

TCN® WHITE PAPER

Tactical Component Network (TCN®) is a registered trademark of Raytheon Solipsys.

Cleared for public release DFOISR Reference 07-S-3040.

Prepared by:

Raytheon Solipsys

8170 Maple Lawn Blvd

Fulton, MD 20759

V: 240.554.8100

F: 240.554.8101

www.solipsys.com

DN: RS-TCN-2007-WP

Date: 31 July 2007

Cornerstones of Sensor Networking

The Raytheon Solipsys (Solipsys) Tactical Component Network (TCN[®]) has been designed and implemented to meet the basic tenets of sensor networking, as described in the following sections.

- **Network extensibility must be minimally impacted by the number of network participants** - This cornerstone is of premier importance if one is to be capable of including a wide array of participants in networks of various topologies. The telephone and bank ATM systems conform to this tenet.
- **Network participants must maintain physical and functional independence** - This cornerstone is to ensure that a change to- or addition of any network element does not force change at any other element(s) in the network. As networks grow to include more and a greater diversity of element participants, this becomes essential from joint acceptance, life cycle, acquisition and test standpoints. Again, this is like the phone system, which is the model of an extensible and independent network.
- **Information exchanged on the network must support and be responsive to the needs of the network users** - Extraneous and otherwise unnecessary network sensor exchanges should not be permitted to undermine the ability of the network to meet legitimate and widely differing user needs. In such a network data earns its way onto the network by satisfying stated goals.
- **The network communications structure must seamlessly include all communications systems acting in concert** - DDS, JTIDS, MIDS, PADL, SADL, EPLRS, SINCGARS¹ and any other desired communications system must collectively form the wireless travel paths for data between networked elements if the network is to include all players. The network structure must accommodate the differences in radio throughput in a seamless and fully interoperable manner. The radio(s) used at any point in the network are selected according to the needs of the user.
- **The sensor network must support multiple levels of exchange security while maintaining needed concurrency** - Inclusion of coalition elements (and their willingness to be included) requires that elements or groups of networked elements be able to control access to their information without undermining legitimate user needs.

This is, in a nutshell, the minimal set of tenets or cornerstones that we have used in addressing the general sensor-networking problem for the joint and coalition world of today. The Solipsys Collaborative Sensor Networking concept addresses the problem by adhering to the following “derived” cornerstones:

1 Data Distribution System (DDS), Joint Tactical Information Distribution System, (JTIDS), Multi-Function Information Distribution System (MIDS), Prototype Air Defense Location (PADL), Situation Awareness Data Link (SADL), Enhanced Position Location and Reporting System (EPLRS), Single-Channel Ground and Airborne Radio System (SINCGARS)

- **Sensor elements must act in concert to meet user-specified objectives for each object** - This rule states that redundant and otherwise useless information should never be exchanged on the network. Exchanges must be made collaboratively and within the context of information provided by other contributing elements. In this way, the sensor networking objectives of extensibility and responsiveness to user needs are met. This also supports the sensor- networking objective of communications system inclusiveness.

- **All element-specific processing must be performed at the originating elements and not at the recipients** - This derived cornerstone is intended to eliminate coupling between elements that can lead to the need for one or more element(s) to change due to a change by some other element. Adherence to this rule supports the independence and, to a good extent, the multi-level security objectives of the sensor networking cornerstones.

Those two rules for collaborative sensor networking address the general sensor networking problems. The challenge is in the implementation of a capability that meets both the general and derived cornerstones. That is precisely what the Solipsys TCN method and enabling product accomplishes.

It should be mentioned that the Solipsys Collaborative Sensor Networking concept results in other derived benefits. Among them are the elimination of limitations on information input due to computational complexity growth and the ability to use low cost PC's and operating systems right out of the box. This is the nature of the functional allocations in TCN derived from the cornerstones. This, in turn, enables the progressive modernization and acquisition reform that is often spoken of in defense circles, but not often achieved.

Introduction to TCN

The TCN concept provides an architectural framework that directly addresses desired system characteristics for collaborative sensor networking. A Combat or Mission system architect can define specific objects that are required to meet operational requirements. These element objects are then coupled with supporting objects and networked together as building blocks for more complex tactical structures. These blocks are bound together through collaboration enabler and portability applications, which ensure component independence. The objective is a framework, based on common components, which can be easily tailored to meet unique system requirements. The foundation elements of TCN provide the method by which these objects, components in TCN, are connected to meet the overall operational requirements for a system.

