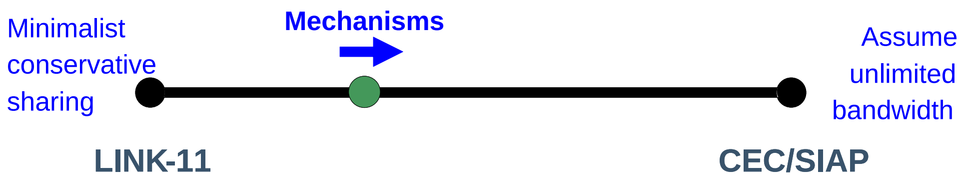

Figure 1: Conservative Enhancement to the COP from R^2 Baseline

As depicted in Figure 1, the research reported here assumes as its point of departure the minimalist approach characterized by TADILs such as LINK-11 rather than the superabundance approach characterized by more progressive capabilities such as those envisioned by the cooperative engagement capability (CEC)[5] and single integrated air picture (SIAP).[6]

Our research shows that an auction mechanism can efficiently allocate a finite but selectable amount of bandwidth, above and beyond that already used in a conventional R^2 approach, to improve the COP. We assume that platforms are rational and therefore liable to deceptive behavior if it furthers their objectives. This might, in fact, be a reasonable assumption in TADILs that operate under multiple coalition flags. In any event, the assumption of rationality is central to mechanism design and only serves to broaden the applicability of the results.

2.2 EMULATING R^2 ON A LINK-11 TADIL

The most direct way to highlight the key concepts of the tactical data networks and auction mechanisms explored in this research is to examine the research application framework piece by piece.

Figure 2 shows the main track display. In this run of the application, there are four active platforms or participating units (PUs). Each PU has a sensor envelope depicted as a circular region on the display. The track picture is quite confused in Figure 2. Each platform is reporting its contact data on the TADIL but is doing so from its own frame of reference, without accounting for navigation errors, orientation of radar from true north, and so forth.

5 For more information, see http://www.globalsecurity.org/military/systems/ship/systems/cec.htm.

6 For more information, see http://www.dtic.mil/ndia/systems/Hobart.pdf.

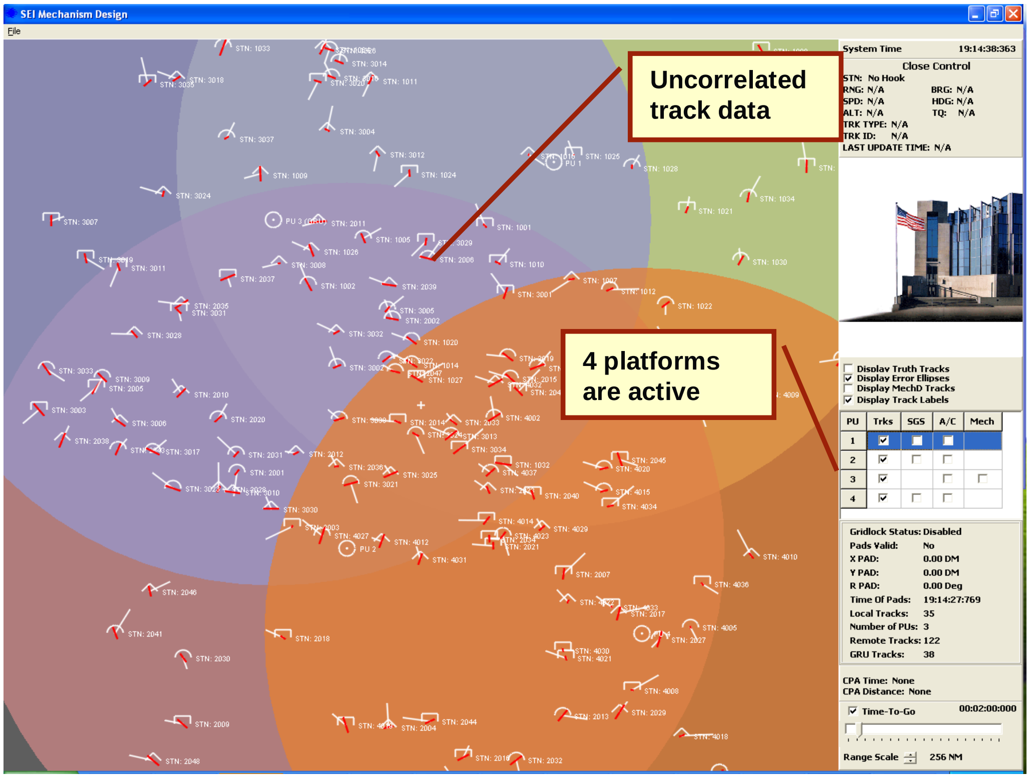

Figure 2: Track Display Prior to Track Correlation

Radar contacts appearing within this region of observation (RO) are depicted using standard symbols, although only a small subset of these symbols is needed in the prototype. A brief explanation of the most important symbols is provided in Figure 3. Time to arrival shows where a contact will be at the end of some time duration, assuming it holds a constant speed and bearing.

The error ellipses are not part of the standard notation but are displayed to emphasize the sensor error associated with each track. The ellipse is computed from the covariance of radar range and bearing errors, as well as navigation errors. The covariance data of a PU's simulated radar is configurable. As discussed later, sensor quality plays an important role in the auction mechanism.

A minor point worth noting is that the first digit of the track number encodes which PU is reporting a track on the link. For example, PU 3 is reporting a hostile track in Figure 3.

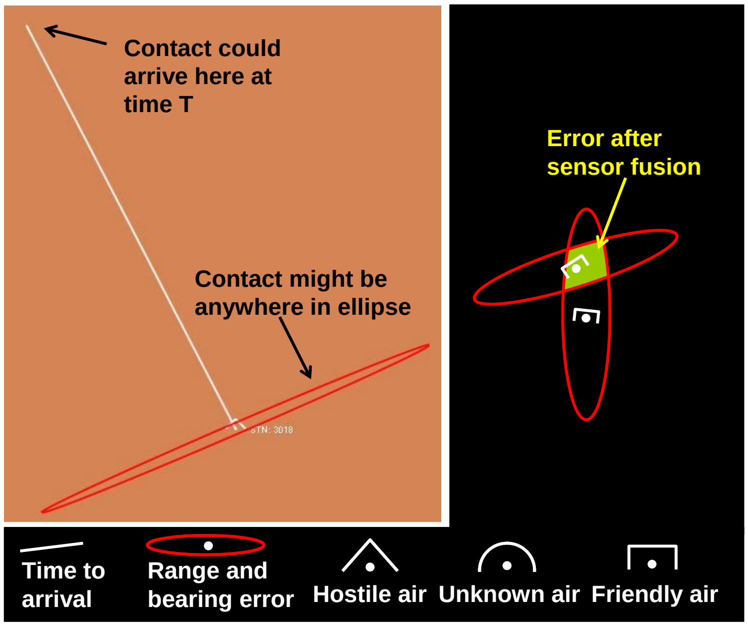

Figure 3: Display Symbols and Fusion Goal

In Figure 4, the effects of gridlock and correlation are depicted. It is possible in the research application to separately select when each platform chooses to perform gridlock and when each chooses to perform correlation.[7] Correlation requires gridlock, and gridlock without correlation is not particularly useful. Both are typically performed.

Note that in Figure 2 and Figure 4, PU 3 does *not* have a gridlock "checkbox" because it has been assigned the role of the "grid reference unit" (GRU). Typically, the role of the GRU in Navy tactical networks is played by Aegis Class ships, as these generally possess the highest quality track data. The application framework uses a relative gridlocking algorithm, and therefore the GRU's coordinate system is adopted as "truth."

The GRU imparts unique characteristics to the mechanism by virtue of the asymmetry in track quality between the GRU and other PUs. The GRU will almost universally acquire R^2 for any track in its RO. As we'll show later in this report, this asymmetry influences auction payoff, since the GRU always shares all of its data with all the other PUs and therefore has no additional "goods" to offer when it comes time for the auction.

7 The display uses SGS and A/C to denote gridlock and correlation, respectively. Historically, SGS is "shipboard gridlock system" …. A/C is "auto-correlation." The acronym SGS/AC is typically used.

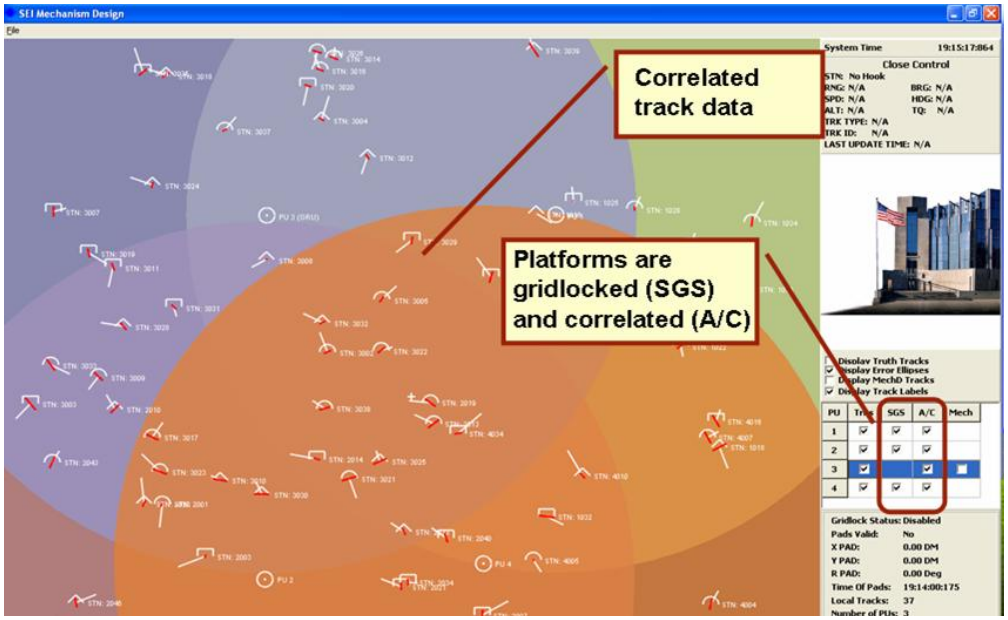

Figure 4:   Track Display After Track Correlation

The "minimalist bandwidth" effects of R^2 assignment are shown in Figure 5, which depicts a (clipped) portion of the network control station (NCS), another display provided by the application framework. The TADIL emulated in this application framework uses a round-robin approach to data transmission. The NCS indicates to each PU when it has a "transmit opportunity;" it is the task of the PU to transmit its R^2 data during its transmit opportunity. The time it takes to complete each round of communication is called the network cycle time (NCT).

The *Net Cycle Time* strip chart in the lower left of the display in Figure 5 shows the total network cycle time dropping from slightly more than 4 seconds to slightly less than 3 seconds after gridlock and correlation. A corresponding drop in the amount of track data (bytes per second) is shown on the *Bytes Sent* strip chart in the lower right of the display.

