WHITEPAPER

Resilient Cloud Infrastructure

Uncertainty of reliability is frequently cited as a primary hurdle in adopting cloud-based systems for live media production. This concern is based primarily on unfamiliarity with the technology — not surprising at this stage in the adoption curve — rather than inherent instability in cloud technology. While use of cloud technologies for mission-critical functions may be new to the media & entertainment industry, it is widely used in many other industries.

Combining existing practices from both IT and M&E experience can provide highly available production systems measured in multiple nines of reliability. This paper proposes several alternative system architectures for high availability and introduces AMPP Hub, a propriety solution from Grass Valley that maintains system function for AMPP — even if internet connectivity is temporarily lost.

9's	Availability
One	90%/yr
Two	99%/yr
Three	99.9%/yr
Four	99.99%/yr
Five	99.999%/yr
Six	99.9999%/yr

Reviewing the Architecture

Availability is the probability that a system is operating correctly and is able to perform its function at a given instant of time. Availability includes ensuring all authorized users have access to the systems and the resources they need to perform their task. Failures are unplanned outages of a system component.

The computing architecture model, as illustrated in the table below, is useful in reviewing possible failures and how they are recovered. All layers of a computing architecture in a serial relationship must be functioning for the system to be fully available. This means that all the system's application building blocks are working as expected and can communicate together correctly. This is true for systems that run on local hardware as well as in the cloud.

The layers have similar functions regardless of where they operate.*[1] Where a system runs affects how those functions are performed. As illustrated in the table, who provides that function may also be different.

*1 For an overall discussion of broadcast system availability see Jay Bergman, Achieving High Availability in Television Broadcast Systems, December 1, 2021

Computing Architecture Layers	Technology Platform		
	Hardware SDI	Hardware IP	Cloud
Control	Vendor 2	Vendor 2	Vendor 1 or 2
Communication	Vendor 1	Vendor 3	Vendor 1, 3 and 4
Application	Vendor 1	Vendor 1	Vendor 1 or 2
Operating System	Vendor 1	Vendor 1	Vendor 1
Hardware	Vendor 1	Vendor 1	Vendor 5

Vendor 1 = Broadcast manufacturer 1

Vendor 2 = Broadcast manufacturer 2

Vendor 3 = IP switch manufacturer

Vendor 4 = Internet provider

Vendor 5 = Cloud provider

Take for example a production switcher:

- ***Hardware SDI:*** The processing engine with its base OS, as well as software features such as EMEMs are connected to a control panel through a custom cable. All are provided by a single vendor. The switcher vendor may provide a control protocol or GPI I/O for control by third-party automation.
- ***Hardware IP:*** The switcher vendor still provides the processing engine with its base OS, as well as software features such as EMEMs. Connection to a control panel, however, is managed over an IT LAN network. Third-party automation protocols remain available.
- ***Cloud:*** The switcher processing is now transferred to a cloud data center where the virtualized switcher functionality runs as a software application. The communication between the virtualized components is typically developed by the switcher vendor. Third-party applications could be developed if the operating system were opened to other vendors. Communication from the end user's on-prem control panel would be transmitted over internet. On-prem as well as hybrid cloud would have a similar architecture.

As shown in the computing architecture chart, the number of possible vendors involved in a system is higher when operating in the cloud. ***Reducing the number of vendors by adopting common platforms and solutions rather than "cherry picking" solutions and integrating them in-house will tend to simplify troubleshooting and recovery.***

Calculating Availability

To calculate the availability of a system, we need to know the MTBF*[2] and MTTR*[3] of each component.*[4] The MTBF values are usually obtained from the vendor.

Because they represent the time required for a service person to get the system running, MTTR values vary by location. MTTR for hardware components can be estimated on a time and part availability basis. MTTR for software is usually the time it takes to reset the software.

With the MTBF and MTTR values we can calculate system availability. Note that each layer of the computing architecture has its own MTBF and MTTR.*[5]

$$Availability = \frac{MTBF}{MTBF + MTTR}$$

A quick way to estimate the total availability of the system is to use the lowest level of availability of any component. Because all layers must be working for the system to function, the availability of the entire system will be slightly lower than "the weakest link."

