GRIDOBSERVER® DETECTS PHANTOM ADJACENCIES

While working on a mid western smart grid network, a curious anomaly arose. Base routing tables were growing at an unexpected rate.

Routing tables are pieces of information each router holds that not only tell them who they are directly connected (adjacent) to, but also allow the routers to perform complex algorithms on this information to determine the paths to all other routers on the network. As these tables grow, the calculations become increasingly more complex and the router processing load is increased. If these tables grow too large, the routers will become overwhelmed trying to perform the calculations, begin to fail, and make parts of the network unavailable.

In this network, all equipment was sized properly, configurations were checked and found to be correct, and connection counts were well within normal operating parameters. But still … routing tables were increasing at an unexpected rate.

Introduce: Grid Observer

Grid Observer (GO) is a different kind of Network Management System (NMS).   Most network management tools watch equipment for failures and possibly even anomalies. Most NMS's would have seen the processing loads increasing on the routers, and may have even seen the routing tables increasing. But only GO would have – and did – show *why*.

The Grid Observer NMS is an atomic and geospatial detective. That means GO not only watches the network for up/warning/down states (like everyone else), not only tracks and graphs critical parameters (like everyone else), but it goes even deeper and models the deep inter-relationships for network equipment. Forget looking at network elements … we need to look at – *and map* – the atomic inter-relationships of those elements.

When GO mapped the network elements in this case, it revealed that a particular type of wireless equipment wasn't performing as advertised. This equipment was point-to-multi point wireless equipment. This topology was supposed to look like a hub and spoke conversation with no spokes being able to directly communicate – only through the hub. But GO discovered that all spokes were communicating with each other. And, to make it worse, they were communicating badly.

This phantom adjacency caused the routing tables to increase dramatically.   Lets look at the simple math on a single hub and spoke tower:

§ A tower with N spokes would have N adjacencies or 16 spokes would have 16 network adjacencies.

HOWEVER …

§ With the spokes communicating directly with each other the adjacency count is N*(N-1)/2 or for a tower with 16 clients, 120 adjacencies with 104 in a constant state of flux.

Multiply that by the number of towers on the network, and *stability is seriously compromised.*

In this case, the vendor was contacted, given the definitive proof that GO supplied, and a patch was quickly provided.

--

Multiscopic Network Infrastructure Architect