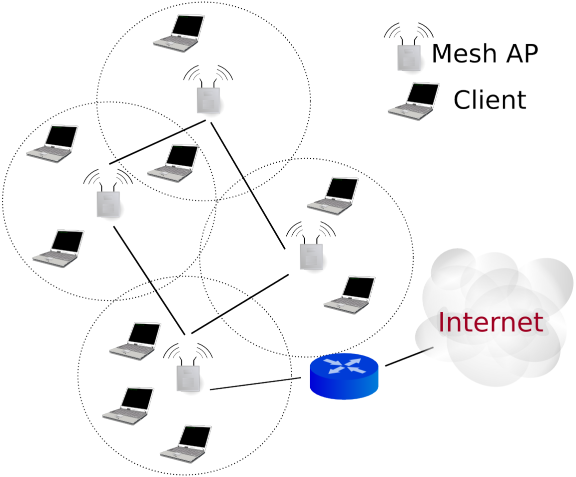
Introduction to Wireless Mesh Networks

Leonardo Maccari
DISI, University of Trento

November 15, 2017

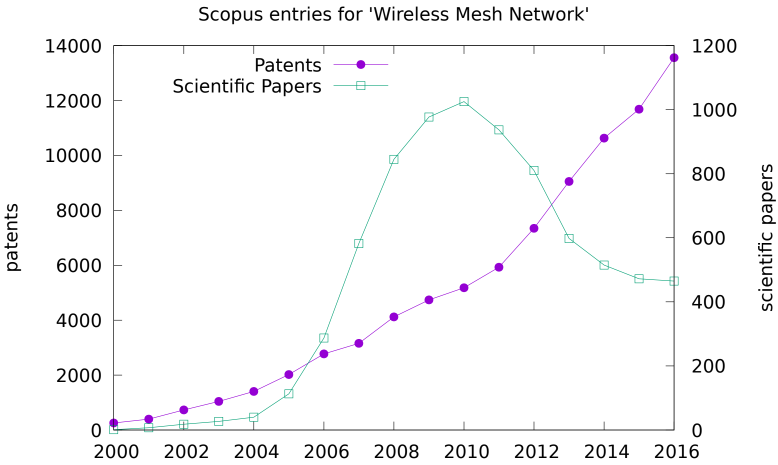
- A wireless Mesh Network (WMN) is a multi-hop network made of wireless nodes and wireless links.
- A wireless router is an embedded device, or a set of embedded devices
- A wireless link is generally obtained using 802.11, but this is not strictly necessary

Wifi Mesh network



Relevant features of Wireless Mesh Networks

- Multi-hop: cooperative routing to reach the destinations
- Automatic organization: no need for pre-deployment work
- Scalability: networks can be easily extended without planning (up to some limit)
- Automatic reconfiguration: recovery from loss of links or failure of nodes



. . . mature technology

When is a Wireless Mesh Networks useful?

- when you want to give coverage without wiring

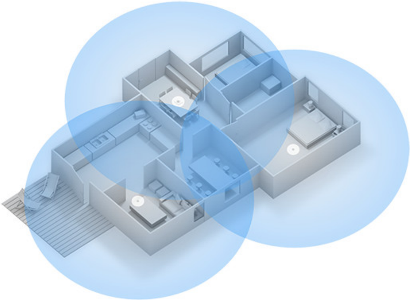
When is a Wireless Mesh Networks useful?

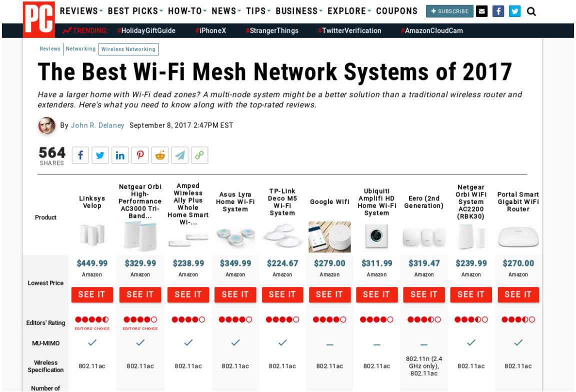
- when you want to give coverage without wiring
- when there is nothing better to use ... (happens more often than you think)

When is a Wireless Mesh Networks useful?

- when you want to give coverage without wiring
- when there is nothing better to use . . . (happens more often than you think)
- when you want to build an *alternative network*

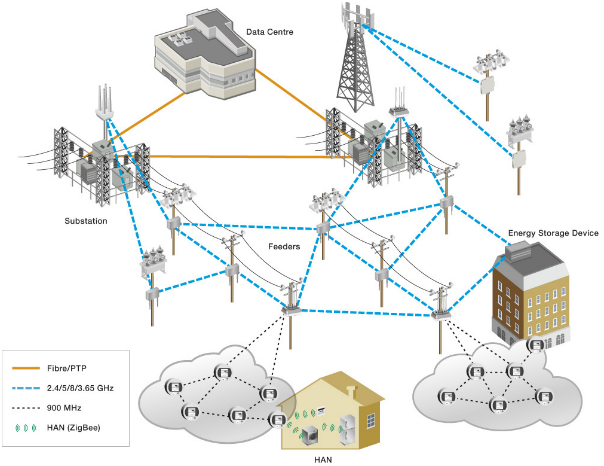
- 802.11ac brings gigabit connectivity to Wi-Fi
- But it reduces the range and it is sensitive to NLOS scenario
- Home Mesh networks are now on the rise to give connectivity to large houses





