



November 20, 2009

Overview

RadBlue has concluded their message sizing analysis, preparing the accompanying spreadsheet that documents the estimated G2S message flow between Video Lottery Terminal (VLT) devices and a central lottery monitoring system in the surveyed jurisdictions. The analysis is in two parts

- the communication start-up algorithm, which occurs rather infrequently (only when the VLT is dramatically reconfigured, powered up, or resets its communication with the central system for other reasons, and
- the daily activity model, which details the daily activity that was reported by the interviewed parties in our G2S over WAN analysis.

This document discusses the assumptions used in the sizing model.

Each VLT device is assumed to have 10 active games, each of which includes 6 available denominations (any of which can be enabled by the host to be available to the player). All games (actually gamePlay devices) are in the VLT’s G2S configuration, which is always reported to the Central System for review at start-up. In addition to these gamePlay devices, the following additional G2S devices will also be included in the G2S data model. The data model describes the communication and control capabilities, along with available event and meters that can be generated by each VLT:

G2S Classes whose devices are used frequently

G2S Class	Purpose in the WAN scenario
communications	Manages G2S communications with the host
cabinet	Reports tilt events, VLT summary meters, and is used by the Central System to disable the VLT if needed
eventHandler	Used infrequently by the host to manage event subscriptions, and regularly by the VLT to generate event reports
meters	Used infrequently by the host to manage daily and periodic meter subscriptions, and regularly by the VLT to generate meter reports
idReader	Used by the VLT to convey information about an ID presented at the VLT, and by the host to indicate information about the player to whom this ID belongs
player	Used by the VLT to convey player session information to the host

G2S Class	Purpose in the WAN scenario
gamePlay	No daily activity, other than meters (which are sent via the meters class). At start-up, the host may use this class to adjust active games and / or denominations.
noteAcceptor	Cabinet level currency and total meters are sent to the host as part of the meter reports. This class may be used as part of an RG solution (G2S control commands can be used to disable the note acceptor in order to restrict player buy-ins)
printer	Included so printer tilts can be reported via event reports.
voucher	Used by the VLT to report voucher meters to the host (via meter reports), and also to convey voucher transaction information to the host. Since the voucher database is maintained on the central system, when the VLT issues a voucher, it must be conveyed to that data store, and when the voucher is redeemed, the redemption is also processed against that store. Additionally, each VLT will need to request a set of validationIds each day to use when issuing vouchers.
GAT	GAT is included as it was determined that the central system may want to validate one or more software components each day.

Additional G2S Classes whose devices are used infrequently

G2S Class	Purpose in the WAN scenario
download	Manages the downloading and installation of software packages (estimated use is an annual distribution of one or more packages of updated or new game software)
optionConfig	Used by the central system to change configuration parameters within the EGM (e.g., operating schedules, titles printed on cash-out tickets, etc.). This class will probably be used on a monthly or quarterly basis to consistently adjust parameters in all VLTs within the jurisdiction.
commConfig	This class may be needed to allow for remote configuration of the communication structure of each VLT. This will become useful if an additional server needs to be added to the central facility, or if other communication parameters need to be adjusted over time (without visiting each VLT in the jurisdiction).

A day in the life of a VLT

Communications Start-up

At startup, each VLT sends a *commsOnline* command to the central system to announce that it is present and operating on the network, provides the URL it uses for communications to that host, and indicates to the host whether that host's subscriptions are still intact. This command is immediately followed by a *commsDisabled* command that indicates the communication device used for this host is currently disabled (the VLT will not send any additional unsolicited messages to the host).

Device status - The central host will investigate the status of G2S devices within the VLT at a gross level by requesting the G2S *descriptorList*, which reveals the G2S classes and devices supported by the EGM (including ownership and guest status of each), along with the following additional information about each device:

- Manufacturer, model number, serial number, and firmware version information (if appropriate)
- The current set of configuration parameters used by each device (is the noteAcceptor using the current configuration #43257, or does it need to be updated?)
- Information about device availability (is it active, enabled, and not locked by the EGM and Host)?

