White Paper

A Practical Approach to Wireless 2.0

Aerohive

Aerohive Networks, Inc.
3150-C Coronado Avenue
Santa Clara, California 95054
Phone: 408.988.9918
Toll Free: 1.866.918.9918
Fax: 408.492.9918
www.aerohive.com

Table of Contents

Introduction

If you haven't heard about the all-wireless enterprise, then you've missed an important networking paradigm shift. Wi-Fi networks offer secure mobility and a tremendous ROI. In the early days of Wi-Fi, significant challenges included management, security, mobility, deployment, reliability, and radio resource management. Yes, basically everything was a challenge. Those were certainly interesting times for the Wi-Fi market. Now, Wi-Fi vendors regularly tout utility-like reliability, stating that their infrastructure platforms can support mission-critical applications and be deployed and managed with minimal expertise.

Rhetorical Question: Have *you* ever seen or experienced a Wi-Fi network that was as reliable as a light switch? Me either.

That brings me to the point of this whitepaper: Wireless 2.0. First, let's define what Wireless 2.0 is. Perhaps an easy way to do that is to give you a list of characteristics of what we think Wireless 2.0 is:

- The pervasive, flexible, and self-healing primary access layer
- As deterministic as Ethernet, but with higher reliability and ROI
- As resilient and scalable as the Internet
- Secure and seamless mobility that is invisible to the user
- Integration with and leverage of the existing Ethernet infrastructure (as opposed to an overlay)
- Powerful, easy-to-use, and virtualized platform management that is available in a variety of delivery modes and cost models, including cloud-based.

To summarize Wireless 2.0 in a single statement, you could say it's, "Wi-Fi that works, so you don't have to." In order to meet these requirements, Aerohive has taken the approach of simplifying the WLAN through:

- Creation of a resilient, flexible, and intelligent Wi-Fi infrastructure platform that replaces WLAN controller hardware with link-state protocols that operate between intelligent access points (APs).
- Creation of a robust, low-cost, and adaptable virtual Wi-Fi management system that is available in a variety of modes and cost models.

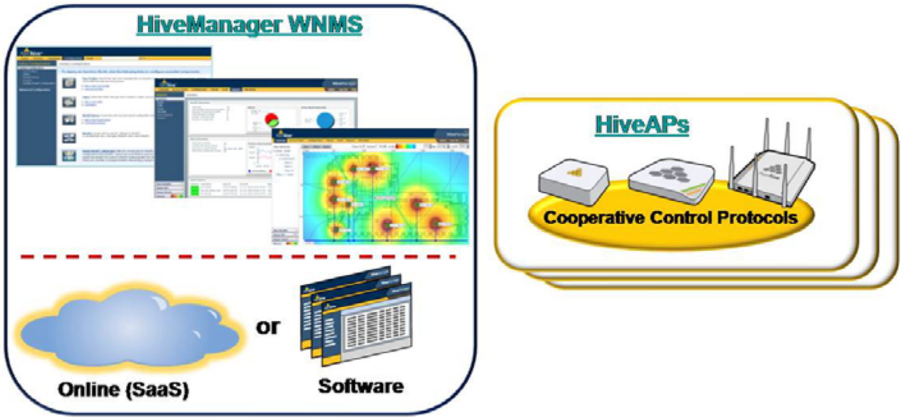


Figure 1. **Simplifying the WLAN**

Creating "Wi-Fi that works", even with minimal requirements, is a tall order given the breadth of client and application types that must perform well over the wireless infrastructure, but when adding in the speed and complexity of 802.11n, a variety of demanding applications, high-density environments, and tricky deployment scenarios, controller-based vendors cannot live up to their promises of Ethernet-like determinism. Creating Ethernet out of Wi-Fi takes a feature-rich platform built on a resilient, fully-distributed architecture. Anything less than controller-less simply isn't capable of Wireless 2.0.

A fully-distributed architecture removes bottlenecks, single points of failure and unnecessary costs while introducing linear and unlimited scalability, stronger security, and deployment simplicity.

