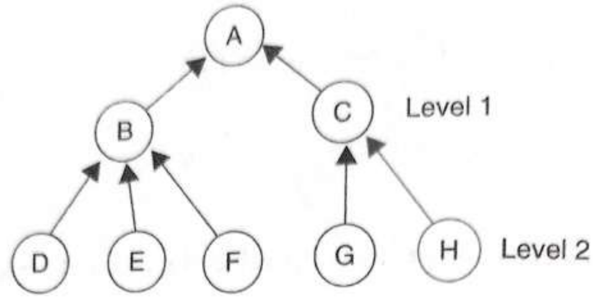

FIGURE 6.7 A network of time-servers used in NTP. The top level Server A (level 0) directly receives the UTC signals.

Network time protocol. Network Time Protocol (NTP), is an elaborate external synchronization mechanism designed to synchronize clocks on the Internet with the UTC despite occasional loss of connectivity, and failure of some of the time-servers, and possibly malicious timing inputs from untrusted sources. These time-servers are located at different sites on the Internet. NTP architecture is a tiered structure of clocks, whose accuracy decreases as its level number increases. The primary time-server that receives UTC from a dedicated transmitter or the GPS satellite belongs to level 0; a computer that is directly linked to the level 0 clock belongs to level 1; a computer that receives its time from level **i** belongs level (**i** + **1**), and so on. Figure 6.7 shows the hierarchy with a single level 0 node.

A site can receive UTC using several different methods, including radio and satellite systems. Like the GPS system in the United States, a few other nations have their own systems of signaling accurate time. However, it is not practical to equip every computer with satellite receivers, since these signals are not received inside buildings. Cost is another factor. Instead, computers designated as primary time-servers are equipped with such receivers and they use the NTP to synchronize the clocks of networked computers.

In a sense, NTP is a refinement of Cristian's method. NTP provides time service using the following three mechanisms:

Multicasting. The time-server periodically multicasts the correct time to the client machines. This is the simplest method, and perhaps the least accurate. The readings are not compensated for signal delays.

Procedure call. The client processes send requests to the time-server, and the server responds by providing the current time. Using Cristian's method, each client can compensate for the propagation delay by using an estimate of the round-trip delay. The resulting accuracy is better than that obtained using multicasting.

Peer-to-peer communication. Network time protocol allows for several time-servers to communicate with one another to keep better track of the real time, and thus provide a more accurate time service to the client processes. This has the highest accuracy compared to the previous two methods.

Consider the exchange of a pair of messages between two time-servers as shown in Figure 6.8. Define the offset between two servers **P** and **Q** as the difference between their clock values. For each message, the sending time is stamped on the body of the message, and the receiver records its own time when the message was received. Let $\mathbf{T_{PQ}}$ and $\mathbf{T_{QP}}$ be the message propagation delays from **P** to **Q** and **Q** to **P**, respectively. If server **Q**'s time is ahead of server **P**'s time by an offset δ

$$\mathbf{T_2 = T_1 + T_{PQ}} + \delta \tag{6.1}$$

$$\mathbf{T_4 = T_3 + T_{QP}} - \delta \tag{6.2}$$

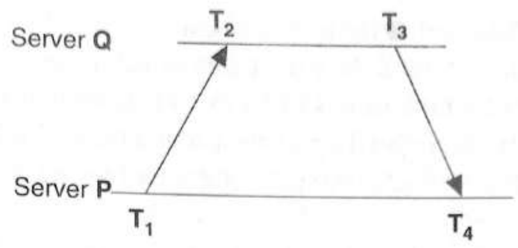

FIGURE 6.8 The exchange of messages between two time-servers.

Adding Equation 6.1 and Equation 6.2,

$$T_2 + T_4 = T_1 + T_3 + (T_{PQ} + T_{QP})$$

The round-trip delay

$$y = T_{PQ} + T_{QP} = T_2 - T_1 + T_4 - T_3$$

Subtracting Equation 6.2 from Equation 6.1,

$$2\delta = (T_2 - T_4 - T_1 + T_3) - (T_{PQ} - T_{QP})$$

So, the offset

$$\delta = (T_2 - T_4 - T_1 + T_3)/2 - (T_{PQ} - T_{QP})/2$$

where $T_{PQ} = T_{QP}$, δ can be computed from the above expression. Otherwise, let $x = (T_2 - T_4 - T_1 + T_3)/2$. Since both T_{PQ} and T_{QP} are greater than zero, the value of $(T_{PQ} - T_{QP})$ must lie between $+y$ and $-y$. Therefore, the actual offset δ must lie between $x + y/2$ and $x - y/2$. Therefore, if each server bounces messages back and forth with another server and computes several pairs of (x, y), then a good approximation of the real offset can be obtained from that pair in which y is the smallest, since that will minimize the dispersion in the window $[x + y/2, x - y/2]$.

The multicast mode of communication is considered adequate for most applications. When the accuracy from the multicast mode is considered inadequate, the procedure call mode is used. An example is a file server on a LAN that wants to keep track of when a file was created (by communicating with a time-server at a higher level, i.e., a lower level number). Finally, the peer-to-peer mode is used only with the higher-level time-servers (level 1) for achieving the best possible accuracy. The synchronization subnet reconfigures itself when some servers fail, or become unreachable. NTP can synchronize clocks within an accuracy of 1 to 50 msec.

6.5 CONCLUDING REMARKS

In asynchronous distributed systems, the absolute physical time is not significant, but the temporal order of events is important for some applications. In replicated servers, each server can be viewed as a state machine whose state is modified by receiving and handling the inputs from the clients. In order that all replicas always remain in the same state (so that one can seamlessly switch to a different server if one crashes) all replicas must receive the inputs from clients in the same order.

The performance of a clock synchronization algorithm is determined by how close two distinct clock times can be brought, the time of convergence, and the nature of failures tolerated by such algorithms. The adjustment of clock values may have interesting side effects. For example, if a clock is advanced from **171 to 174** during an adjustment, then the time instants **172** and **173** are lost. This