WHITEPAPER

UNDER THE HOOD — A TECHNICAL OVERVIEW OF THE INETCO INSIGHT® APM SOLUTION

INETCO Insight

INETCO

A Whitepaper by Angus Telfer & Marc Borbas

INETCO® Systems Ltd.

September 2012

info@inetco.com

P. 604.451.1567

www.inetco.com

WHITEPAPER OVERVIEW

INETCO Insight® is a new kind of application performance monitoring software for IT operations teams. **INETCO Insight's** transaction decoding, semantic correlation and statistical analysis engines can be rapidly configured to monitor any application type (custom, packaged, or industry-specific) and can simultaneously monitor hundreds of distinct applications and transaction flows.

Regardless of protocol, platform, or application, the result is a single transaction record containing business, application, infrastructure, and network performance information that can be sliced and diced in real-time and output to downstream systems.

The product is built on **INETCO's** core technology platform, a modular, real-time, distributed event processing system written in C/C++. It turns the network into a rich source of intelligence on business service usage and performance and has a myriad of applications from systems management to business analytics to telemetry and intercept.

This paper will explore the key challenges in monitoring application performance from the network, explain how **INETCO Insight** overcomes these challenges, and highlight key integration points for partners who want to enhance their products with I**NETCO Insight** capabilities.

KEY CHALLENGES

Organizations are increasingly focused on meeting customer service expectations, guaranteeing end-to-end business process delivery and modernizing their IT infrastructures. This is driving many IT operations teams to consider new requirements and methodologies when it comes to performance monitoring.

Historically, applications were monitored in one of two ways:

1. **By instrumenting the application code and measuring processing times**
2. **By collecting and correlating events from application and infrastructure components**

The first approach is effective for applications with relatively few moving parts that reside in the same data center because the instrumentation task is feasible and the network has very little impact on overall response times.

The second approach is effective for applications with relatively static configurations and small numbers of servers because there is a strong relationship between the underlying infrastructure and the applications running on it.

Unfortunately, modern applications rarely fall into either of these buckets. As applications become more distributed, heterogeneous, and sophisticated, new ways of monitoring their performance are required

The network provides a viable, "always on" source of information about application usage and performance. After all, it is the vehicle for transporting instructions from component to component in a distributed application.

However, tapping into a fire hose of traffic traveling across a modern network, and transforming it into useful information about application usage and performance, presents a number of key challenges:

- **Message and Transaction Recognition.** Protocol messages and transactions typically have very poorly defined, often ad-hoc, application content and interaction. Communications consists more of a "conversation" than a clear cut transaction. As such, there is a significant challenge in determining when a captured message is relevant, when a transaction starts, when it ends, and how to find relevant contents. This is challenging even when all the data involved is visible (i.e. has all been decoded) and no packets have been lost or are out of sequence. In the case of heavy data loads it is also vital that message processing for transaction recognition be done at the earliest possible point and with the least processing and memory possible.
- **Transaction Characterization.** New transaction types must be quickly configured into the system. One challenge is establishing methods for users to characterize transactions rapidly and provide reliable recognition between different types of transactions with only the most essential features of a transaction.
- **Large Fan-in.** Web-based transactions are often in the thousands of bytes in length, may involve large numbers of messages, and may include multiple data formats such as video, audio, etc. Even in the case of early transaction recognition on CPUs directly monitoring the network, reducing the amount of data sent for real-time upstream processing is a major challenge.

- **Multi-Link Transaction Correlation.** Delays in a transaction system may come from any part of that system. In a service oriented architecture, these delays may come from the network link between the web browser and the web server on the client side; from the web server itself; from any of the "back-end" network links to internal components; on networks linking internal infrastructure; internal applications; networks to external services (i.e. order fulfillment, payment processing); or the external services themselves. For a monitoring technology to be useful, it will be necessary to correlate single link transactions (e.g. such as on a client link) with communications on related back-end components and vice versa. There are major challenges in doing this in the presence of encryption, caching, and the simple fact that there may be few (not easily determined) common features or feature dependencies between two or more links.