*2 MTBF = Mean Time Between Failures (measured in hours)

*3 MTTR = Mean Time To Recovery (measured in hours)

*4 For a detailed explanation of how to calculate availability, see Hoda Rohani, Azak Kamali Roosta, Calculating Total System Availability

*5 Standard practice for availability ratings assumes that the hardware or software is operated in an environment conducive to proper function. This means that the probability of hardware or software failing does not include the probability of such factors as power outage, cooling failure, misconfiguration by the administrator or even operator error. The availability of these components should be taken into the account separately.

Traditional Redundancy

The calculation above assumes a single system with serial dependencies where a single failure can lead to a cascading failure across the system. However, the availability for the total system can be improved by pairing it with a parallel system. Because parallel configurations offer more opportunity for maintaining availability while a system recovers, the total system availability is greater than the availability of each of its components.

The general formula for adding n parallel components is:

$$Parallel\ Availability = 1 - \prod_{i=1}^{n} (1 - Availability_i)$$

Traditionally, the M&E industry has installed a fully-duplicated system wherever practical. Ideally these systems were also geographically separated. An automatic fail-over mechanism would switch between the systems when a fault was detected.

Where fully duplicated systems weren't practical, lower-cost workarounds such as a router panel backing up a production switcher were installed.

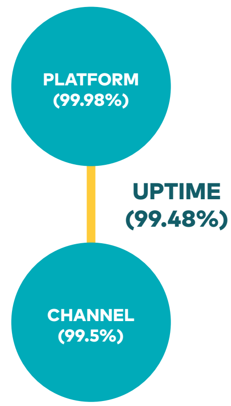
These same principles work in cloud redundancy. Just as with a traditional system, there are a variety of options based on preference.

Cloud Redundancy Options

The following three examples show how system availability may be calculated.

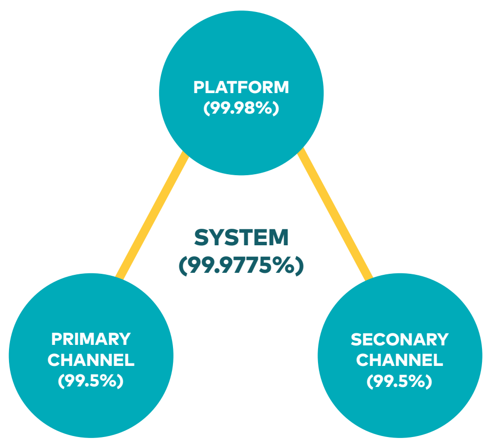
Architecture #1: Single Platform + Single Channel

- Platform (Uptime=99.98%)
- Primary Playout Channel (Uptime=99.5%)
- Outage happens with Platform ***OR*** Channel outage
- Uptime Math: 1-Uptime = (1-Platform)+(1-Channel)
  - Uptime = 1 – [0.0002+0.005]
  - Uptime = 99.48%



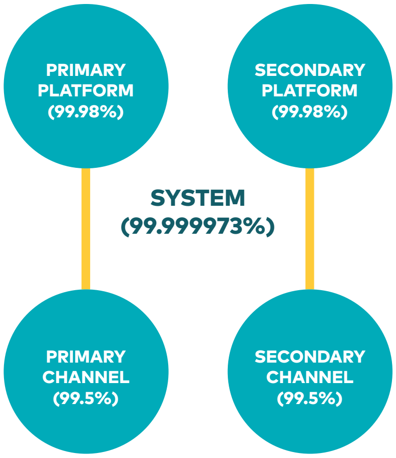
Architecture #2: Single Platform + Redundant Channels

- Platform (Uptime=99.98%)
- Primary Playout Channel (Uptime=99.5%)
- Secondary Playout Channel (Uptime=99.5%)
- Outage happens when platform ***OR*** both channels fail
- Uptime Math: 1-Uptime = (1-Platform) + [(1-Primary_Channel) * (1-Secondary_Channel)]
  - Uptime = 1 – {0.0002 + [0.0000025]}
  - Uptime = 99.9775%



