

# INDEX

Page numbers in italics refer to figures and tables.

## Symbols

#, xxxiv  
\$ prompt, xxxiv

## Numbers

2.4 GHz band. *See also* Bluetooth  
802.11 WLAN standards, 8  
antennas, 29, 29–30  
broadcast range, 8–9, 9  
channels on, 15, 15–16  
3D printing  
air quality station enclosures,  
233, 252  
Pwnagotchi cases, 266, 266  
3.5 mm jack to USB adapters, 124, 124  
5 GHz band  
antennas, 29, 29  
channels on, 16–17

## A

A2DP protocol. *See* Advanced Audio  
Distribution Profile protocol

access points (APs)  
compromised, 204  
creating, 96  
defined, 6  
monitors for, 86–90, 87  
Raspberry Pi–based, 92  
unauthorized, 207

actor–critic process loop, 276–278, 277

AC Wave 2 wireless (MU-MIMO;  
Next-Gen AC wireless), 18

Adafruit IO

defined, 254  
publishing data with, 254–257, 257  
adafruit\_blinka library, 78, 80

Adafruit Mini PiTFT, 70–71, 71

attaching, 73, 76, 76

pinout, 75, 75–76

starting, 86–87, 87

adaptive data rate algorithm, 184

adaptive frequency hopping (AFH), 37

Advanced Audio Distribution Profile

(A2DP) protocol

defined, 137

sinks

activating, 140

configuring, 137–138

configuring automatic pairing,  
141–142

controlling volume, 145

troubleshooting, 151–152

Advanced Linux Sound Architecture

(ALSA) package

Bluetooth speakers, 133

overview, 126–128

PulseAudio, 128

Spotifyd, 148–149

Advantage Actor–Critic (A2C)

algorithm, 275–278

agetty process, 195–196, 257, 300–301

AI. *See* artificial intelligence

AirPlay

casting, 347–348

connecting to using macOS, 367

reverse engineering of, 367

RPiPlay implementation of, 367–369

air quality monitoring (AQM) stations,  
227–260

adding sensors, 258–260

adjusting alert thresholds, 258

calibrating, 258

dashboards, 245–251

adding data sources, 245,  
245–246, 246

creating alerts, 249, 249–250, 250

- air quality monitoring stations (*continued*)
    - dashboards (*continued*)
    - editing, 248, 248–249
    - importing model dashboard, 246–247, 247
    - testing alerts, 250–251, 251
    - using, 247–248
  - enclosures, 251–254, 252, 253
  - gathering sensor data, 239–244
    - creating virtual project
      - environment, 239–240
    - enabling service, 244
    - installing InfluxDB and Grafana, 239
    - installing pySerial, 240–241
    - obtaining geohash, 241
    - starting sampler, 242–243
    - verifying sample data, 243–244
  - GPIO, 235–238, 236, 237, 238
  - hardware, 230, 230–233, 231, 232
  - prerequisites, 233–235
  - publishing data, 254–257, 257
  - software, 233
  - technologies, 228
  - troubleshooting, 257–258
  - use cases, 228, 228–230, 229
  - airtime (dwell time; time on air), 177
  - alsamixer command, 133, 142, 151
  - ALSA package. *See* Advanced Linux Sound Architecture package
  - alsa-utils package, 127
  - Amazon Fire TV, 373
  - amplitude, 4, 4
  - Android support for wireless displays, 346, 348, 350, 357
  - ANonce (authenticator nonce), 34, 35, 293
  - antennas, 27–31
    - effective distance, 10
    - LoRa, 171, 187–188, 188
    - signal boosting, 10
    - SYN43436 chipset and, 27–28
    - Wi-Fi, 28–31, 29, 30, 31
  - Apple Bonjour, 367–368
  - Apple TV, 247–248, 350, 373
  - application programming interfaces (APIs), 42
  - application-specific integrated circuits (ASICs), 92
  - Applied Minds* (Madhavan), 377
  - APs. *See* access points
  - APSP00F (Evil Twin) attacks, 207, 218, 222, 222
  - AQI (World Air Quality Index), 229, 229
  - AQM stations. *See* air quality monitoring stations
  - ARM Cortex-R4 (ARMCR4) processor, 25, 270
    - firmware, 269–271
  - artificial intelligence (AI) in Pwnagotchi, 274–279
    - actor–critic process loop, 276–278
    - reward function, 278–279
  - association frames, 33, 225, 293–294, 305
  - audio. *See also* Bluetooth; Spotify
    - displaying properties of, 153–154
    - lossy vs. lossless, 146
    - troubleshooting, 151–152
  - Audio/Video Remote Control Profile (AVRCP), 144–145, 153
  - authentication
    - authenticators and supplicants, 34, 35
    - key based, 44–45
    - mesh networks, 335
    - wireless connections, 33–36, 35
  - authenticator nonce (ANonce), 34, 35, 293
  - automatic content recognition (ACR), 372–373
  - Avahi, 367
- ## B
- balenaEtcher tool, 272
  - bandwidth, 10–11
  - bandwidth monitor (bmon) tool, 336
  - basic service set (BSS), 14
  - batctl ping utility, 321–322
  - batctl utility, 314, 316, 318, 322–324, 328–332, 339
  - BATMAN (Better Approach to Mobile Ad hoc Networking) protocol, 313–314
  - batman-adv kernel module, 314, 316, 318–321, 321, 324–329, 331–335, 339–341