This paper explains how **INETCO Insight** addresses each of these key challenges within high-volume application environments.

INETCO INSIGHT OVERVIEW

INETCO Insight is a software-based passive network monitor for application transactions. It decodes transaction messages in real-time from raw network traffic forwarded by a managed Ethernet switch, packet capture equipment, or application servers.

The **INETCO Insight** architecture is made up of five major components:

- **INETCO Insight Agentless Event Collectors –** for monitoring data activity
- **INETCO Insight Decoders –** for message decoding
- **INETCO Insight Processors –** for transaction correlation and analysis
- **INETCO Insight Data Historian –** for real-time storage and querying of transaction data
- **INETCO Insight Console, Web Server and Reporting Server –** for user access points and output to 3rd party systems

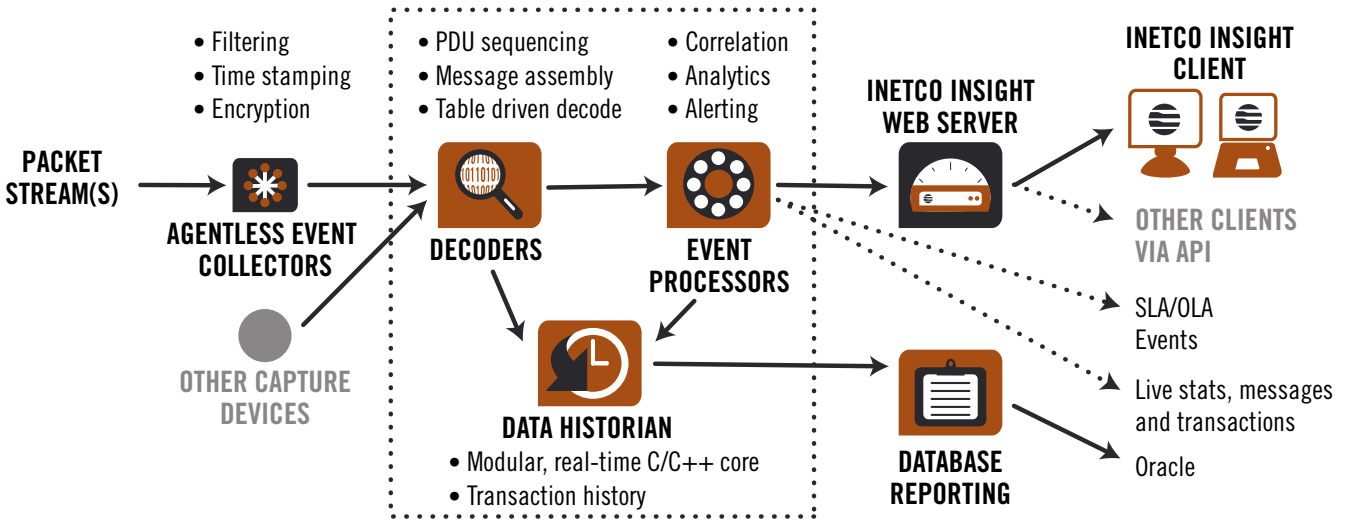

Figure 1 **INETCO Insight** operation. This diagram shows the major components of the **INETCO Insight** system.

Between each of the five major **INETCO Insight** components, there is either a documented protocol or API for interfacing to the **INETCO Insight** system. For instance, the decoder and processor communicate using AMQP, an open standard, wire-level protocol for reliable messaging that support pub/sub architectures. The **INETCO Insight** client uses a ReSTful API to request information from the web server.