Architecture #3: Dual Platform + Redundant Channels

- Primary Platform (Uptime=99.98%)
- Primary Playout Channel (Uptime=99.5%)
- Secondary Platform (Uptime=99.98%)
- Secondary Playout Channel (Uptime=99.5%)
- Failure happens when either primary platform or channel fails AND secondary platform or channel fails
- Uptime Math: 1-Uptime = [(1-Primary_Platform) + (1-Primary_Channel)] * [(1-Secondary_Platform) + (1-Secondary_Channel)]
  - Uptime = 1 – {[0.0002+0.005] * [0.0002+0.005]}
  - Uptime = 99.999973%



As illustrated in Architecture #3, systems based in the cloud using a 1+1 hot standby can match the high availability of terrestrial systems. Unlike traditional systems, where providing separate systems in different regions is complex, providing geographically separated systems in the cloud is a common practice for enterprise-level systems.

Note 1

In the generic systems shown above, the failover arbiter system is not defined. The performance of the arbiter depends on the cloud provider. In our examples, the MTTR when switching from one system component to another is not included. However, it is likely to be one or more video frames. Specific information on expected failover MTTR should be confirmed with the cloud provider.

Note 2

The calculations above assume a worst case scenario with uptime being the guaranteed minimum. In actual practice, the chance of a major failure simultaneously happening across two independent systems is actually much lower than the probability would suggest.

AMPP Hub

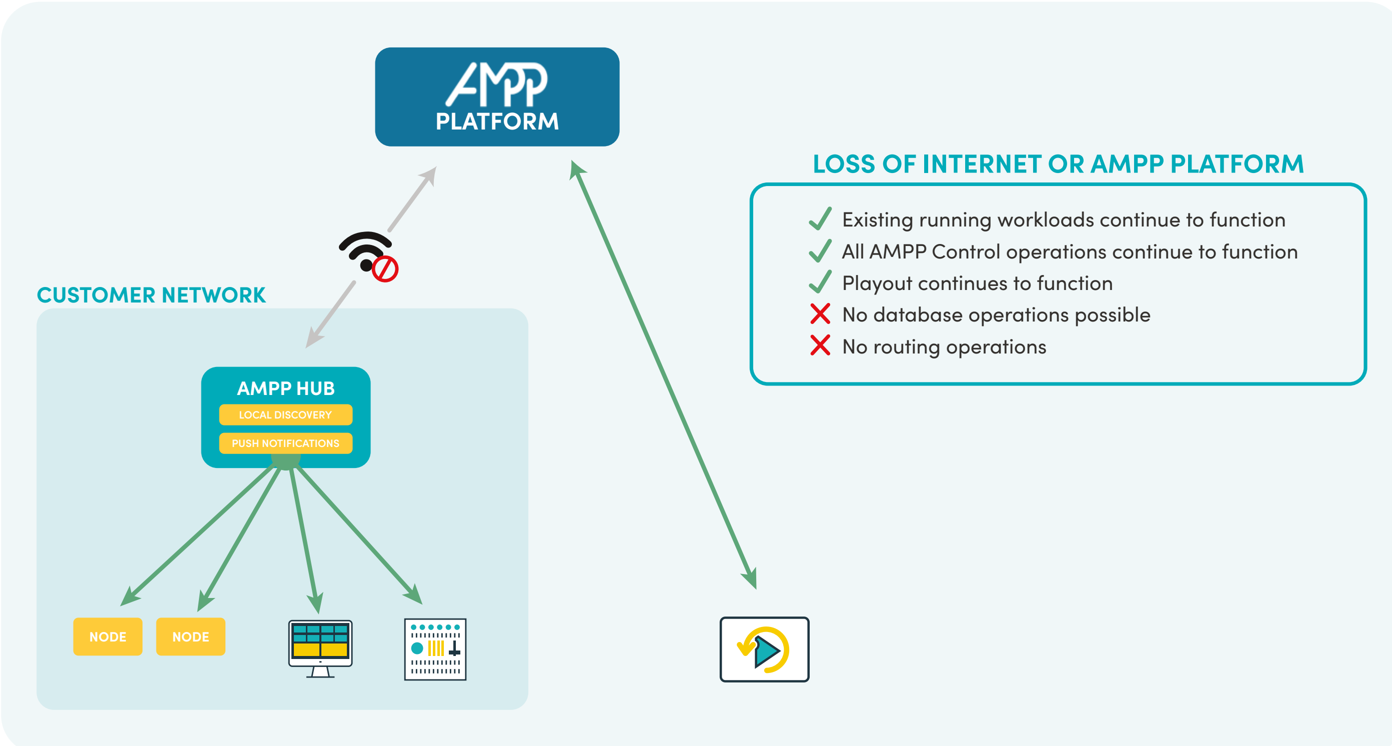
The generic information on cloud resiliency as described above is applicable to Grass Valley's AMPP platform. To provide even further availability in this environment, AMPP offers a unique feature called AMPP Hub. ***AMPP Hub temporarily masks a loss of connection with the cloud platform by maintaining local functionality.***