- batteries
  - external modules, 263–264, 264
  - LiPo, 71, 71–72, 72
  - air quality monitoring stations, 254
  - attaching, 76–78
  - benefits of, 183
  - LoRa, 162, 162, 183
  - monitoring, 88
- beam frames, 33, 351
- beamforming, 10
- Better Approach to Mobile Ad hoc Networking (BATMAN) protocol, 313–314
- Bettercap
  - accessing web interface, 291–292
  - defined, 269
  - scripting with caplets, 289–291
  - starting interactive sessions, 289
- BinAuth script, 96, 116
- Bluetooth, 36–37, 37, 121–155
  - Bluetooth classic, 37
  - connecting speakers, 131–137
    - enabling snapd, 136
    - getting device properties, 133
    - installing ncpot, 136–137, 137
    - pairing, 132–133
    - playing music from audio library, 134–135, 135
    - streaming internet radio from console, 134
    - streaming Spotify from terminal, 135–136, 136
    - testing audio output, 133
    - using Bluetooth controller, 131–132
  - displaying audio properties, 153–154
  - hardware, 122–126
    - audio DAC SHIM, 124, 124
    - Bluetooth-ready speaker, 123
    - compatible Raspberry Pi models, 123
    - digital-to-analog converter, 123
    - home audio components, 126
    - Raspberry Pi IQaudio DAC+ HAT, 125, 125
    - USB adapter, 123, 123
    - USB to 3.5 mm jack adapter, 124, 124
  - integrating with TFT display, 154
  - Linux audio, 126–128, 127
  - monitoring sources with Kismet, 217
  - prerequisites, 128–131
  - receivers, 137–145
    - activating sink, 140
    - conducting audio test, 142–143, 143
    - conducting pairing test, 140–141
    - configuring A2DP sink, 137–138
    - configuring automatic pairing, 141–142
    - connecting DAC to hi-fi equipment, 144
    - controlling volume, 144–145
    - initiating audio playback, 144
    - integrating DAC, 143–144
    - using external USB adapter, 138–140
  - software, 126
  - Spotify Connect boxes, 145–151
    - configuring Spotifyd, 147–148
    - controlling playback via Spotify app, 150, 150–151
    - creating systemd service, 148–149
    - installing Spotifyd, 146–147
    - Spotify Connect overview, 145–146
    - starting Spotifyd service, 150
  - troubleshooting, 151–152
  - use cases, 121–122
- bluetoothctl utility, 131, 140–142, 144, 151, 153–154
- BlueZ, 129–130, 139, 141, 153
- bluez-alsa Bluetooth audio backend, 128
- bluez-tools package, 141
- bmon (bandwidth monitor) tool, 336
- Bonjour, 367–368
- Bosch BME680 Breakout sensor, 232, 232, 258–260
- Bosch BME688 Breakout sensor, 252
- brcm80211 driver, 32–33
- brcmfmac driver, 19, 271
- brcmsmac driver, 32–33, 302, 370, 391
- br-lan network interface, 93, 98, 100
- broadband wireless metropolitan area networks (MANs), 7
- broadcast range, 8–10, 9
- Broadcom, 32

Broadcom BCM2711 chipset, 345, 361  
Broadcom BCM43455 chipset. *See*  
    CYW/BCM43455 chipset  
btmon command, 139  
bytecode, 237–238, 238

## C

C\* Music Player for Linux (cmus),  
    134–135, 135  
C programming language, 237  
caplets, 289–291  
captive portal detection (CPD), 94–95  
captive portal identification (CPI), 101  
captive portals, 91–120  
    basic, 100–105  
        configuring openNDS, 100  
        connecting vouchers, 102–103  
        creating voucher systems,  
            103–105, 105  
        customizing, 103  
        starting openNDS, 100–101  
    case for, 92  
    creating access points, 96  
    dynamic, 105–114  
        connecting client devices,  
            112–113, 113  
    implementing forward  
        authentication, 107–108  
    installing Lighttpd, 108–110  
    managing clients, 113–114  
    setting up openNDS, 106  
    using Lighttpd with stand-alone  
        RaspAP, 110–112, 111  
    hardware, 92–93, 93  
    installing openNDS, 97–100  
    overview, 93–95, 94  
    prerequisites, 96–97  
    security measures, 95–96  
    setting quotas, 114–116  
    software, 93  
    splash page sequence, 95  
    traffic shaping, 117–118  
    troubleshooting, 118–119  
    use cases, 91–92  
casting, 347–350  
    AirPlay, 347–348  
    Chromecast, 348–349, 350  
    comparing devices, 349–350, 350  
    Miracast, 348–349  
    streaming devices and consumer  
        privacy, 373  
channel bonding, 14  
channel interference, 9  
channel surfing, 13–17  
    channels on 2.4 GHz band, 15–16  
    channels on 5 GHz band, 16, 16–17  
    channel widths, 14–15  
checksum, 20  
chip enable (CE) pins, 164  
chip rate, 176  
chirp pulses, 175  
chirp rate, 178  
chirps (sweep signals), 175–176, 176  
chirp spread spectrum (CSS)  
    modulation, 175, 175–176  
Chromecast, 348–349, 350  
CircuitPython library, 78  
citizen science, 230  
CLIs (command line interfaces), 42  
CM4 (Raspberry Pi Compute Module 4),  
    23, 24  
cmdline.txt file, 234  
cmus (C\* Music Player for Linux),  
    134–135, 135  
Coder.com, 47  
code-servers, self-hosted, 46–49, 47, 49  
coding rate, 178  
Coffee Grind Size software,  
    375–377, 376  
color e-ink displays, 264  
command line interfaces (CLIs), 42  
command line tools for  
    troubleshooting, 382, 382–383  
Common Sense Media, 373  
Common Vulnerabilities and Exposures  
    (CVE) system, 387  
COMPUTE! magazine, 378  
Computer Fraud and Abuse Act  
    (CFAA), 294  
concentrator boards, 198  
constraints, 377–378  
containers, 56–57  
continuous mode operation, 173  
core snap, 190  
CPD (captive portal detection), 94–95  
CPI (captive portal identification), 101

- cron daemon, 85
- crontab command, 85
- cyclic redundancy checks (CRCs), 20
- CYW/BCM43455 chipset
  - block diagram of, 26
  - Bluetooth support, 37
  - MIMO, 18
  - overview, 24, 24–27
  - speed limitations, 25

## D

- DACs. *See* digital-to-analog converters
- Data Link layer, 322
- data rate, 114
- data volume quotas, 114
- dBi, 29
- dd utility, 272
- deactivate command, 80
- deauthenticating client stations, 293
- death flood attacks, 218
- Debian Bookworm, 57
- Debian Bullseye, 57, 361
- debouncing, 169–170
- debug messages, 381–382
- decibel milliwatts (dBm), 12
- declarative constraints, 378
- decomposition of systems, 379
- dedicated short-range communications (DSRC), 17
- deep neural networks (DNNs), 268
- default gateway (default route), 94, 97
- Defense Advanced Research Projects Agency (DARPA), 310
- development environments
  - choosing, 42–51
    - direct access via terminal, 42–44, 44
  - editor choices, 45, 45–46
  - key-based authentication, 44–45
  - remote development, 49–51, 50
  - self-hosted code-servers, 46–49, 47, 49
- defined, 42
- LoRa, 167–168
- virtual project environments, 239–240