Chronological Progression

The Wi-Fi infrastructure industry has moved through several distinct phases, and is now entering a new phase. The order of this product progression is shown below, ending in where the industry is now headed:

1) **Autonomous** – Standalone APs, managed individually
2) **Autonomous** – Standalone APs, managed with a Wireless Network Management System (WNMS)
3) **Controller-based** – Dependent APs, coordinated and managed by a single controller
4) **Controller-based** – Dependent APs, coordinated by one or more controllers, which are managed by WNMS
5) **Controller-based** – Dependent APs, coordinated by a group of controllers, managed by WNMS, that are adapted to help transition the WLAN to high-throughput, mission-critical support
6) **Controller-less** – Intelligent APs, managed by WNMS, that use inter-AP protocols to coordinate all control tasks for high throughput, deployment flexibility, and mission-critical support

It no longer seems visionary to publicly announce that WLAN infrastructure will ultimately have fully-distributed intelligence. In fact, many vendors have already shown enough of their cards that it's simple to see where their roadmap leads. Some have even been so bold as to release their product roadmaps publicly, confirming their long-term intentions of having a fully-distributed architecture. The latest industry reports show that a significant portion of the market has still not adopted the controller-based model, continuing to buy autonomous APs while seemingly awaiting a more robust and economical solution.

For controller-based vendors, the march toward a decentralized architecture began with distributed data forwarding, eventually evolving into branch office solutions. It's in the branch office where they realized significant technical and financial challenges such as fast/secure roaming and redundant controllers for high availability. This immediately sparked additional development efforts to push more features back into the AP itself, but this too presented a roadblock in that controller-based APs were not designed with the processing power to replace the controller itself. In the near future, this will mean re-architecting AP hardware and software in order to be competitive with fully-distributed products, which takes time and introduces considerable developmental growing pains.

The Edge of Intelligence

There are multiple important facets to WLAN networking, and a very important one is system architecture. There are currently three kinds of architecture available in the market: 1) centralized (where products have been), 2) hybrid (where most products are now), and 3) distributed (where most products will be in the future). The Internet. Your routed and switched network. Personal computers. They're all shining examples of where distributed computing had the last word.

Rhetorical Question: Why would you want only one device performing a set of functions when you could have hundreds, or even thousands of devices performing those functions simultaneously?

Example: You could have one controller doing all of the processing tasks for 1,000 APs or you could have 1,000 APs processing tasks for themselves. Which scenario offers more computing power? Which is more scalable? Which is more resilient?

Of course, the first argument against the distributed approach is cost, but since controllers arrived on the market in 2003, Moore's law has been hard at work reducing AP component prices and increasing computing power. Today, high-powered CPUs and high-speed memory chips are commodities that can be purchased very inexpensively. In this paradigm shift, it's now the controller's lack of computing power and high cost that is being called into question by system integrators and customers.

With Wi-Fi network reliability being repeatedly touted by analysts as the most important criteria among customers, the question looms large, "why would a customer accept anything less than superior reliability?" Every Wi-Fi infrastructure vendor is now pitching their solution as the most reliable (you've probably heard phrases like, "Wi-Fi as a Utility"), even though the architecture on which most vendors have built their solution is far less than dependable. Every Wi-Fi infrastructure vendor knows that a fully-distributed architecture will win out in the end, and because of this, they have begun moving key features into their APs within their latest code revisions.

Performance

The controller-based, centralized forwarding model introduced a significant bottleneck, but in the days of 802.11a/b/g, it was rarely noticeable because Wi-Fi networks were just getting their start. Today, with 802.11n as the leading Wi-Fi technology, Wi-Fi vendors have to build colossal controllers to keep up with the throughput demands and even still, they aren't fast enough.

Controller-based vendors are addressing this problem with distributed data forwarding (also called local forwarding), stating that this helps alleviate controller bottlenecks. In reality, this causes worse problems than it fixes because forwarding data directly from the AP without applying *stateful* security and QoS policy to it is just irresponsible and promises to disrupt application performance. Having all intelligence in the AP allows for fully-distributed data forwarding with security, QoS, and other policies applied at the edge so that customers can leverage the speed of their Ethernet infrastructure.