AMPP Hub requires a Linux-based server connected to the local network that is configured as a secure SSL endpoint for AMPP. Commands from user interfaces connected to the local network never go to the cloud but are immediately redirected locally. Remote operators who don't have access to the local network interact with the cloud instance.

If there is a temporary break in internet service, the server continues to function. Despite the lack of communication to the cloud, all running workloads remain. Control commands from soft panel UIs or connected devices are executed. The interfaces continue to display new keyframes. While no system changes can be made during the time the connection is lost, the local creative team can continue to produce a show. Remote team members will still have access to the cloud instance but they will no longer see the work of their local colleagues until communication is restored.

Once connection to the platform has been re-established, the MTTR for resynchronizing the system between AMPP Hub and the rest of the AMPP platform is within a matter of seconds.

Emergency Operations



A Practical Illustration

To see how the availability is maintained, let's assume that we are producing a live event from a truck located outside a stadium. IP cameras from the stadium are being fed to the truck, which is creating a live production using AMPP solutions. If the signal between the truck and the cloud platform went down, here is how the various functions would be affected:

Routing of signals to each work station	Routing would remain in the state when the signal was lost. No changes could be made to routing until the system is recovered.
Camera shading	Cameras connected to control panels would still have shader control. Would not be able to switch control panel to another camera.
Production switching	Production switching would continue as normal. Program and Aux outputs would reflect all actions taken on the switcher. Clip Store may be unavailable if using cloud rather than on-prem server for Clip Store service.
Highlight marking and recording	Replay is based on Elastic Recorder X. Therefore unavailable until system is recovered.
Replay contribution back to the production switcher	Replay is based on Elastic Recorder X. Therefore unavailable until system is recovered.
Elastic recorder	Unavailable until system recovered. This could affect not only replay but also recording of switcher PGM and AUX outs, as well as editing and asset management.
Graphics contribution to the production switcher	If using HTML5-based graphics, these would be unavailable until system is recovered.
Editing	Unavailable until system recovered.
Asset management workflows	Unavailable until system recovered.
Playout	If playout is generated from the truck, it will continue. If playout is generated from the cloud, the program upload would be lost from the truck and the signal would require management from an operator at a different location.

