



O'Reilly Network oreilly.com Safari Bookshelf Conferences

[Sign In/My Acc](#)

Articles Weblogs Books Learning Lab eDocuments Podcasts

advertisement

SAP TechEd '05. Learn to innovate. Register now & save!

Search



Advanced Search »

Login
 Register
 Manage Newsletters
 Register Your Books

Sponsored Developer Resources

Altova Developer Zone
 ASP.NET 2.0 Training Center
 Driver Updates
 GoToMyPC® Remote Access
 Inside Lightroom

Sites

codezoo.com
 Databases
 Digital Media
 Emerging Telephony
 LinuxDevCenter.com
 MacDevCenter.com
 Mozilla DevCenter
 ONJava.com
 ONLamp.com
 Apache
 BSD
 MySQL
 PHP
 Python
 Security
 OpenP2P.com
 Perl.com
 Policy DevCenter
 Ruby
 SysAdmin
 What Is...
 WindowsDevCenter.com
 Wireless DevCenter
 XML.com
 WebServices.XML.com

Affiliate Sites

When Is 54 Not Equal to 54? A Look at 802.11a, b, and g Throughput



Tags: [wireless](#)
[Bookmark with del.icio](#)

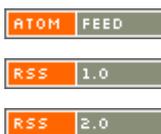
by [Matthew Gast](#), author of [802.11 Wireless Networks: The Definitive Guide](#)
 08/08/2003 (updated: 8/14/2003)

Now that the 802.11g standard has been finalized, comparisons with the other standards in the 802.11 family are inevitable. One conclusion that is frequently drawn is that 802.11g offers similar speeds to 802.11a. After all, both products are advertised as having a data rate of 54 Mbps.

This article develops a simple model for the maximum TCP throughput of 802.11 networks so that a comparison can move beyond a simple comparison of nominal bit rates. According to the model, 802.11g is significantly faster than 802.11b. In a network consisting only of 802.11g clients, it is even slightly faster than 802.11a. However, "protection" mechanisms added to 802.11g to ensure backwards compatibility with legacy 802.11b clients can cut the throughput by 50 percent or more.

Technology	Transactions per second	Mbps of TCP payload throughput	Transactional speed relative to 802.11
11b, 11 Mbps	479	5.6	1.0
11a, 54 Mbps	2,336	27.3	4.9
11g, 54 Mbps/no protection	2,336	27.3	4.9

LinuxQuestions.org
OSDir.com
Servlets.com



11g, 54 Mbps/CTS-to-self protection	1,113	13.0	2.3
11g, 54 Mbps/RTS/CTS protection	750	8.8	1.6

Related O'Reilly Books

- ▶ [802.11 Security](#)
- ▶ [802.11 Wireless Networks: The Definitive Guide](#)
- ▶ [BlackBerry Hacks](#)
- ▶ [Building Wireless Community Networks](#)
- ▶ [Linux Unwired](#)
- ▶ [Mac OS X Unwired](#)
- ▶ [Palm OS Network Programming](#)
- ▶ [Palm OS Programming](#)
- ▶ [Programming Visual Basic for the Palm OS](#)
- ▶ [Talk Is Cheap](#)
- ▶ [Trucos Redes Inalambricas](#)
- ▶ [Windows XP Unwired](#)
- ▶ [Wireless Hacks](#)
- ▶ [Wireless Java](#)

The Transactional Model

The basic transactional model assumed by this article is a maximum-length frame containing a single TCP segment followed by a TCP acknowledgement. To cope with the inherent unreliability of radio waves, the 802.11 MAC requires positive acknowledgement of every transmission. Each TCP packet must therefore be wrapped up in a frame exchange. The complete transaction consists of the two 802.11 data frames carrying TCP embedded in their respective exchanges:

The TCP data segment:

- Distributed Interframe Space (DIFS): this interframe space indicates that an exchange has completed, and it is safe to access the medium again
- The data frame containing the TCP segment.
- A Short Interframe Space (SIFS), which is a small gap between the data frame and its acknowledgement.
- The 802.11 ACK frame.

The TCP ACK:

- DIFS
- The data frame containing the TCP ACK
- SIFS
- 802.11 ACK frame

For this model, I will assume that the host operating system limits the outgoing frame size to 1,500 bytes. 802.11 permits much larger frame sizes, but this flexibility has not traditionally been used by client products. Most access points connect to existing networks with Ethernet, and therefore limit the payload size to the maximum Ethernet payload size. (In fact, this simple precaution is required to obtain Wi-Fi certification.)