The nature of defense system design differs from commercial systems in that the defense systems are required to combine information from widely varying sources into a single, unambiguous data set. TCN has, as its foundation, a collection of generic software applications including Data Conditioner, Current Observation-Related Estimates (CORE) Synthesis, Reporting Need Manager, and SoliNet. Other available applications include Tactical Display Framework (TDF) and Multi Source Correlator Tracker (MSCT). A notional TCN structured segment is shown in Figure 1.

The principal components within a TCN segment involved in sensor data processing and data fusion are as follows:

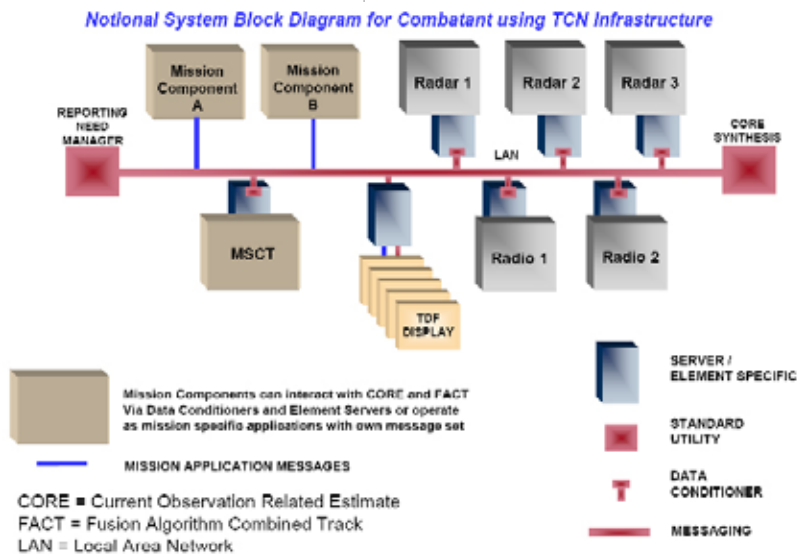


Figure 1 - Notional TCN Structured Segment

1) Sensor Element Server

- Manages correlation, association, and tracking for the local sensor
- Processing is optimized for the specific sensor
- Providing the Data Conditioner with Associated Measurements (AMRs) as well as new tracks
- Developed by the element owner because the domain knowledge of the server is highest at the source

2) Communications Element Server

- Formats messages for communication medium
- Provides communication device interface and management
- May perform functions associated with encryption devices

3) Data Conditioner

- Provides generic interface for exchange of sensor information with element servers
- Provides needs/accuracy-based data distribution
- Accumulates and distributes AMRs as CORE to local and remote CORE Synthesis utilities, based on needs expressed by the end user and the resulting improvement in the network track state
- Performs sensor data alignment functions

4) CORE Synthesis

- Fuses CORE data with the network track state into a Fusion Algorithm Combined Track (FACT) for use by element servers, Data Conditioners and Mission Components
- Manages local distribution of TCN data

5) Report Need Manager

- Maintains Report Need requests from local data users
- Manages local and network distribution of Reporting Need requests

6) TDF

- Provides a standard utility for display of FACT data and amplifying track information
- A browser-based design allows element-specific display requirements to be provided by the element developer as plug-ins, or accessed directly from the server using mainstream technology such as the Hypertext Transfer Protocol (HTTP) or Extended Markup Language (XML)
- Other displays can be used in place of or in conjunction with TDF

7) MSCT

- Communicates with non-TCN track source information and correlates and associates data with TCN FACTs
- Distributes track information to display applications
- Provides interface to simulation data
- Other legacy system interfaces can be used in place of MSCT

8) Mission Specific Messaging

- Mission applications exchange messages via a defined set of message formats and potentially by a variety of protocols
- Element Servers may process Mission specific data that can bypass Data Conditioner function
- SoliNet can be configured to support mission specific messaging layers

9) TCN Messaging

- Distributes local CORE and FACT
- Receives and distributes remote CORE
- Receives and distributes remote and local reporting needs
- SoliNet messaging is required by TCN foundation components

TCN Technical Overview

The TCN provides an architecture for an extensible, evolvable, multi-level access, sensor track fusion process that creates a distributed fused track state database from multiple RADAR or other multi-dimensional sensor position and rate measurements from multiple sites. These measurements are processed, and resulting information is exchanged over one or more distribution networks. The networks can be constrained in bandwidth, but typically should have transmission delays that are on the order of the measurement period of the sensors or smaller to derive the best performance from the process.