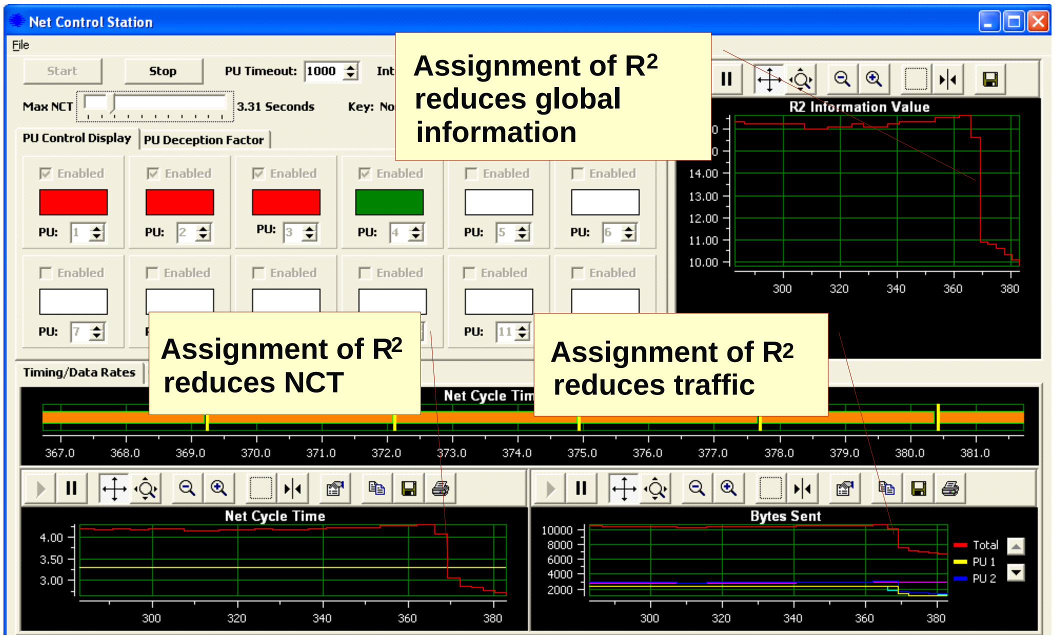

Figure 5: Effect of Reporting Responsibility on Bandwidth

A different measure is depicted in the upper right, where a sharp drop in R^2 information value is shown. This information value is computed as a function of the covariance data for all track data on the link. For our purpose, it serves as an indirect, if somewhat blunt, measure of the total information available on the link. When a track is no longer transmitted, the opportunity to fuse its data—an opportunity whose value will be inversely related to its covariance—is also lost. We use an auction mechanism to recover the most valuable gain in information for a given quantum of extra bandwidth.

2.3 AUCTIONING BANDWIDTH FOR IMPROVEMENTS IN THE COP

Assuming perfect track correlation, one effect of assigning R^2 is that the total network cycle time is reduced to its minimum. An auction would permit us to efficiently allocate an additional quantum of bandwidth to improve the quality of the COP. By “efficiently,” we mean that the auction will choose the track data that, when fused, will result in the largest possible gain in overall COP quality when subjected to a maximum network cycle time constraint.

Figure 6 shows a (clipped) portion of a network control station after a successful auction of TADIL bandwidth. Assigning R^2 had a reduced NCT from over 4 seconds to just under 3 seconds. In Figure 6, we set a target maximum NCT to 3.34 seconds. Increasing this value will allow additional track data to be transmitted and fused but will also result in increased latency between track updates. The mechanism allocates bandwidth equal to the difference between the selected maximum NCT and the actual NCT assuming only R^2 reporting.

Each cycle is demarked by a vertical yellow line on the *Net Cycle Time* strip chart (horizontal bars near the midpoint of the graphic). As time progresses, the breakdown display moves from right to left, one cycle at a time. Traffic generated by the auction protocol itself is displayed in blue. The auction protocol requires three cycles to complete; in the figure, only the last two cycles of an auction round are shown. Traffic generated by the normal R^2 track data is displayed in orange. Traffic generated by the additional track data allocated by the auction is displayed in red.

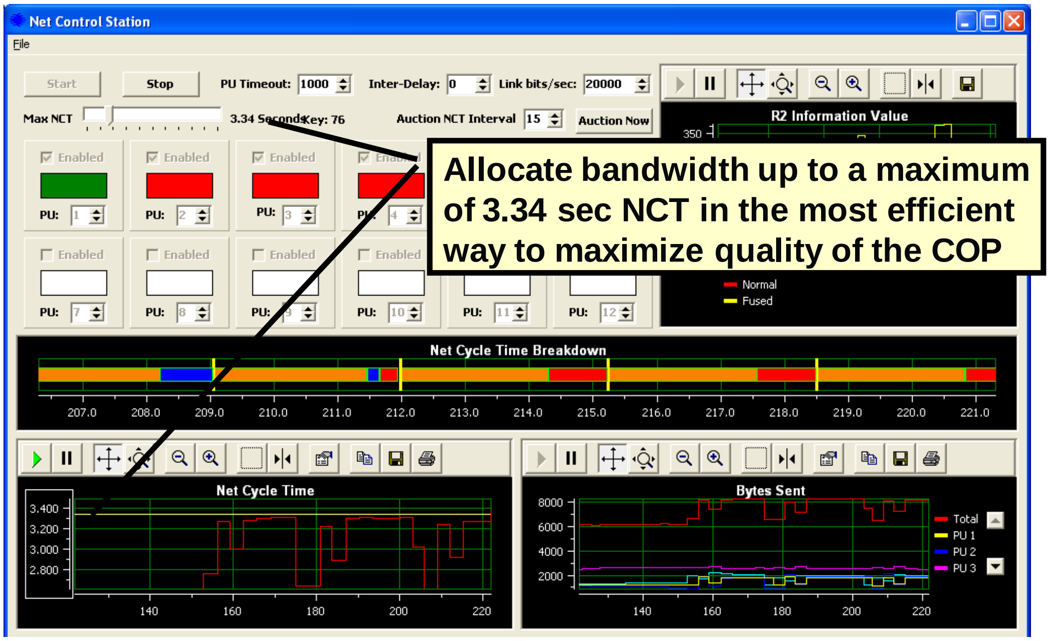

Figure 6: NCT After a Successful Bandwidth Auction

What has been described so far is, at root, the solution to an optimization problem using a simple knapsack algorithm. Given a bag of a finite capacity (here, a total amount of network bandwidth determined in part by the maximum NCT), find the collection of items that fills the bag to its maximum capacity (here, a collection of track descriptors), so that the solution maximizes an objective function (here, the quality of the COP).

The algorithmic heart of the auction we implemented is, in fact, a knapsack optimization algorithm. The mechanism goes beyond a mere knapsack, however, because it provides incentives for each PU to truthfully reveal information that is known only to each PU and that is required to construct an optimal knapsack solution. Specifically, each PU has information about its own track quality that it derives from the covariance of its radar apparatus; the mechanism provides incentives for each PU to truthfully reveal its track quality.

Intuitively, PUs will value track data of higher quality (i.e., lower covariance) over track data with lower quality (i.e., higher covariance). However, if we assume that PUs are rational, they will truthfully reveal their track data only if it is in their best interest to do so. Perhaps by overestimating its track quality, a PU might manage to "sell" its poor quality data at a higher price (in a sense of "sell" that is defined later in this report). Or, perhaps by underestimating its track quality, a PU might manage to have bandwidth allocated to transmit some other PU's tracks to it rather than send its own. Deceptive behavior among participants in a tactical data network might seem farfetched but is certainly not inconceivable in coalition rather than single-flag settings. In any case, the assumption of rational behavior is essential for mechanism design.

The task of the auction designer for this application setting is to ensure that each PU has an incentive to truthfully reveal its radar covariance.

We now turn to the details of the incentive compatibility and other theoretical aspects of mechanism design in Section 3 and to the precise details of how radar covariance relates to incentives and platform payoff in Section 4. We will return to the application framework in the Section 5 discussion on empirical results.

3 Mechanism Design

Mechanism design concerns designing institutions (or protocols) that govern the interactions of rational individuals with private preferences in a way that leads to a collective decision. The institution generally incentivizes the participants to reveal their own private preferences and guides the group decision so that it satisfies some global criterion. In this section, we will describe mechanisms and describe a specific mechanism, the VCG auction, in more detail.

3.1 SOME CONCEPTS OF MECHANISM DESIGN

Assume that there are n individuals 1,…, n (or participants or agents) who must collectively make a decision that affects each of them. The decision is to choose from a set, *A*, of alternatives. Example situations include

- Example (1): Individuals are citizens of a community. The decision is whether to spend community funds on a community project. The alternatives are "yes" or "no." The mechanism is voting.
- Example (2): Individuals are bidders at an auction. The decision is how to allocate items X, Y, and Z to bidders B_1, B_2, … B_n. The alternatives are every possible allocation of items to bidders (e.g., B_1 gets items X, Y, and Z or B_2 gets X and Y, and bidder B_9 gets item Z). The mechanism is an English (open-cry, first-price) auction.

Of course we are investigating mechanism design in a computational setting. The example that we elaborate later in this paper is

- Example (3): Individuals are computational agents representing ships in a battle group. The decision is how to allocate spare network bandwidth to send additional track data from one ship to the other ships. The alternatives are the various ways to allocate bandwidth. The mechanism is the VCG (sealed-bid, second-price) auction.

3.1.1 Individual Preferences

Group decisions are guided by individuals' preferences, which vary by situation. Mechanisms must be able to handle all combinations of those preferences.

Individuals participating in a group decision observe the situation and gather all the information they need to guide their preferences. For example, individuals are likely to value an umbrella higher and sunscreen lower when it is raining than when it is sunny. This observed information is represented abstractly by a parameter known as *type*. An individual's type is typically denoted by θ_i and the type profile (the vector of all individuals' types) as $\theta = (\theta_1, \theta_2, \ldots \theta_n)$.

3.1.2 Social Choice Function

The collective decision is represented as function $f(.)$ that maps the set of all type profiles to the set of all alternatives. This function is known as the social choice function. In other words, the social choice function chooses an alternative from the set of alternatives when given information that determines everyone's preferences. For example, θ_i might represent the track data that is currently owned by each ship i. This type then determines this ship's preference for how network bandwidth should be allocated in order for it to receive additional track data.

The social choice function is guided by an evaluation criterion, which is usually a function of the utility that the individuals accord to each alternative. Utility is a way to quantify preference and is a function of the alternative and the type. Given a situation characterized by type θ_i, $u_i(x_1, \theta_i) > u_i(x_2, \theta_i)$ indicates that individual i prefers alternative x_1 to x_2 in the situation abstractly described by θ. Designing a mechanism involves determining a measure for utility. In our example, information gain due to acquiring track data from another ship is our measure of utility.

A social choice function is said to be efficient if it maximizes the total utility of all individuals when "choosing" an alternative. The goal of mechanism design is to implement a social choice function that satisfies a criterion such as efficiency. For example, an efficient social choice function for Example (3) is one that allocates bandwidth for sending tracks in a way that maximizes the total utility of all ships.

3.1.3 The Induced Game

Mechanism design has an intimate relationship with game theory. A *game* is a formal representation of a situation in which a number of individuals strategically interact; that is, each individual's ultimate welfare not only depends on one's own actions, but also on the actions of the other individuals. Strategy is an important concept for games. A *strategy* is a "complete contingent plan or decision rule that determines how a player will behave in every possible situation." For example, each ship will choose a strategy that determines whether to accurately reveal its track data.