DevEUI, 192–193

Device and Service Discovery (DSD) protocol, 351

DHCP (Dynamic Host Control Protocol), 55–56, 325

dhdtutil utility, 270

Digital Rights Management (DRM), 347

digital signal processors (DSPs), 25

digital-to-analog converters (DACs), 123–126

- audio DAC SHIM, 124
- connecting to hi-fi equipment, 144
- home audio components, 126
- integrating into Raspberry Pi, 143–144
- Raspberry Pi IQaudio DAC+ HAT, 125
- USB to 3.5 mm jack adapter, 124

dipole size, 29, 29

directories in Linux, 65

direct-sequence spread spectrum (DSSS), 16

distributed denial-of-service (DDoS) attacks, 207

dmesg tool, 382, 390–391

DNS (Domain Name System), 99–100

dnsmasq package, 99, 326–327, 332

Docker, 58–61

- docker compose up command, 58
- docker-compose.yaml* file, 59

Docker Engine, 57–58

Domain Name System (DNS), 99–100

dot files, 43

drivers, 31, 126

DRM VC4 V3D driver, 355–356

duty cycle, 174

dwelt time (airtime; time on air), 177

dynamic bucket size, 115

dynamic captive portals. *See also* captive portals

- creating, 105–114
  - connecting client devices, 112–113, 113
  - implementing forward authentication, 107–108
  - installing Lighttpd, 108–110
  - managing clients, 113–114
  - setting up openNDS, 106
  - using Lighttpd with stand-alone RaspAP, 110–112, 111
- troubleshooting, 385–387

dynamic frequency selection (DFS), 19  
Dynamic Host Control Protocol  
(DHCP), 55–56, 325  
dynamic splash pages. *See* captive portals

## E

edge detection, 169–170  
Edimax BT-8500 Nano USB Adapter, 123  
Edimax EW-7811Un Nano, 205, 205,  
312, 312, 319–320  
editors, 45, 45–46  
e-ink display modules, 264–265  
electromagnetic spectrum (EMS), 4–5, 5  
Emacs editor, 46  
encrypted connections, 34  
engineer’s mindset, 375–393  
    Coffee Grind Size software,  
    375–377, 376  
    developing, 377–379  
    problem-solving methodology,  
    379–387, 380  
    applying fixes, 384  
    defining problems, 380  
    forming hypotheses, 383  
    iterating, 384  
    making observations, 380–383, 382  
    narrowing focus, 383–384  
    real-world example, 385–387  
    querying system logs, 388–391  
    tracing socket conflicts, 391–392  
eth0 network interface, 93  
Ethernet standard, 7  
EvilSocket (Simone Margaritelli), 276  
Evil Twin (APSP00F) attacks, 207, 218,  
222, 222  
experience plug-in, 304–305, 305  
extended unique identifier  
(EUI; DevEUI), 192–193

## F

FAS (forward authentication service),  
106–108  
*fas-aes.php* file, 113  
faskey values, 106, 385–387  
fg (foreground) command, 64  
fiber optics, latency in, 11  
files in Linux, 66  
filesystem in Linux, 65–66, 65

fingerprint alerts, 218  
FiraCode font, 82  
firewalls, 206  
firmware, 126  
flash patches, 271  
foreground (fg) command, 64  
forward authentication service (FAS),  
106–108  
four-way handshake, 34, 35,  
292–293, 293  
four-wire serial bus. *See* serial peripheral  
interfaces  
FQDNs (fully qualified domain  
names), 99  
free and open source software (FOSS),  
202–203  
Free Lossless Audio Codec (FLAC), 146  
Freifunk Community API, 341  
Freifunk Paderborn project, 310,  
310–311, 340, 340–341  
frequencies, 4, 7  
frequency-hopping spread spectrum  
(FHSS), 36  
full-upgrade command, 51  
fully qualified domain names  
(FQDNs), 99  
fuser tool, 382

## G

Gagné, Jonathan, 375–377  
gain in antennas, 29  
gateways, 325–328, 325  
    adding to mesh networks, 326–328  
    announcing bandwidth, 334  
    default gateway, 94, 97  
    managing mesh network nodes with,  
    331–332  
    role of, 326  
Geerling, Jeff, 27–28  
general-purpose input/output (GPIO)  
    defined, 69  
    edge detection, 195  
    faults, 299–301  
    headers, 231, 231, 267  
    pins, 165, 235–238  
    Raspberry Pi, 73, 73–76  
    Zero Python library, 73  
generators, 301

- geohashes, 241
- getty service, 166–167, 195–196, 300–301
- global translation tables, 339
- GND (ground) pins, 237
- GNU troubleshooting tools, 382, 382
- Google Cast, 349
- Google Chromecast, 348–349, 350
- Google TV, 373
- GPIO. *See* general-purpose
  - input/output
- gpioinfo tool, 382
- GPS, 160, 171, 303–304
- Grafana
  - dashboards, 245–251
    - adding data sources, 245, 245–246, 246
    - creating alerts, 249, 249–250, 250
    - editing, 248, 248–249
    - importing model dashboard, 246–247, 247
    - testing alerts, 250–251, 251
    - using, 247–248
  - installing, 239
- graphical processing units (GPUs), 297
- Groove Salad station, 134
- ground (GND) pins, 237
- group temporal key (GTK), 34
- GStreamer dependency, 363
- Gym toolkit, 278

## H

- H.264 video codec, compiling, 370
- half-duplex systems, 6
- handshakes
  - collecting, 292–298
    - creating target networks, 294–295
    - deauthenticating client stations, 293
    - passively, 294
    - processing PCAP files, 295–297
    - sending association frames, 293–294
    - using Hashcat, 297–298
  - four-way handshake
    - overview, 34
    - process, 35, 292–293, 293
  - mapping with GPS, 303–304, 304
  - in wireless connections, 34

