CHIPKIN

White Paper

Node Status recovery with Change of State systems and alarm deluge prevention

By Peter Chipkin, Chipkin Automation Systems Inc
whitepapers@chipkin.com

Abstract

Many changes to the state reporting system do not provide a connection recovery mechanism.

> For example, using the Rockwell consumer/producer model, if a produced change in IO state message is lost (because of a connection failure) the consumer will not see the change and the producer will not know the message was lost.

CAS Gateway (CASgw)'s COS engine provides several configurable and robust recovery mechanisms. They are outlined in this paper.

Many changes of state reporting systems can produce alarm message deluges when a single major event produces thousands of subsequent events.

> For example, the loss of a major communication connection can produce a deluge of alarms from nodes using the connection if the node count is high. Systems which employ AI and other technologies to filter and prioritize alarms generally require that all alarms are known before the analysis and filters can be applied thus still necessitating large volumes of alarm messages.

CAS Gateway (CASgw) provides a flexible architecture which can be used to intelligently reduce alarm deluges and ensure only critical messages reach the front end (MMI) while still providing a trace of all original events. This architecture is discussed in this paper.

The problem with many COS systems.

COS systems free bandwidth when a system is operating under normal conditions and often swamp available bandwidth with a significant event that triggers thousands of subsidiary events.

In the system below

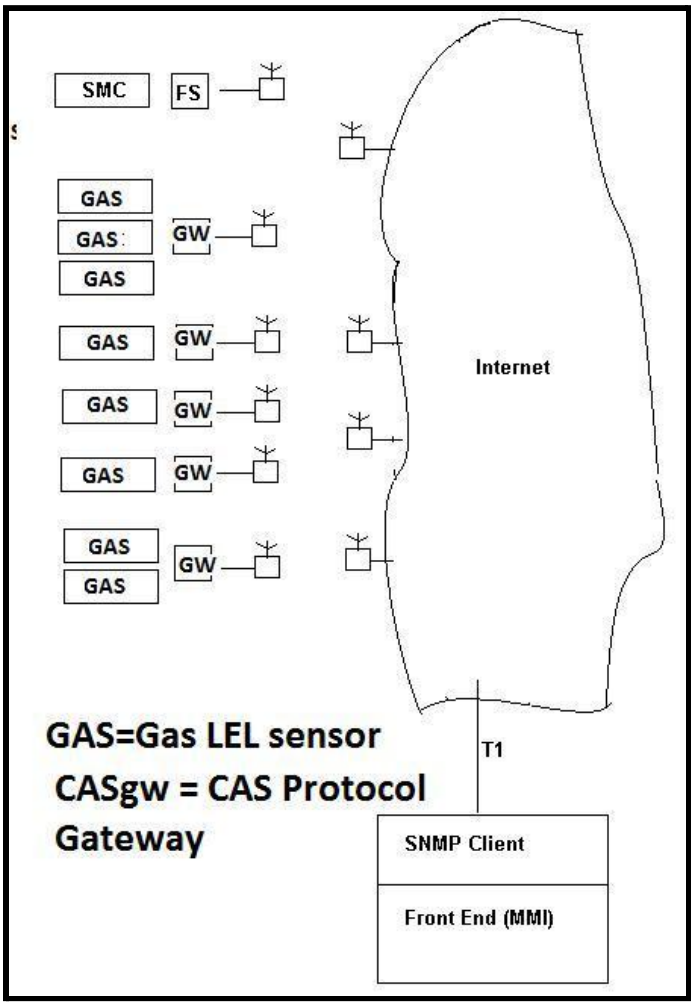
SMC=Environmental Controller

1. CASGW = FS = Protocol Converter Gateway . Collects data from a field based data gathering device. In the case of this paper, for the purposes of example a SMC Gas Sentry is referred to. It is a device which measures LEL of various gases. It produces SNMP traps in response to alarm conditions.

When the T1 connection fails the two classic COS problems are highlighted.

1. The published data does not reach the client. The MMI cannot alarm the events as the alarms do not reach the MMI. Worse still, the MMI as a passive client relies on published COS data, does not know the connection to each / all nodes have been lost.

2. When the node is recovered an alarm deluge can be produced as each node reports its status.



CAS Gateway (CASgw)'s Change of State Technology – A robust and flexible solution.

Connection Recovery Using CAS Gateway (CASgw)'s COS technology.

The CAS Gateway (CASgw) technology providing COS support is known as Subscribe / Publish. There are several different implementations. A general description is provided below.