Formally, a *game* (in a game theoretic sense) consists of a set of players (or individuals), a set of strategies for each player, and a payoff function for each player that determines the payoff associated with every possible strategy profile (that is, the vector of strategies chosen by every player) [Mas-Colell 1995].

A central goal of game theory is to determine which strategy profile will be chosen by the set of individuals in some specific game. The central goal of mechanism design is to design the strategy set, rules of interaction, and payoff function so that the desired social choice function is implemented for all type profiles. A mechanism induces a game in the sense that, for a given situation, the type profile determines a strategy profile for the induced game. If the *equilibrium* associated with that game is consistent with the social choice function, the mechanism has achieved its goal. If the mechanism achieves this goal for all possible type profiles, it is said to implement the social choice function; for example,

- for any given situation described by what ships own what track data (*the type profile)*
- there exists a profile of ship strategies for what track data to reveal *(the strategy profile)*
- so that the allocation of bandwidth *(the outcome)*
- resulting from the auction *(the mechanism)*
- is the one with the highest total utility *(an efficient social choice)*

3.2 SOME DETAILS ABOUT THE VCG AUCTION

Auctions are a common type of mechanism, and the VCG auction is one of the most studied types of auctions. Another name for the VCG auction is the *second-price, sealed-bid* auction.

3.2.1 Single-Item Auction

Consider the single-item VCG auction where there is a single item up for auction. The participants submit a bid. The highest bidder wins but pays the amount of the second highest bid.

The key thing to notice about the VCG auction is that the winner's bid does not affect the price paid by the winner. The auction incentivizes bidders to directly reveal the true value they place on the item, even in this competitive setting, and ensures that the bidder who values the item the most wins it. To see this, order the bidders by their true values where V_i denotes bidder i's true value: $V_1 < V_2 <, \ldots, < V_{N-1} < V_N$. If all bidders bid their true values then bidder N wins, pays V_{N-1} and gains $V_N – V_{N-1}$. If bidder N bids higher than their value, the result remains the same. If bidder N bids lower than their true value, but still greater than $V_{N-1,}$ the result remains the same. If bidder N bids lower than V_{N-1} they lose the auction and gain zero. Therefore, bidder in is quite content to bid their true value. If some other bidder, bidder i, bids higher than V_N, they would win the auction, pay V_N and have a net "gain" of $V_i – V_N$, which is negative and therefore a loss.

This mechanism realizes the principle: "lying doesn't pay." Bidding something other than the bidder's true value was never beneficial and sometimes was detrimental. This mechanism is *incentive compatible*; it leads to bidders truthfully revealing the true values they place on items.

3.2.2 Multi-Item Auction

In the multi-item auction, there are many items up for auction, and each bidder bids on every subset of these items.[8] The desired outcome of the auction is to maximize the total value to all the bidders resulting from the auction, which is how we define the optimal allocation. Just as for the single-item VCG auction, optimality is achieved by constructing a payment scheme that incentivizes bidders to reveal their true values for items. The payment bidder i makes in a VCG auction is

8 Later, we make an important simplifying assumption about the linearity of track values to avoid combinatorial complexity.

(1) $$\sum_{j \neq i} v_j(x^*_{-i}) - \sum_{j \neq i} v_j(x^*)$$

and the utility gain for bidder i as a consequence of the auction is

(2) $$u_i(x^*) = v_i(x^*) - \left[\sum_{j \neq i} v_j(x^*_{-i}) - \sum_{j \neq i} v_j(x^*) \right]$$

where

- x^* denotes the optimal allocation of a collection of goods to the participants in the auction.
- x^*_{-i} denotes the optimal allocation of the collection goods when bidder i is excluded from the auction.
- $v_i(x)$ denotes the value that bidder i places on the allocation.

The utility (the term on the left-hand side of the equation) in Equation (2) is the value associated with the allocation minus the payment (which is the term in brackets). The first term in the payment is the value of the allocation for an auction that excludes bidder i. The second term uses the optimal allocation when i is included in the auction. The second term is necessarily less than or equal to the first term, since the first term is based on the optimal allocation when i is excluded. This means that the total value to all bidders excluding bidder i decreases as a consequence of i's participation in the auction. This decrease in value can be viewed as the opportunity cost of i's participation. Hence i's utility gain from the auction is its value for the allocation minus the opportunity cost for its participation.

Another way to look at this is that bidder i's payoff (that is, utility gain) is the incremental value of bidder i's participation as shown by slightly rewriting Equation (2) as follows:

(3) $$u_i(x^*) = \sum_{i=1}^{n} v_j(x^*) - \sum_{j \neq i} v_j(x^*_{-i})$$

3.2.3 Proof of Incentive Compatibility

This payment scheme can be shown to incentivize bidders to reveal true values. Let's say that bidder i misrepresents its true valuation function. This could result in some other allocation, $\hat{x}$. Subtract the utility resulting from the false valuation from the utility with the true valuation.

(4) $u_i(x^*) - u_i(\hat{x}) = \sum_{j=1}^{n} v_j(x^*) - \sum_{j=1}^{n} v_j(\hat{x})$

Since the optimal allocation yields a sum of bidder values that is the highest, the difference in Equation (4) must be greater than or equal to zero. Therefore, the payoff resulting from an untruthful valuation is less than for the truthful valuation; "lying doesn't pay."

What is the intuition for VCG incentive compatibility? Why would bidder i lie? Lying does not change the bidder's true valuation function; lying can only change bidder i's allocation, which in turn can change the value it accrues due to a "better" allocation. However, bidder i's actual goal is to maximize its payoff—the valuation of bidder i's allocation is only part of the payoff. Increasing bidder i's valuation draws utility from other bidders, but we can see from Equation (3) that payoff depends on the sum of all bidders' valuations, not just on bidder i's valuation. When bidder i lies, its valuation might increase, but all other bidders' might decrease even more, thereby decreasing bidder i's payoff. The optimal allocation occurs when bidder i tells the truth.

3.2.4 Revisiting the Single-Item Auction

It helps to explain the single-item auction from the more general point of view of the multi-item auction.

Continue to assume that there are n participants, that x denotes an allocation of items to participants, and that x^* denotes the optimal allocation. Since, in the single-item case, only one participant is allocated an item, every possible allocation results in, at most, one participant having a positive valuation while the remaining participants have zero valuations. Therefore, the first term in expression (1) above has only one non-zero term, and the second summation is equal to zero. Assume that participant i makes the highest bid and participant m makes the second to the highest bid. Expression (1) simplifies to $v_m(x^*)$, which is the second highest bid.

This perspective shows that the single-item VCG payment is indeed a special case of the more general multi-item VCG payment.

4 “Mechanism Engineering”

By “*mechanism engineering*,” we mean the practical use of the science of mechanism design in the construction of an engineering—here a *software-intensive system* engineering—artifact. It is a phrase that captures the essence of what we are exploring. This section discusses some of the issues that one will face when developing a mechanism for use in a realistic complex system. Of course, our context is the sensor data fusion application described in Section 5.

4.1 RELEVANCE OF AUCTIONS TO SYSTEM DESIGN

One might ask what auctions have to do with systems design. An auction is an allocation of a desired resource to a collection of self-interested participants, where the allocation optimizes the total value (or utility) of the participants. Allocation of resources, both human and computational, is central to system design. The opportunities for intentionally and unintentionally conflicting self-interest increases as systems grow larger. Therefore, incentivizing behavior that can lead to optimal resource allocation when deception is possible and the level of global control and awareness is limited is important. Mechanisms (auctions, in this case) do that.

4.2 REQUIREMENTS FOR DESIGNING A MECHANISM FOR SENSOR FUSION

We are not designing a mechanism from scratch. We decided to use a variant of the VCG auction for several reasons: (1) we read about an example of its use for sensor fusion in work by both Rogers and Dang and (2) it is a well-known mechanism with well-known prosperities [Rogers 2006, Dang 2006]. When considering an auction, one must answer the following questions:

- Who are the participants and what are their incentives?
- What items (or resources) are being auctioned (or allocated)?
- What are the valuation functions?
- What are the desired properties of the social choice?

4.2.1 Participants

Each ship is a participating unit (PU). Each PU has a set of objects within its region of observation that it is tracking. Regions of observation overlap, and therefore more than one PU can track the same object. When PUs share data about common objects, their tracking accuracy increases, and therefore each PU’s data can be of value to other PUs.

4.2.2 Resources

The scarce resource is network bandwidth. The network is allocated as shown in Figure 7 below. In this example, there are four PUs; accordingly, one network cycle comprises four transmission periods, one period for each PU. If auctions are excluded, each PU in turn broadcasts its track data, that is, the tracks for which the PU has been assigned R^2. Network cycle time is defined as the time required to transmit track data from all PUs.

In tactical data networks such as the one emulated here, network *bandwidth* is expressed as tracks per network cycle time. Bandwidth is intimately related to latency of track updates. When setting up the auction, we choose a maximum latency (Max NCT). The difference between Max NCT and network cycle time is what is being auctioned.

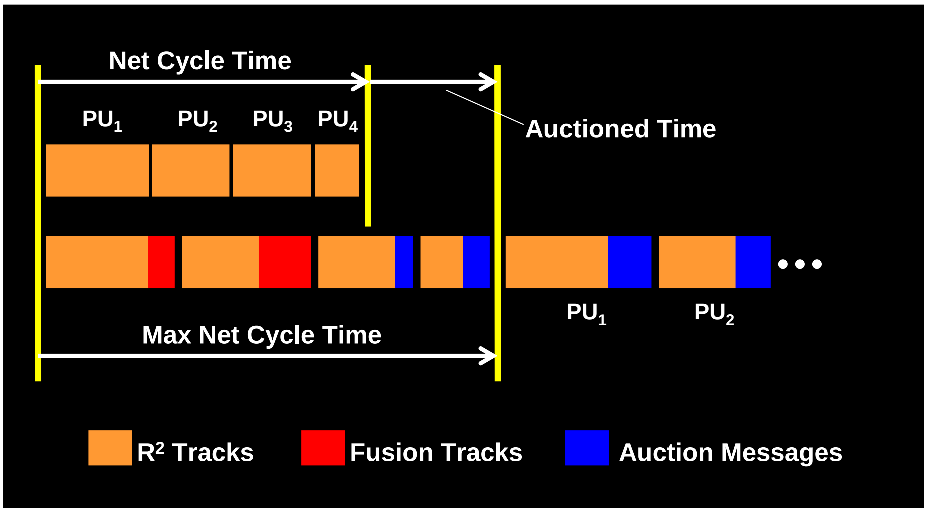

Figure 7: How Network Bandwidth Is Consumed

The auction mechanism selects the additional (non-R^2) tracks that will yield the most valuable improvements in the common operating picture. Bandwidth is also consumed by the auction protocol itself. Each auction requires three net cycles to complete. In our implementation, no non-R^2 tracks are transmitted during an auction, although this is not essential. Auctions are initiated periodically, by default once every 15 net cycles, but this can be adjusted. In Figure 7, an auction is being initiated by PU_3. It will continue for two additional cycles, after which each PU will transmit its R^2 tracks and any other tracks that have been selected as fusion tracks, until the next auction is initiated.