While you might hear the term "airtime fairness" tossed around a bit these days, it's a term that means different things to different vendors. The concept generally means giving clients an amount of transmission airtime rather than just an opportunity to transmit one or more frames onto the wireless medium. By dealing in airtime instead of in 802.11 frames, an appropriate (or purposefully inappropriate) amount of throughput can be given to each client based on its current data rate. Aerohive calls this feature Dynamic Airtime Scheduling, and it's a powerful feature on top of which our SLA engine operates. Without solid and granular airtime control, it's impossible to offer SLAs, and Aerohive's Dynamic Airtime Scheduling has allowed us to be the first Wi-Fi vendor to offer SLAs as a system feature. It's a giant leap toward Ethernet-like, deterministic performance.

Question to Ponder

How many controllers will you need to meet your applications' high-availability throughput requirements 3 years from now?

Reliability

While performance is very important, reliability is even more important. Who cares how fast your network is if it's constantly offline, right? ISPs selling ADSL service learned this valuable lesson in the 1990's, prioritizing reliability (moving from ~80% to ~99% uptime) over speed (moving from 1.5 Mbps to 6 Mbps). Here it is 2010, and we're just now starting to see ADSL transition to VDSL, with speeds up to 18 Mbps. Apparently they made the right decision because the ISPs who prioritized reliability are dominant players now.

In a fully-distributed WLAN, failover/failback, best-path forwarding, and dynamic mesh routing features allow for a self-healing, and thus highly-reliable, infrastructure. By routing around failures, such as APs, Ethernet switches/routers, and cut cables, and leveraging multiple Ethernet backhaul paths and stateful forwarding protocols, a fully-distributed Wi-Fi network can actually be <u>more</u> reliable than its Ethernet counterpart.

Since the invention of the controller, Moore's law has made it possible to introduce extremely powerful AP hardware at low prices, enabling the removal of the previously necessary controller hardware. Now, the control plane can exist as protocols operating between APs, and as the Internet has shown us, protocols are both resilient and free. Elimination of controller hardware removes a performance bottleneck and a single point of failure while increasing security and significantly decreasing costs.

Question to Ponder

Given the option, would you prefer that your Wi-Fi infrastructure be <u>more</u> reliable than your existing Ethernet infrastructure?

Scalability

A controller-less architecture allows for linear and unlimited scalability, eliminating the stair-step model where additional or larger controllers must be introduced, along with the complexity and expense to configure and manage them.

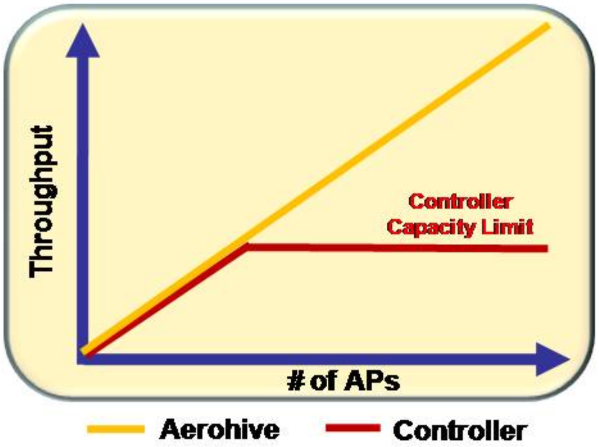


Figure 2. **The Controller Stair-Step Model Versus Aerohive's Controller-less Model**

In a fully-distributed model, a network design can start as small as a single AP, and linearly scale by simply adding more APs. Since processing, filtering, and forwarding are fully-distributed, scalability is unlimited and minimal network planning is required for deployment. Additionally, network upgrades are as simple as 1:1 replacements and/or deploying additional APs. Centralized architectures impose unnecessary design restrictions and cost structures, which complicate deployments and make budgeting difficult at best.

Controllers suffer from all sorts of limitations that cause scaling problems, some of which you probably wouldn't consider until you ran headlong into it. For example, suppose that you had two large controllers, in high-availability configuration, supporting 1,000 APs (just to have a round number). If each AP has dual radios and support 3 SSIDs per radio (e.g. HQ with 802.1X/EAP, Guest with Captive Portal, WPA2-PSK for clients that don't support 802.1X/EAP), then each AP will build 3,000 tunnels to the primary controller and 3,000 tunnels to the backup controller. That's 6,000 tunnels on the network, all of which are unnecessary. Why should the same traffic traverse the network infrastructure twice (once in a tunnel, and once outside of a tunnel)? Controller-less WLANs simply don't have to deal with this kind of unnecessary overhead.

Question to Ponder

Given a choice of unlimited & linear or limited & non-linear scaling, which would you choose?