Product	Linksys Velop	Netgear Orbi High-Performance AC3000 Tri-Band...	Amped Wireless Ally Plus Whole Home Smart Wi-...	Asus Lyra Home Wi-Fi System	TP-Link Deco M5 Wi-Fi System	Google Wifi	Ubiquiti Amplifi HD Home Wi-Fi System	Eero (2nd Generation)	Netgear Orbi WiFi System AC2200 (RBK30)	Portal Smart Gigabit WiFi Router
Lowest Price	$449.99 Amazon SEE IT	$329.99 Amazon SEE IT	$238.99 Amazon SEE IT	$349.99 Amazon SEE IT	$224.67 Amazon SEE IT	$279.00 Amazon SEE IT	$311.99 Amazon SEE IT	$319.47 Amazon SEE IT	$239.99 Amazon SEE IT	$270.00 Amazon SEE IT
Editors' Rating	EDITORS' CHOICE	EDITORS' CHOICE								
MU-MIMO	✓	✓	✓	✓	✓	–	–	–	✓	✓
Wireless Specification	802.11ac	802.11ac	802.11ac	802.11ac	802.11ac	802.11ac	802.11ac	802.11n (2.4 GHz only), 802.11ac	802.11ac	802.11ac
Number of										

- High throughput
- Small scale



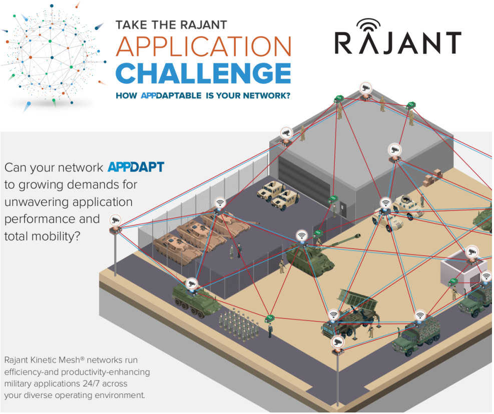
see http://new.abb.com/network-management/communication-networks/wireless-networks


ABB

Are by far the most developed one, see what these companies do:

- Firetide
- Strix
- MeshDynamics
- FluidMesh

Military Applications



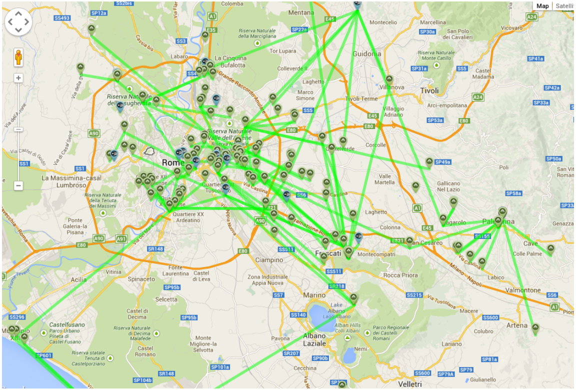
see
`https://www.rajant.com/applications/federal-military-civilian/`

see https://unifi-mesh.ubnt.com/

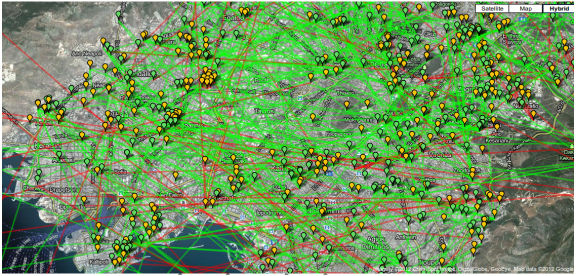
- The largest well documented examples of mesh network deployments are from bottom-up communities:
- Community Networks

Community networks are bottom-up networks organized by a local community of people for two reasons:

- Using local services
- Accessing the Internet (when necessary)