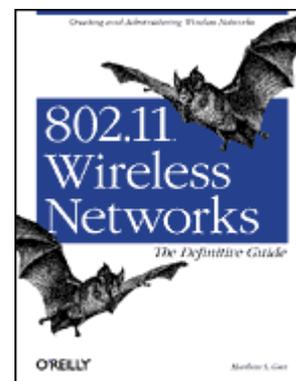
Encapsulation

In addition to the payload data, there are 36 additional bytes of data added in the encapsulation process. The 802.11 MAC header adds 28 bytes of data for various control and management functions, error detection, and addressing. A further eight bytes are added by the SNAP encapsulation header to identify the network layer protocol. (For further encapsulation details, see Figures 3-9 and 3-13 of *802.11 Wireless Networks*.) The total size of the MAC payload for the TCP data segment is 1,536 bytes (12,288 bits). For the TCP ACK, the 40 bytes of TCP/IP header information also gets a further 36 bytes of SNAP header for a total of 76 bytes (608 bits).

Throughput

This article measures the throughput of the

Related Reading



[802.11 Wireless Networks: The Definitive Guide](#)
Creating and

network at the TCP payload layer. Each transaction delivers one data frame. There are 1,500 bytes of Ethernet payload, but 40 bytes must be subtracted for headers. By adding up the total time required for each component of the transaction, a transaction rate can be derived. For the throughput, simply multiply by 1,460 bytes.

Weaknesses of the Model

The transactional model is quite simple, and neglects several important effects. First of all, it assumes a steady stream of well-ordered frames with no contention for the medium. 802.11 implements collision avoidance and exponential backoff, so in reality, the time between frame exchanges will be longer than one DIFS. Exponential backoff the presence of contention will further decrease throughput. My initial back-off the envelope estimate is that contention for the medium would reduce the maximum throughput figures above by 25% to 50%, depending on the exact assumptions made.

Furthermore, there is always the possibility of simultaneous transmission. 802.11 will order layer two frames and prevent collisions, but higher-layer packets may be transmitted simultaneously. 802.11 treats TCP segments and acknowledgements as unrelated higher-layer frames, so a certain number of collisions are expected. (One engineer has told me that up to 10% retransmissions should be expected for TCP/IP on 802.11 due to this effect.)

Furthermore, the model ignores the sophistication in the TCP acknowledgement model. To avoid constraining throughput, TCP uses "sliding windows" and allows multiple outstanding frames to be transmitted before acknowledgement. In practice, TCP acknowledgements can apply to multiple segments, so this model overstates the impact of higher-layer protocol acknowledgements.

The Baseline: 802.11b at 11 Mbps

The baseline speed comes from 802.11b. It is not as fast as the newer specifications, but it's what most of the 802.11 cards out in the world do. I'll do the calculation for 802.11b first, and then compare everything else to the results for 802.11b.

First off, the basic timing numbers for 802.11b:

- SIFS = 10 μ s
- Slot time = 20 μ s
- DIFS = 2 x Slot time + SIFS = 50 μ s

802.11b requires that a preamble be prepended to every frame before it is transmitted to the air. That preamble may be either the traditional "long" preamble, which requires 192 μ s for transmission, or it may be an optional "short" preamble that requires only 96 μ s. (For a diagram of the physical layer headers in 802.11b, see Figure 10-26 in *802.11 Wireless Networks*.) Support for the long preamble is mandatory, and is the default setting on most devices. In the interest of reducing the number of calculations, this article will perform calculations using only the long preamble.

802.11b running at the max speed divides data up into 8-bit symbols. There are 1,536 8-bit blocks in the TCP data segment, and 76 8-bit blocks in the TCP acknowledgement. The ACK does not have SNAP headers, and is only 14 bytes long.

Administering Wireless Networks By [Matthew Gast](#)

[Table of Contents](#)

[Index](#)

[Sample Chapter](#)

[Read Online--Safari](#)

Search this book on Safari:

Only This Book

Code Fragments only

Encoding the MAC frames is easy. 802.11b divides up the MAC frame into a series of 8-bit "symbols," and then transmits 1.375 million symbols per second. So add up the individual components of the transaction to get the total duration.