Security

In today's market, some security features are now table stakes. 802.1X/EAP and PSK authentication, CCMP/AES encryption, secure management, fast/secure roaming, and rogue detection/mitigation are among these "must have to survive" features. More advanced security features include Private PSK, VPN, policy-based stateful firewall, and directory system integration, and those items are a few where vendors strive to differentiate themselves. Not all vendors have each of these types of security, and what's more, not all vendors implement these features in the right way.

What is "the right way" to implement security? It certainly depends on the type of security you're discussing, but as an example, let's talk about stateful firewalls. Firewall policies are applied to users or groups of users and follow them wherever in the network they may roam, regardless of which AP the client is connected to. Both controller-based and controller-less implementation of firewalls have this in common, but there is an important difference between the two implementations: where filtering takes place. With the controller-based model, traffic traverses the entire network, all the way to the network core, before it's filtered. With the controller-less model, the AP does all filtering.

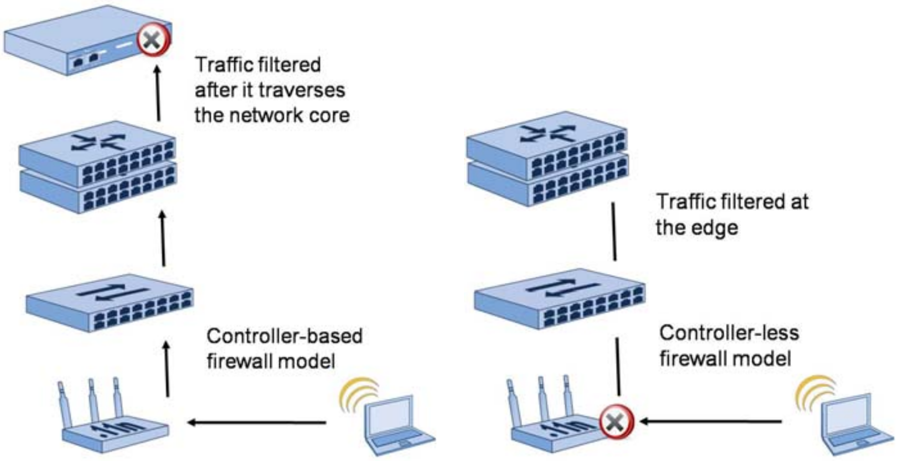

Traffic filtered after it traverses the network core
Controller-based firewall model
Traffic filtered at the edge
Controller-less firewall model

***Figure 3.* Where Should Filtering Take Place?**

In order to grasp the difference, you simply need to ask yourself, "would I place my Internet-facing firewall in my network core?" It's a rhetorical question of course because everyone implements firewalls at the border between the LAN and the Internet, and so it should be with Wi-Fi.

Question to Ponder

Would you say that having traffic that will be dropped traversing access and core network layers is a waste of network resources and an unnecessary security risk?

Flexibility

The introduction of the controller considerably constrained the Wi-Fi infrastructure designer, introducing one or more “single points” to which APs had to logically connect for control-plane sharing. First generation autonomous APs had very nice deployment flexibility because they could be placed anywhere coverage was needed, without regard for how they would connect back to one or more controllers (e.g. across WAN links, across Layer-3 boundaries, etc). Since the controller-less architecture also used autonomous APs as the starting point, not a controller-based architecture, controller-less maintains this deployment flexibility advantage along with other advantages that autonomous APs had over controllers, such as low-cost, linear scalability, and network resilience.

With the absence of a controller, Access Points (APs) can play the same role in enterprise Wi-Fi that routers play on the Internet: *independent-yet-coordinated*. A controller-less architecture is a coordinated system that can lose a body part and still maintain full-featured operation.

In addition to the advantageous scalability, reliability, security, and cost factors, a controller-less architecture introduces some additional key advantages. Intelligent APs can discover and self-configure for whatever role they need to play within the network without manual intervention. Consider the deployment/role differences between an AP that is deployed in a large, single-building network and one that is deployed in a large, distributed (100+ remote offices) network.

Question to Ponder

Why should there be “AP modes” (as implemented by controller-based APs) when all features should be available in all deployment scenarios?