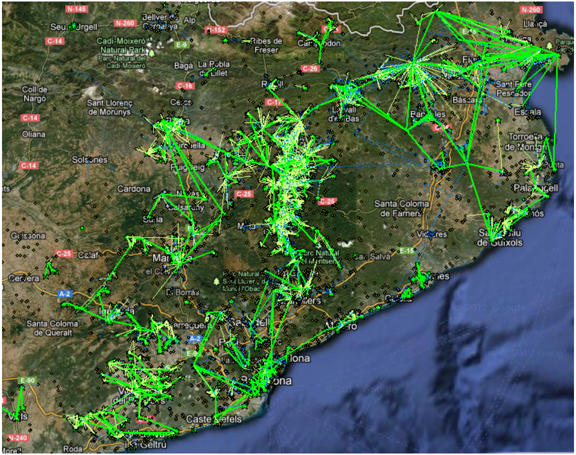

Map
Colombo
Mentana
Montecelio
Marcellina
San Polo dei Cavalieri
Vicovaro
Labaro
La Cinquina
Bufalotta
Tor Lupara
Guidonia
Colleverde II
Laghetto
Marco Simone
Villanova
Tivoli
Tivoli Terme
Castel Madama
Villaggio Adriano
Arci-empolitana
Castelverde
Corcolle
Rome
La Massimina-casal Lumbroso
Ponte Galeria-la Pisana
Quartiere XX Ardeatino
Valle Martella
Gallicano Nel Lazio
Poli
Laghetto
Colonna
Palestrina
Genazzano
Cave
Colle Palme
Ciampino
Frascati
Montecompatri
Rocca Priora
Labico
Valmontone
Acilia
Vitinia
Spinaceto
Fonte Laurentina
Castel di Leva
Zona Industriale Appia Nuova
Selcetta
Castel di Decima
Marino
Macere
Artena
Castelfusano
Monte Migliore-la Selvotta
Castel Romano
Pavona
Albano Laziale
Nemi
Colubro
Lariano
Cecchina
Velletri
Giulianello

Ninux (Italy)



AWMN (Greece)

Huge networks



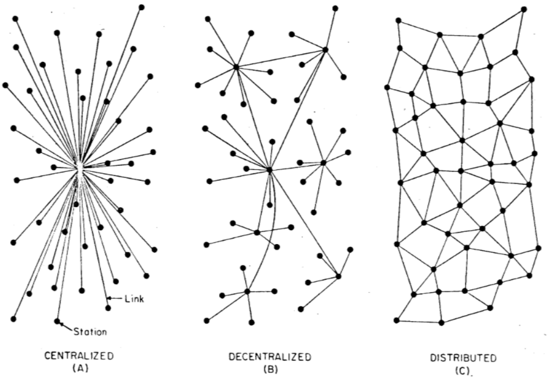
Guifi (Spain)

- A mesh network is a multi-hop network.
- To go from A to B you have to “bounce” though multiple hops.
- Each node in the network is a router for some other node’s traffic

- A mesh network is generally unplanned
- When a new node is added, the rest of the nodes re-configure themselves to reach the new node, in a distributed way
- Similarly, when a node is removed the other nodes network re-configure themselves to router around the failure

Robust Networks

- A typical communication network is designed to resist up to a certain number of failures
- A mesh network is not "designed", so it tries to compensate the lack of robustness by-design with redundancy
- "Robust" and "distributed" have a lot in common
- Let's open a parenthesis...

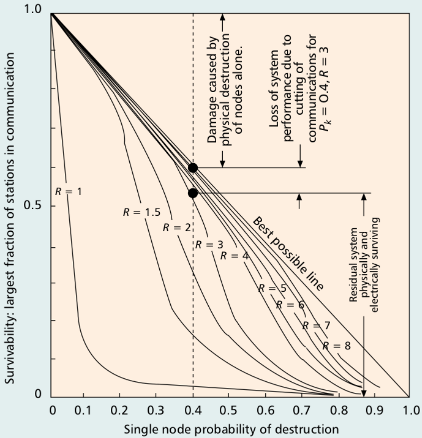


Paul Baran, *"On distributed communications networks"* IEEE transactions on Communications Systems 1964

- *(a) The centralized network has all its nodes connected to a central switching node to allow simple switching, giving it a single point of high vulnerability.*
- *(b) The decentralized network, representative of the AT&T Long Lines network at the time, is better. Instead of a single central switching node, the network comprises small centralized clusters, with most traffic going to nearby neighbors, and only the longer distance traffic routed to the longer links.*
- *(c) The distributed network is a network without any hierarchical structure; thus, there is no single point of vulnerability to bring down much of the network.*

Shortly after I arrived at RAND I began to study the behavior of distributed networks with different levels of redundant connections. In Fig. 2, we see a network with its nodes tied together with the minimum possible number of links. This is called a network of redundancy level 1. A network of redundancy level 2 looks like a fishnet, with horizontal and vertical links. When we reached redundancy levels on the order of 3 an interesting phenomenon occurred: the network became extremely robust.

- Redundancy = Ratio of links to node
- Survivability = Largest fraction of stations in communication after a failure
- What is the relationship between redundancy and survivability?



Perfect switching in a distributed network — sensitivity to node destruction, 100% of links operative.

Paul Baran, *"The beginnings of packet switching: some underlying concepts"* IEEE Communications Magazine, 2002

This meant that it would be theoretically possible to build extremely reliable communication networks out of unreliable links, by the proper use of redundancy.
In other words, if a redundantly connected node survived the physical attack, there is a high probability that this node, at least on paper, was somehow connected to all the other surviving nodes. "Somehow" was the issue, and was the motivation for packet switching.

- At the time telephony was circuit-based. A physical analog circuit was created between the endpoints

- At the time telephony was circuit-based. A physical analog circuit was created between the endpoints
- But how to re-route in case of failure in a distributed way with such a system?

- At the time telephony was circuit-based. A physical analog circuit was created between the endpoints
- But how to re-route in case of failure in a distributed way with such a system?
- Packet switching was born. It did not require global network synchronization, allowed different bit-rate per link, and needed distributed routing intelligence. It needed standard formats, layered addressing and distributed routing. The Internet was born.

