In previous chapters, we learned how to find and establish connections to peers through a variety of methods using both Game Center and Game Kit. In this chapter, we will look at how to design a networking experience for not just an iOS game but also a game on any platform. This chapter is designed slightly different than the previous chapters you have encountered in this book. Primarily, there will be no associated source code with this chapter and we will only briefly touch on Game Kit networking topics themselves. This chapter will focus on the concepts of network design, as opposed to actually implementing the network itself. In the next chapter, you will discover how to tie everything together and have your peers begin to communicate with each other.

While it is entirely possible (and often done) to go ahead and just start writing your network logic, it is probably not a great idea. After all, you wouldn’t begin writing a new app or game without first planning out how it will function. Networking is a complex topic and you should approach it with a plan; otherwise, you could find yourself rewriting the entire system after you put a lot of work and effort into it. You don’t want to find yourself up against a wall because the approach you took limited your options for future expandability. Just like with software, you shouldn’t jump right into writing code on the first day. You should whiteboard things out a little and get a feel for the requirements of the project.

Take, for example, a desktop role-playing game called Clan Lord that was written in the late 1990s for the Mac. Clan Lord has maintained a very dedicated fan base that has kept the game active and continual to the present day. However, when the game was originally written, many network-related issues were not properly thought through.

Clan Lord uses frame-by-frame syncing for all of its network calls. This means that every frame, every element visible on the player’s screen, has to be transmitted. This approach works, and works well while you have a small game, a small user base, and limited functionality. However, when you are designing software, you cannot have a limited vision for the future. Always plan for the best, or depending on your perspective, the worst case. When designing a network, you must take into account what you will want to do six months, a year, or even ten years from now with your game or app.
Clan Lord now suffers from long-ingrained problems, such as an eight frames per second rendering engine, due to the fact that you cannot sync more than eight full frames of data per second on the average home network. This could have been prevented by implementing some logic into the client when the project was first started; for example, it would have been much more efficient to inform the client where objects are and when they move, as opposed to fully syncing everything each frame. In addition, player movement is limited to eight frames per second because actions have to be synced back to the server, making it hard to react to events. This also could have been prevented by using prediction algorithms, discussed later in this chapter, to determine where a player will end up during movement.

Clan Lord is one example of a game that was much more popular than planned, and lived a lot longer than anyone expected. Sadly, when this happens, you are limited to the vision and design that you had when the project was first started. It is much harder to undo something later than to do it initially. When designing your network, take time to do it carefully and intentionally, as it can follow you around for a very long time.

### Three Types of Networks

Although there are many different types of network designs available, there are three primary kinds of networks that you can implement when designing your network. Picking a primary type of network is a good place to start, as it will guide you toward the next step in the design process.

We will be focusing on just three of the primary types of network designs, but keep in mind that there are dozens of other well known network configurations, some of which we will briefly touch on in this section. The three types of networks we will be discussing in length throughout this chapter are peer-to-peer, client-to-host, and ring networking.

### Peer-to-Peer Network

A peer-to-peer network (see Figure 7–1) is the most common network that you will see on the iOS platform. No device is treated any differently than any other device, and each device is in charge of sending and receiving data to all the other peers it wishes to communicate with.