- Hardware Attached on Top (HATs), 71
- Hashcat, 271, 296–298
- hashcat-data package, 298
- hashcat-nvidia tool, 297
- hcxpcapngtool utility, 296–297
- Hcxtools, 271
- HDMI connections, 345, 345, 354
- headless access, 53, 208, 233–234
- hertz (Hz), 4
- High-bandwidth Digital Content Protection (HDCP), 347
- hop penalty, 333–334
- hopping phenomenon, 313, 333
- hops, 11
- hostapd service, 14–15
- host controller interface (HCI), 159
- htop tool, 152, 382
- HTTP
  - 511 response, 99
  - requests on port 80, 94, 98
  - reverse-HTTP protocol, 368
- http-ui caplet, 291
- Hz (hertz), 4

## I

- I<sup>2</sup>C (inter-integrated circuit), 73, 81–82, 88, 259
- I<sup>2</sup>S (inter-IC sound) audio, 124
- iC880A-SPI concentrator board, 198
- ICMP (Internet Control Message Protocol), 321
- IDEs (integrated development environments), 42
- IEEE (Institute of Electrical and Electronics Engineers), 6–8, 7, 8
- iftop tool, 336
- imperative constraints, 378
- INA219.py* program, 88
- InfluxDB, 239, 245
- InfluxDB Query Language (InfluxQL), 248
- information security (InfoSec), 262
- infrastructure mode, 8
- initialization vector (IV), 107
- in-kernel drivers, 92
- Institute of Electrical and Electronics Engineers (IEEE), 6–8, 7, 8

- integrated development environments (IDEs), 42
- IntelliSense, 46
- inter-IC sound (I<sup>2</sup>S) audio, 124
- inter-integrated circuit (I<sup>2</sup>C), 73, 81–82, 88, 259
- Internet Connection Sharing (ICS), 301–302
- Internet Control Message Protocol (ICMP), 321
- internet radio, 134
- inter-process communication (IPC), 391
- intrusion detection, 201–226, 202
  - hardware, 204–206, 205
  - headless access, 208
  - installing Kismet, 208–212
  - monitoring traffic, 214–217, 215, 216
  - parsing Kismet logs, 223–224
  - software, 206
  - use cases, 204
  - wardriving, 202–203, 203
  - WIDS, 206–207, 218–219
  - wireless adapter, 212–214
  - WLAN security test bench, 219–223, 221, 222
- iostat tool, 382
- IoT devices, 197–198, 207
- iPerf tool, 21–23, 22, 119
- iptables tool, 382
- ip tool, 382
- isotropic radiators, 29
- iTerm2, 42
- iw utility, 13
- iwlist utility, 17

**J**

- Java language, 238
- Java Virtual Machine (JVM), 238
- Jayofelony, 272
- jitter, 11
- Jobs, Steve, 347
- jobs command, 338
- journalctl command, 380–381, 382, 388–390
- journal service, 388–390
- Jython, 238

**K**

- Kastrau, Sarah, 266
- Keras, 268–269
- kernel
  - console, 195–196
  - drivers, 31–33
  - land (kernel space), 314
  - updates, 51–52
- key-based authentication, 44–45
- Kismet, 202, 206–224
  - builds, 208
  - dashboard, 216
  - Data Sources dialog, 216
  - initial login screen, 215
  - installing
    - monitoring traffic, 214–217, 215, 216
  - monitor interfaces, 214
  - overview, 206
  - parsing logs from, 223–224
  - PCAP files size, 219
  - suid-root*, 210
  - testing Pwnagotchi with Kismet, 305–306
  - WIDS, 218–219
  - WLAN security test bench, 219–223
- kismetdb\_statistics tool, 223
- kismetdb\_to\_pcap tool, 217

**L**

- lambda ( $\lambda$ ), 4
- Lassam, Curtis, 45
- latency (lag; round-trip time), 10–12, 371
- Layer 3 routing solution, 326
- laziness factor, 278–279
- lazycast, 353–361
  - building binaries, 356
  - connecting sources, 357–359, 358
  - connecting to Wi-Fi Direct group, 358
  - disabling onscreen prompt, 354
  - expected performance levels, 371, 371
  - implementing Miracast, 359–361
  - installing dependencies, 354–355
  - overview, 346
  - RPiPlay vs., 362
  - starting, 357
  - updating drivers, 355–356

- leaky bucket algorithm, 115
- lgpio library, 169–170
- libmicrohttpd (MHD) library, 96–97
- Librespot project, 146
- light scattering technique, 242
- Lighttpd
  - installing, 108–110
    - using with stand-alone RaspAP, 110–112
- link-local addresses, 321
- link quality, 20, 83–84, 319, 334
- Linux, 63–66
  - architecture, 314
  - audio, 126–128, 127
  - configuring Pwnagotchi on, 280–282
  - directories, 65
  - files in, 66
  - filesystem, 65, 65–66
  - inter-process communication, 391
  - key shortcuts, 63–64
  - networking errors, 370
  - troubleshooting, 382, 388–390
- listening mode (monitor mode; promiscuous mode), 205
- lithium-ion polymer (LiPo) batteries, 71, 71–72, 72
  - attaching, 76–78
  - benefits of, 72
  - external battery modules, 263–264
  - LoRa, 162, 162, 183
  - monitoring, 88
- logs
  - parsing Kismet logs, 223–224
  - querying system logs, 388–391
- long range (LoRa) wireless, 38–39, 39, 157–199. *See also* LoRaWAN
  - adding IoT sensors, 197–198
  - building LoRaWAN gateways, 198
  - comparison of spreading factors, 177
  - connecting to LoRaWAN gateways, 185, 185–194
    - adding end device, 192–193
    - authenticating, 191
    - configuring Pi Supply IoT pHAT, 187–190, 188
    - generating configuration file, 191
    - installing The Things Stack CLI, 190–191
    - joining network, 194
    - obtaining device identifiers, 191–192
    - participating in TTN, 186–187, 187
    - verifying device in console, 193, 193–194
- distance record, 180, 181
- expansion boards, 159–161
- hardware, 158–162
  - compatible Raspberry Pi models, 159, 159
  - expansion boards, 159–161
  - LiPo battery HAT, 162, 162
  - Pi Supply IoT LoRa/LoRaWAN pHAT, 161, 161
  - Waveshare LoRa/LoRaWAN Node, 160–161
- impact of spreading factor on transmission, 178
- LoRa physical layer, 38–39
- modulation, 174–176, 175, 176
- packets, 179, 179–180
- parameters, 176–178, 177, 178
- prerequisites, 164–171
  - attaching antenna, 171
  - enabling SPI port, 164, 164–165, 165
  - enabling UART port, 165–167, 166
  - installing Python packages, 169–170
  - preparing environment, 167–168
- running example, 171–174
- Semtech SX1262 overview, 163, 163–164
- software, 162–163
- testing, 180–185, 181
  - distance record, 180–181
  - ideal conditions, 181–182
  - parameter tuning, 183–185, 184
  - technique for evaluating range, 182–183
- troubleshooting, 194–197
- use cases, 158
  - using RAK811 examples, 197
- LoRa Basics Station, 198
- LoRa-RF Python library, 169–170