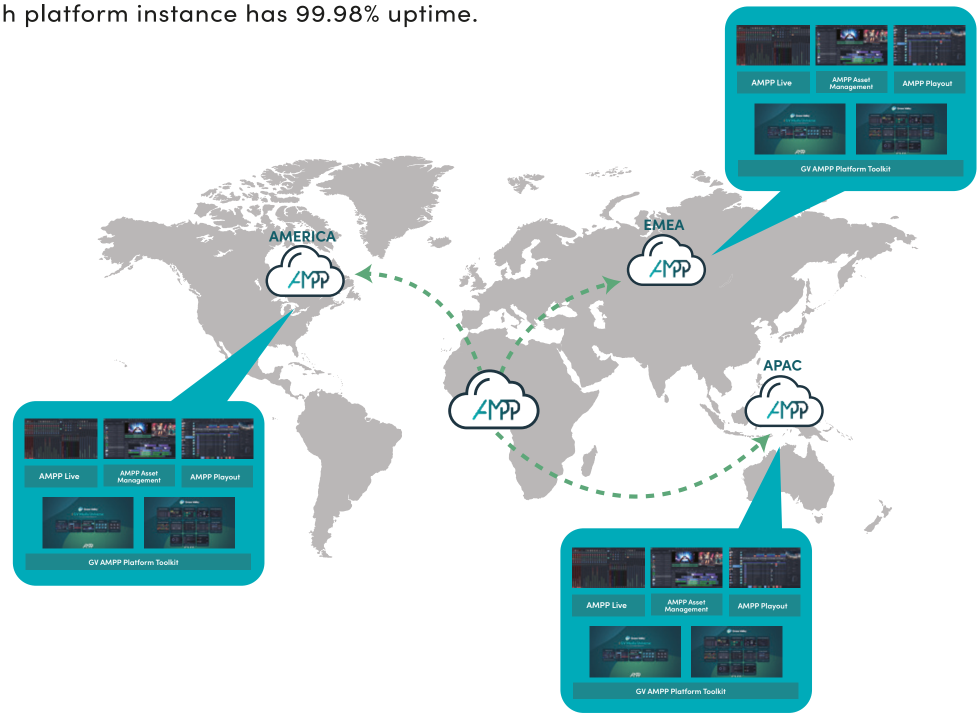
As the table illustrates, it is possible, with correct system configuration, to keep the program running even if the connection between the truck and the cloud platform is temporarily lost.

Deploying AMPP as a High Availability Platform

Even though a connection loss can be hidden, keeping the connection loss as short as possible is the reason for the generic cloud models shown earlier.

To back up the platform, AMPP can be deployed as two or more instances with redundancy on a global scale. AMPP users may deploy media workloads wherever they wish as the platform functionality may be fully replicated in each zone. Each platform instance has 99.98% uptime.

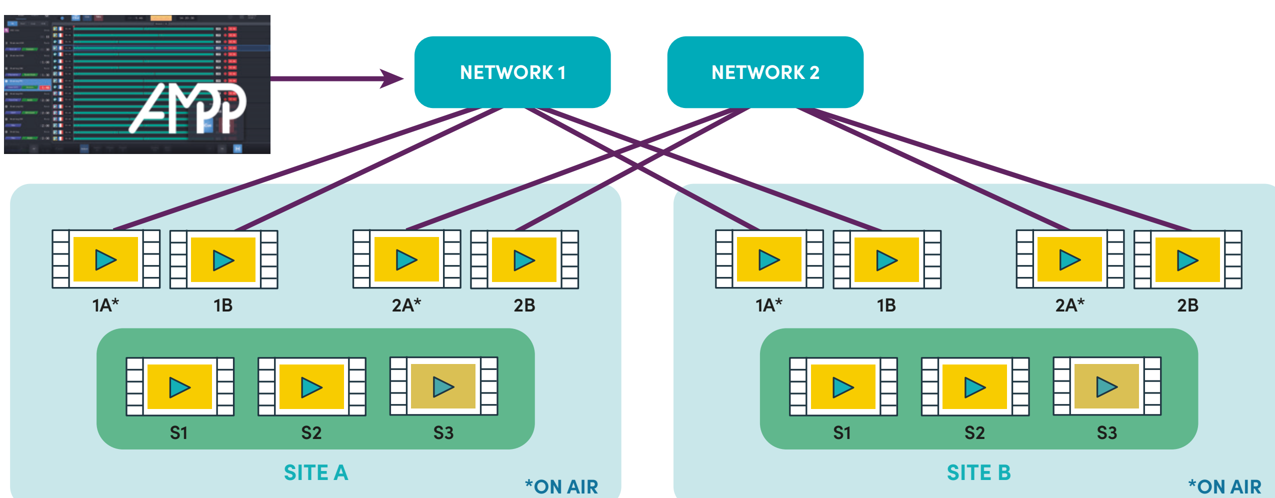
Because the loss of the connection may be due to the failure of the local internet service provider, a 5G connection may be used to maintain connection to the platform. In our truck example used earlier, this would be enough bandwidth to maintain routing and a 720p program feed uploaded to the cloud. A 1080p program feed would require bonded 5G cellular.