Figure 7 is slightly misleading in one sense. Our implementation does not include the cost of conducting an auction (the auction messages) in the allocation of additional bandwidth beyond the network cycle time. The Max NCT will be exceeded during the auction if the difference between the Max NCT and network cycle time is less than the auction overhead. This is not a flaw in the implementation, but rather a tradeoff. If auction intervals are sufficiently far apart, the transient overload is compensated by the value of the extra track data that can be sent.

4.2.3 Valuation

PUs are interested in acquiring and computing the most accurate information about tracked targets and in using this information to ensure their success in any engagement. However, PUs are also offered doctrinal-based incentives to contribute as best they can to the overall information accuracy of the battle group. A PU's mission success can be correlated to the overall information accuracy of the battle group, and "after action rewards" will be based on a PU's contribution to overall information accuracy.

The valuation function converts all the data available to a ship—data from other ships combined with its own data—into a measure of track data quality. Each PU tracks multiple objects. Each PU derives a total value based on the information it receives for all the objects in its region of observation (RO).

4.2.4 Social Choice Function

Mechanisms implement social choice functions. The one implemented by this auction is to produce an allocation that optimizes the sum of the valuation functions of all the PUs.

4.3 DESIGNING THE MECHANISM

We will make the following assumptions:

- A cyclic bandwidth allocation protocol is used. Network bandwidth is limited due to a maximum net cycle time (NCT). The maximum net cycle time (Max NCT) is determined by the maximum latency requirements associated with how old data is allowed to be.
- A broadcast transmission protocol is used.
- PUs are rational and therefore can be expected to act in their own best interest.

4.3.1 Auction Protocol

Conducting the auction entails the following steps:

1. The auctioneer initiates a new auction round by requesting that each PU send "non-R^2 track data." Each PU broadcasts, for each track in its RO for which it does not have R^2, the range and bearing covariance for that track. Each PU broadcasts this data at its next transmit opportunity (TO); therefore, this step consumes one network cycle.
2. The auctioneer requests that each PU send its track valuations for its "coveted tracks." Non-R^2 track data from Step 1 is coveted by a PU if the track is correlated to a track within the PU's RO. Tracks are valued according to the potential information gain that will follow from some later data fusion of this track data if it is chosen to be sent. In the current auction, information gain is expressed by the range and bearing covariance of track data. This step consumes one network cycle.
3. The auctioneer uses the track values from the PUs to determine the total information gain associated with each track if that track is sent to each coveting PU. The auctioneer then uses

a knapsack algorithm to allocate bandwidth to PUs to send the tracks that maximize total information gain, assuming

- PU information gain is a linear function, that is, for tracks A and B, IG({A,B}) = IG(A) + IG(B).
- Each track requires a constant amount of network bandwidth.

4. The auctioneer announces the "winning tracks," that is, the set of tracks selected by the knapsack algorithm in Step 3. The auctioneer also records the information gain of and payment made (in units of information gain) by each PU (see Equation (5)), which will be used as the basis for after-action rewards (explained in Section 4.3.3). Steps 3 and 4 together consume one network cycle. This completes the auction round initiated in Step 1.
5. Each PU broadcasts, at each transmit opportunity, its R^2 track data as well as any non-R^2 track data that have been identified as "winning tracks" in Step 4.

4.3.2 Optimization

Let $v_i(Z,F)$ denote PU i's information gain for the track data represented by the matrix Z and the bandwidth allocation represented by the matrix F. You can think[9] of Z as a matrix where Z_{ij} represents PU i's track data for track object j. Naturally, some of these entries will be null, since not all PUs have information about all tracks.

F denotes the amount of bandwidth dedicated to sending each track. To understand F, think of it as a matrix where the number of rows is equal to the number of PUs and the number of columns is equal to the number of track objects. F_{ij} represents the amount of network time devoted to PU i for broadcasting track j. Naturally, some of these entries will be zero, since not all tracks are seen by all PUs. Others are zero as a consequence of the allocation that results from the auction.

Our goal is to optimize the use of the additional network latency (defined in Section 4.2.2 as the difference between the Max NCT and NCT) to transmit additional track data. This additional latency represents the size of the "knapsack." We need to fill the knapsack by sending the appropriate track information in a way that maximizes the total information gain across all the PUs.

We now define the payoff and payment functions for this mechanism.

4.3.3 Incentives for the Mechanism

If the problem were strictly an optimization problem, we would be done. However, it is actually a problem of joint decision making amongst a collection of self-interested PUs that hold private information that affects the outcome. The goal of designing a mechanism is to incentivize behavior in a way that leads to a desirable outcome. For this auction, this means defining and designing the right payment structure.

[9] We say "you can think of" because we do not need to probe into the details of how Z and F are represented.

What does payment mean in this setting? In a traditional (non-computational) setting, participants value items being auctioned in terms of money and make bids in terms of money. The VCG auction incentivizes bidders to reveal the bidder's true value for the items being auctioned. Payment is made in terms of money. Payoff is then the difference between the value of the allocated item(s) (again, in terms of money) and the payment.

Also, recall that the incentive compatibility of the VCG auction arises because, from any participant's point of view, maximizing its payoff is equivalent to maximizing total value.

In our computational setting, information gain is the measure of utility. To create an incentive-compatible mechanism for our computational setting, we have to make sure that maximizing payoff is equivalent to maximizing total information gain. This equivalence reflects an important aspect of "mechanism engineering." Computational participants inherit their "computational incentives" from humans. This notion is captured nicely by Rosenschein and Zlotkin in *Rules of Encounter* [Rosenschein 1994]:

> *"We are interested in social engineering for machines. We want to understand the kinds of negotiation protocols, and punitive and incentive mechanisms, that would cause individual designers to build machines that act in particular ways. Since we assume that the agents' designers are basically interested in their own goals, we want to find interaction techniques that are 'stable', that make it worthwhile for the agent designer not to have machine deviate from the target behavior."*

The structure of the auction has to incentivize humans to design the agents to behave in an incentive-compatible manner. Therefore, mechanism engineering not only involves designing computational protocols, but also involves designing and/or changing social institutions, understanding the relationship between such social institutions and the designers of computational protocols, and understanding the behavior of designed computational protocols.

In our setting, the agent designers have two driving incentives, which are based on the doctrine of the armed services:

- Survivability of the individual PU depends on the survivability of the battle group, which in turn depends on maximizing the information gain of the whole group. This logic incentivizes every PU to maximize their contribution to the group's information gain. Increasing one's own information gain is not always the best route to one's own survival.
- "After-action reviews," which lead to promotions and other rewards, use contribution to total information gain as an important evaluation criterion. In effect, the auction also acts as an accounting device that records the marginal contribution of each PU to the group's information gain. Payment occurs outside of the auction mechanism itself.

These incentives are consistent with viewing information gain as our measure of utility in this setting. Each PU tries to both increase its own information gain and minimize its effect on the group's information loss. This is consistent with a traditional VCG auction in which all individuals try to increase their own value by bidding on items they greatly value but at the same time reducing their payment, which is the total value loss associated with their participation in the auction.

4.3.4 Payoff and Payment for the Mechanism

Equation (5) represents the payment made by ship i to the auctioneer.

$$(5)\quad \sum_{j\neq i} v_j(Z, F^*_{-i}) - \sum_{j\neq i} v_j(Z, F^*)$$

Equation (5) represents the lost opportunity for information gain due to PU i's participation in the auction. In other words, PU i's participation cost the other participants this much information gain—this is a payment in terms of information gain since PU i's "account" is, in effect, debited this amount for its effect on the total information gain.

The first term in the expression represents the total information gain due to an optimal allocation when ship i **does not participate** in the auction (F_{-i}*). The second term in the expression represents the total information gain minus PU i's contribution due to an optimal allocation when ship i **does participate (F*).**

There is an important subtlety in this situation that was not true for the traditional VCG auction. Expression (5) might be negative. In other words, PUs might make negative payments.

To see this subtlety, note that PU i's presence in the auction contributes to the total information gain by

- broadcasting its track data to the other ships and
- receiving track data from other ships. This is accounted for in v_i(Z, F).

The first form of contribution is reflected in the second term of expression (5). If this contribution is large enough, it can cause the payment to be negative. This can be interpreted as an addition to PU i's account for sending valuable track data to the other PUs. If we only consider the value of ship i receiving data, the second term would have to be less than or equal to the first term, as we discussed earlier.

The second form of contribution is not reflected at all in the payment. Rather, it is reflected in the first term of the payoff, which is

$$(6)\quad u_i(Z, F^*) = v_i(Z, F^*) - \left[\sum_{j\neq i} v_j(Z, F^*_{-i}) - \sum_{j\neq i} v_j(Z, F^*)\right]$$

4.3.5 Incentive Compatibility

A mechanism is incentive compatible when directly revealing the truth for all participants is an equilibrium. (That is, truthfully revealing the participants' preferences is an equilibrium strategy profile.) We use exactly the same argument as the traditional VCG incentive-compatibility argument shown above to show that this mechanism is incentive compatible.

The difference between this mechanism and a "pure VCG auction" is that each participant stands to benefit from both receiving and sending track data. The traditional VCG is usually applied to auctions (where participants are buyers) or to reverse auctions (where participants are sellers), but not when participants are both buyers and sellers.

Hence, in principle,[10] there are two opportunities to lie. A PU can lie about its value function and/or can lie about the quality of its track data. Lying about the value function might cause more track data to be sent to a PU. Lying about track quality might make it seem more desirable for the PU to send its track data to others. Neither of these "incentives" is really an incentive. (Note that in our current implementation, it is impossible for a PU to lie about its value function. A PU's information gain only depends on the track data that it receives, not on any private information.)

Let Equation (7) below, which is simply a rewriting of Equation (6), represent the payoff to PU i as a consequence of participating in the auction when it tells the truth. Z denotes the true track data, and F^* denotes the optimal allocation, that is, the allocation of bandwidth that results in the highest overall information gain.

$$(7)\quad u_i(Z,F^*) = \sum_{i=1}^{n} v_j(Z,F^*) - \sum_{j \neq i} v_j(Z,F^*_{-i})$$

PU i's lying about valuation or track data cannot affect the second term in the above equation, since this allocation assumes that PU i is not participating and is therefore also neither sending nor receiving track data. PU i's lying about valuation or track data can result in an allocation other than F^* shown in the first term in the above equation. When PU i lies, the allocation changes but the valuation is still determined by the true valuation functions and the actual track data that is sent. Since F^* results in the highest total valuation, any other allocation must result in a total valuation that is less than or equal to the total valuation resulting from F^* and consequently a payoff to PU i that is less than or equal to the payoff resulting from F^*. Therefore, it is not rational for any PU to lie—at least from the point of view of increasing the PU's payoff.

10 Later, we will explain that in our implementation, presented here, there really is only one opportunity to lie, not two. A more general implementation would provide two opportunities, and therefore we will assume this in our proof.