- LoRaWAN, 39, 39
  - end device activity problems, 196–197
  - end device classes, 161
  - gateways
    - building, 198
    - connecting to, 185, 185–194, 187, 188, 193
    - finding, 194
    - overview, 157–158
  - lossy vs. lossless audio, 146
  - low-pass filters, 143
  - low-power wide area networks (LPWANs), 38, 158
  - lsmcmod command, 32
  - ls of package, 257
- M**
- MAC (Media Access Control) layer, 26, 185
- macOS
  - casting, 349
  - configuring Pwnagotchi on, 278–280, 279
  - RPiPlay project, 346
- Madhavan, Guru, 377
- make utility, 96–97, 364–365, 369–370
- man command, 382
- Margaritelli, Simone (EvilSocket), 276
- maximum transmission unit (MTU)
  - values, 319–320
- Media Access Control (MAC) layer, 26, 185
- Meraki, 310
- mesh11sd project, 119
- mesh-capable adapters, 312
- mesh networking, 309–342
  - BATMAN protocol, 332–335, 333
  - configuring gateway and allowing access, 325, 325–332
  - adding gateway, 326–328
  - managing nodes with gateway, 331–332
  - rebooting mesh network, 328
  - verifying integrity of mesh network, 329–331
  - creating mesh network, 316–325
    - adding nodes, 320
    - creating mesh hostnames, 322–323
    - diagnosing mesh network
      - connectivity, 320–322, 321
    - extending mesh network, 324–325
    - MTU values, 319–320
    - running at boot, 323–324
  - hardware, 311–313, 312, 313
  - history of, 310, 310–311
  - monitoring, 335–339
  - node mobility and hopping, 333
  - preparing nodes, 315–316
  - software, 313–315
  - troubleshooting, 339–340
  - use cases, 311
- message integrity check (MIC), 34
- MHD (libmicrohttpd) library, 96–97
- Microsoft 4K Wireless Display Adapter, 349
- Microsoft Windows
  - configuring Pwnagotchi on, 280
  - wireless displays, 346, 350
- micro USB cables, 265–266, 266
- MIMO (multiple-input multiple-output), 10, 18
- Mini PiTFT. *See* Adafruit Mini PiTFT
- Miracast, 348–349, 359–361
- Miracast over Infrastructure (MICE), 346
- Mirai botnet, 207
- MISO protocol, 165
- monitor command, 55
- monitor mode (listening mode; promiscuous mode), 205
- MOSI protocol, 165
- mpg123 audio player, 134
- mtr tool, 11–12
- MTU (maximum transmission unit)
  - values, 319–320
- multicast (one-to-many) traffic, 36
- multidimensional data array, 269
- multipath phenomenon, 18
- multiple-input multiple-output (MIMO), 10, 18
- multiuser MIMO (MU-MIMO; AC Wave 2 wireless; Next-Gen AC wireless), 18

## N

Nano text editor, 45  
NASA, 311  
NAT (network address translation), 282, 326  
ncspot player, 135–137, 136, 137  
ndsctl auth command, 116  
ndsctl utility, 113–114  
*netactivity.py* script, 337–338  
Netgear A6210 wireless adapter, 92–93  
NetSpot, 371  
netstat tool, 382, 392  
network address translation (NAT), 282, 326  
network-attached storage (NAS), 135  
network interface cards (NICs), 23  
Network layer, 322  
NetworkManager, 54–63  
    command line interface, 54, 89  
    modifying settings, 55–56  
    monitoring connections, 55  
network mapper (nmap) utility, 332  
network speed testing, 20, 21  
network zones, 119  
Nexmon, 269–271, 302  
Next-Gen AC wireless (MU-MIMO; AC Wave 2 wireless), 18  
nmap (network mapper) utility, 332  
nmcli command, 54, 89  
node mobility, 333  
Nova SDS011 particle sensor, 231, 231  
    connecting to Pi Zero 2 W, 238, 238  
    enabling service, 244  
    laser diode service life, 243  
nslookup tool, 97  
nu (*v*), 4  
NumPy library, 83  
Nvidia Shield TV, 373

## O

Ogg Vorbis, 146  
OGM (originator message), 313–314, 332–333  
one-to-many (multicast) traffic, 36  
one-to-one (unicast) traffic, 36  
OpenAQ, 258  
OpenMAX (OMX) standard, 362  
OpenMAX player (omxplayer), 361–362

openNDS, 93–119  
    authentication process, 94, 94  
    connecting client devices, 112–114  
    creating voucher system, 103–105, 105  
    description of, 93  
    dynamic landing pages, 113  
    installing, 97–100  
    preinstalled splash pages, 100, 102–103  
    quota setting, 114–116  
    security measures, 95–96  
    setting up for dynamic captive portals, 106  
    starting, 100–101  
    troubleshooting, 118–119, 385–387  
        background, 385  
        hypothesis and fix, 386–387  
        problem statement and observations, 385–386  
        proximate vs. ultimate causes, 387  
    web server compatibility, 108  
Open Network Demarcation Service.  
    *See* openNDS  
Open Systems Interconnect (OSI)  
    model, 38, 39, 322  
originator message (OGM), 313–314, 332–333  
originators, 313  
Orion spacecraft, 311  
orthogonal frequency division  
    multiplexing (OFDM), 16  
orthogonal signals, 178  
OSI (Open Systems Interconnect)  
    model, 38, 39, 322  
over-the-air activation (OTAA), 190  
over-the-air packet inspection, 207  
over-the-air sniffing, 207