Economics

Much has been written on the economic advantages of Wi-Fi technology in general as compared to Ethernet or other wireline network transports. Rather than repeat that, I will build on it. It’s economically advantageous to have 50-100 users (on average) connected to a single access point that is connected to a single Ethernet cable because the Ethernet port’s bandwidth capacity is used much more efficiently. That applies to all Wi-Fi vendors, but where the economics start to break down is at the controller, which usually adds 40% or more to the total cost of the WLAN infrastructure.

In a controller-less model, APs use protocols, much akin to the well-known OSPF link-state routing protocol, between each other to perform control-plane tasks. In comparison to protocols, which are free, controllers are quite expensive – especially when multiple controllers are needed for scalability, high availability, or branch office solutions. Removing the controller from the enterprise Wi-Fi network also removes the need to license features, which is often more expensive than the controller hardware itself.

As previously mentioned, the stair-step controller model has scalability ramifications, but controllers also face specific cost drawbacks as well. Controllers are CAPEX-based and have to be conjoined into a hard-to-manage mixture of controller sizes as organizations scale their Wi-Fi infrastructure. For example, suppose that you had purchased a controller capable of supporting 16 APs, but then you found yourself needing 20 APs at a later time. You then decided to keep the 16-AP controller and to purchase another 32-AP controller in hopes that the two controllers could handle your needs. After your organization surpassed the 48-AP limit, now you decided to purchase a 128-AP controller.

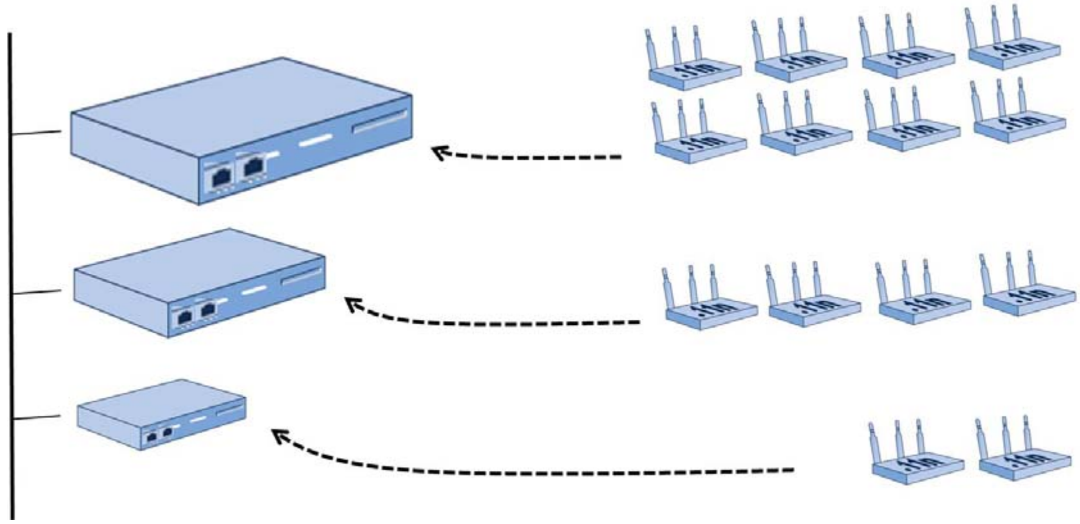


***Figure 4.* Is Your Controller Stack a Hodgepodge?**

You can now either roll all 49+ APs into one controller and buy many new AP and feature licenses for the 128-AP controller or more likely, use all three controllers in some mishmash configuration. This means adding a WNMS for management of three controllers, more data center rack space, cooling, and power, more single points of failure, and an inflexible and expensive scaling model. Why do this, when you could just buy APs and let self-configuring protocols operate the control plane for you?