	TCP data	TCP ACK
DIFS	50 μ s	50 μ s
802.11 Data	192 μ s + 1536/1.375 Msps = 192 μ s + 1,118 μ s = 1,310 μ s	192 μ s + 76/1.375 Msps = 192 μ s + 56 μ s = 248 μ s
SIFS	10 μ s	10 μ s
802.11 ACK	192 μ s + 14/1.375 Msps = 192 μ s + 11 μ s = 203 μ s	= 203 μ s
Frame exchange total	1,573 μ s	511 μ s
Transaction Total	2,084 μ s	

Each transaction requires 2,084 μ s. At that duration, 479 exchanges can complete per second. With a TCP payload of 1,460 bytes per exchange, the throughput is 5.7 Mbps.

802.11a

802.11a is faster than 802.11b for two reasons: timing relationships between frames in the exchanges are tighter, and the encoding used by 802.11a does not require such long preambles for synchronization.

Here are the basic timing numbers for 802.11a:

- SIFS = 16 μ s
- Slot time = 9 μ s
- DIFS = 2 x Slot time + SIFS = 34 μ s

Like 802.11b, 802.11a divides data up into a series of symbols for transmission. However, the encoding used by 802.11a uses much larger symbols. At 54 Mbps each symbol encodes 216 bits. (For a full listing of encoding block sizes, see Table 11-3 in *802.11 Wireless Networks*.) The OFDM encoding used by 802.11a adds six bits for encoding purposes to the end of the frame, so our maximum length frame of 1,536 bytes becomes a series of 12,288 data bits plus the six tail bits. The total bit string of 12,294 bits can be encoded in 57 symbols. The 608 bit TCP ACK plus tail bits requires four symbols, and the 802.11 ACK also requires just one symbol.

Each frame is prepared for transmission in the air with a 20 μ s header to synchronize the receiver. Following the 20 μ s header is a series of symbols, each requiring 4 μ s for transmission.

	TCP data	TCP ACK
DIFS	34 μ s	34 μ s
802.11 Data	20 μ s + 57 * 4 μ s/symbol = 20 μ s + 228 μ s = 248 μ s	20 μ s + 3 * 4 μ s/symbol = 20 + 12 μ s = 32 μ s
SIFS	16 μ s	16 μ s
	20 μ s + 1 * 4 μ s/symbol	

802.11 ACK	= 20 μ s + 4 μ s = 24 μ s	= 24 μ s
Frame exchange total	322 μ s	106 μ s
Transaction Total	428 μ s	

At 428 μ s per transaction, it is possible to complete 2,336 exchanges per second. That corresponds to a throughput of 27.3 Mbps.

Pages: [1](#), [2](#)

[Next Page](#)

[Contact Us](#) | [Advertise with Us](#) | [Privacy Policy](#) | [Press Center](#) | [Jobs](#)

Copyright © 2000-2006 O'Reilly Media, Inc. All Rights Reserved.
All trademarks and registered trademarks appearing on the O'Reilly Network are the property of their respective owners.

For problems or assistance with this site, email help@oreillynet.com



O'Reilly Network

oreilly.com

Safari Bookshelf

Conferences

[Sign In/My Acc](#)[Articles](#)[Weblogs](#)[Books](#)[Learning Lab](#)[eDocuments](#)[Podcasts](#)

advertisement

SAP TechEd '05. Learn to innovate. Register now & save!

Search

Go

[Advanced Search »](#)

Login

[Register](#)[Manage Newsletters](#)[Register Your Books](#)**Sponsored Developer Resources**[Altova Developer Zone](#)
[ASP.NET 2.0 Training Center](#)[Driver Updates](#)[GoToMyPC® Remote Access](#)[Inside Lightroom](#)**Sites**[codezoo.com](#)[Databases](#)[Digital Media](#)[Emerging Telephony](#)[LinuxDevCenter.com](#)[MacDevCenter.com](#)[Mozilla DevCenter](#)[ONJava.com](#)[ONLamp.com](#)[Apache](#)[BSD](#)[MySQL](#)[PHP](#)[Python](#)[Security](#)[OpenP2P.com](#)[Perl.com](#)[Policy DevCenter](#)[Ruby](#)[SysAdmin](#)[What Is...](#)[WindowsDevCenter.com](#)[Wireless DevCenter](#)[XML.com](#)[WebServices.XML.com](#)**Affiliate Sites****When Is 54 Not Equal to 54? A Look at 802.11a, b, and g Throughput**Pages: [1](#), [2](#)[Print](#)Tags: [wireless](#)[Bookmark with del.icio](#)**802.11g**