Active Game denominations – Since the active denominations of each gamePlay device are not reported via the descriptorList, the host will probably also ask the VLT for the status of each of the bet denominations (active/inactive) of each of the gamePlay device to ensure they are configured appropriately.

Verifying subscriptions – to ensure that the VLT's subscriptions for this host are current, the host will probably just update the hourly and daily meter subscriptions, and then check the event subscription currently recorded for this host in the VLT, making adjustments as needed.

Verifying software inventory – we assume that several commands in the G2S GAT class will be used to review the VLT's software inventory, and also to reverify selected components to ensure that they are the approved versions.

If any adjustments are needed to any of the G2S devices, , the host can then use additional G2S control commands to make the appropriate adjustments, but in most cases, the *descriptorList* will indicate all is well, so the host can just set the *keepAlive* frequency and then enable the communication device and the VLT can begin its normal functioning.

Frequent Communications

During the course of each gaming day, the following activities will happen on a regular basis:

keepAlive – the communications heartbeat – we assume that the VLT will initiate a communications check with the central system after 5 minutes of no other G2S activity (though this value is configurable). The *keepAlive* lets both the VLT and the host know that the other is alive and well, and that communications are active.

events – during the course of the gaming day, a number of G2S *eventReports* will be sent from the VLT to the host. These fall into the following categories:

- *noteAcceptor* events – the Note Stacked event can be used to report player buy-ins (via currency) at the VLT. Current estimate is 2 note insertions per hour, as tracked note activity may only be relevant for *carded* players.
- Cabinet events – these report tilts, door activity, etc. to the host
- Printer events – used to report printer malfunctions to the central host (especially important if vouchers are the only method to cash out at a VLT)

Voucher activity – In the G2S model, validation numbers used on vouchers are provided by the host system responsible for voucher activity (maintaining the voucher database). The *getValidationData* and *validationData* command pair are used by the VLT to request additional IDs as needed. With proper data management, the VLT can request the IDs it needs for the upcoming day during the period of the facility being closed, and then consumes them throughout the day. In our model, we assume this to be the case, and that 10 vouchers (on average) will be produced by each VLT, and then redeemed using the network as well.

Player Id Activity – Since the lottery retail establishments are typically small facilities with a minimal number of gaming devices, we assume that tracked players will probably be comfortably seated at a VLT for an extended period of time. Player ID activity and the related player session information reports are needed for RG tracking and player activity accumulation.

Software verification – Since the capability is available, the central system may choose to validate one or more software modules each day, especially after door activity occurs.

The G2S activity has been presented using an Excel spreadsheet to make it easy to adjust the model. Message sizes will remain consistent, but the number of transactions per day of each type may adjust based on revised estimates of activity.

EGM	Dir	Host	gzip'ed Size	Quantity	Total Size	Comments
commsOnline	→		500	1	500	Can be used to determine if subscriptions are still intact, and if device list has changed since last startup.
	←	commsOnlineAck	478	1	478	
commsDisabled	→		502	1	502	
	←	commsDisabledAck	480	1	480	
	←	getDescriptor	517	1	517	
descriptorList	→		1342	1	1,342	Can be used to verify status and configuration ID of all devices
	←	getGameDenoms	462	10	4,620	verify active denoms on each game, repeated for every active game in cabinet
gameDenomList	→		546	10	5,460	(Assumes 10 games / VLT)
	←	setMeterSub (Periodic)	635	1	635	Rather than requesting current meter subscriptions, just set correct subscription.
meterSubList	→		659	1	659	meters = gamePlay, voucher, noteAcceptor, gameDenoms at cabinet level
	←	setMeterSub (EOD)	635	1	635	
meterSubList	→		659	1	659	
	←	getEventSub	525	1	525	Get current event subscription to verify current sub, since set and clear commands are additive
eventSubList	→		1125	1	1,125	
	←	setEventSub	1014	1	1,014	Only needed if event subscription needs to be "tuned" Setting appropriate events
setEventSubAck	→		504	1	504	
	←	clearEventSub	613	1	613	Only needed if event subscription needs to be "tuned". Clearing 10 events
clearEventSubAck	→		503	1	503	
	←	setDateTime	481	1	481	Not needed if local time source is in use, or if VLT clock is within 5 seconds
cabinetDateTime	→		508	1	508	
	←	getComponentList	464	1	464	At startup, request component list for daily GAT activity
componentList	→		760	1	760	