The TCN architecture provides the capability for data exchange to create a track picture that is in a common frame of reference (Common Frame Track Picture) for all Data Sources and Data Users in a set of data distribution networks. Each Data User requests the level of track accuracy it needs according to track identity, type, category, status, geographic location and other doctrine as specified by the Mission System's ID doctrine. These requests are mapped to the user's Reporting Need for each track. The appropriate level of data is distributed to support the requested Reporting Needs, while controlling the growth in distribution bandwidth requirements to a complexity that can be supported in large, widely distributed multi-level user networks. Data sources can constrain the data sent via communication networks, or point-to-point links, based on the user population of these communication links. In this way coalition networks can operate within an overall TCN architecture with data controlled at its source. The functional components are structured to allow single-point additions of or changes to data sources and users. The addition of new sources and users does not affect the computational complexity of the existing network components for a given number of supported tracks. The de-coupled component structure is designed in such a way that existing components need no modification for new components to use and contribute to network data.

Components of widely distributed user networks may not have bandwidth available to share a set of identical complete track data throughout the networks at the accuracy required by the most demanding data consumers. However, they can share highly accurate track data requested by specific users while supporting other levels of track accuracy according to user requests. In this approach, users request data based on their Mission System ID doctrine at the situation awareness level, the tactical level, or the engagement level. Data are accumulated, condensed, registered, and distributed as needed in a common geodetic coordinate frame with corresponding ellipsoidal covariance information. These messages are CORE. The CORE Synthesis function at each tactical segment assembles the Common Frame Track Picture from this low bandwidth data without the need for source location or sensor information. The ID doctrine and engagement functions are performed by Mission System components and may be different for each user. They are functions of an overall system that are integrated with TCN via defined Application Program Interfaces (API).

TCN Foundation

Figure 2 depicts the interactions among the TCN foundation components and other components via TCN message exchange. CORE are created from associated measurement data at each Sensor Data Conditioner and Comms Data Conditioner and distributed in a common recognizable form throughout the network for use by CORE Synthesis functions. Network data distribution is managed by the Comms Data Conditioners in conjunction with the Comms Servers. CORE Synthesis functions create FACTs for use by local platform functions that need target track states. These functions include Sensor Servers, Data Conditioners and Mission components.

