Users of G2S and S2S can tune protocols to meet bandwidth requirement

Marc McDermott

Only large casinos and large casino companies will benefit from the G2S and S2S protocols because of the bandwidth required. Right? The good news is: wrong! Let's start with the basics. The bandwidth of a system is a statement of how much information can flow through it in a specified amount of time. For example, a cable rated at 1 megabit per second means that 1 million bits of information can flow through it in one second. Switches, routers and most other data transmission equipment have a similar specification. XML and SOAP are widely used communications standards that, while being incredible useful, have always been associated with high bandwidth requirements. These standards are also two key building blocks of the GSA's G2S and S2S protocols, which also have been deemed to require a lot of bandwidth.

The "high bandwidth" association has raised questions as to the suitability of G2S and S2S for small and distributed locations. Recently, though, the assumption that G2S and S2S require high bandwidth has come under closer scrutiny and is being found not to be the case in many applications. The "low bandwidth" capability lies in the understanding and proper use of several key factors in the GSA protocols that allow the operator and manufacturer to "tune" the protocol to meet their bandwidth requirements.

The first bandwidth reduction factor is gzip. Gzip is a compression algorithm that can reduce message size by up to 90 percent. G2S and S2S support gzip either in the http stack or in a "gzip payload" option, which was provided to allow compression of the payload information for applications that do not support gzip in the http stack. While all messages are not reduced by 90 percent, the overall reduction in message size is significant.

Another key feature in reducing bandwidth in the G2S protocol lies in philosophy of the protocol itself. Primarily, G2S is an event-driven protocol, i.e. if there are no requests for data, there is no data sent and, therefore, no bandwidth used.

To break this down into more detail, in the G2S protocol, there are requests and events. Requests come from a host to an EGM and ask for information or instruct the EGM to perform certain functions. If the request is, for example, for meter data, the EGM will return the meter information requested. Obviously, the more frequently a meter report is requested, the more bandwidth will be consumed. The point is that the amount of bandwidth consumed is controlled by the operator. The operator may choose to conserve bandwidth by making fewer requests for information.

Events are the other mechanism used to return data. Events are the result of an occurrence on an EGM. This could be a door opened, a game started, a note inserted in the acceptor and so on. Everything that happens on an EGM has an associated event that can be sent to notify an operator that the corresponding action has occurred. To receive this information, the operator sets "subscriptions" for events, and the resulting event reports. Each time an event occurs, the EGM checks to see if there are any subscriptions for that event. If

there are, the event report gets sent to the subscribing host. More event subscriptions require more bandwidth. If there are no subscriptions for events, no event reports are sent and no bandwidth is used. By managing the type and number of event subscriptions set at any one time, the operator can also manage the bandwidth used to report events.

Going one step further with event reporting, events can be sent with "associated data" such as all the meters that could be affected by the "thing" that caused the event.

An event report without associated data is small; an event report with associated data can be large. Large event reports use much more bandwidth than small event reports. Again, there is a choice to be made. And again, the operator, or manufacturer, can directly determine the amount of bandwidth used.

Bandwidth restricted applications may also use the S2S protocol. In a distributed gaming environment, where many small locations report to a central location, a common configuration includes several EGMs that connect to a small site controller. The site controller, in turn, reports EGM activity to the central location.

The S2S protocol is perfectly capable of providing low bandwidth communications in this situation. Like G2S, S2S also has events and requests and, though the event reporting function is different than G2S, the concept is similar: the more subscriptions, the more bandwidth used and the fewer subscriptions, the less bandwidth used.

The same is true for requests for information. However, S2S has another feature that enhances the low bandwidth requirements: S2S wild cards. With S2S, requests can be directed at a specific EGM, using the site controller as a conduit between the host and the target EGM. However, in other cases, S2S wild cards may be used to instruct the site controller to perform functions on all the EGMs connected. To disable all the EGMs, for example, requires a single command to the site controller. The advantage, then, is that S2S uses wild cards and the use of the site controller for an additional optimization in information being sent. This feature becomes more efficient as more EGMs are connected to a site controller.

The result of the intrinsic capabilities of the G2S and S2S protocols is that the users of the protocols actually determine whether high or low bandwidth is required. The protocols themselves are capable of successfully conveying information in either situation

The key for low bandwidth operations, then, lies in exercising the restraint necessary to request only the information needed instead of the information desired. However, because of the incredible array of information that the GSA protocols are capable of providing, staying with what is "needed" may not be an easy task. **SlotManager**

Marc McDermott is technical director of the Gaming Standards Association. For more information on the association, visit www.gamingstandards.com. McDermott may be reached at mmdermott@gamingstandards.com.