- At the time telephony was circuit-based. A physical analog circuit was created between the endpoints
- But how to re-route in case of failure in a distributed way with such a system?
- Packet switching was born. It did not require global network synchronization, allowed different bit-rate per link, and needed distributed routing intelligence. It needed standard formats, layered addressing and distributed routing. The Internet was born.
- The first proposed routing protocol was “hot potato”: no buffering, always forward to “someone”.

- If you want to dig into it: check the definition of graph Percolation and Resilience.
- In a nutshell, random graphs become robust when the link/node ratio reaches a certain "percolation threshold"
- If your technology is capable of routing around failures, you can exploit this concept to have a robust network even without planning.
- PS: note that networks are not just "random graphs"

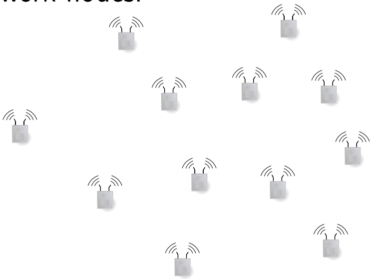
- We see that the highest the redundancy, the highest the robustness
- Redundancy = links/nodes
- To have a robust network you want to have many links
- To have many links you want to use antennas with a high aperture (i.e. broadcast antennas)
- Broadcast antennas produce high interference

To have large networks, you have to use high transmission power:

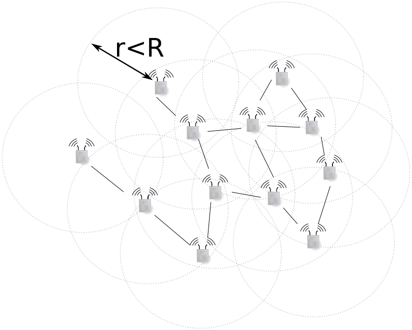
- High power produce high interference
- Law limits EIRP transmission in ISM bands:

Frequency (GHz)	Power
2,4 – 2,4835	20 dBm EIRP
5,15 – 5,35 (indoor)	23 dBm EIRP
5,47 – 5,725	30 dBm EIRP

Consider these network nodes:

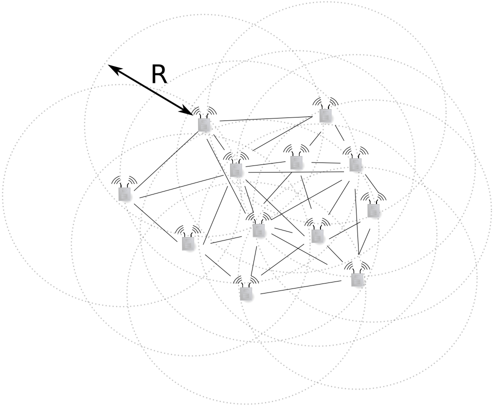


Set the radio range to r:



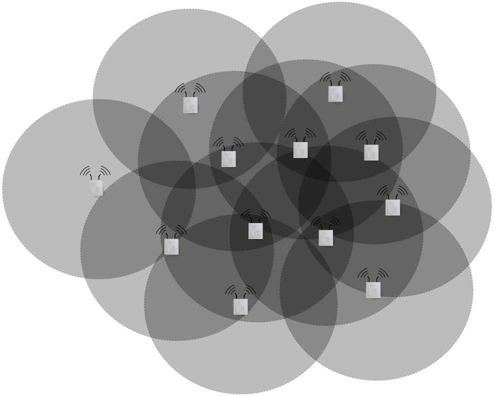
Leaf nodes present → one failure may partition the network.

Set the radio range to R:



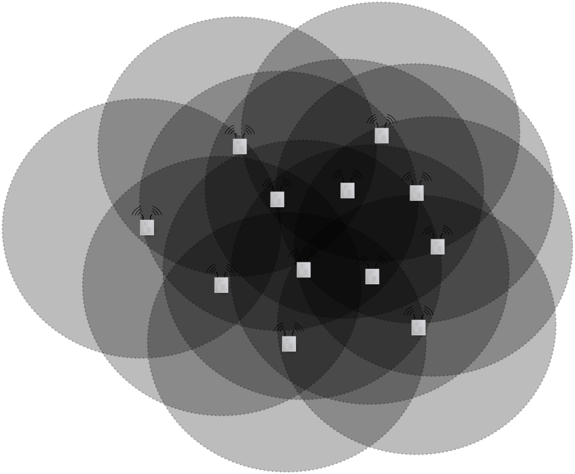
Many links → no leaf nodes → tolerant to one failure.

Now consider interference:



Less overlapping → higher throughput

Now consider interference:



More overlapping → lower throughput

- Initially, mesh networks were realized with router with 1 wireless card and an omnidirectional antenna
- Then, multi-radio devices emerged
- Then multi-radio MIMO devices emerged
- Then, professional outdoor devices emerged, which can be composed
- Today you can have several radio/devices with MIMO and beamforming

- Linksys WRT54
- CPU: Broadcom BCM4702 @ 125 MHz
- Flash: 16 MB
- RAM: 4 MB
- Wireless: 1 802.11g Radio

- CPU: Atheros AR9344@560MHz
- Flash: 8MB
- RAM: 128MB
- Ports: 4x1 GigE, 1x1 GigE WAN, x2 USB v2.0
- Wireless: 2x 802.11n Radio (5GHz and 2.4GHz)