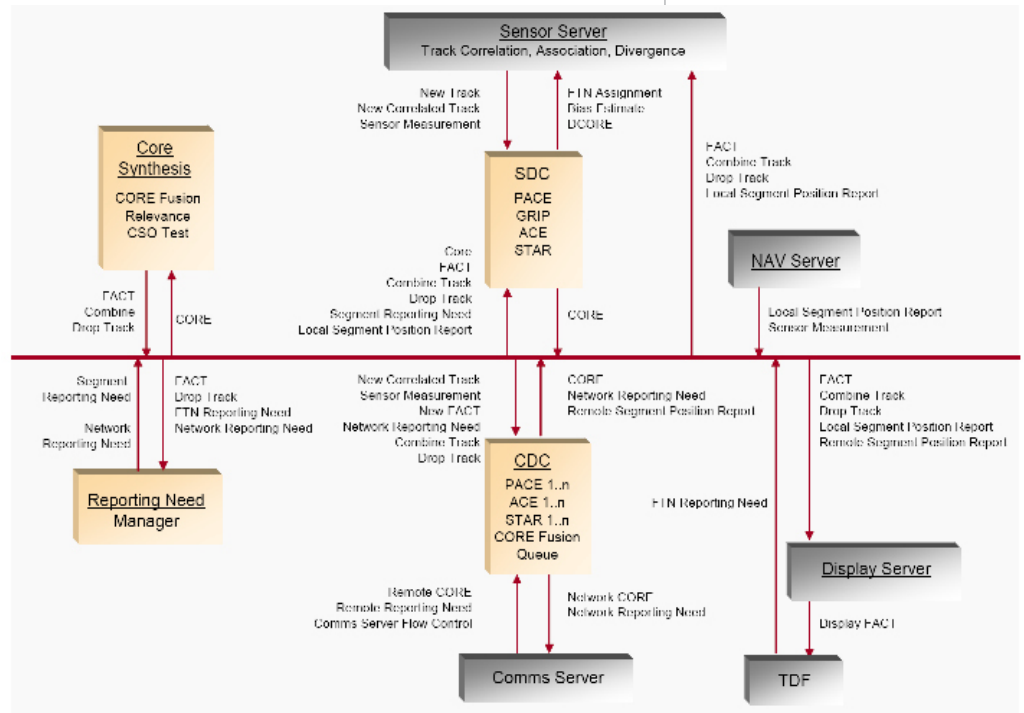


Figure 2 - Interactions Among Components

Sensor Server

Figure 3 is an information flow diagram showing the interaction between key functions within a Sensor Server. Sensor measurement data are corrected for angular bias as a function of relative target-sensor position using accumulated data. Correlation, Association, and Tracking (CAT) validates the association of measurement data to a Sensor Measurement Association & Resolution Track file (SMART) used for local hypothesis testing and gating. Resulting unambiguous AMRs are sent to a Sensor Data Conditioner and the Comms Data Conditioners. The Sensor Servers, the primary data sources, are responsible for the correlation and association of track data to the network track file. The Data Conditioners and Core Synthesis support the correlation function of the Sensor Server but do not make correlation decisions.