As a result, partners can interface with the **INETCO Insight** system to extract a wide range of data for various purposes, including:

Data	Interface	Potential Application
Individual application messages	**INETCO Insight Decoder**	Forward every authorization request so a fraud monitoring system could check it
Decoded transactions and response time information via TCP	**INETCO Insight Processor**	Incorporate into load balancer policies to prioritize high-value transactions
Decoded transactions and response time information via database	**INETCO Insight Reporting**	Build a long-term profile of application performance
Count of transactions by type for a group of nodes or endpoints over an interval	**INETCO Insight Processor (Analytics)**	Displaying a summary of transaction activity for a particular user or device
Recent transactions for a specific customer	**INETCO Insight Data Historian**	Display transaction details in a customer portal
Application/transaction dependency information	**INETCO Insight Data Historian**	Update application dependency models
Events/Alerts	**INETCO Insight Processor**	Forward SLA violations to alarm consoles and reporting systems

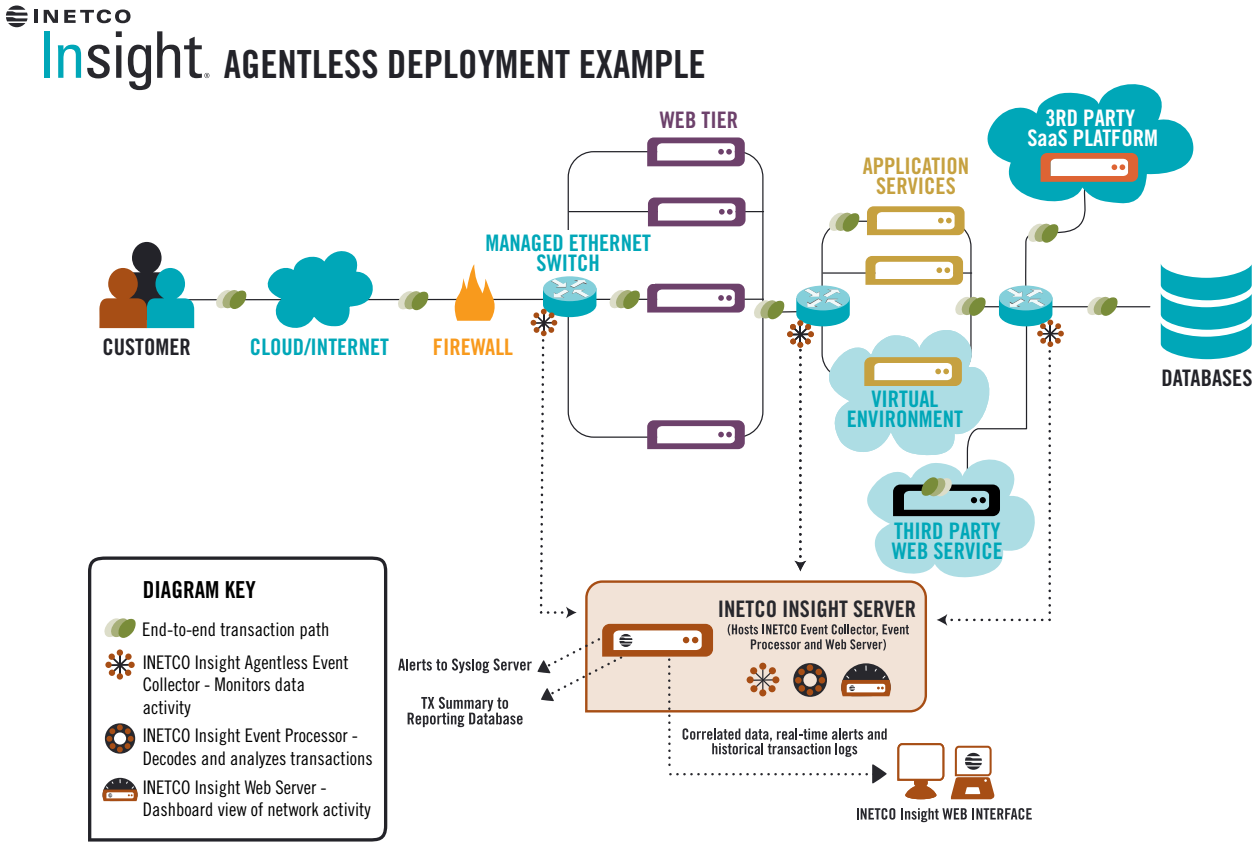

Figure 2: INETCO Insight deployment. This diagram illustrates how **INETCO Insight** can be deployed in an n-tier application environment.