Performing the same calculations for 802.11g is a bit tricky. 802.11g operate the same frequency band as 802.11b, and is required to remain backwards-compatible. The encoding used by 802.11g will not be recognized by 802.11b stations, so "protection" mechanisms are defined to limit the cross-talk in mixed b/g environments. Essentially, the protection mechanisms require that 802.11g stations operating at high rates pre-reserve the radio medium by using slower 802.11b-compatible reservation mechanisms.

802.11g SIFS = 10 μ s802.11g short slot time = 9 μ s (802.11g-only mode with no legacy stations)802.11g long slot time = 20 μ s (mixed mode requires slow slot time)

802.11g uses many of the same timing parameters as 802.11a. However, it is saddled with backwards compatibility requirements. It inherits the short 10 μ s SIFS time from 802.11b, but the high-rate coding in 802.11g needs additional time. Therefore, 802.11g adds a 6 μ s "signal extension" time at the end of every frame.

802.11g-only BSS

When no 802.11b stations are present, no protection is required. This situation is extremely unlikely, given the huge installed base of 802.11b cards.

As always, start with the basic timing parameters:

- 802.11g SIFS = 10 μ s
- 802.11g fast slot time = 9 μ s (can only be used when no 802.11b stations are present)
- 802.11g DIFS = 2 x Slot time + SIFS = 28 μ s

The 802.11g ERP-OFDM PHY is nearly identical to the 802.11a PHY, except that it operates in a different frequency band and uses a shorter SIFS time. Physical layer headers are identical, as is the coding. Therefore, the calculation for the time required to transmit a frame is nearly identical, with only minor changes to the interframe space times.

LinuxQuestions.org
OSDir.com
Servlets.com



Related O'Reilly Books

- ▶ [802.11 Security](#)
- ▶ [802.11 Wireless Networks: The Definitive Guide](#)
- ▶ [BlackBerry Hacks](#)
- ▶ [Building Wireless Community Networks](#)
- ▶ [Linux Unwired](#)
- ▶ [Mac OS X Unwired](#)
- ▶ [Palm OS Network Programming](#)
- ▶ [Palm OS Programming](#)
- ▶ [Programming Visual Basic for the Palm OS](#)
- ▶ [Talk Is Cheap](#)
- ▶ [Trucos Redes Inalambricas](#)
- ▶ [Windows XP Unwired](#)
- ▶ [Wireless Hacks](#)
- ▶ [Wireless Java](#)

	TCP data	TCP ACK
DIFS	28 μs	28 μs
802.11 Data	20 μs + 57 * 4 μs/symbol + 6 μs = 20 μs + 228 μs = 254μs	20 μs + 3 * 4 μs/symbol + 6 μs = 20 + 12 μs = 38 μs
SIFS	10 μs	10 μs
802.11 ACK	20 μs + 1 * 4 μs/symbol + 6 μs = 20 μs + 4 μs + 6 μs = 30μs	= 30 μs
Frame exchange total	322 μs	106 μs
Transaction Total	428 μs	

The transaction length of 802.11g is identical to 802.11a. The interframe spacing is slightly shorter, but this is offset exactly by the signal extension field.

Protection 1: CTS-to-self

Once the first 802.11b station associates with an 802.11g access point, however, protection is required. The minimal protection contemplated by the standard is that 802.11g stations will protect the fast 802.11g frame exchange with a slow Clear To Send (CTS) frame that locks out other stations access to the medium. Protection dramatically reduces the maximum theoretical throughput because the additional CTS transmission is required with its long 802.11b headers.

Longer interframe spacing is required when legacy clients are connected and protection is engaged. The short slot time is only available when no 802.11b stations are present. Once they are present, the frame spacing reverts to the 802.11b standard:

- SIFS = 10 μs
- Slot time = 20 μs
- DIFS = 2 x Slot time + SIFS = 50 μs

A CTS frame is 14 bytes. It will be transmitted at the highest rate understood by all stations attached to the access point, which will be at most 11 Mbps. The CTS frame will be encoded quickly, but only after its long header. Following the CTS will be the Data-ACK sequence at the high 802.11g speeds.