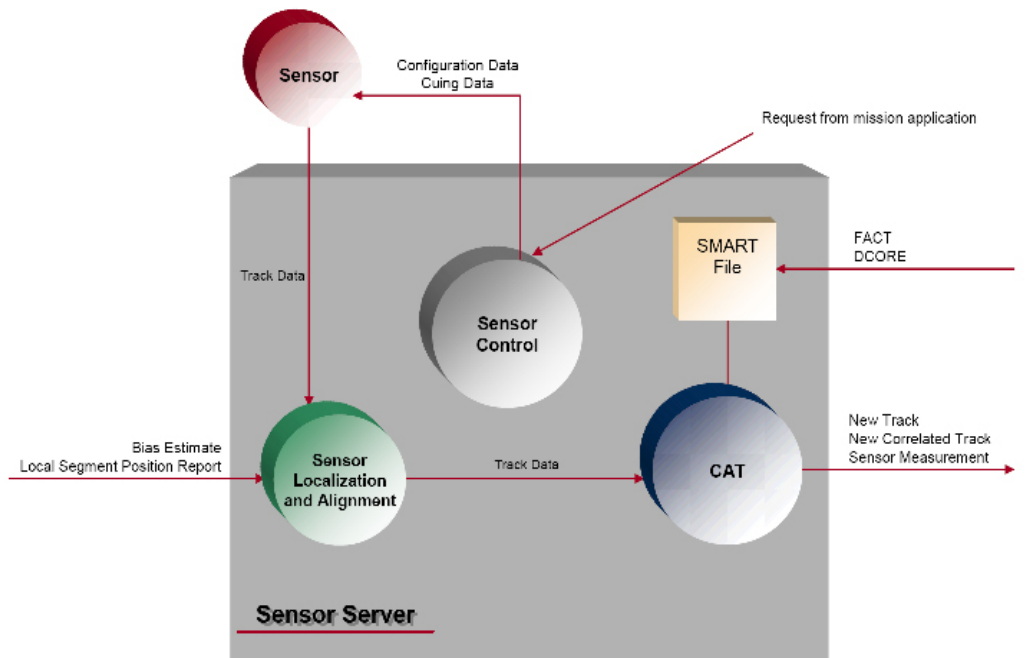


Figure 3 - Sensor Server Interaction

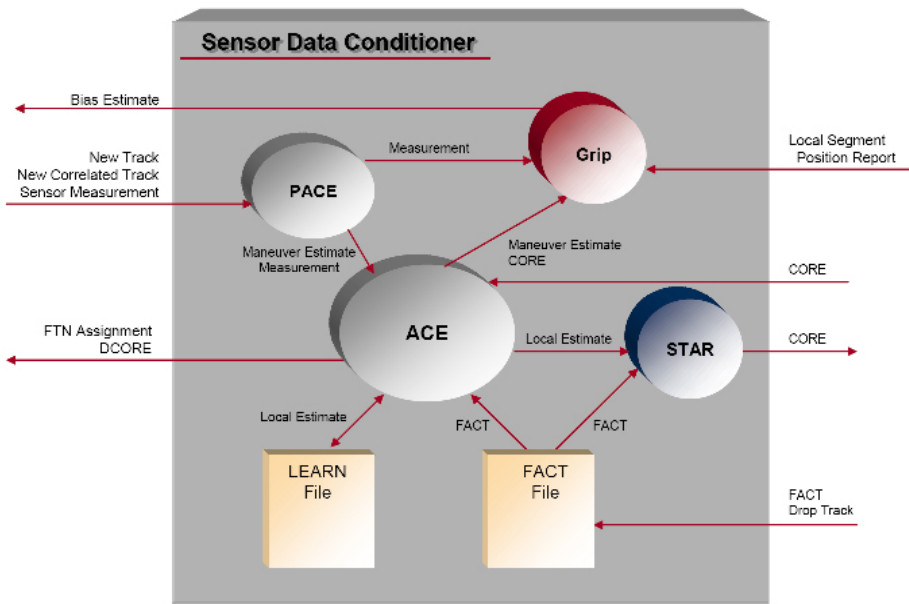
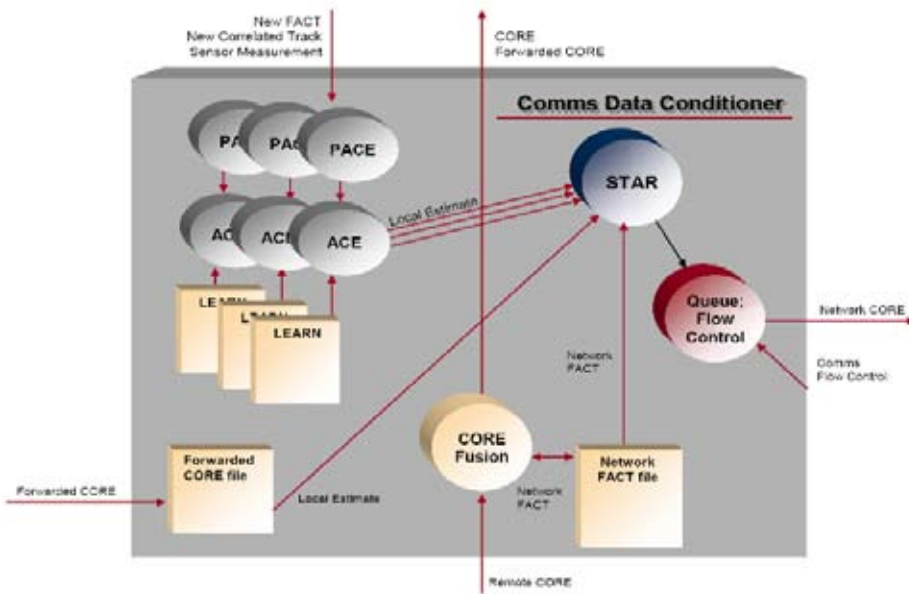


Figure 4 - Sensor Data Conditioner Processing Flow

Figure 5 - Comms Data Conditioner Processing Flow



Adaptive Curve Estimation (ACE) where they are used to form straight trajectory and maneuvering target hypotheses as Local Estimates (LEs) for each Reporting Need. ACE selects the best hypotheses at the addition of each AMR. The hypotheses selected by ACE is used by the State Testing and Adaptive Reporting (STAR) function to determine if a CORE should be created from the selected LE. In STAR the corresponding FACT covariance is predicted to the current time and used to determine the immediate need for sensor data. The value of distributing a