DATA ACQUISITION

The first operation in **INETCO Insight** involves getting the raw transaction data. This is the job of the **INETCO Insight** Event Collector software. For the purpose of this discussion the **INETCO Insight** Event Collector can be thought of as an advanced network sniffer or datascope.

INETCO Insight currently allows two ways of getting raw data. One method is to obtain data directly from the network using a router's SPAN port, also called "port mirroring." In virtualized environments, equivalent features are available for mirroring network traffic.

The use of SPAN ports is not always possible or desirable. In this case the **INETCO Insight** Event Collector can be loaded directly onto packet capture equipment or application servers to gather the required data. In these deployments the **INETCO Insight** Event Collector acts as a passive "shim" between the application services and the communications channel of interest. This method can also be used to collect network data in virtualized environments or to receive a conditioned packet stream from existing network instrumentation.

In either case, capture can be performed "out-of-band", without introducing agents or code changes to application components.

***NOTE: INETCO Insight** can also collect raw data via the diagnostics port on many proprietary network gateways and from third party network "sniffers." Contact INETCO regarding such support.*

Upon collection the raw network data is time stamped. **INETCO Insight** can integrate with GPS-based packet capture equipment, where precise time-stamping is critical**.** A filter is then applied to remove any data with IP addresses and ports not involved with transaction processing (e.g. video, voice, SNMP, etc.). This dramatically reduces the volume of data forwarded for analysis by eliminating extraneous data. At this stage a sequence number is also added to the data to allow downstream detection of any missing or duplicated data.

The filtered data is then encrypted before being sent to the **INETCO Insight** Server. This ensures that sensitive transaction information is never available "in the clear." Integration with hardware security modules is supported**.** Encrypted checksums are also included with every message to ensure data between the **INETCO Insight** Event Collectors and the **INETCO Insight** Server is not modified added to or deleted. This ensures data integrity that is critical in environments where **INETCO Insight** is to be used to create financial transaction audit trails.

Finally the **INETCO Insight** Event Collectors forward encrypted data to the **INETCO Insight** Server. It does this via a reliable flow controlled TCP data channel to ensure the **INETCO Insight** Server is not overloaded. While this may result in minor delays in data delivery, timing information is maintained since all data is time stamped when first collected.

Once received by the **INETCO Insight** Server, the stream of real-time network data is then decoded by the **INETCO Insight** Server as explained in the next section.

MESSAGE DECODING

The second major operation of **INETCO Insight** involves decoding the incoming raw network data to extract application message information. This is the first job of the **INETCO Insight** Server. It involves sequencing the incoming raw network data, assembling application messages from the data, decoding the individual data fields within the application messages, and masking or deleting any sensitive fields (e.g. credit card numbers for financial transactions, patient numbers for healthcare records flows, etc.)

To process the incoming data the **INETCO Insight** Server must first sequence the incoming raw network data units (i.e. "protocol data units" or PDUs) from the **INETCO Insight** Event Collectors to ensure they are analyzed in the correct order. This involves the use of the time stamps and internal event sequence numbers added to the data when it is first collected. With multiple **INETCO Insight** Event Collectors, this information may not be sufficient as the real-time clocks of the distributed collectors will vary even when the Network Time Protocol is used for time synchronization. In such cases the **INETCO Insight** Server may also use contextual information within the raw network data to properly sequence PDUs.