Question to Ponder

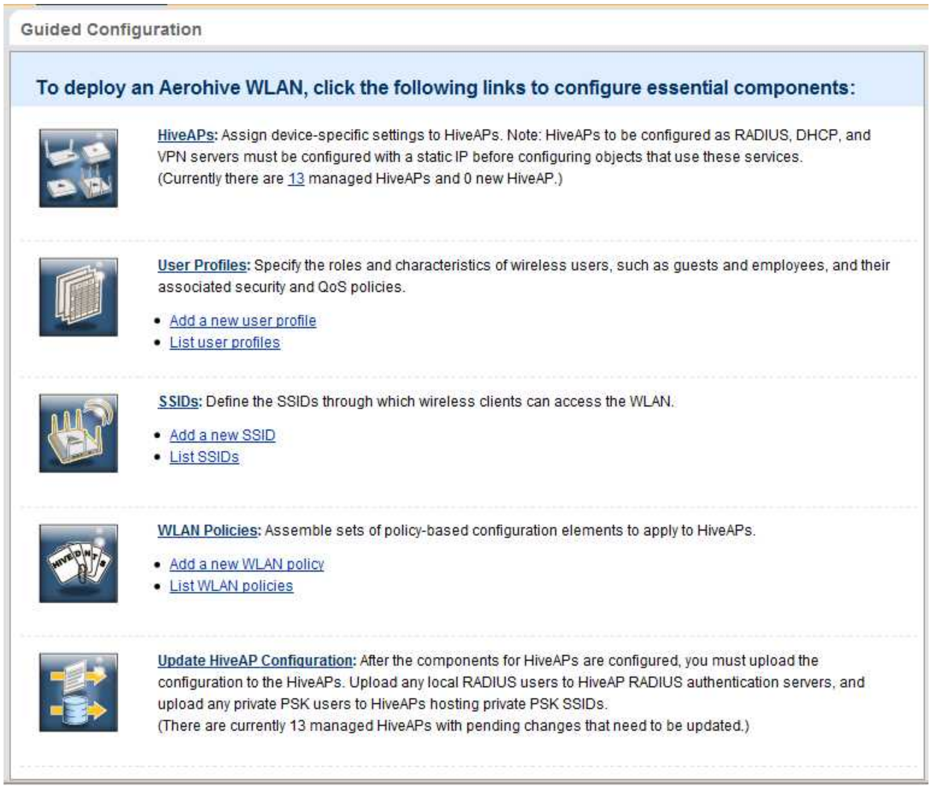
In order to achieve the low cost, deployment & scaling simplicity, and superior ROI in a Wireless 2.0 world, how will controller-based vendors lower their cost without lowering product quality?

Aerohive asserts that it <u>must</u> be through removal of unnecessary WLAN components, and there's only one component in a 3-component WLAN system that can be made optional: controllers.

Usability

IT administrators often perform the standard move/add/change operations, and usability is a key component of a good networking system in general. Less components usually means less complexity, and unfortunately for many of today's IT administrators, they have a controller-based infrastructure with a plethora of components (e.g. core controllers, redundant controllers, branch controllers, 8-10 types of licenses per controller, APs, and a WNMS). In contrast, a controller-less architecture has 2 general components: WNMS & APs. This simplification makes both the sales and management processes quite simple.

There's no doubt that today's leading enterprise-class Wi-Fi platforms are feature-rich. In fact, vendors often focus so intently on adding new features that they forget about important concepts like feature fatigue. Usability is usually an afterthought, and when users begin to complain, vendors have to create object wizards as a backpedaling attempt to shield users from their platform's complexity. The effect on ease-of-use should be top-of-mind when adding any new feature, let alone the high number of features regularly added to high-end systems. From a high level, Graphical User Interfaces (GUIs) should logically and seamlessly flow through the configuration process to the provisioning of APs.



Guided Configuration

To deploy an Aerohive WLAN, click the following links to configure essential components:

HiveAPs: Assign device-specific settings to HiveAPs. Note: HiveAPs to be configured as RADIUS, DHCP, and VPN servers must be configured with a static IP before configuring objects that use these services.
(Currently there are 13 managed HiveAPs and 0 new HiveAP.)

User Profiles: Specify the roles and characteristics of wireless users, such as guests and employees, and their associated security and QoS policies.

- Add a new user profile
- List user profiles

SSIDs: Define the SSIDs through which wireless clients can access the WLAN.

- Add a new SSID
- List SSIDs

WLAN Policies: Assemble sets of policy-based configuration elements to apply to HiveAPs.

- Add a new WLAN policy
- List WLAN policies

Update HiveAP Configuration: After the components for HiveAPs are configured, you must upload the configuration to the HiveAPs. Upload any local RADIUS users to HiveAP RADIUS authentication servers, and upload any private PSK users to HiveAPs hosting private PSK SSIDs.
(There are currently 13 managed HiveAPs with pending changes that need to be updated.)