- You can separate the access network from the backbone network
- You can use the 5GHz band for backbone which should be (slightly) less crowded than 2.4GHz
- You have a backup interface if one breaks down

NB: these products are indoor products, if you want to use them outdoor you have to box them

- Ubiquiti is the first manufacturer that sold 802.11n products for prices lower than 100$, for outdoor use
- Angle: from 360 to 15 deg
- Distance: from 100m to 20km
- Throughput: from 100Mb/s to 1Gb/s

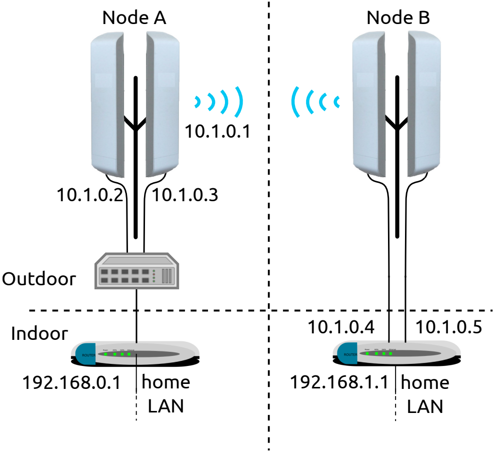
- The latest standards and hardware give you:
- Directional antennas: longer range, higher throughput
- (MU-)MIMO: different polarization schemes produce less interference
- Beamforming: limits interference on non-necessary directions

Today, networks made of hundreds of nodes are possible

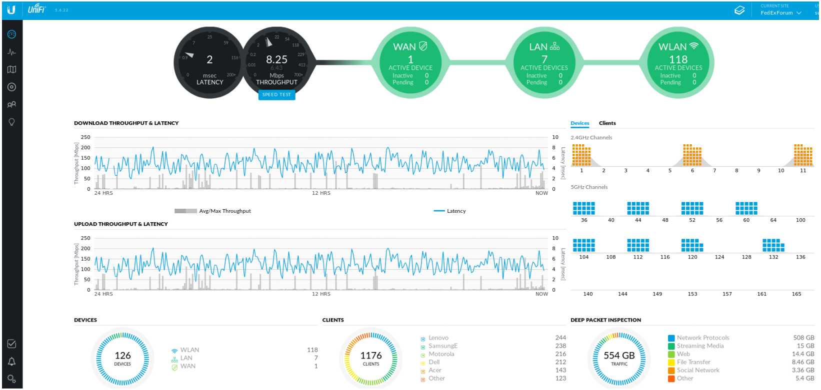
- All the vendors now ship their products with proprietary modifications to the MAC layer to achieve better performance
- ex: Ubnt AIRMAX, Tp-Link Pharos MAXtream TDMA ...
- TDMA (Time-Division Multiple Access) makes AP-station performance generally better than plain CSMA/CA
- Proprietary extensions make Wi-Fi non compatible with other vendors. You will have vendor lock-in.

- To achieve increased coverage, you will can compose multiple antennas:

- Multiple outdoor devices
- Bridge configuration for the radios
- One router to power them



- All the devices come with their proprietary OS
- the OS give you also bells and whistles:
  - Proprietary extensions (TDMA)
  - Fancy configuration tools
  - Monitor and Remote Control (critical)
- They also give you integration with other components (firewalls, RADIUS, VoIP etc. . .)
- They **lock you in**.



UniFi 5.6.22
CURRENT SITE
FedExForum
2
msec
LATENCY
8.25
Mbps
THROUGHPUT
SPEED TEST
WAN
1
ACTIVE DEVICE
Inactive 0
Pending 0
LAN
7
ACTIVE DEVICES
Inactive 0
Pending 0
WLAN
118
ACTIVE DEVICES
Inactive 0
Pending 0
DOWNLOAD THROUGHPUT & LATENCY
Throughput [Mbps]
Latency [msec]
24 HRS
12 HRS
NOW
Avg/Max Throughput
Latency
UPLOAD THROUGHPUT & LATENCY
Devices
Clients
2.4GHz Channels
5GHz Channels
DEVICES
126
DEVICES
WLAN 118
LAN 7
WAN 1
CLIENTS
1176
CLIENTS
Lenovo 244
SamsungE 238
Motorola 216
Dell 212
Acer 143
Other 123
DEEP PACKET INSPECTION
554 GB
TRAFFIC
Network Protocols 508 GB
Streaming Media 15 GB
Web 14.4 GB
File Transfer 8.46 GB
Social Network 3.36 GB
Other 5.4 GB

Open Vs Proprietary Approach: Lock-In

Ubiquiti Networks Community > airMAX > airMAX AC >

LiteBeam 5AC 23 - disable airMax

Reply Topic Options

alexatkinuk
New Member
Posts: 22
Registered: 11-05-2016

LiteBeam 5AC 23 - disable airMax
11-05-2016 10:07 AM

Options

Is it not possible to disable airMax on this device?

I had been looking at various Ubiquiti products and chose this because it was priced more or less the same as the M5 for newer technology.

As it was advertised as being usable for an outdoor to indoor AP I expected this feature to be available, as what use is a building-wide WiFi that only works with one specific vendors clients? But I see no option for this in the UI.