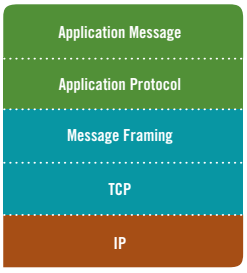

Figure 4: Protocol levels for application messages. This diagram illustrates the different protocol levels involved in a simple application-level message delivery using TCP/IP.

Next, application messages are assembled. Application messages typically do not fit within a single low level PDU. Instead they are often spread out over multiple PDUs as shown in Figure 5**.** For this reason decoding of the application message stream includes not only reading data embedded in a PDU at each layer, but also includes assembling PDUs to create higher level messages.

The **INETCO Insight** Server run time architecture allows message assembly to occur in real-time. The event collection component of the **INETCO Insight** Server receives the relevant PDUs (i.e. network data fragments) and stores them in the high performance **INETCO Insight** data store. All PDUs that could potentially become part of an application level message are grouped together in sequence so that application level messages can be assembled from more than one PDU.

Once a complete application message is assembled, a new application level message entry is inserted into the data store and information is associated with this entry for upstream correlation. This entry is also linked to the PDUs in the data store to support a key feature of **INETCO Insight**, the ability to drill down through the various protocol layers. This mechanism is further extended to support more complex correlation of multi-hop transaction flows and drill-down from a high-level business transaction to a low-level link transaction.

Once a complete application message has been assembled, it must be decoded. This is not a trivial task, and a variety of methods must be used. Three general types of transaction message formats prevail. One is a field separated one of which there are many, many variants (e.g. Visa II financial messages). The second one is a bit field oriented one of which there are also many, many variants (e.g. ISO 8583, FIX FAST, etc.). These are often used in high-speed, low latency environments where communication efficiency is paramount. The third is a tagged field one commonly implemented using XML or HTTP, of which, again, there are many, many variants, but they are often self-describing.

Decoding each of these transaction message formats presents it's own set of challenges. Field-separated messages require prior knowledge of where to find specific data elements, the ability to recognize missing elements, and the ability to apply a semantic model to add meaning to the raw data fields extracted. Bit field oriented messages are often tuned and adjusted by various parties to meet stringent performance criteria and often include unassigned fields that application developers can use to pass proprietary details. Tagged field messages can be extremely large and contain a lot of "non-transactional" data. In all cases, there are endless variants, even in tightly standardized application protocols.

Hard coding all of the many variants is not a practical solution. As such, decoder components in the **INETCO Insight** Server use a "table driven" approach to quickly define the decode tables for each message type within a transaction. Once this is done the table is compiled and installed on the **INETCO Insight** Server for fast access.

When a message is received the appropriate decode type is detected automatically for each incoming message and the message is forwarded to the appropriate decoder for decoding in real-time. Note that **INETCO Insight** contains a wide variety of transaction decode tables for common message formats. New tables are continuously added to the product and custom formats can be produced by **INETCO's** service organization.