4.3.6 Optimal Bandwidth Allocation

While participants are involved in independent, distributed decision making, the auctioneer must perform an optimization computation based on the participants' reported values. This computation can be done by interpreting the bandwidth optimization problem as a knapsack problem and using a 0-1 knapsack algorithm.

The knapsack problem assumes that you have a collection of items that need to be carried in a knapsack. Each item has a weight and a value. The carrier of the knapsack establishes a maximum weight limit for the knapsack. The problem is to maximize the total value of the knapsack contents while adhering to the weight limit.

In our problem, the bandwidth for PU i to send track data for track j, F_{ij} is the weight. The information gain to the system due to PU i sending it track data is the value. The algorithm requires an integer weight; therefore, w_i = ceiling(100*f_i) and the weight limit is 100.

Let F[i,w] denote the maximum value of a knapsack when considering items 1 through i and a maximum weight of w where i varies from 0 to N and w varies from 0 to Wmax. The recursive expression for the algorithm is

- $F[i, w] = F[i-1, w]$ if $w > Wmax$
- $F[i, w] = \max(F[i-1,w-w_i] + v_i, F[i-1,w])$ if $w \leq Wmax$

A "C" implementation of the 0-1 knapsack with pseudo-polynomial runtime complexity is provided in Appendix A.

5 Exploring the Mechanism

The previous section discussed various pragmatic considerations of mechanism engineering, at least as those pragmatics were exposed by our choices of application area and mechanism. In this section, we discuss our use of the application framework, briefly introduced in Section 2, to study the behavior of the mechanism in a demanding setting.

5.1 USING THE APPLICATION FRAMEWORK TO STUDY THE MECHANISM

At the core of the emulated tactical data network is a track data generator. The track generator provides the raw track data that is then consumed by each PU. Each PU then adjusts the raw track data to reflect the range and bearing error that is defined for the radar apparatus hosted by that PU, which may vary from PU to PU.

Figure 8 shows the main track display for a scenario with four PUs, prior to gridlock and correlation. The total set of tracks generated includes both those that are displayed in white icons and those displayed in grey. Those in white are visible to at least one PU, while those in grey are not visible. A track will not be visible if it is outside of the region of observation of all PUs. Also, a track will not be visible if it is within the region of observation of a PU but it falls beneath the tangent plane or if it is too close to the PU given the pulse repetition frequency (PRF) of the radar. Track data is refreshed every four seconds, which corresponds to the scan rate of SPS-48C radar.[11]

[11] For more information, see https://wrc.navair-rdte.navy.mil/warfighter_enc/weapons/SensElec/RADAR/sps48.htm. This system is pronounced “forty-eight Charlie” by aficionados.

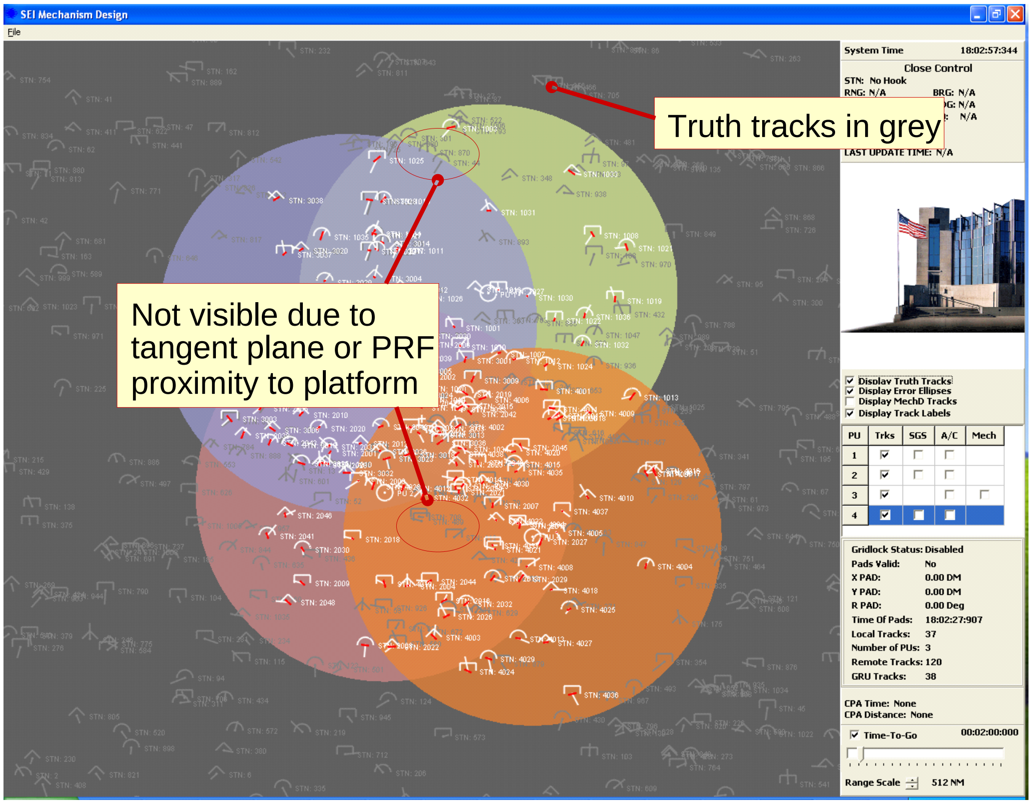

Figure 8: Uncorrelated Tracks with Truth Tracks Revealed

The point to observe in
Figure 8 is that there is significant dynamism in the application framework (many tracks, realistic refresh rate). Figure 9 shows features of the framework that can be used to vary the characteristics of scenarios. Beginning with “Link characteristics” in Figure 9 and working counterclockwise

- The capacity of the TADIL can be varied from 5,000-100,000 bits per second. Greater capacity results in shorter network cycle time.
- Auction frequency can be varied from 1 to 50 cycles, where N means conduct an auction every N cycles. Since each auction requires three cycles to complete, a value greater than or equal to three results in continuous auctions being conducted.
- The number of PUs can vary from 1 to 12. However, if there are fewer than two PUs, there are no opportunities for data fusion (or gridlock or correlation), and additional PUs are easily accommodated.
- The NCT that is required to transmit all R^2 data (the “Base Load NCT”) will depend on both the selected capacity of the TADIL and the specific configuration of tracks produced from the track generator.
- The maximum NCT can be varied from 0 to 20 seconds and determines how much bandwidth is allocated (Max NCT – Base Load NCT) to transmitting additional tracks to improve the COP.

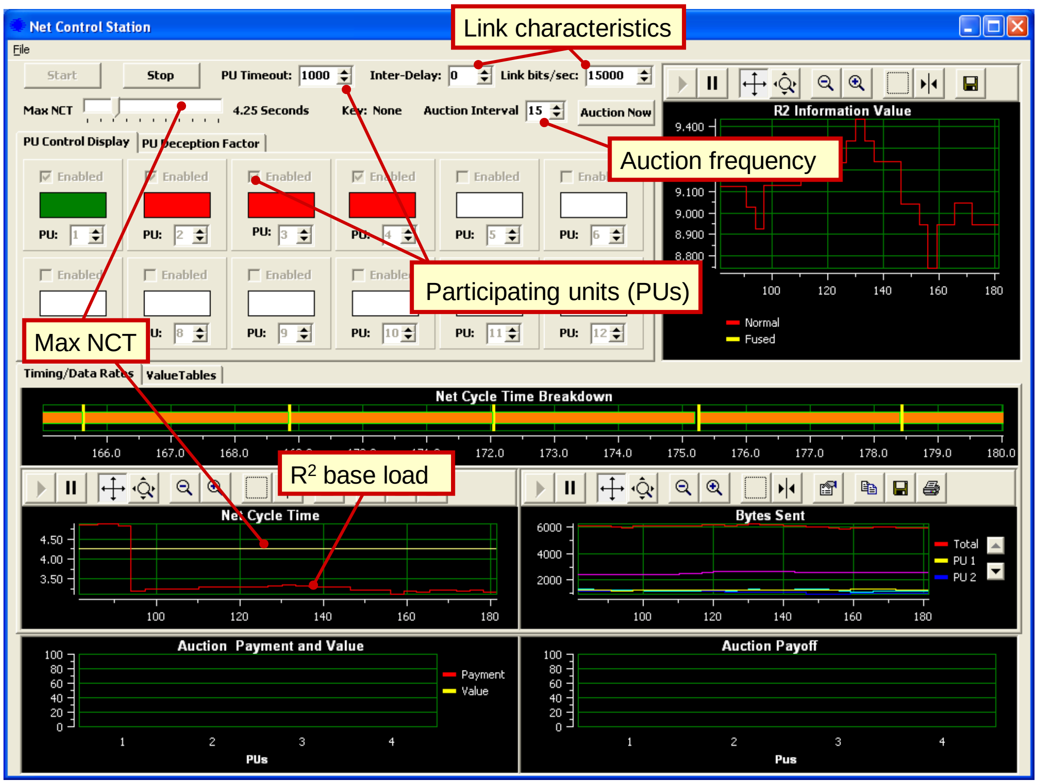

Figure 9: Framework Features to Study Mechanism Performance

When auctions have been enabled (see Figure 4, to the right of the SGS gridlock and A/C correlation checkboxes), an auction is conducted at the selected auction interval (by default, every 15 cycles). Figure 10 shows the features of the application framework that allow us to study the behavior of the mechanism within different scenarios. Examining the highlights from the top down and left to right

- The NCT is broken down into three constituents: (1) for the baseline R^2 reporting, (2) for the overhead of conducting the auction, and (3) for transmitting the highest value tracks identified by the auction.
- The target maximum NCT (horizontal yellow line) and actual net cycle times are displayed. Note the transient phases where the maximum NCT is exceeded, as permitted by this mechanism (see the discussion of the alternative definitions of spare bandwidth in Section 4.3). For example, see "Transient overrun" on the NCT, where the red "transmit" line crosses the yellow "Max NCT."
- The payment (Equation (5), pg. 24) and utility (information gain) for each auction round are displayed on the lower left, and the overall payoff (Equation (7), pg. 25) for each PU is displayed in the lower right. Note the "negative payment" for PU 2 in the figure—a possibility that is discussed in Section 4.3.