	TCP data	TCP ACK
DIFS	50 μs	50 μs
CTS	192 μs + 14/1.375 Msps = 192 μs + 11 μs = 203 μs	= 203 μs
SIFS	10 μs	10 μs
802.11 Data	20 μs + 57 * 4 μs/symbol + 6 μs = 20 μs + 228 μs + 6 μs = 254 μs	20 μs + 3 * 4 μs/symbol + 6 μs = 20 + 12 μs + 6μs = 38μs
SIFS	10 μs	10 μs

802.11 ACK	$20 \mu s + 1 * 4 \mu s/\text{symbol} + 6 \mu s$ $= 20 \mu s + 4 \mu s + 6 \mu s$ $= 30 \mu s$	= 30 μs
Frame exchange total	557 μs	341 μs
Transaction Total	898 μs	

The total transactional time is over twice as long because of the protection mechanism. At 898 μs per transaction, only 1,113 transactions can complete second, and the throughput drops dramatically to 13.0 Mbps.

Protection 2: RTS-CTS

Using only a CTS frame to reserve the medium is the minimum requirement, it may fail in some cases where there are so-called "hidden nodes" that do not see the CTS. To fully reserve the medium, the initial edition of the 802.11 standard included a two-frame exchange that would fully announce the impending transmission composed of a Request To Send (RTS) frame followed by the CTS frame. Although the standard requires only a CTS-to-self, using a full RTS/CTS will better protect the inner exchange from interference. The final calculation is quite similar to the previous one, with the addition of the RTS frame at the start:

	TCP data	TCP ACK
DIFS	50 μs	50 μs
RTS	$192 \mu s + 20/1.375 \text{ Msps}$ $= 192 + 15 \mu s$ $= 207 \mu s$	= 207 μs
SIFS	10 μs	10 μs
CTS	$192 \mu s + 14/1.375 \text{ Msps}$ $= 192 \mu s + 11 \mu s$ $= 203 \mu s$	= 203 μs
SIFS	10 μs	10 μs
802.11 Data	$20 \mu s + 57 * 4 \mu s/\text{symbol} + 6 \mu s$ $= 20 \mu s + 228 \mu s + 6 \mu s$ $= 254 \mu s$	$20 \mu s + 3 * 4 \mu s/\text{symbol} + 6 \mu s$ $= 20 + 12 \mu s + 6 \mu s$ $= 38 \mu s$
SIFS	10 μs	10 μs
802.11 ACK	$20 \mu s + 1 * 4 \mu s/\text{symbol} + 6 \mu s$ $= 20 \mu s + 4 \mu s + 6 \mu s$ $= 30 \mu s$	= 30 μs
Frame exchange total	774 μs	558 μs

The total transactional time is even longer because a more robust (and hence time-consuming) protection mechanism is used. A full RTS/CTS exchange makes the TCP ACK require more time than a straightforward 11 Mbps transmission using older 802.11b encoding. Therefore, a transaction would consist of the TCP data protected by RTS/CTS at 774 μs followed by a TCP ACK at the 511 μs

previously calculated for 802.11b. The total transaction time is 1,285 μ s per transaction, so only 778 transactions can complete per second, and the throughput drops back into single digits--9.1 Mbps

Final Thoughts

No matter how you look at it, 802.11g is significantly faster than 802.11b. However, once an 802.11b station associates to an 802.11g network, the throughput drops dramatically, because protection must be activated. The 802.11b station does not need to actively send data to cut the throughput; it just needs to be associated, so that protection is enabled. Mixed 802.11b/g deployments are likely to be common for the foreseeable future, especially in situations where there is no control over client adapters. 802.11a networks can sustain much higher data rates than 802.11g networks with protection enabled and 802.11a offers the added advantage of more radio channels for easier layout of high-density deployments. 802.11g offers a worthwhile speed advantage over 802.11b, but it does not challenge 802.11a for the performance crown.

Matthew Gast is the author of [802.11 Wireless Networks: The Definitive Guide](#), [Network Printing](#), and [T1: A Survival Guide](#).

Return to the [Wireless DevCenter](#)

Are you running a mixed 802.11b and g network? If so, do you have any real world experiences to add?