## P

P2P (peer-to-peer), 350–353  
packages, updating, 51–52  
packet capture (PCAP) files, 217, 219, 224, 295–297  
Packet Radio Network (PRNET), 310  
packet shaping (traffic shaping), 117–118  
pair command, 132

- pairwise master key (PMK), 34, 293
- pairwise master key identifier (PMKID), 293
- pairwise transient key (PTK), 34
- particle sizes, measuring, 242
- passwords, generating encrypted, 354
- PCAP (packet capture) files, 217, 219, 224, 295–297
- PCBs (printed circuit boards), 24
  - antennas, 30, 30
- PCM (pulse code modulation), 148
- pdb tool, 382
- peer-to-peer (P2P), 350–353
- PHP, 108–110
- Physical (PHY) layer, 25–26, 185, 322
- PID (process ID), 381
- Pillow library, 79, 83
- Pimoroni Audio DAC SHIM, 124, 124
- ping tool, 11–12, 332, 382
- pinout guides
  - Mini PiTFT, 75–76
  - Raspberry Pi, 73–75
- pinouts, 73
- pinout utility, 73–74, 235–236, 236
- pinout.xyz* website, 74
- PiSugar 2 battery, 263–264, 264, 267–268
- PiSugar plug-in, 286–287
- Pi Supply IoT LoRa/LoRaWAN pHAT, 161, 161, 187–190, 188
- Pi Zero 2 W
  - air quality monitoring stations, 230, 230, 235–238, 236, 238, 258
  - antenna, 30, 30
  - LoRa, 159, 159
  - mesh networking, 311–312, 312
  - Mini PiTFT, 70
  - Pwnagotchi, 263, 263, 267–268
  - radiograph of, 28
- Pi Zero W, *xxxii*
  - Bluetooth, 123
  - Mini PiTFT, 70
- PMK (pairwise master key), 34, 293
- policy functions, 276
- pollution. *See* air quality monitoring stations
- power pins, 236
- preauthentication state, 95
- printed circuit boards (PCBs), 24
  - antennas, 30, 30
- Proant AB, 30
- probe requests and responses, 33
- problem-solving methodology, 379–387, 380
  - applying fixes, 384
  - defining problems, 380
  - forming hypotheses, 383
  - iterating, 384
  - making observations, 380–383, 382
  - narrowing focus, 383–384
  - real-world example, 385–387
- process ID (PID), 381
- programming languages, 237–238
- promiscuous mode (listening mode; monitor mode), 205
- proximate causes, 387
- proxy certificates, 335
- ps tool, 382
- psutil package, 338
- PTK (pairwise transient key), 34
- PulseAudio, 128–131, 149, 152
- pulse code modulation (PCM), 148
- pulse width modulation (PWM), 142–143, 143
- PuTTY, 42
- Pwnagotchi, 261–307
  - AI in, 275–279
    - actor-critic process loop, 276, 276–278
    - reward function, 278–279
  - assembling components, 266–268, 267, 268
  - Bettercap, 289–292
  - collecting handshakes, 292–298, 293
    - creating target networks, 294–295
  - deauthenticating client stations, 293
  - mapping handshakes with GPS, 303–304, 304
  - processing PCAP files, 295, 295–297
  - sending association frames, 293–294
  - using Hashcat, 297–298
- configuring, 272–275, 274, 275

- customizing personality and expressions, 302–303
- experience plug-in, 304–305, 305
- hardware, 262–266
  - 3D-printed cases, 266, 266
  - compatible Raspberry Pi models, 263, 263
  - e-ink display module, 264, 264–265, 265
  - external battery module, 263–264, 264
  - micro USB cable, 265–266, 266
- Kismet and, 225, 305–306
- MANUAL mode, 283–288, 284
  - default file and folder locations, 306
  - installing PiSugar plug-in, 286–287
  - performing full backup and restore, 288
  - plug-ins, 285, 285–286
  - saving and restoring configurations, 287–288
- preparing SD card, 271–272
- short stack unit, 268
- software, 268–271, 270
- streamlining data collection, 306, 306–307
- troubleshooting, 299–302
- USB Ethernet gadget mode, 279–283
  - configuring device on Linux, 280–282
  - configuring device on macOS, 279–280, 280
  - configuring device on Windows, 280
  - connecting via SSH, 283
  - sharing internet connectivity, 282
  - use cases, 262
- pwning Wi-Fi. *See* Pwnagotchi
- pySerial, 240–241
- Python, 237, 238
  - packages, 79, 169–170
  - virtual environments, 79–80
- Python Virtual Machine (PVM), 237–238

## Q

- quality of service (QoS), 116
- queue length, 115
- Qwiic/STEMMA QT-compatible sensors, 232

## R

- radio frequencies (RFs), 5
- RAK2247 Pi HAT, 198
- RAKWireless RAK811 module, 161, 188–190, 197
- raspap-docker package, 60
- RaspAP project, xxx–xxxi, xxxi, 56–63
  - accessing web UI, 62–63
  - deploying with Docker, 58–61
  - installing Docker Engine, 57–58
  - introducing containers, 56–57
  - manipulating AP settings, 220–222
  - monitor mode, 112
  - PHP and, 110
  - using Lighttpd with, 110–112, 111
  - using RaspAP Quick installer, 61–62
- Raspberry Pi 3 Model B, 23
- Raspberry Pi 3 Model B+, 23
- Raspberry Pi 4 Model B
  - overview, 344, 344–345
  - video specs, 345
  - WLAN chipsets, 23
- Raspberry Pi 5, 23
- Raspberry Pi 400, 23
- Raspberry Pi Compute Module 4 (CM4), 23, 24
- Raspberry Pi Imager tool, 78, 208, 234, 272, 316
- Raspberry Pi IoT LoRa pHAT, 161
- Raspberry Pi IQaudio DAC+ HAT, 125
- Raspberry Pi OS Lite, 51, 233–234, 353
- Raspberry Pi Zero W model, 23
- Raspberry Pi Zero 2 W model, 23
- raspi-blinka.py* script, 81–82
- raspi-config command, 19, 52
- real-time clock (RTC), 264
- received signal strength indicator (RSSI), 13
- receiving antennas, 28
- reception (Rx), 6
- recomposition of systems, 379
- reference monitors, 258