Data Conditioner

Figure 4 and Figure 5 are information flow diagrams showing the interaction between the key functions within a Sensor Data Conditioner and Comms Data Conditioner respectively. The primary function of the Sensor Data Conditioners and Comms Data Conditioners is to create CORE from its corresponding sensor's data based on identified Reporting Needs. When AMRs are received from the Sensor Server, they are first sent to Parametric Adaptive Curve Estimation (PACE) where they are used to detect and estimate significant target accelerations. Then the AMRs are sent to Adaptive Curve Estimation (ACE) where they are used to form straight trajectory and maneuvering target hypotheses as Local Estimates (LEs) for each Reporting Need. ACE selects the best hypotheses at the addition of each AMR. The hypotheses selected by ACE is used by the State Testing and Adaptive Reporting (STAR) function to determine if a CORE should be created from the selected LE. In STAR the corresponding FACT covariance is predicted to the current time and used to determine the immediate need for sensor data. The value of distributing a CORE produced from the LE is calculated. The immediate need and the added value are compared to determine if a CORE is to be produced. The manner of comparison ensures that a CORE is produced only if it is needed and if the LE will currently offer the greatest improvement of all sources in the network. When a CORE from a registered sensor is distributed, the corresponding LEs are cleared. Therefore, a series of CORE for any target has independent sensor-related covariances.

Sensor Server Data Conditioner

The secondary role of the Sensor Data Conditioner is to provide alignment data to the Sensor Server. Angular sensor bias and sensor localization are estimated by the Geodetic Registration Information Processing (GRIP) function within the Sensor Data Conditioner. GRIP combines various levels of registration solutions, depending on the available registration data sources, such as navigational instrument (NAV) data (including local platform GPS and reference track GPS), local sensor data and remote CORE. De-coupling in the structure of the CORE Synthesis process results in large part from the de-coupling of the registration and fusion processes.

Localization and alignment occur in the Sensor Server. Data are then distributed in a geodetic frame using full covariance information, thus they are readily interpreted by all users without knowledge of the source location or type. Registration occurs at multiple levels simultaneously to ensure a robust solution. Registration results from all available data are combined to provide the most accurate solution.

Comms Data Conditioner

The STAR function is of primary importance to the Comms Data Conditioner which monitors and prioritizes the use of bandwidth across its associated wireless communications link. The STAR function in the Comms Data Conditioner compares the Local Estimates to the Network FACT file maintained by the Comms Data Conditioner. The Network FACT file is generated using only the CORE that are distributed across the wireless communication network associated with the CDC. These Network CORE are fused using the same tracking filters used in CORE Synthesis.

Figure 6 depicts the operation of the STAR reporting decision. The quantity in the left box represents the magnitude of the predicted FACT rate error in its largest dimension. The quantity in the right box represents what would be the magnitude of the FACT rate error in its largest dimension if a CORE were distributed based on the LE. This is the improved FACT error. The predicted FACT rate error is compared against two thresholds determined by the requested Reporting Needs. If above the upper threshold, then a CORE is sent to reduce FACT error. Typical operation is below this threshold. Errors below the lower threshold indicate that no unit should send CORE, as the FACT is very accurate. Instead, ACE continues to accumulate sensor data in the form of LEs with lower variances. Between these two

thresholds, the improved FACT error (quantity on the left) is computed and subtracted from the predicted FACT error (quantity on the right). The difference is compared against a subsequent threshold to determine if a CORE should be sent. It is this latter comparison that ensures that STAR distributes the most valuable data, thereby preempting the prolific distribution of lower value data. This algorithm reduces computational and data distribution complexity by a full computational order. It thereby enables networks to meet otherwise unattainable specifications for network size and track capacity, accuracy, and concurrency. It also provides a greater flexibility in the selection of appropriate data distribution equipment by driving bandwidth requirements down to the realm of commercial radios. Where track accuracy improvement is constrained to single sensor performance by security concerns, filtering methods such as covariance intersection can be applied.

The magnitude of the predicted FACT rate error ellipsoid's largest dimension at the time of the LE is calculated by predicting the FACT to the time of the LE, and deter-