0 Kudos Reply

flipper
SuperUser
Posts: 15,030
Registered: 06-18-2010
Kudos: 4436

Re: LiteBeam 5AC 23 - disable airMax
11-05-2016 11:44 AM

Options

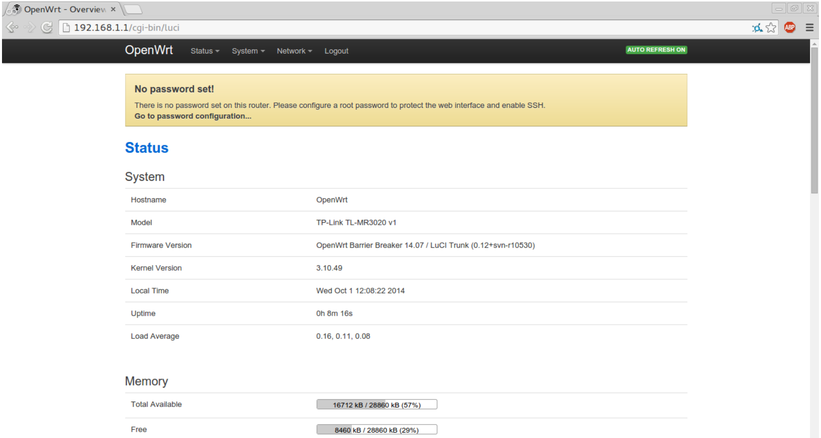
The AirMax AC radios are compatible only with other AirMax radios; they will not connect with 802.11ac wifi clients.

No, you can not turn off AirMax. They're useful for creating point to point and point to multipoint networks, not for serving wifi clients.

Hi, I'm @flipper, member of the Ubiquiti Community, and I am Not a Bot

- OpenWRT is an open source OS for embedded devices
- OpenWRT is community-maintained, and supports hundreds of different devices
- Most of the Proprietary OSs, are based on OpenWRT
- OpenWRT is extremely feature rich, it is a linux distribution with thousands of packages

OpenWRT LuCI Interface



OpenWRT pros and cons

- + Open source
- + Free
- + No lock-in
- + Supports many different devices
- + You can extend it
- + Does only what you want

- - Your feature may be unmaintained
- - Not including the latest, newest features
- - No commercial support
- - Not tailored for your product
- - No orchestration out of the box

As Shakespeare said:

“All that glisters is not gold”

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

AirOS Vulnerability Issue Update, 3/18/17 [Edited] Options

Friday - last edited Friday by UBNT UBNT-cmb

Hi All -- Wanted to give an update here:

1. UniFi, EdgeMAX and AmpliFi are not affected. This issue is limited to AirOS and associated products (toughswitch,airgateway,etc)

2. The issue has been addressed as follows:
AirOS v8.0.1 — already available since Feb 3, 2017 (release notes here)
AirOS v6.0.1 — released today (release notes here)
AirGateway v1.1.8 - Service release —released today (release notes here)
TOUGHSwitch v.1.3.4 - Service — released today (release notes here)
airFiber v3.2.2 and v3.4.1 - released today (release notes here)

3. While we acknowledge all vulnerabilities are serious, we believe this issue rates fairly low in terms of threat severity compared to past patched vulnerabilities

4. Ubiquiti has a dedicated Security Director 100% focused strictly on Ubiquiti software vulnerabilities @UBNT-rubens along with a very strong supporting group of engineers.
In addition, we participate in 3rd party vulnerability assessment programs such as Hackerone.com where we have given out significant rewards to date.
Finally, we have significant investments in a retained 3rd party external security audit company who reviews our software solutions on a frequent basis.

5. The php2 code concern we are already addressing and it will be easily eliminated from applicable code bases within the next few weeks

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

AirOS Vulnerability Issue Update, 3/18/17 [Edited]
Friday - last edited Friday by UBNT UBNT-cmb

Options

Hi All -- Wanted to give an update here:

1. UniFi, EdgeMAX and AmpliFi are not affected. This issue is limited to AirOS and associated products (toughswitch,airgateway,etc)

2. The issue has been addressed as follows:
AirOS v8.0.1 — already available since Feb 3, 2017 (release notes here)
AirOS v6.0.1 — released today (release notes here)
AirGateway v1.1.8 - Service release —released today (release notes here)
TOUGHSwitch v.1.3.4 - Service — released today (release notes here)
airFiber v3.2.2 and v3.4.1 - released today (release notes here)

3. While we acknowledge all vulnerabilities are serious, we believe this issue rates fairly low in terms of threat severity compared to past patched vulnerabilities

4. Ubiquiti has a dedicated Security Director 100% focused strictly on Ubiquiti software vulnerabilities @UBNT-rubens along with a very strong supporting group of engineers.
In addition, we participate in 3rd party vulnerability assessment programs such as Hackerone.com where we have given out significant rewards to date.
Finally, we have significant investments in a retained 3rd party external security audit company who reviews our software solutions on a frequent basis.