***Figure 5.* Logical, Seamless Configuration and Provisioning Process**

There are two kinds of administrators in today's enterprise: 2D and 3D. This has been a constant for a number of years and shows no signs of changing. 2D administrators prefer an interface that keeps the parts of a WLAN profile grouped together. 3D administrators prefer an interface that lets you configure objects within the interface that can be later used to build WLAN profiles. There's no shoe-horning these two types of thinkers into a single interface type, so it's just easier to build an interface to suit each user type. HiveManager Express (and its cloud-based twin HiveManager Online Express) is built for the 2D administrator while HiveManager Enterprise (and its cloud-based twin HiveManager Online Enterprise) is built for the 3D administrator. Now each administrator type can have what they've always wanted – an interface customized for their way of thinking.

Question to Ponder

How successful will controller-based vendors be at re-architecting their entire feature set for a fully-distributed architecture while simultaneously rebuilding their GUI to make it user-friendly?

The Cloud

Software-as-a-Service (SaaS) is all the rage these days and for good reason: it causes a cost-model paradigm shift, and it offers ubiquitous and simplified access. Another term, *cloud-based*, is quickly gaining popularity in the networking world as well, but the terms are somewhat synonymous. Cloud-based computing has reached a level of maturity where analysts are now more concisely defining two types of “clouds”: private and public. In order to address both markets, Aerohive has introduced cloud-based solutions for organizations that prefer that their management platforms live in their data center and for organizations that prefer that their management platforms live in a data center managed by a service provider.

Aerohive’s architecture puts both the data and the control plane into the APs, leaving only the management plane as centralized. The beauty of this model is that while management visibility is nice, it’s non-essential for network operation. More concisely, if the HiveManager (WNMS) is disconnected from the APs, there’s no network outage. Other vendors have tried to copy this approach by putting their controller in the cloud, but this introduces a slow and unstable link between APs and their control plane which can (and usually does) lead to Wi-Fi outages and/or limited AP functionality. Essential network infrastructure functions, such as the control plane, belong in the network, ***not in the cloud***.

Aerohive has introduced two different cost models, CapEx and OpEx, to address the various situations in which customers find themselves at a given point in time.

The Private Cloud (CapEx Model)

Private, cloud-based management is deployed with either Aerohive’s hardware appliance or VM-based HiveManager platform. Either way, it can be carved up into many virtual HiveManager WNMS engines, each managing their own APs. HiveManager is a software or hardware/software platform that lives in the customer’s or Managed Service Provider’s (MSP’s) data center and is sold as a CapEx model.

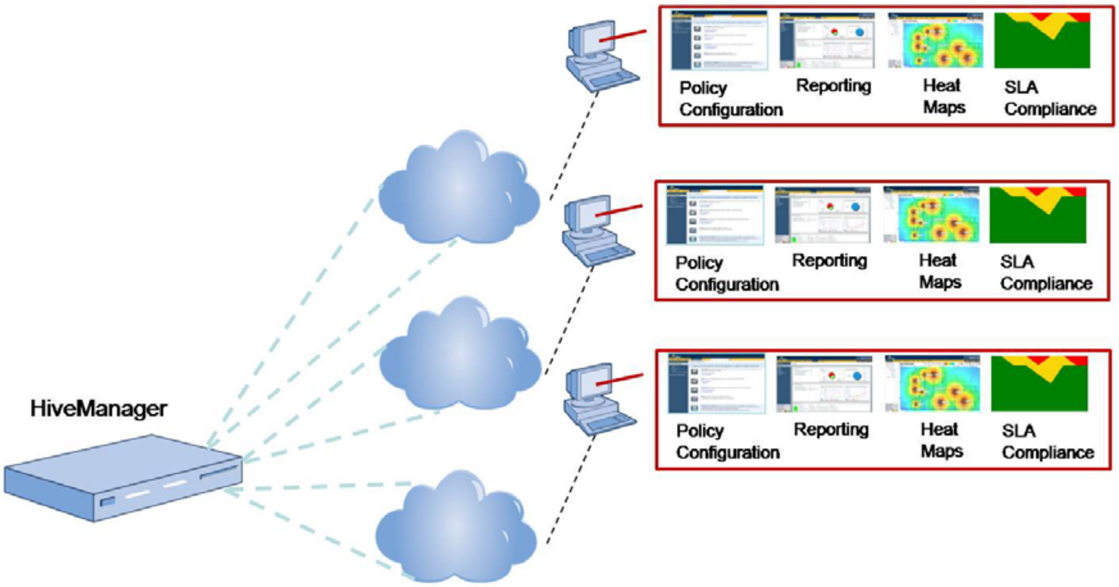


Figure 6. **The Private Cloud: Virtualizing HiveManager**

The Public Cloud (OpEx Model)