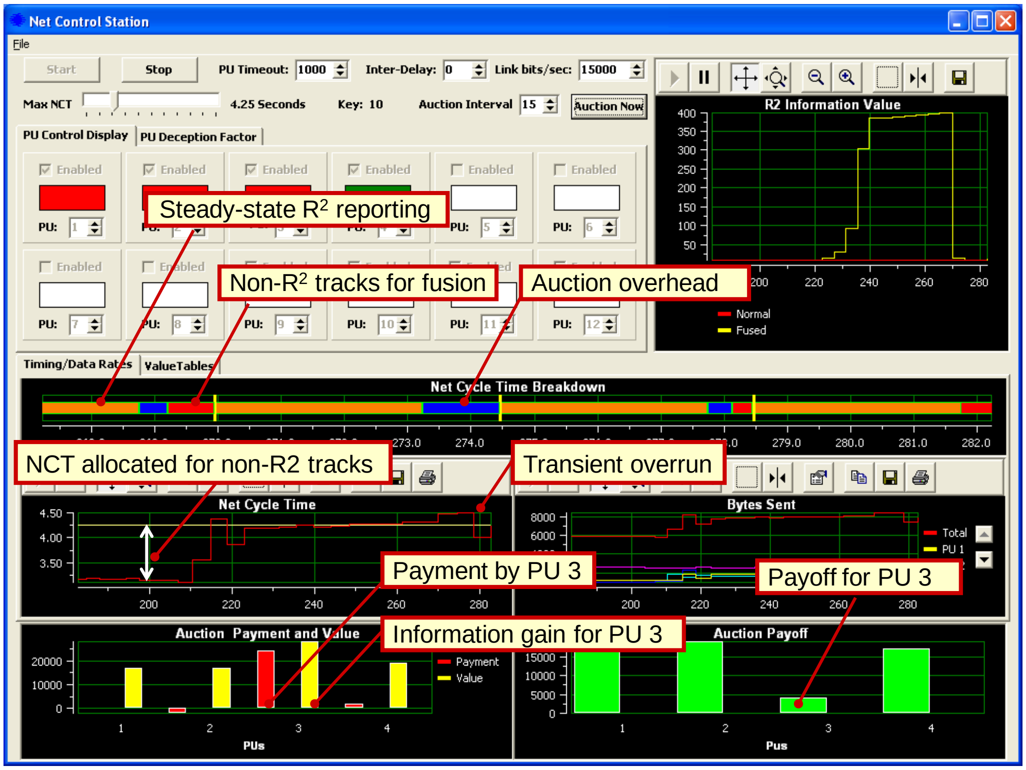

Figure 10: Framework Features to Study Auction Payoff and Efficiency

One interesting aspect of the auction is its effect on the platform that serves as a GRU (grid reference unit). As noted in Section 2, the GRU (in this scenario, PU 3 plays the role of GRU) is typically assigned to the platform with the most capable radar. Because of this, the GRU assumes R^2 for all tracks within its range of observation.

Within the auction, however, the GRU has already published all its track data, and hence it cannot add further value by sharing additional track data. This is indicated by the relatively low (to other PUs) payoff for PU 3 and by the high payment it makes by virtue of being a net recipient of non-R^2 track data from other PUs. The asymmetric role of the GRU and its impact on the auction mechanism produced a number of interesting real-world seemingly anomalous (but yet, correct) behavior that required close study to understand.

A formal argument of "incentive compatibility" for the mechanism we implemented was provided in Section 4.3. The upshot of this argument is that any PU that might otherwise be disposed to lie about its track quality for the purpose of improving its payoff would be dissuaded from doing so, assuming, of course, that the PU is "rational." Nonetheless, it is interesting and useful to study the behavior of the auction mechanism when one or more PUs lie about their track quality, notwithstanding the formal argument.

We studied the behavior of the mechanism when PUs lie so that we could observe the stability of the mechanism when the underlying assumptions of purely rational PU behavior have been violated. As we later show, there were some unanticipated, and positive, results from undertaking this additional aspect of the investigation.

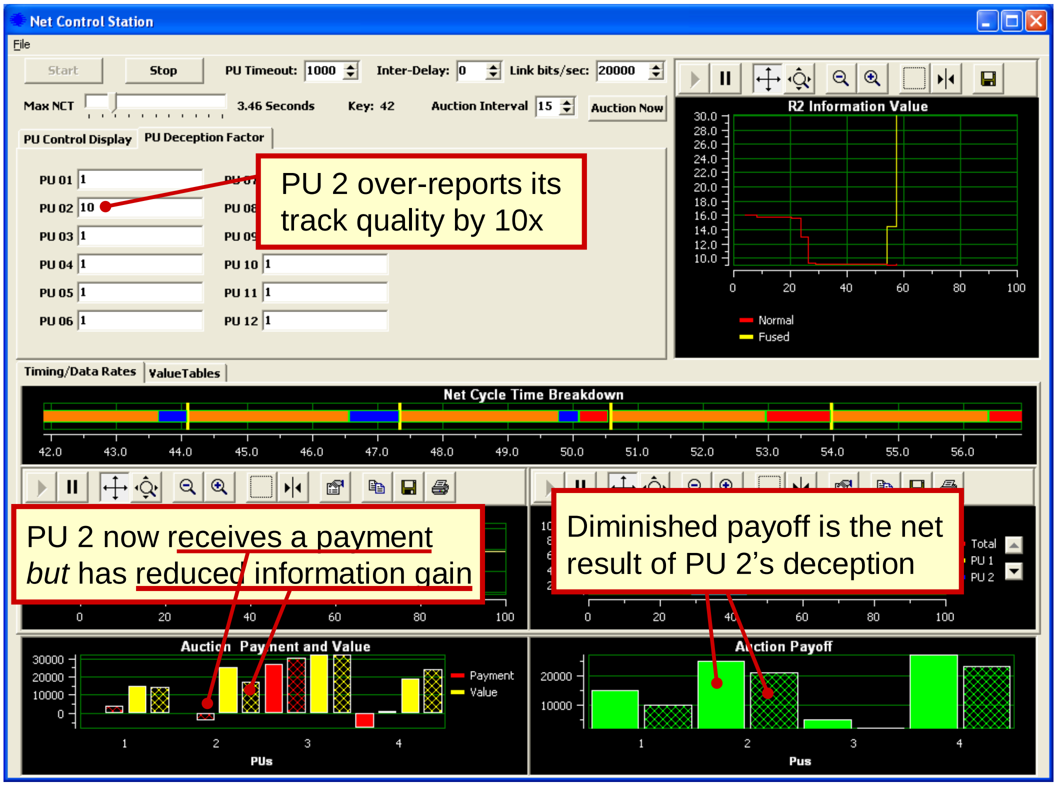

Figure 11: Adverse Effects of Deceptive Bidding in the Auction

The application framework supports the investigation of deceptive bidding by allowing any PU to over- or underestimate the quality of *all* its track data by a *constant* deception factor. The presence of a lie can be detected by the framework but is not directly visible to other PUs, of course. If the framework detects *any* deception, it runs two auctions—one auction assuming that all PUs are truthful about their track quality and one auction with any deceptive reports of track quality.

The "Auction Payment and Value" panel, located in the lower left of Figure 11, displays the *payment* and *information gain* of each PU in both the lying and truthful auctions, that is, each PU has four bars in this bar graph. The results from the lying auction are displayed with crosshatching, while results from the truthful auction are displayed in solid colors. In Figure 11 (the meaning of which is discussed in the next paragraph), PU 4 makes a negative payment in the truthful auction (i.e., receives a payment), while it makes a positive payment in the lying auction. PU 1, on the other hand, makes no payment in the truthful auction (it has no solid red bar). The "Auction Payoff" in the lower left of Figure 11 displays lying and truthful payoff, as discussed in Equation (5).

Figure 11 is a snapshot of a scenario in which PU 2 lies about its track quality by over-reporting the quality of its track data by a factor of 10. This lie should have the effect of making this PU's track data more appealing to other PUs, and therefore we would expect the result of an auction to be that more network bandwidth will be allocated to PU 2 track data, which will lessen PU 2's payment. In the truthful auction, PU 2 makes no (and receives no) payment; in the deceptive auction, it made a negative payment, which is the equivalent of receiving a payment. On the other hand, since more bandwidth is allocated to track data already in PU 2's possession, it might obtain a diminished information gain as a result of the auction. Again, this expectation is confirmed in the scenario in Figure 11.

However, the important point to observe is that PU 2's *payoff*, which is the difference between information gain and payment, is reduced in the lying auction, which suggests that the loss of information gain by PU 2 was not offset by the gain in payment. This matters because the argument for incentive compatibility refers to payoff. Either the payment or information gain, but never both, might be enhanced by deception.

As expected, then, PU 2 was worse off in the lying scenario than it would have been in the truthful scenario. Here, at least, is one empirical validation of incentive compatibility.

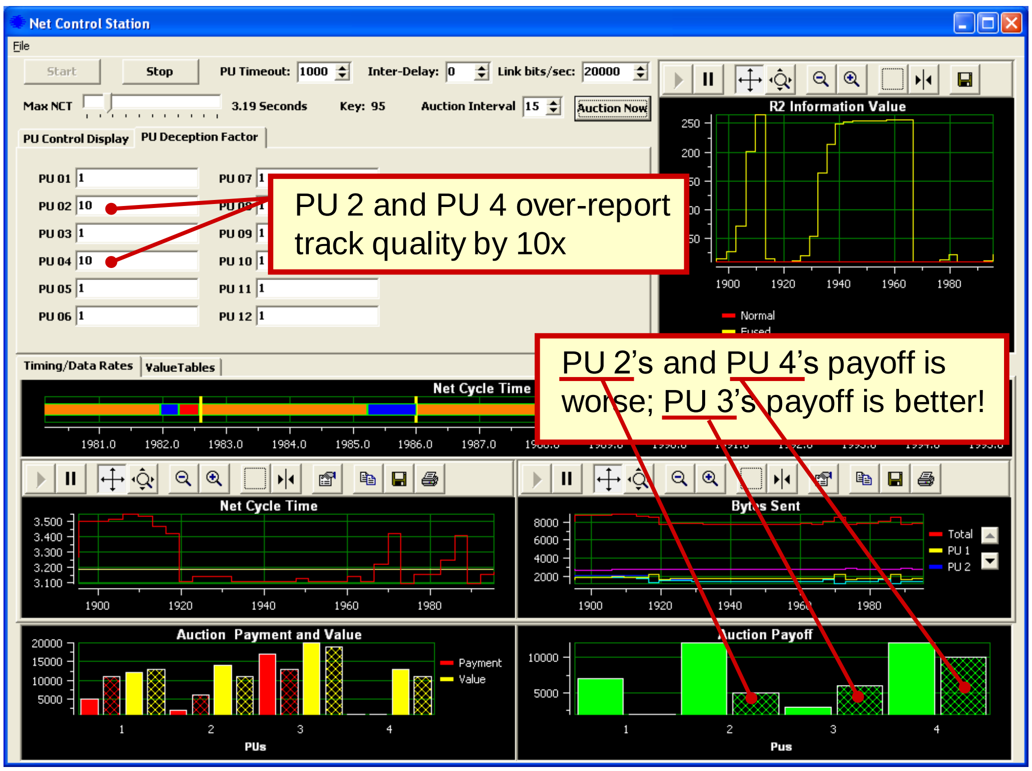

Figure 12: Potential for Bidder Collusion in VCG Auctions

Our study of deception also produced results that were, on first examination, entirely unexpected. For example, consider the snapshot in Figure 12 where two PUs are deceptive. Per incentive compatibility, both were expected to obtain worse payoffs than if they had been truthful, and indeed this is what occurs. However, PU 3's payoff is *improved*. Is this a problem?

It turns out that the outcome of the Figure 12 scenario does *not* violate incentive compatibility, since both deceptive PUs were worse off. However, the scenario *does* demonstrate the well-documented susceptibility of the VCG mechanism to *bidder collusion.*[12]

Bidder collusion is a form of strategic manipulation where two (or more) PUs may arrange a deception so that their joint outcomes are better than they would be in a truthful auction (for example, where a lying PU obtains a worse payoff while its truthful confederates receive an improved payoff). We might imagine[13] the situation depicted in Figure 12 as engaging in collusion: the two lying PUs 2 and 4, and the truthful PU 3.