Flexibility

Not all channels or events are created equal. The loss of a marquee event, such as a championship-deciding football game, would have much wider commercial and reputational damage than the loss of a late-night soap opera re-run. AMPP offers a wide choice of resiliency strategies for playout that can be applied on a per channel or event basis.

Let us look at a real-world use case where GV Playout X uses AMPP's ability to separate roles from physical compute. The customer was providing hundreds of channels for a live global sporting event. Any failure would have had severe consequences to their business and brand. For simplicity, the diagram below shows only two of the channels.



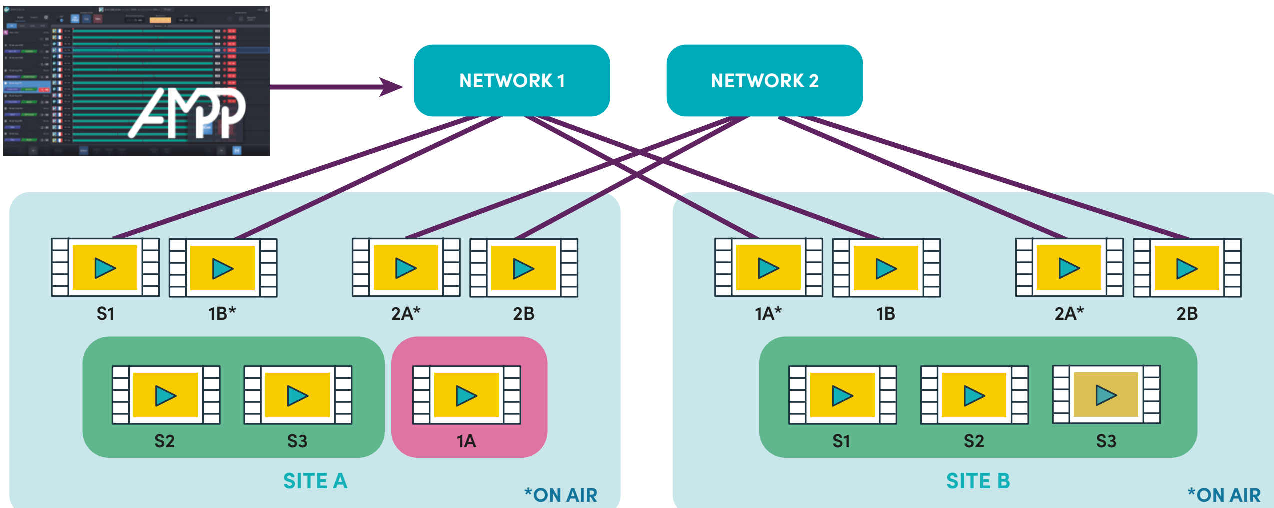
The customer has their playout in two mirrored sites, one in the UK and one in the Netherlands. It was their decision to have both sites on-premise, however either or both sites could have been hosted in a public or private cloud.

The operators were based in France, though with AMPP's HTML5-based UI they could have been anywhere in the world — including their homes.

The channels were configured to have a 1+1 hot backup. In addition, they chose to have a small pool of spare channels on each site.

In the case of a primary failure, in this case channel 1A, several things happen automatically:

- The hot standby 1B immediately becomes the on-air channel and all video routing is performed
- A spare is automatically promoted from the pool to become the new hot standby
- The failed channel is moved to a failed pool so remedial action can be taken and it can be moved into to the spares pool for future use



The user has not only maintained on air continuity, but they have also maintained their resiliency.

This kind of real-time reconfiguration and assignment of roles is not possible with traditional solutions tied to hardware.

Depending on the risk profile, other solutions are available:

- Single site or a site in a private/public cloud
- More than one hot standby per channel, offering double or triple hot redundancy
- Using a shared spares pool for redundancy with no hot backup

The choice of which strategy to pursue can be based on business need rather than any hardware or architectural limitation. This strategy is not then fixed in stone, and can be changed without huge engineering effort.

Summary

While new technology can make users nervous, the fundamental principles behind creating M&E systems have not changed. Finding the right level of system redundancy can provide the high availability required for high-profile projects. Because cloud solutions allow systems to quickly scale up and down with a range of options, right-sizing the level of redundancy is easier than ever before.