Public, cloud-based management is deployed by Aerohive in its own redundant data centers and sold as an OpEx model, where VARs/MSPs can resell Aerohive's cloud-based management service. Some Aerohive customers who began with HiveManager Online's OpEx model have transitioned to HiveManager's CapEx model as they've grown. HiveManager Online introduces a simple and inexpensive way for the mid-market and distributed enterprise to deploy enterprise-class Wi-Fi rather than relying on feature-poor SOHO-class devices at each location.

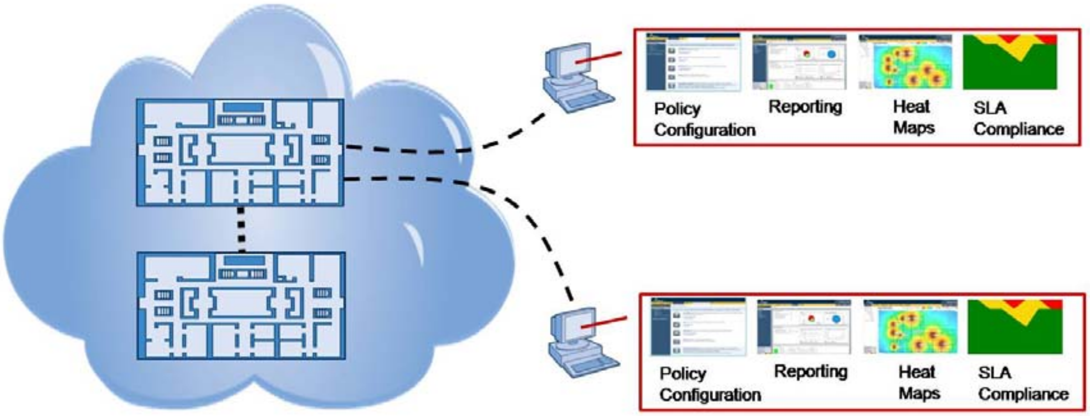


Figure 7. **The Public Cloud: HiveManager Online**

Question to Ponder

Would cloud-based management and/or a flexible cost model help your organization transition from SOHO-class to Enterprise-class Wi-Fi?

Conclusion

The best proof-of-concept for Aerohive's architecture is the Internet itself. Vint Cerf, VP and Chief Internet Evangelist at Google (often called the "Father of the Internet"), said in a 2009 interview, "Part of my motivation when I was working on the Internet was exactly to build a system that did not have any central control. Recall that this was being supported by the US defense department, and one of the things that the defense department wants is highly reliable and resilient systems. One way to achieve that is to not have any central place that could be attacked and destroyed in therefore interfere with the operation of the net. So the consequence of this, I would say 'decentralized architecture' is that it is highly resilient to a variety of impairments and in consequence of that, it's very hard for anybody to shut the Internet down entirely."

While you probably hear, "The Internet is down" all the time at home from your family members, if the Internet were really down, there would be global chaos. Livelihoods depend on the resilience of the Internet, and it was designed for maximum availability through a distributed architecture. To most users, the Internet is, "just there" – an invisible entity that "just works." Aerohive is building the Wi-Fi infrastructure version of the Internet based on the same fully-distributed architecture.

While all Wi-Fi vendors understand the need for and are taking steps in the directions of a fully-distributed architecture, crossing the bridge from controller-based to controller-less takes a significant amount of time and effort because all system features and the user interface must be re-architected for a fully-distributed platform. Make no mistake, all vendors who want to remain in business will eventually approach Wireless 2.0, but the looming questions are, "How long will it take?" and "How much will their customers suffer during the transition?"

"***Problems cannot be solved at the same level of awareness that created them.***" *Albert Einstein*

We invite you to take the road -less traveled: controller-less.