Summary:

"Subscription" is jargon for the action taken by a remote client to inform the publisher of its interest in receiving unsolicited messages. 'Publication' is jargon for the action taken by a COS capable device in sending messages with COS data to subscribers.

A client subscribes for data. Subscription can be an active process (message from client) or a configuration process. Publishing means the act of sending a message that contains COS data.

Subscribing:

There are two main attributes of a subscription

Scope of Subscription: Defines what data the remote client is interested in. May be defined on a

- Per node basis
- Per data object basis
- Per data range basis

Type of Subscription: A subscriber tells the publisher some specifics of the subscription or these attributes are taken from the CAS Gateway (CASgw) configuration. The types of subscription allow the driver to override defaults and config file settings for ALL the publication rules.

Publishing:

Publishing COS Data: A subscription not only provides info about the subscriber but also tells the publisher the rules that trigger COS data and rules for publishing this data.

There are two main attributes groups for Publishing.

COS Trigger Rules

These rules include one or more of the following

- Delta – Determines how much a value must change by before a COS is produced.
- Deadband – Determines how much a value must change by before a subsequent COS is produced.
- Hysteresis – Determines how much a value must rewind too before a new alarm COS is produced.
- Change from Normal – Defines normal. When a value becomes abnormal A COS is produced.
- Change of State -
- Min Update – Determines a time interval during which, if no COS is produced then by a value change then a COS is produced based on the interval.
- Max Update - Determines a time interval that must pass after a COS is produced before which no subsequent COS is produced.
- High Alarm – Determines a threshold a value must reach to produced COS data.
- High High Alarm – Determines a threshold a value must reach to produced COS data.
- Low Alarm – Determines a threshold a value must reach to produced COS data.
- Low Low Alarm – Determines a threshold a value must reach to produced COS data.

Publishing Process, Rules and Recovery

COS data is not sent directly to the subscriber node. It is sent to the SNCQ (**Subscriber Node COS Queue**). From this queue messages are sent to the subscriber if the subscriber node is online or are held / processed / dropped for later publishing if the subscriber node if offline. This queue is thus used to prevent alarm deluges and to recover from a loss of a connection .

A large number of configurable publishing rules may be defined for each subscriber node. The rules are applied according to the state of the remote subscriber node's connection – three states are recognized – offline, probation (recovery from offline) and online.

The rules include

- Subscriber Node COS Queue (SNCQ) Length of Queue full rule
- Node Bandwidth Rules (Online)
- Priority Rules (Online)
- On ‘transition to offline’ rule
- LIFO/FIFO Rules for offline publishing
- Max Queue Age (Recovery)
- Priority Rules (Recovery)
- LIFO/FIFO/FILO Rules (Recovery)
- Node Bandwidth Rules (Recovery)
- On ‘ transition to online’ Rule for recovery publishing

Example:

A remote client is subscribed for COS data associated with a process variable monitored by a CAS Gateway (CASgw). Each time that the monitored process variable changes by its ‘delta’ rule a COS event is queued. As the CAS Gateway (CASgw) processes the queue it applies the node bandwidth publication rules ensuring that the remote node is not swamped Now the connection is lost. Events are queued. Now the connection is re-established. The CAS Gateway (CASgw) processes the queue applying the recovery rules. In this case the recovery rules may say that the subscriber node is only interested in the current state or the subscriber node may want all messages so that it can track the change of state for its data log application.

Node Status Monitoring

Customer client applications are often Change of State Data Subscribers. This often makes them passive clients and as such they don't know whether a remote server node is disconnected / failed or whether there haven't been any events to publish.

There are several solutions to this problem.

- By getting the client to poll each remote server node on a periodic bass the client can report the status of the connection to the remote node. This is possible if the client is an application like NetCool from Micromuse as well as for most other client applications.

- By configuring a heartbeat monitor in the client. The client watches for published data which contains a heartbeat from each node. If the heartbeat stops updating the client can report the connection to that node as having failed.

- By having an independent client node monitor the remote server nodes and having a direct connection between the front end client the monitor node it is possible to relieve the client of extra work and only monitor a reduced point list from the monitor node.

In the diagram below a monitor node has been added. The node plays two roles. Firstly is polls for a single data element from each remote server node. If the connection to a node is lost the MMI can report this by monitoring the node status as made available in the monitor node. Because the MMI client and the monitor node are on the same network they are not subject to the connection problems that can occur with the remote server nodes. In addition the monitor mode can act as a responder to periodic polls from each remote server node. This allows the remote server nodes to 'know' if they still have a connection to their subscriber.