5. The php2 code concern we are already addressing and it will be easily eliminated from applicable code bases within the next few weeks

Ubiquiti Vulnerability

AirOS Vulnerability Issue Update, 3/18/17 [Edited]

Options

Friday - last edited Friday by UBNT UBNT-cmb

Hi All -- Wanted to give an update here:

1. UniFi, EdgeMAX and AmpliFi are not affected. This issue is limited to AirOS and associated products (toughswitch,airgateway,etc)

2. The issue has been addressed as follows:
AirOS v8.0.1 — already available since Feb 3, 2017 (release notes here)
AirOS v6.0.1 — re
AirGateway v1.1.8
TOUGHSwitch v.1
airFiber v3.2.2 anc

5. The php2 code
weeks

3. While we ackno ... fairly low in terms of threat severity compared to past patched vulne

4. Ubiquiti has a de ... ware vulnerabilities @UBNT-rubens along with a very strong suppo
In addition, we par ... Hackerone.com where we have given out significant rewards to date.
Finally, we have significant investments in a retained 3rd party external security audit company who reviews our software solutions on a frequent basis.

5. The php2 code concern we are already addressing and it will be easily eliminated from applicable code bases within the next few weeks

Ubiquiti Vulnerability

PHP

From Wikipedia, the free encyclopedia

This article is about the scripting language. For other uses, see PHP (disambiguation).

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994,[4] the PHP reference implementation is now produced by The PHP Development Team.[5] PHP originally stood for *Personal Home Page*,[4] but it now stands for the recursive acronym *PHP: Hypertext Preprocessor*.[6]

PHP code may be embedded into HTML or HTML5 code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.[7]

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.[8]

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a *de facto* standard. Since 2014 work has gone on to create a formal PHP specification.[9]

Contents [hide]

PHP

Paradigm	Imperative, functional, object-oriented, procedural, reflective
Designed by	Rasmus Lerdorf
Developer	The PHP Development Team, Zend Technologies
First appeared	June 8, 1995; 21 years ago[1]
Stable release	7.1.3[2] / March 16, 2017; 5 days ago
Typing discipline	Dynamic, weak, gradual (as of PHP 7.0.0)[3]

History [edit]

Early history [edit]

PHP development began in 1995 when Rasmus Lerdorf wrote several Common Gateway Interface (CGI) programs in C,[10][11][12] which he used to maintain his personal homepage. He extended them to work with web forms and to communicate with databases, and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

PHP/FI could help to build simple, dynamic web applications. To accelerate bug reporting and to improve the code, Lerdorf initially announced the release of PHP/FI as "Personal Home Page Tools (PHP Tools) version 1.0" on the Usenet discussion group *comp.infosystems.www.authoring.cgi* on June 8, 1995.[13][14] This release already had the basic functionality that PHP has as of 2013. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax resembled that of Perl but was simpler, more limited and less consistent.[5]

Lerdorf did not intend the early PHP to become a new programming language, but it grew organically, with Lerdorf noting in retrospect: "I don't know how to stop it, there was never any intent to write a programming language [...] I have absolutely no idea how to write a programming language, I just kept adding the next logical step on the way."[15] A development team began to form and, after months of work and beta testing, officially released PHP/FI 2 in November 1997.

Pre-installed Backdoors

sign in | become a supporter | subscribe | search | find a job | dating | more | International edition

theguardian

UK world sport football opinion culture business lifestyle fashion environment tech travel all

home › world › US americas asia australia africa middle east cities development europe

NSA

Glenn Greenwald: how the NSA tampers with US-made internet routers

The NSA has been covertly implanting interception tools in US servers heading overseas - even though the US government has warned against using Chinese technology for the same reasons, says Glenn Greenwald, in an extract from his new book about the Snowden affair, No Place to Hide

Pre-installed Backdoors

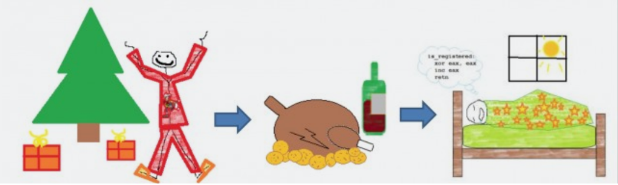
ars TECHNICA   BIZ & IT   TECH   SCIENCE   POLICY   CARS   GAMING & CULTURE   FORUMS   SIGN

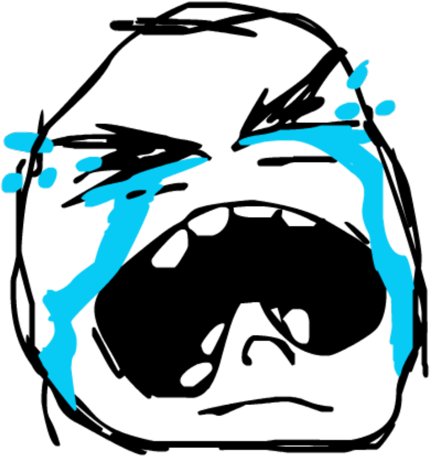
BIZ & IT —

Backdoor in wireless DSL routers lets attacker reset router, get admin

A quick Christmas hack uncovers a vulnerability in Linksys, Netgear, others.