Auctions that are resistant to bidder collusion are well documented; Sandholm recommends the use of first-price auctions as an alternative if collusion is a primary concern and possibly other remedies, but the investigation of coalition-proofing was beyond the scope of this initial study [Sandholm 1996].

Another form of strategic manipulation that we observed in the framework is sometimes referred to as *spiteful bidding*. Spiteful bidding can be modeled where a bidder's payoff is the difference between the winning bidder's utility and the utility of the spiteful bidder. For example, a PU might lie about its track quality, and therefore accept a diminished payoff, if the payoff of some other truthful PU is also diminished more severely than the payoff of the lying PU. Again, the susceptibility of the VCG is well documented [Brandt 2007]. As with collusion, remedies are also well documented but beyond the scope of this study.

It is worth noting that we did not guide the application framework in any way to expose these known vulnerabilities of the VCG auction. Doing so would have required considerable effort to develop PU bidding strategies that could "speculate" on the global state of the system.

5.2 OBSERVATIONS AND RESULTS FROM THE MECHANISM IN ACTION

The previous discussion provided a brief overview of both the research application framework used to study an auction mechanism for tactical data networks and the behavior of a particular mechanism in that framework. Our empirical studies using the framework were by no means exhaustive, but a few observations are worth emphasizing:

- The application framework has sufficient dynamism and scale to demonstrate, and more importantly to study, the behavior of computational mechanisms.
- The mechanism we implemented can operate in a performance-critical application, both in terms of the computational complexity of "winner determination" (the pseudo-polynomial 0-1 knapsack) and network overhead.

---

12 Note that the information gain for the entire system is diminished in the case of any deception, though this is not displayed.

13 There was no intentional collusion.

- The mechanism we implemented exhibits known vulnerabilities to strategic manipulation, but these particular vulnerabilities are not likely to be exploited in the operational context of a tactical data network operating under one flag. This would not be the case in coalition (multi-flag) settings, which would necessitate changes in the mechanism.
- The implementation of the mechanism itself was straightforward and, all things considered, quite compact. On the other hand, there are many variants of the mechanism, each of which may address, or introduce, subtle effects.

Admittedly, we have barely scratched the surface of the potential uses of mechanism design in tactical data networks. Nevertheless, the empirical aspects of the study suggest (to us) that computational mechanisms can be used in demanding settings such as tactical data networks. However, our experience also suggests that what might at first appear to be only slight variations in the application setting—for example the use of point-to-point rather than broadcast communication—might require significant changes to the mechanism. There are also other classes of mechanisms—for example bilateral exchange markets—that may be more suitable to the particulars of this tactical data network or others.

Having only scratched the surface, we only outline those areas for exploration that build most directly on our existing prototype.

5.3 AREAS FOR FUTURE EXPLORATION

Inserting a VCG auction into a realistic sensor data fusion problem has provided us with a rich environment for exploring mechanism design. The following four areas of findings and future research represent just a sampling of the explorable issues of mechanism engineering. We are confident that other issues will also surface as we think further about the practical and theoretical aspects of moving from mechanism design to mechanism engineering. The four areas are

- dynamic environments
- agent externalities
- exploring alternative mechanisms
- preference elicitation, currency, and budget constraints

5.3.1 Dynamic Environments

The platforms in our mechanism research application represent a dynamic environment changing from cycle to cycle. To date, we have only treated the problem as a sequence of static problems.

The case in which more than one network cycle needs to be considered when making decisions needs to be explored. This situation might arise when there is a need to take sensor readings for more than one network cycle or there is some other need to plan across multiple time periods. It could also arise if the readings from a past cycle are good enough approximations for the current cycle, thus obviating the need to spend network bandwidth resending data.

5.3.2 Agent Externalities

Our allocation problem exhibits *externalities*—that is, the valuation of each platform for an allocation of bandwidth depends on the details of the sensing and data fusion actions that will be performed by the other platforms. The information gain "credits" accorded to one platform depend on the number of other platforms that value its data. In an ordinary VCG auction, allocative externalities do not exist. Individuals only care about their valuation of the resources that they are personally allocated.

Another imposed externality is the decision about the network cycle time: this is a single decision that affects all participants and cannot be set separately for each participant. This situation presents us with a classic engineering tradeoff between timeliness and information gain. On one side, the value of information can decrease as latency increases, that is, as the NCT increases. On the other side, as NCT increases, more participants can share more information, which contributes to overall information gain. Methodical exploration of the tradeoff space will be an important aspect of moving from mechanism design to mechanism engineering.

5.3.3 Exploring Alternative Mechanisms

Another view of our problem is that it is a problem of economic exchange, in the sense that each platform is interested in "buying" information from other platforms and also capable of "selling" information to other platforms. While a VCG mechanism can be used in an exchange environment, perhaps other mechanisms such as the following work just as well or better:

- budget-balanced, approximately efficient truthful [Babaioff 2005, Gonen 2007] and approximately truthful [Parkes 2001] exchange mechanisms
- methods to redistribute payments back to participants [Cavallo 2006]
- auctions with distributed auctioneers
- broader market-based [Wellman 1993, Parkes 2004] and negotiation protocols [Fatima 2007]

5.3.4 Preference Elicitation, Currency, and Budget Constraints

Incentives and decision preferences in computational agents are ultimately derived from human agents. Therefore, a very important part of mechanism engineering is to understand the contextual social institutions, appropriately elicit human preferences, and ensure that they are properly codified in the computational mechanisms.

For example, in our application, it was important to decide on the incentives of each platform and the social choice criterion, and to make sure that they were consistent with the VCG auction. To ensure consistency, each platform is ultimately rewarded for its contribution to the group's information gain rather than its own information gain. However, in the scenario adopted for this report, this transaction occurs outside of the mechanism; the mechanism maintains accounting records. There is motivation for further exploring the whole area of virtual currency to make the transaction part of the mechanism proper and to understand the practical issues of using virtual currency to incentivize individuals and organizations.

Another related factor that affects this mechanism is the designation of which platform holds reporting responsibility. This designation has a strong influence on which tracks are eligible to be sent. We were able to construct cases in which the platform holding reporting responsibility would receive a negative payoff. This was a result of not considering the implicit information gain contributed by that platform. This "additional" information gain is implicit in the sense that it did not require any additional bandwidth to be allocated through the auction. Rather, it changed the preconditions of the auction. The situation is an example of an interesting interaction between the auction mechanism and the specifics of this domain, which need to be methodically identified and considered when carrying out mechanism engineering.

Currently, information gain is the only "measure of merit" that we consider. In a sense, it serves as the virtual currency for our application. It is not hard to envision other measures of merit that are important, such as relevance. For example, some tracks, such as enemy tracks, might take tracking precedence. Adding this factor could introduce tension between two sources of incentives: relevance and information gain, which deserve further exploration.

6 Conclusions

Our research provides strong evidence that

- Computational mechanism *design* provides new and useful design principles for the design of complex systems, especially those that support users who may have distinct incentives.
- Computational *mechanisms* can be used in performance-critical, highly dynamic settings such as those found in tactical data networks, with behavior that is predictable using strong underlying game and microeconomic theory.

These results, we believe, are applicable to a much broader class of system than tactical data networks; in fact, the research is primarily intended to study mechanism design and only secondarily to study its use in a particular setting.

Nonetheless, the research also demonstrates that the well-known VCG auction can be useful in existing DoD tactical data networks as a tool for providing incremental improvements in the quality of a common operating picture. In addition, we have identified avenues for further refining the VCG auction and for using market mechanisms to embrace different tactical data network settings.

Finally, we have provided the research community with a robust application framework for studying computational mechanisms. This sort of framework—and others like it—will be enormously useful to close the gap between the sometimes alien research traditions of game theory and microeconomics and the practical requirements of software and systems engineering of complex systems.

Appendix A: Acronyms

A/C	auto-correlation
CEC	Cooperative Engagement Capability
COP	common operational picture
DoD	U.S. Department of Defense
GRU	grid reference unit
NCS	network control station
NCT	network cycle time
PRF	pulse repetition frequency
PU	participating unit
R^2	reporting responsibility
RO	region of observation
SGS	shipboard gridlock system
SGS/AC	shipboard gridlock system/auto-correlation
SIAP	Single Integrated Air Picture
TADIL	tactical data information link
TO	transmit opportunity
VCG	Vickrey-Clarke-Groves (auction mechanism)

Appendix B "C" Implementation of 0-1 Knapsack

```
// knapsack.h
#ifndef _KNAPSACK_H_
#define _KNAPSACK_H_
/*
 *     Adaptation of work by Dr. Steve Goddard (goddard@cse.unl.edu).
 *     See http://www.cs.unl.edu/~goddard/Courses/CSCE310J.
*/
#include "dlist.h"   // just a list abstraction, defines TDlist
typedef void *TItem; // things you put in the knapsack
// these callbacks must return -1 on error and >= 0 on success
typedef long (*TWeight)(TItem, void *);    // item weight
typedef double (*TBenefit)(TItem, void *);// item benefit
/*
 *     knapsack -    find a solution that puts the most items in the
 *                   knapsack and maximizes the total benefit
 *     return   -    0 on success, -1 on error.
 *                   If success:
 *                    -  solutionWeight, weight of knapsack
                      *  solutionBenefit, benefit of solution
 *                    -  solution, TDlist list of items in knapsack
*/
int knapsack(
       TItem *items,               // an array of items
       long numItems,              // number of items in array
       long maxWeight,             // knapsack capacity, 0..maxWeight
       TWeight getWeight,          // callback to get item weight
       void  *getWeightData,       // callback data
       TBenefit getBenefit,        // callback to get item benefit
       void  *getBenefitData,      // callback data
       long *solutionWeight,       // solution weight (return data)
       double *solutionBenefit,    // solution benefit (return data)
       TDlist solution);           // solution list (return data)
#endif
```

```
// knapsack.c
#include <stdlib.h>
#include <error.h>
#include "knapsack.h"
int knapsack(
       TItem *items,               // an array of items
       long numItems,              // number of items in array
       long maxWeight,             // knapsack capacity, 0..maxWeight
       TWeight getWeight,          // callback to get item weight
       void  *getWeightData,       // callback data
       TBenefit getBenefit,        // callback to get item benefit
       void  *getBenefitData,      // callback data
       long *solutionWeight,       // solution weight (return data)
       double *solutionBenefit,    // solution benefit (return data)
       TDlist solution);           // solution list (return data)
{
       double **B;                 // knapsack B matches published algorithm
       long i, k;                  // ranges through items
       long w;                     // ranges through weights
       long wi;                    // ith weight
       double bi;                  // ith benefit
       long n = numItems;          // n, W match published algorithm
       long W = maxWeight;
       long solWeight = 0;         // temporaries to compute the solution
       double solBenefit = 0;
       *solutionWeight = 0;
       *solutionBenefit = 0;
       // allocate n + 1 so array can be indexed from 1..n
       // with 0 used as the base case 0 solution
       B = (double **)calloc( (n + 1), sizeof(double *) );
       if (B == NULL) { // a serious error occurred
              perror("B[#items]");
              return -1;
       }
       // allocate W + 1 so array can be indexed from 0 to W
       for (i = 0; i < n + 1; i++) {
              B[i] = (double *) calloc(W + 1, sizeof (double));
              if (B[i] == NULL) { // a serious error occurred
                     perror("B[][maxweight]");
                     return -1;
              }
       }
       // the knapsack algorithm
       // items range from 1..n
       // weights range from 0 to W
       for (i = 1; i <= n; i++) {
              for (w = 0; w <= W; w++) {
                     wi = getWeight(items[i-1], getWeightData);
                     bi = getBenefit(items[i-1], getBenefitData);
                     if (wi < 0 || bi < 0) {
                            return -1; // bad error; could free B here
                     }
                     if (wi <= w) { // item i can be part of the solution
                            if (bi + B[i-1][w-wi] > B[i-1][w]) {
                                   B[i][w] = bi + B[i-1][w-wi];
                            }
                            else {
                                   B[i][w] = B[i-1][w];
                            }
                     }
                     else { // wi > w
```

```
                        B[i][w] = B[i-1][w];
                }
            }
        }
        // construct a list of the items in the knapsack
        i = n; // i, k are used to match published algorithm
        k = W;
        while (i > 0) {
            if (B[i][k] != B[i-1][k]) {
                if (solution != NULL) {
                    if (Dlist_insert(solution, items[i-1])==NULL){
                        return -1; // bad error; could free B
                    }
                }
                if (getWeight(items[i-1], getWeightData) < 0 ||
                    getBenefit (items[i-1], getBenefitData) < 0) {
                    return -1; // bad error; could free B here
                }
                solWeight += getWeight(items[i-1], getWeightData);
                solBenefit += getBenefit(items[i-1], getBenefitData);
                k = k - getWeight(items[i-1], getWeightData);
                i = i-1;
            }
            else {
                i = i-1;
            }
        }
        *solutionWeight = solWeight;
        *solutionBenefit = solBenefit;
        for (i = 0; i < n + 1; i++) {       // cleanup
            free(B[i]);
        }
        free(B);
        return 0;
}
```