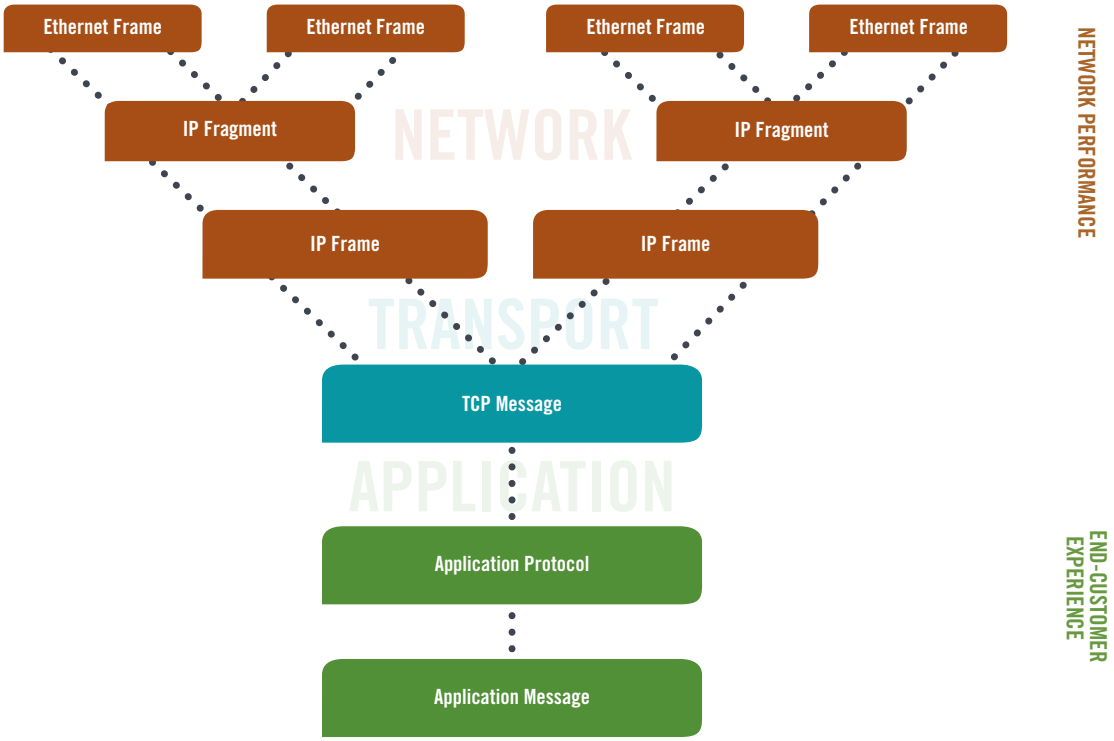

Figure 5: PDUs assembled into application messages. This diagram illustrates how ethernet packets are re-assembled into application messages to transition from a network view of traffic to an end-user view of application performance.

As part of the decoding operation special entries in the decode tables allow masking and dropping of sensitive application information. In the case of financial messages this includes masking the credit card number and dropping other sensitive information such as the PIN block.

INETCO Insight provides decoding capabilities for the following protocols:

- IP, TCP (v4 and v6), UDP and other network layer protocols.
- HTTP, HTML, SOAP, XML, and other "Internet" protocols.
- AMQP, CLNP, TPDU, NIST, SQL, and other transport layer protocols.
- ISO 8583, FIX, IFX, OFX, VISA, and other application layer protocols both within and external to the financial segment.
- X.25, SNA, Bisync, dial, and other "legacy" communications protocols

Finally once a message has been decoded it is passed on to the **INETCO Insight** Server's correlation component to be correlated with other relevant messages into a transaction.

At this point, an interface is available for third parties to intercept decoded messages for security, policy, or performance monitoring purposes.

TRANSACTION CORRELATION

The third operation of **INETCO Insight** is the correlation of messages into single link transactions and transactions into higher level transactions, eventually resulting in multi-hop application and business transactions.

Two things make **INETCO Insight** unique in this respect. First, it includes semantic models for various application protocols that capture how messages are exchanged in order to execute certain transactions. This goes beyond the simple atomic transaction modeling most network-based APM systems use, where a response is simply paired to the originating request and allows the assembly of messages into much more complex multi-step transactions.

Secondly, **INETCO Insight** uses an analytic framework called the **Unified Transaction Model** to organize application and network messages into a higher level business transaction — which is analogous to a user task or business process.

This model is driven by a set of Application Category definitions that describe the structure, relationships and message events of interest for different styles of applications (e.g. an n-tier J2EE application, a redundant ATM or POS application, etc.).

Correlation, even at a single link level, is not easy since all possible error conditions that may be encountered in real life protocols, including timing errors, must be considered. A business transaction in a modern, distributed application consists of a large number of discrete steps that span many different application components and 3rd party services.

The first step in transaction correlation is to identify the PDUs that make up a transaction. This is done by the "transaction identification" component of **INETCO Insight** (i.e. TransIdent) as shown in Figure 7. This component uses a "Transaction Definition" specification for two operations.