- reinforcement learning (RL), 275
- Remote - SSH extension, 49–51, 50
- remote development, 49–51
- Remote Network Driver Interface
  - Specification (RNDIS), 279
- reward function, 278–279
- RFs (radio frequencies), 5
- rgb\_display\_minipitfttest.py* script, 83
- RNDIS/Ethernet Gadget service, 279, 280
- Robust Security Network (RSN)
  - protocol, 36, 293
- Roku Stick, 373
- Roofnet project, 310
- root users, 58
- routers, 93, 97–98
- routing tables, 98, 314
- RPi.GPIO library, 78, 169–170
- rpi-igpio library, 169–170
- RPiPlay, 347, 362–369
  - building RPiPlay binaries, 364–365
  - command line options, 366
  - connecting sources, 366, 366–367
  - implementing AirPlay, 367–369
  - installing prerequisites, 363
  - lazycast vs., 362
  - starting RPiPlay, 365–366
- rpi-update*, 362
- RSN (Robust Security Network)
  - protocol, 36, 293
- RSSI (received signal strength indicator), 13
- RTC (real-time clock), 264
- Rx (reception), 6

## S

- Sautner, Roland, 252
- SBCs (single-board computers), xxxii
- Scientific American* magazine, 378
- SCLK function, 165
- screen mirroring, 344
- SD cards
  - flashing
    - with Bullseye Lite, 315
    - with Raspberry Pi OS Lite, 208, 233–234
  - preparing
    - for Mini PiTFT, 78
    - for Pwnagotchi, 271–272
- Secure Shell (SSH), 42, 44, 53, 283
- self-healing networks, 310
- self-hosted code-servers, 46–49, 47, 49
- Semtech, 38
  - SX1262 LoRa RF chip, 163–164
- serial peripheral interfaces (SPIs), 73, 81–82, 164, 164–165, 165
- serial ports
  - getty service, 300–301
  - history of, 166–167
  - preparing for Nova sensor, 234–235
- service set identifiers (SSIDs), 13
- SF (spreading factor), 177, 177–178, 178, 184, 184
- shared medium, 16
- shells, 42
- SHIM (“shove hardware in middle”), 124
- short stacks, 267–268, 268
- signal boosting, 10
- signal strength, 12–13, 13, 83–85, 88–89
- single-board computers (SBCs), xxxii
- single-input single-output (SISO), 26
- Sixfab 3G-4G/LTE Base HAT, 232, 232–233
- smart queue management (SQM), 116
- smart TV spying, 372–373, 373
- snapped daemon, 136
- snappy packages, 190
- snaps, 136
- SNonce (supplicant nonce), 34
- SoC (system on a chip), 24
- Soma FM, 134
- sound cards, 126
- sound servers, 127
- spatial multiplexing technique, 18
- speakers
  - connecting via Bluetooth
    - enabling snapd, 136
    - getting device properties, 133
    - installing ncspt, 136–137, 137
    - pairing, 132–133
    - playing music from audio library, 135, 135–136
    - streaming internet radio from console, 134
    - streaming Spotify from terminal, 135–136, 136

- testing audio output, 133
  - using Bluetooth controller, 131–132
- troubleshooting, 151
- speaker-test command, 133
- speed, network, 7–10
- SPI (serial peripheral interface), 73,
  - 81–82, 164, 164–165, 165
- spidev library, 78, 169
- splash pages, 95, 100, 102–103. *See also*
  - captive portals
- Spotify. *See also* Spotify Connect;
  - Spotifyfyd
  - Spotify Music Pro, 146
  - streaming from terminal, 135–136, 136
- Spotify Connect, 145–146
- Spotifyfyd, 146
  - configuring, 147–148
  - installing, 146–147
  - playback, controlling, 150, 150–151
  - Spotify Connect box, creating with,
    - 145–151
  - starting service, 150
  - systemd service, creating, 148–149
- spreading factor (SF), 177, 177–178,
  - 178, 184, 184
- spread spectrum modulation, 174–176
- spring pins, 264
- SQM (smart queue management), 116
- Squid RGB, 312–313, 313, 336–337
- SSH (Secure Shell), 42, 44, 53, 283
- SSIDs (service set identifiers), 13
- ss tool, 382
- stackable, 76
- Stallman, Richard, 46
- station (STA), 33
- strace utility, 381, 382
- streaming devices and consumer
  - privacy, 373
- supplicant nonce (SNonce), 34
- supplicants, 34
- sweep signals (chirps), 175–176, 176
- SX1262 LoRaWAN Node expansion
  - board, 160, 160
- symbolic links (symlinks), 286
- Synaptics SYN43436 chipset, 27–28, 28
- sync word, 179
- systemctl tool, 382
- systemd daemon, 196

- system journal, 380–381
- System Management Bus (SMBus)
  - protocol, 88
- system on a chip (SoC), 24
- systems-level thinking, 378–379

## T

- T x R : S* notation, 26
- TCP (Transmission Control Protocol), 11
- tcpdump tool, 224, 382
- TensorFlow, 268–269
- tensors, 268–269, 269
- terminal, 42–44, 44
  - emulators, 42
  - multiplexers, 43–44
- Terminus, 42
- Texas Instruments PCM5122 DAC
  - chip, 125
- TFT display. *See* thin-film transistor display
- ThemeSpec splash page, 102
- The Things Network (TTN),
  - 186–187, 194
- The Things Stack CLI (ttn-lw-cli)
  - adding end device, 192–193
  - authenticating, 191
  - generating configuration file, 191
  - installing, 190–191
  - obtaining device identifiers, 191–192
  - verifying device in console, 193–194
- The Things Stack Sandbox (TTSS),
  - 186–187, 187, 193
- thin-film transistor (TFT) display, 69–90
  - assembling components, 76,
    - 76–78, 77
  - customizing AP monitor, 89–90
  - customizing signal monitor, 88–89
  - GPIO, 73, 73–76, 74, 75
  - hardware required, 70–72, 71, 72
  - integrating Bluetooth audio with, 154
  - monitoring LiPo battery, 88
  - preparing SD card, 78
  - running AP monitor, 86–88, 87
  - running signal strength monitor,
    - 83–85
  - software, 72–73
  - use cases, 70

thin-film transistor display (*continued*)  
 writing to display, 78–83  
   automating installation, 80–81  
   checking I<sup>2</sup>C and SPI, 81–82  
   creating virtual project  
     environment, 80  
   prerequisites, 79–82  
   Python setup, 82–83  
   using Python virtual environments,  
     79–80