EGM	Dir	Host	gzip'ed Size	Quantity	Total Size	Comments
	←	doVerification	547	5	2,735	Probably want to check a few modules at start-up
verificationStatus	→		559	5	2,795	so this sequence may be repeated multiple times
verificationResult	→		664	5	3,320	[includes SHA256 result]
	←	verificationResultAck	532	5	2,660	
	←	setKeepAlive	478	1	478	Set EGM keepAlive frequency to central (maybe 5 minute heartbeat?)
setKeepAliveAck	→		505	1	505	
	←	setCommsState	500	1	500	
commsStatus	→		580	1	580	
Total Size (bytes)					36,557	

Additional *setDevice* State commands may be needed if devices are not already enabled

Additional option configuration commands may be needed if configurationId is not as expected

EGM	Dir	Host	gzip'ed Size	Frequency /VLT/Day	Total Size (bytes)	Comments
keepAlive	→		502	96	48,192	assumes a heartbeat every 5 minutes; value to right reduces keepAlive count by the count of other messages sent from the VLT throughout the day (a keepAlive is sent after 5 minutes of no activity).
	←	keepAliveAck	467	96	44,832	
eventReport	→		785	10	7,850	cabinet events - tilts - with status, no meters
eventReport	→		882	10	8,820	cabinet events - door activity with status + meters
	←	eventAck	472	20	9,440	(assumes 20/day total)
eventReport	→		1050	40	42,000	note acceptor events for RG purposes (modeled using Note Stacked, includes note meters + status)
	←	eventAck	471	40	18,840	(assumes 2/hour)
eventReport	→		735	1	735	printer events (tilts)
	←	eventAck	471	1	471	(assumes 1/day)
meterInfo	→		1225	24	29,400	Periodic Meter Reports - 1/hour
meterInfo	→		1225	1	1,225	EOD Meters 1/day
meterInfoAck	←		476	1	476	EOD Meters are acknowledged
getValidationData	→		562	1	562	VLT gets voucher IDs from central host
	←	validationData	1053	1	1,053	(assumes 1/day)
issueVoucher	→		684	10	6,840	10 vouchers issued and redeemed per day
	←	issueVoucherAck	479	10	4,790	
redeemVoucher	→		563	10	5,630	
	←	authorizeVoucher	615	10	6,150	
commitVoucher	→		660	10	6,600	
	←	commitVoucherAck	480	10	4,800	
getIdValidation	→		520	40	20,800	Player Id inserted for RG
	←	setIdValidation	672	40	26,880	(assumes 2/hour)
playerSessionStart	→		566	40	22,640	

EGM	Dir	Host	gzip'ed Size	Frequency /VLT/Day	Total Size (bytes)	Comments
	←	playerSessionStartAck	576	40	23,040	
playerSessionEnd	→		782	40	31,280	May be a session end command or a PRE101 (Interval Rating) event
	←	playerSessionEndAck	485	40	19,400	However, there is no buy-in info in either, so this may not be the correct solution for RG.
	←	doVerification	547	5	2,735	On a daily basis, probably want to recheck a few modules using GAT
verificationStatus	→		559	5	2,795	
verificationResult	→		664	5	3,320	[includes SHA256 result]
	←	verificationResultAck	532	5	2,660	
Total Size (bytes)					404,256	
Size per Hour					16,844	
Size per Minute					281	