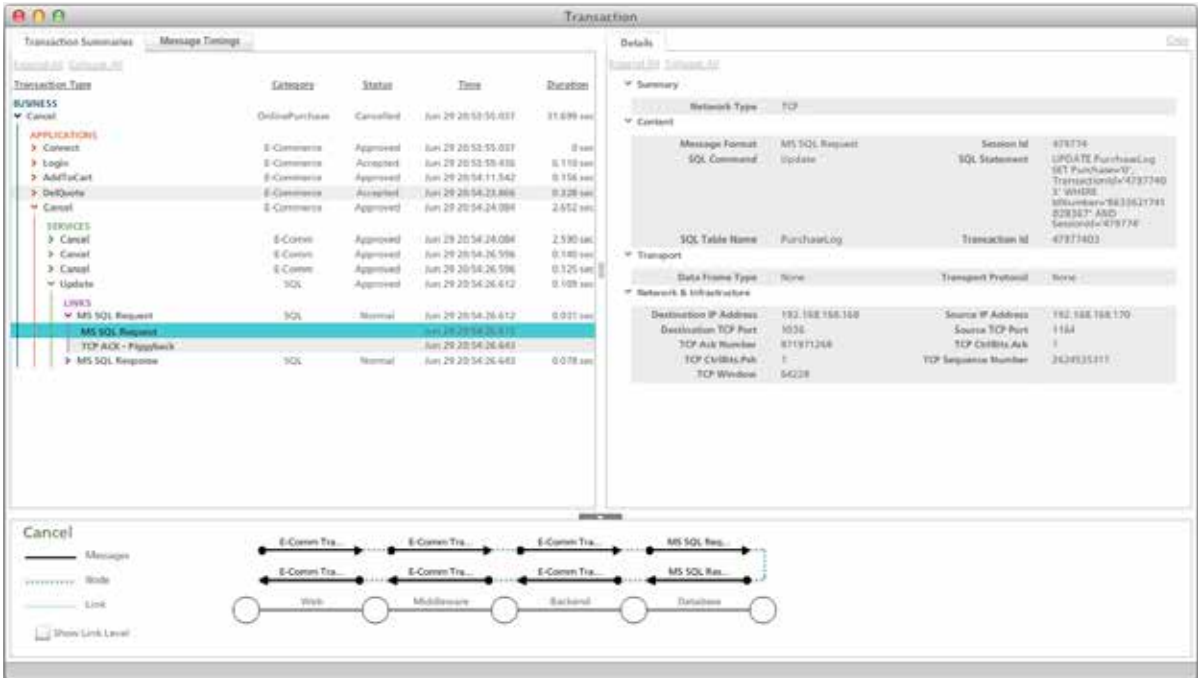

Figure 6: Online purchase transaction. This screenshot illustrates how **INETCO Insight** ties together dozens of application messages into a single business transaction, while preserving all the important details of how the transaction is executed at a technical level.

First it determines the logical boundaries of a transaction (i.e. the messages or events that start and end a transaction). For a given network type a set of rules is defined in the Transaction Definition to determine the bounds of the transaction. These rules allow the recognition of single or multiple transactions within a network connection, successful transactions, and failed transactions. Using this mechanism all messages are flagged with a unique transaction identifier to identify the one they belong to.

Knowing something is a transaction is not of much use unless transaction specific information is also readily available. The "Transaction Info" component of **INETCO Insight** is responsible for mining relevant transaction information from the decoded fields and configuration information**.** This includes the type of transaction (i.e. withdrawal, add to cart, update customer information, etc.), whether the transaction is successful, error information, and any other information of importance. The result is a Service Transaction (i.e. a SQL insert, a card authorization, an HTTP request/response).

Service Transactions are then assembled into Application and Business Transactions according to the Application Category definitions present in the system.