Thingiverse, 252

throughput, 10–12

time on air (ToA; airtime; dwell  
 time), 177

Tmux, 43–44, 44

Tom’s Obvious Minimal Language  
 (TOML), 171

traceroute tool, 11–12, 382

traffic shaping (packet shaping), 117–118

transmission (Tx), 6

Transmission Control Protocol (TCP), 11

transmitting antennas, 28

trend-based monitors, 218

troubleshooting. *See also* problem-  
 solving methodology  
   air quality monitoring stations, 257–258  
   Bluetooth audio, 151–152  
   captive portals, 118–119  
   command line tools for, 382, 382–383  
   LoRa, 194–197  
   mesh networking, 339–340  
   openNDS, 118–119  
   Pwnagotchi, 299–302  
   real-world example, 385–387  
   wireless displays, 369–372

TTN (The Things Network),  
 186–187, 194

ttn-lw-cli. *See* The Things Stack CLI

TTN Mapper, 194

TTSS (The Things Stack Sandbox),  
 186–187, 187, 193

ttyAMA0 file, 166–167

Tx (transmission), 6

**U**

UART (universal asynchronous receiver-  
 transmitter) interface, 73, 165,  
 165–167, 235–237, 237

UDP (User Datagram Protocol), 11, 348

ultimate causes, 387

uncontrolled ports, 35

unicast (one-to-one) traffic, 36

Unified Configuration Interface  
 (UCI), 100

UNII channel blocks, 17

uninterruptible power supply HAT, 313

USB adapters, external, 138–140

USB On-The-Go port, 279

USB to 3.5 mm jack adapters, 124, 124

userland (user space), 314

userland libraries, 353

## V

value function, 276

VESA-compatible project case, 346, 346

VideoCore, 361–362

Video for Linux API version 2 (V4L2)  
 API, 362

video processing units (VPUs), 362

Vim editor, 46

virtual project environments, 167–168,  
 239–240

Visual Studio Code (VS code), 46

vmstat tool, 382

volume, controlling, 144–145

voucher system, 103–105

VS Code (Visual Studio Code), 46

VS Code Remote - SSH extension,  
 49–51, 50

## W

walled gardens, 95, 119

wardialing, 202

Wardriver plug-in, 304

wardriving, 202–203, 203

warwalking, 304

waveform elements, 4

wavelength  
   calculating, 28–29  
   defined, 4

Waveshare 2.13-inch E-Ink display HAT,  
 264, 264–265, 267–268

Waveshare 2.13-inch E-Paper HAT+,  
 265, 265

Waveshare Li-polymer Battery HAT,  
 71, 72

- Waveshare LoRa/LoRaWAN node, 160–161
- Waveshare Pi Zero UPS HAT, 76–77, 77
- Waveshare SX1262 LoRa expansion board, 160, 160, 163
- Waveshare UPS HAT, 71, 71, 162, 313
- WFD (Wi-Fi Direct), 348, 350–353, 351
- WIDS (wireless intrusion detection systems), 206–207, 218–219
- Wi-Fi
  - antennas, 28–31, 29, 30, 31
  - CERTIFIED seal, 5
  - country code, 52
  - hotspots, creating, 54
  - landing pages. *See* captive portals
  - networks, number of, 201–202, 202
  - splash pages. *See* captive portals
- WiGLE site, 203
- Windows, Microsoft
  - configuring Pwnagotchi on, 280
  - wireless displays, 346, 350
- wireless adapters
  - configuring for intrusion detection, 212–214
  - defined, 6
  - hardware, 23–31
    - BCM/SYN43436 chipset and antenna, 27–28, 28
    - CYW/BCM43455 chipset, 24, 24–27, 26
    - Wi-Fi antennas, 28–31, 29, 30, 31
    - WLAN chipsets used in popular Pi models, 23
  - interface combinations, 352–353
  - interface modes, 352
  - for intrusion detection project, 205–206
- wireless APs, setting up, 54–63
  - NetworkManager, 54–63
    - modifying settings, 55–56
    - monitoring connections, 55
  - RaspAP, 56–63
    - accessing web UI, 62–63
    - deploying with Docker, 58–61
    - installing Docker Engine, 57–58
    - introducing containers, 56–57
    - using RaspAP Quick installer, 61–62
- Wireless Display (WiDi), 347
- wireless displays, 343–374
  - alternative to surveillance, 373–374
  - casting, 347–350, 350
  - hardware, 344, 344–346, 345, 346
  - installing lazycast, 353–361
    - building binaries, 356
    - connecting sources, 357–359, 358
    - disabling onscreen prompt, 354
    - implementing Miracast, 359–361
    - installing dependencies, 354–355
    - starting lazycast, 357
    - updating drivers, 355–356
  - installing RPiPlay, 362–369
    - building RPiPlay binaries, 364–365
    - connecting sources, 366, 366–367, 367
    - implementing AirPlay, 367–369
    - installing prerequisites, 363
    - starting RPiPlay, 365–366, 366
  - smart TV spying, 372–373, 373
  - software, 346–347
  - troubleshooting, 369–372, 371
  - use cases, 344
  - VideoCore, 361, 361–362
  - Wi-Fi Direct, 350–353, 351
- wireless fidelity, 5–6
- wireless intrusion detection systems (WIDS), 206–207, 218–219
- wireless local area networks (WLANs), 8
  - bandwidth tests, 20–23
  - chipsets used in popular Pi models, 23
  - security test bench, 219–223, 221, 222
- wireless regional area networks (WRANs), 7
- wireless regulatory domains, 19
- wireless specialty networks (WSNs), 7
- wireless traffic monitoring, 214–217
- wlan0 network interface, 93
- wordlist.txt* file, 298
- World Air Quality Index (AQI), 229, 229
- wpa\_cli program, 360
- wpa\_supplicant.conf* files, 19, 78
- wpa\_supplicant utility, 53, 315–316, 370
- WRANs (wireless regional area networks), 7
- WSNs (wireless specialty networks), 7