SEAN GALLAGHER - 1/2/2014, 8:05 PM





- Proprietary solutions can be abandonware also
- When you are locked-in to abandonware it's a big pain

- With Proprietary Systems: Buy, plug, open the controller
- With Open Systems: Buy, reflash with OpenWRT, configure, monitor with third-party software: openwisp.org

- Any dynamic networks needs to have a dynamic routing protocol
- With "dynamic" it is intended a routing protocol that dynamically reacts to changes in the network topology
- Dynamic routing is the standard for any complex network (i.e. anything more complex than your home network)

- While in wired networks the conditions are assumed to change slowly (due to physical failure or reboots)
- In WMN the conditions may change frequently. What can change is:
  1. A node can break down/reboot (as in any other network)
  2. A piece of a node can fail (i.e. one radio device)
  3. A link may be obscured: an obstacle makes the link NLOS
  4. A link quality may deteriorate strongly: alignment changes do to wind, starts raining/snowing, antenna gather dirt, reflective surface creates interference (see tide)
- There is no physical connection between nodes, so that "link sensing" can not be achieved easily.

The abstract goals of a routing protocol are:

- To discover the set of reachable IP addresses on the network
- To maintain a *routing table*, a data structure that contains, for each destination address (or block of addresses) known by the router, the outgoing interface over which the router must forward a packet destined to this address.
- The routing table may also contain additional information such as the address of the next router on the path towards the destination or an estimation of the cost of this path

[1] Some text and figures taken from http://cnp3book.info.ucl.ac.be/2nd/html/principles/network.html

- Routing protocols (for WMNs) can be categorized under several dimensions:
  - Proactive/Reactive
  - Link State/Distance Vector
  - Layer II(!?)/Layer III
- We will review Proactive protocols, and analyse some of them in details.

- The protocol is generally silent
- Whenever n_i (node i) needs to communicate with n_j, it will start a discovery phase in which it reconstructs the path to n_j
- The most well known reactive protocol for mesh networks is AODV (Ad hoc On-Demand Distance Vector[2])
- I have never seen anyone using reactive protocols in WMN, so we won't dig into them.

[2] https://tools.ietf.org/html/rfc3561

- Every Router maintains a routing table, i.e. a data structure $R[]$ that contains for each destination d the following information:
  - $R[d].link$: the outgoing link that the router uses to forward packets towards destination d
  - $R[d].cost$: the sum of the metrics of the links that compose the shortest path to reach destination d
  - $R[d].time$: the timestamp of the last distance vector containing destination d

Each node periodically does the following:

```
Every N seconds:
 v=Vector()
 for d in R[]:
   # add destination d to vector
   v.add(Pair(d,R[d].cost))
 for i in interfaces
   # send vector v on this interface
   send(v,interface)
```

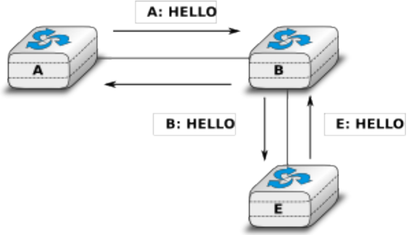
When receiving a message do:

```
def received(V,l): # received vector V from link l
  for d in V[]
    if not (d in R[]): # new route
      R[d].cost=V[d].cost+l.cost
      R[d].link=l
      R[d].time=now
    else: # existing route, is the new better ?
      if (( (V[d].cost+l.cost) < R[d].cost) or
            (R[d].link == l) ):
       # Better route or change to current route
       R[d].cost=V[d].cost+l.cost
       R[d].link=l
       R[d].time=now
```

- This simple algorithm (Bellman-Ford) guarantees convergence with some known problems and limitations (we will see):
- Count to infinity
- Convergence speed
- We will see some variants of DV protocols for mesh networks.

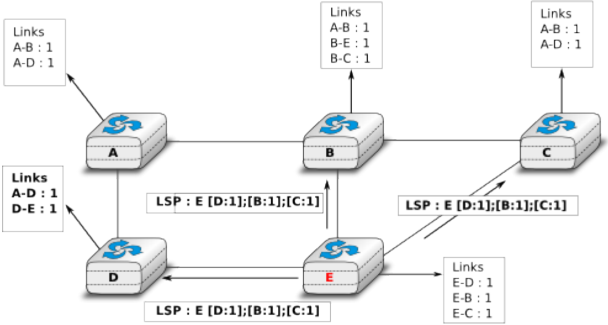
- Distance vector routers use a distributed algorithm to compute their routing tables
- Link-state routers exchange messages to allow each router to learn the entire network topology.
- Based on this learned topology, each router is then able to compute its routing table by using a shortest path computation with Dijkstra's algorithm

- n_i sends periodic HELLO messages containing their IP address(es)
- When receiving an HELLO n_j is added to the neighborhood



Link State: Flooding

- n_i periodically broadcasts its RT to all the others
- n_j collects all the RTs and builds the knowledge of the full graph



- Experimenting with networks: NePA-Test/CORE
- Link State Routing Protocols: OLSR
- Distance Vector Routing Protocols: Babel
- DV Source Routed Routing Protocols: BMX6/7
- Layer II protocols: BATMAN+, IEEE 802.11s
- Parenthesis: Wireless security protocols, IEEE 802.11i

- Experimenting with routing:
  - hard and fun: modify/improve a protocol
  - fair and entertaining: compare/measure protocol performance
  - easy and OK: run and document