Users can also define transaction groups to segment a large application into meaningful buckets. This is done in the "Group Info" component which matches configured decoded fields and their values against user configured groups. The relevant group names are then attached to the associated transaction information.

The final function of the correlation component of the **INETCO Insight** Server is to put the correlated and labeled transaction in a "Transaction Store" where it may be later accessed for queries and drill down.

At this point, an interface is available to query the **INETCO Insight** data store to extract transaction information.

This information is then handed off to the statistics and output component of the **INETCO Insight** Server to enable interactive querying, web-based display, transaction logging, transaction forwarding, event notification, and syslog output.

Within this system (which is not detailed in this paper), there are a range of interfaces, including a web-based API, real-time TCP feeds, and SQL database output.

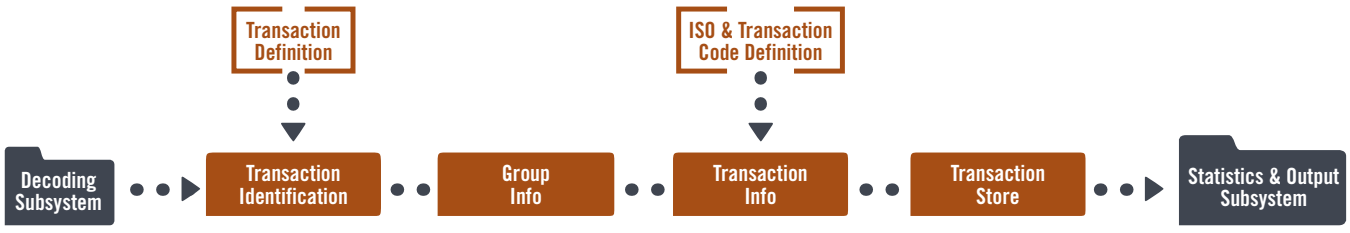

Figure 7: Transaction correlation process. This diagram illustrates the steps used by **INETCO Insight** to correlate transactions.

SUMMARY

INETCO Insight's data collection message decoding and transaction correlation technology enables real-time monitoring of complicated and diverse transaction flows from a handful of non-disruptive network access points.

INETCO Insight's proprietary processing engine inspects every packet related to an application, assembling these packets into messages, decoding the contents of every message, correlating messages into atomic transactions, and then re-constructing multi-tier and multi-hop transactions. At each step relevant header information, message payload details, addressing information, and timing attributes are captured.

As a result **INETCO Insight** is able to dramatically improve visibility into complex application environments, enabling faster problem isolation, improved cross-team collaboration, and end-to-end visibility into business process delivery.

DEFINITIONS

The following terms are used in this document:

PAN Transaction "Primary Account Number"

PDU Protocol Data Unit. A low level data packet received by the **INETCO Insight** Event Collector and forwarded to the **INETCO Insight** Server.

SPAN Switch Probe Analyzer Port. A port on a network switch router or other network device which can be used to monitor traffic on other ports This functionality is also called port mirroring.

INETCO Insight

About INETCO®—Every transaction tells a story™

INETCO® Systems Limited provides transaction-based application performance monitoring solutions to IT operations teams that are looking for a faster, non-invasive way to identify application and infrastructure bottlenecks and ensure optimal service and business process delivery within their production environments. **INETCO's** solutions are currently deployed in over 50 different countries**.** Happy **INETCO Insight®** partners and customers include a variety of global companies spanning the banking, ATM, retail, healthcare, travel, telecommunications and payment processing markets.

www.inetco.com

INETCO

INETCO Insight

INETCO Systems Ltd.
#258 – 4664 Lougheed Highway
Burnaby, BC V5C 5T5 Canada

T 604.451.1567 F 604.451.1565
insight@inetco.com

Twitter @INETCOInsight
Facebook facebook.com/INETCO
LinkedIn linkedin.com/company/inetco
YouTube youtube.com/user/tofinotofin
INETCO BTM BLOG inetco.com/blog

www.inetco.com