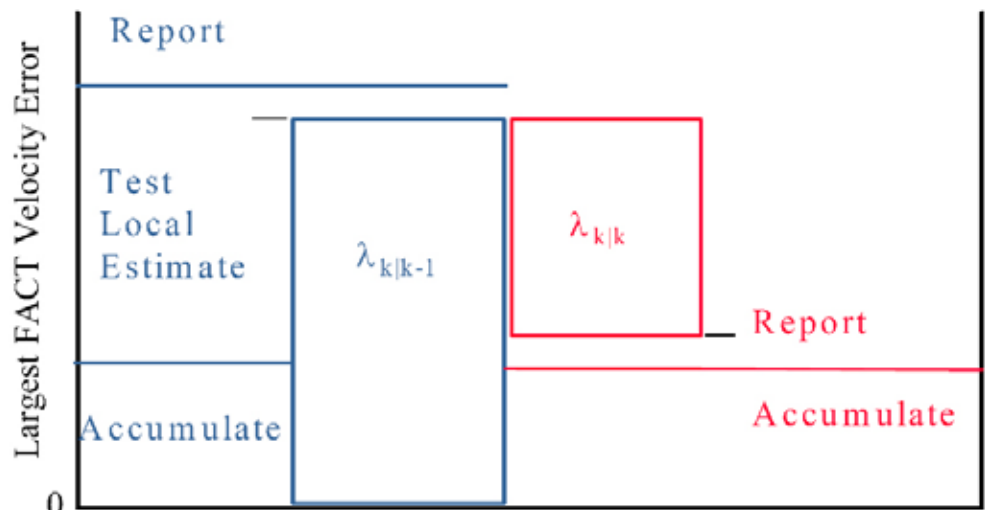


Figure 6 - STAR Reporting Decision

mining the largest eigenvalue of the rate covariance. The magnitude of the FACT rate error ellipsoid's largest dimension if a CORE were distributed is calculated by computing the filtered rate error that results from updating the FACT with the CORE and determining the largest eigenvalue of that rate covariance.

Core Synthesis

Figure 7 is an information flow diagram showing the interaction between key functions within Core Synthesis. CORE from local data sources are distributed by the Sensor Data Conditioner and CORE from remote data sources are distributed by the Comms Data Conditioner to CORE Synthesis. CORE are most often distributed as 6-state estimates. However, 9-state and 3-state CORE are distributed for certain conditions and target dynamics. The CORE are fused by a robust target-oriented filter that produces FACTs. The updated FACT states are then distributed to components on the local segment. The Sensor Data Conditioner uses the FACT states for the generation of additional CORE while the Sensor Server uses the FACT states for correlation and association. Mission components use the FACT states for situation awareness and other mission specific activities.

In addition to fusing FACT data, CORE Synthesis performs sanity checks and alerts local components of closely spaced tracks and dropped tracks. CORE are first validated using a relevance test against the previous FACT state. After the CORE is fused with the FACT state, the updated FACT state is compared to other FACTs in the database for possible redundant or closely spaced tracks. If redundant or closely spaced tracks are detected, CORE Synthesis generates a Combine Track. When network tracks are no longer being updated by any data source in the system, CORE Synthesis drops the network track and notifies all components. CORE Synthesis also can connect networks by fusing FACT data from different networks operating at

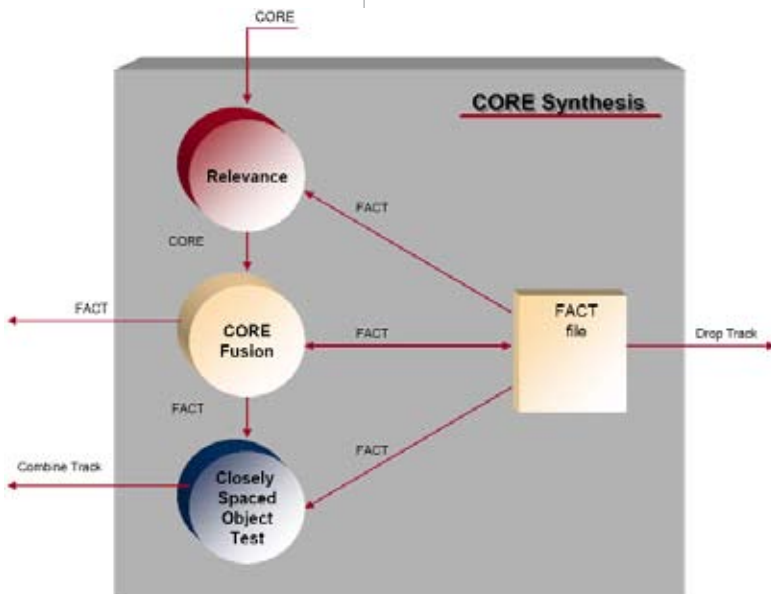


Figure 7 - Core Synthesis Processing Flow

different Reporting Needs. The CORE Synthesis process allows functions to be allocated to software components in a variety of ways, as best suits the system application.