You must be [logged in](#) to the O'Reilly Network to post a talkback.

 Post Comment

 Main Thread Only

 Old First

Showing messages 1 through 9 of 9.

What is speed negotiation's influence

2006-02-28 13:28:03 Dick9 [\[Reply\]](#) | [\[View\]](#)

I understand that weaker stations initiate raw speed negotiations (down from 54mb) that are then used for all others. What is the influence of this downgrading/upgrading process on throughput of 802.11g?

Wireless G in Mixed Mode

2003-12-11 14:38:12 anonymous2 [\[Reply\]](#) | [\[View\]](#)

I'm currently using a LinkSys "G" series in mixed mode. Looks like I can speed up my wireless throughput by changing to G only since I don't have any B devices.

question about cts-to-self

2003-09-08 20:35:43 anonymous2 [\[Reply\]](#) | [\[View\]](#)

In IEEE802.11g specification, cts-to-self mechanism is described as "when a node wants to distribute NAV information, the node may first transmit a CTS frame with the RA address equal to its own mac address and with a duration that protects the pending transmission".

From this article, I don't understand why the author said "Once the first 802.11b station associates with an 802.11g access point, however, protection is required. The minimal protection contemplated by the standard is that 802.11g stations will protect the fast 802.11g frame exchange with a slow Clear To Send (CTS) frame that locks out other stations' access to the medium".

I don't completely understand cts-to-self mechanism. Would you please explain it to me?
my email: stella.li@netac.com

 **Corrections made/new revision date**

2003-08-15 16:15:40 mgast [[Reply](#) | [View](#)]

The date on the article now reads "revised 8/14/03" because two corrections were made.

The smaller correction is that the analysis did not mention the six OFDM bits used in 802.11a and 802.11g. Six bits are added as a pad at the end of the data for use with the convolution code. As it turns out, adding six bits to the end of any of the frames discussed in this article does not change the number of symbols, so it does not change any of the calculations.

More substantially, I had neglected the 6 microsecond signal extension time in 802.11g. One anonymous talkback comment as well as a private e-mail pointed out my error. The tables and text have been revised to correct this oversight. The effect of the correction is that an 802.11g-only network will have identical throughput to an 802.11a network.

 **Apple's Airport Extreme Basestation**

2003-08-12 03:01:27 anonymous2 [[Reply](#) | [View](#)]

I would assume this to be the case. My reasoning is that my linksys B/G AP has the same option and in one of the early reviews (tomshardware I believe) it was mentioned that transfer was slower in mixed mode than when it was set to G only.

 **Correct?? Pure 802.11g Goodput in this article has some errors...**

2003-08-11 03:56:49 anonymous2 [[Reply](#) | [View](#)]

802.11g(D8.2) pure network (ERP-OFDM case) has 'Signal Extension' (=6 times for convolutional decoding per all OFDM packet (All PHY scheme so called XXX-OFDM in 802.11g spec. has Signal Extension.)). 802.11g's goodput almost equals to 802.11a's goodput.

 **Apple's Airport Extreme Basestation**

2003-08-10 17:14:01 anonymous2 [[Reply](#) | [View](#)]

I understand that Apple's Airport Extreme basestation allows you to specify the mode you want it to run in - 802.11b compatible, mixed 802.11b/g, or 802.11g only. If I were to specify the basestation to run in only 802.11g mode, would this allow it to utilize the full speed of 802.11g because it would disregard any attempt for a 802.11b card to be associated with it?

Hope this makes sense.

 **Apple's Airport Extreme Basestation**

2003-08-12 02:57:47 anonymous2 [[Reply](#) | [View](#)]

I would assume this to be the case. My reasoning is that my linksys E AP has the same option and in one of the early reviews (tomshardware I believe) it was mentioned that transfer was slower in mixed mode than when it was set to G only.

 **Apple's Airport Extreme Basestation**

2003-08-10 23:38:27 anonymous2 [[Reply](#) | [View](#)]

I think you are correct but I don't know for sure.

[Contact Us](#) | [Advertise with Us](#) | [Privacy Policy](#) | [Press Center](#) | [Jobs](#)

Copyright © 2000-2006 O'Reilly Media, Inc. All Rights Reserved.
All trademarks and registered trademarks appearing on the O'Reilly Network are the property of their respective owners.

For problems or assistance with this site, email help@oreillynet.com