Bibliography

[Anderson 2005]
Anderson, Edward; Kelly, Frank; & Steinberg, Richard. "A Contract and Balancing Mechanism for Sharing Capacity in a Communication Network." *Computing and Markets, Dagstuhl Seminar Proceedings.* Schloss Dagstuhl, Germany, March-July 2005. Lehman, Muller & Sandholm (eds). Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), 2005.

[Anshelevich 2004]
Anshelevich, Elliot; Dasgupta, Anirban; Kleinberg, Jon; Tardos, Eva; Wexler, Tom; & Roughgarden, Tim. "The Price of Stability for Network Design with Fair Cost Allocation," 295-304. *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04).* Rome, Italy, October 2004. IEEE Computer Society, 2004.

[Babaioff 2005]
Babaioff, Moshe & Walsh, William E. "Incentive-Compatible, Budget-Balanced, Yet Highly Efficient Auctions for Supply Chain Formation." *Decision Support Systems 39*, 1 (2005): 123-149.

[Brandt 2007]
Brandt, F.; Sandholm, T.; & Shoham,Yoav. "Spiteful Bidding in Sealed-Bid Auctions," 1207-1214. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 07)*. Hyderabad, India, January 2007. International Joint Conferences on Artificial Intelligence, 2007.

[Cavallo 2006]
Cavallo, Ruggiero. "Optimal Decision-Making with Minimal Waste: Strategyproof Redistribution of VCG Payments," 882-889. *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS'06)*. Hakodate, Japan, May 2006. Association for Computing Machinery, 2006.

[Chen 2004]
Chen, Ming; Yang, Guangwen; & Liu, Xuezheng. "Gridmarket: A Practical, Efficient Market Balancing Resource for Grid and P2P Computing," 612-619. *Grid and Cooperative Computing: Second International Workshop, GCC 2003. Lecture Notes in Computer Science,* Volume 3033. Shanghai, China, December 2003. Springer, 2004.

[Chen 2005]
Chen, Rachel; Roundy, Obin; Zhang, Rachel; & Janakiraman, Ganesh. "Efficient Auction Mechanisms for Supply Chain Procurement." *Source Management Science 51*, 3 (March 2005): 467-482.

[Dang 2006]
Dang, V. D.; Dash, R. K.; Rogers, A.; & Jennings, N. R. "Overlapping Coalition Formation for Efficient Data Fusion in Multi-Sensor Networks," 635-640. *Proceedings of the National Conference on Artificial Intelligence 21*. Boston, Mass., July 2006. Association for the Advancement of Artificial Intelligence, 2006.

[Edelman 2007]
Edelman, Benjamin; Ostrovsky, Michael; & Schwarz, Michael. "Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords." *Journal of American Economic Review 97*, 1 (March 2007): 242-259.

[Fatima 2007]
Fatima, S. S.; Wooldridge, M.; & Jennings, N. R. "Approximate and Online Multi-Issue Negotiation," 947-954. *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*. Honolulu, Hawaii, May 2007. Association for Computing Machinery, 2007.

[Federico 2003]
Giulio, Federico & Rahman, David. "Bidding in an Electricity Pay-as-Bid Auction." *Journal of Regulatory Economics 24*, 2 (September 2003): 175-211.

[Gerkey 2002]
Gerkey, Brian P. & Mataric, Maja J. "Sold! Auction Methods for Multirobot Coordination." *IEEE Transactions on Robotics and Automation 18*, 5 (October 2002): 758-768.

[Golliday 1985]
Golliday, C. Leslie, Jr. "Data Link Communications in Tactical Air Command and Control Systems." *IEEE Journal on Selected Areas in Communication SAC-3*, 5 (September 1985): 779-791.

[Gonen 2007]
Gonen, Mira; Gonen, Rica; & Pavlov, Elan. "Generalized Trade Reduction Mechanism," 20-29. *Proceedings of the Eighth ACM Conference on Electronic Commerce*. San Diego, Calif., June 2007. Association for Computing Machinery, 2007.

[Hinz 2003]
Hinz, Juri. "Optimal Bid Strategies for Electricity Auctions." *Journal of Mathematical Methods of Operations Research 57*, 1 (April 2003): 151-171.

[Holzman 2003]
Holzman, Ron; & Law-yone, Nissan. "Network Structure and Strong Equilibrium in Route Selection Games." *Journal of Mathematical Social Sciences 46*, 2 (October 2003): 193-205.

[Mas-Colell 1995]
Mas-Colell, Andreu; Whinston, Michael D.; & Green, Jerry R. *Microeconomic Theory.* Oxford University Press, 1995.

[McMillan 1994]
McMillan, John. “Selling Spectrum Rights.” *The Journal of Economic Perspectives 8*, 3 (1994): 145-162.

[Parkes 2001]
Parkes, David C.; Kalagnanam, Jayant R.; & Eso, Marta. “Achieving Budget-Balance with Vickrey-Based Payment Schemes in Exchanges,” 1161-1168. *Proceedings of the 17th International Joint Conference on Artificial Intelligence.* Seattle, Wash., August 2001. American Association for Artificial Intelligence, 2001.

[Parkes 2004]
Parkes, David C. & Shneidman, Jeffrey. “Distributed Implementations of Vickrey-Clarke-Groves Mechanisms,” 261-268. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*. New York, N.Y., July 2004. IEEE Computer Society, 2004.

[Rogers 2006]
Rogers, A. Dash; Jennings, R. K.; Reece, N. R.; & Roberts, S. “Computational Mechanism Design for Information Fusion Within Sensor Networks,” 1463-1464. *Proceedings of the Ninth International Conference on Information Fusion.* Florence, Italy, July 2006. Association for Computing Machinery, 2006.

[Rosenschein 1994]
Rosenschein, Jeffrey S. & Zlotkin, Gilad. *Rules of Encounter*. MIT Press, 1994.

[Sandholm 1996]
Sandholm, T. “Limitations of the Vickrey Auction in Computational Multiagent Systems,” 299-306. *Proceedings of the First International Conference on Multi-Agent Systems*. San Francisco, Calif., June 1996. Victor Lesser, editor. MIT Press, 1996.

[Sandholm 2006]
Sandholm, T. “Expressive Commerce and Its Application to Sourcing,” 349-350. *Proceedings of the Eighteenth Conference on Innovative Applications of Artificial Intelligence (IAAI06).* Boston, Mass., July 2006. AAAI Press, 2006.

[Staib 2001]
Staib, Margaret. “Sustainment Procurement in the Air Force—U.S. Department of Defense.” *Air Force Journal of Logistics 25*, 2 (Summer 2001): 3-15.

[Wellman 1993]
Wellman, M. P. “A Market-Oriented Programming Environment and Its Application to Distributed Multi-Commodity Flow Problems.” *Journal of Artificial Intelligence Research 1* (1993): 1-23.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. **AGENCY USE ONLY** (Leave Blank)	2. **REPORT DATE** January 2008	3. **REPORT TYPE AND DATES COVERED** Final

4. **TITLE AND SUBTITLE**

Using the Vickrey-Clarke-Groves Auction Mechanism for Enhanced Bandwidth Allocation in Tactical Data Networks

5. **FUNDING NUMBERS**

FA8721-05-C-0003

6. **AUTHOR(S)**

Mark Klein, Daniel Plakosh, Kurt Wallnau

7. **PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. **PERFORMING ORGANIZATION REPORT NUMBER**

CMU/SEI-2008-TR-004

9. **SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

HQ ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. **SPONSORING/MONITORING AGENCY REPORT NUMBER**

ESC-TR-2008-004

11. **SUPPLEMENTARY NOTES**

12A **DISTRIBUTION/AVAILABILITY STATEMENT**

Unclassified/Unlimited, DTIC, NTIS

12B **DISTRIBUTION CODE**

13. **ABSTRACT (MAXIMUM 200 WORDS)**

A mechanism is an institution such as an auction, voting protocol, or a market that defines the rules for how humans are allowed to interact, and governs the procedure for how collective decisions are made. Computational mechanisms arise where computational agents work on behalf of humans. This report describes an investigation of the potential for using computational mechanisms to improve the quality of a combat group's common operating picture, in a setting where network bandwidth is scarce. Technical details are provided about a robust emulation of a tactical data network (based loosely on the Navy LINK-11) that was developed for the study. The report also outlines the basic principles of mechanism design, as well as the features of the Vickrey-Clarke-Groves (VCG) auction mechanism implemented for the study. The report describes how the VCG mechanism was used to allocate network bandwidth for sensor data fusion. Empirical results of the investigation are presented, and ideas for further exploration are offered. The overall conclusion of the study is that computational mechanism design is a promising alternative to traditional systems approaches to resource allocation in systems that are highly dynamic, involve many actors engaged in varying activities, and have varying—and possibly competing—goals.

14. **SUBJECT TERMS** computational mechanisms, mechanism design, network, bandwidth allocation	15. **NUMBER OF PAGES** 58

16. **PRICE CODE**

17. **SECURITY CLASSIFICATION OF REPORT** Unclassified	18. **SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	19. **SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	20. **LIMITATION OF ABSTRACT** UL