Reporting Need Management

Local and Network Reporting Needs are selected according to doctrine, request, track resolution requirements, registration requirements, and local sensor surveillance volume. Components on the local segment make reporting need requests and the Reporting Need Manager selects the reporting need for each track based on the request of the most demanding user. These Reporting Needs (RNs) are distributed throughout the network to establish a goal oriented CORE reporting mechanism whereby data is only sent when it meets a stated need – data is “pulled” by users and not “pushed” regardless of its ultimate value to the user’s mission.

STAR selects the data that will most efficiently fulfill the selected Reporting Needs (RN). Through an examination of the FACT rate error, thresholds for the LE data are determined. Since other contributing nodes use the same FACT, the knowledge that

the local Data Source's LE can be used to create and distribute a CORE means that the local source has the best current contribution. If another unit had a better contribution, then it would have been distributed and the FACT accuracy would have induced a stricter threshold on local LE data, thereby preempting its distribution.

Figure 8 illustrates the flow of RN requests to and from the Reporting Need Manager.

Summary

The TCN foundation components described above provide the services and functionality common to all sensor networking systems. These components are merely the starting point for designing and implementing a TCN enabled Mission System. The Mission System architect selects the hardware for hosting the TCN foundation components, software utilities that are portable to an extensive number of hardware configurations. Next the Mission System architect selects the data sources and Sensor Servers of the system. The Mission System architect also selects and integrates the communication resources and Comms Servers. At this point the system consists of the fundamental technology required to create a network track picture. The Mission System architect then designs and implements the mission applications which provide the system unique services for each combatant, such as engagement support, ID processing and non-kinematic data integration. The TCN framework allows the mission systems created by a variety of users to be extensible, interoperable and maintainable: the cornerstones of sensor networking.

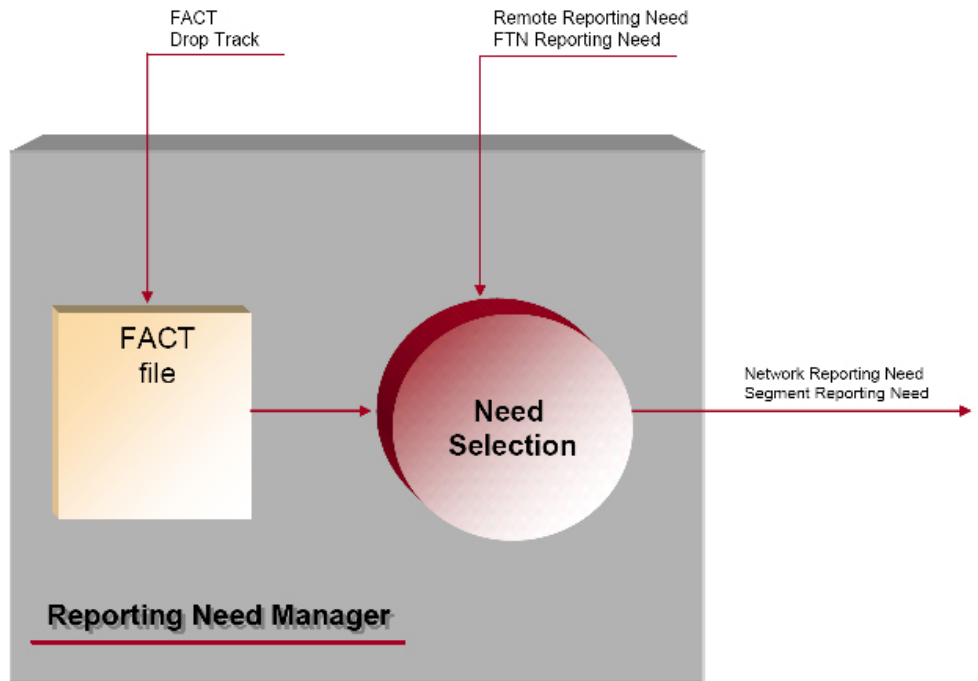


Figure 8 - Reporting Need Request Flow