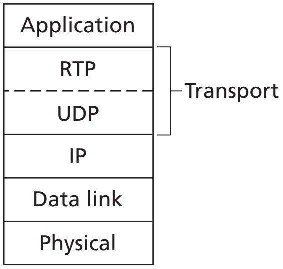

Figure 7.12 ♦ RTP can be viewed as a sublayer of the transport layer.

An alternative approach (not done in the programming assignment) is to use a Java RTP class (or a C RTP library for C programmers) to implement the RTP operations. With this approach, as shown in Figure 7.12, the application developer is given the impression that RTP is part of the transport layer, with an RTP/UDP API between the application layer and the transport layer. Without getting into the nitty-gritty details (as they are class/library-dependent), when sending a chunk of media into the API, the sending side of the application needs to provide the interface with the media chunk itself, a payload-type number, an SSRC, and a timestamp, along with a destination port number and an IP destination address. We mention here that the Java Media Framework (JMF) includes a complete RTP implementation.

7.4.2 RTP Control Protocol (RTCP)

RFC 3550 also specifies RTCP, a protocol that a networked multimedia application can use in conjunction with RTP. As shown in the multicast scenario in Figure 7.13, RTCP packets are transmitted by each participant in an RTP session to all other participants in the session using IP multicast. For an RTP session, typically there is a single multicast address and all RTP and RTCP packets belonging to the session use the multicast address. RTP and RTCP packets are distinguished from each other through the use of distinct port numbers. (The RTCP port number is set to be equal to the RTP port number plus one.)

RTCP packets do not encapsulate chunks of audio or video. Instead, RTCP packets are sent periodically and contain sender and/or receiver reports that announce statistics that can be useful to the application. These statistics include number of packets sent, number of packets lost, and interarrival jitter. The RTP specification [RFC 3550] does not dictate what the application should do with this feedback information; this is up to the application developer. Senders can use the feedback information, for example, to modify their transmission rates. The feedback

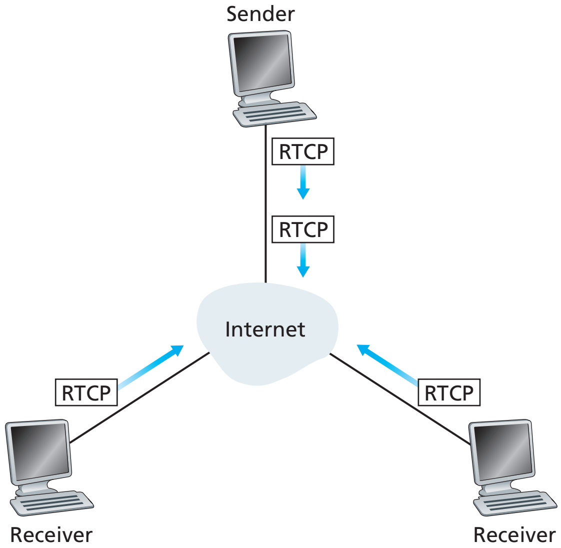

Figure 7.13 ♦ Both senders and receivers send RTCP messages.

information can also be used for diagnostic purposes; for example, receivers can determine whether problems are local, regional, or global.

RTCP Packet Types

For each RTP stream that a receiver receives as part of a session, the receiver generates a reception report. The receiver aggregates its reception reports into a single RTCP packet. The packet is then sent into the multicast tree that connects all the session's participants. The reception report includes several fields, the most important of which are listed below.

- The SSRC of the RTP stream for which the reception report is being generated.
- The fraction of packets lost within the RTP stream. Each receiver calculates the number of RTP packets lost divided by the number of RTP packets sent as part of the stream. If a sender receives reception reports indicating that the receivers are receiving only a small fraction of the sender's transmitted packets, it can switch to a lower encoding rate, with the aim of decreasing network congestion and improving the reception rate.
- The last sequence number received in the stream of RTP packets.
- The interarrival jitter, which is a smoothed estimate of the variation in the interarrival time between successive packets in the RTP stream.

For each RTP stream that a sender is transmitting, the sender creates and transmits RTCP sender report packets. These packets include information about the RTP stream, including:

- The SSRC of the RTP stream
- The timestamp and wall clock time of the most recently generated RTP packet in the stream
- The number of packets sent in the stream
- The number of bytes sent in the stream

Sender reports can be used to synchronize different media streams within an RTP session. For example, consider a video conferencing application for which each sender generates two independent RTP streams, one for video and one for audio. The timestamps in these RTP packets are tied to the video and audio sampling clocks, and are not tied to the *wall clock time* (i.e., real time). Each RTCP sender report contains, for the most recently generated packet in the associated RTP stream, the timestamp of the RTP packet and the wall clock time when the packet was created. Thus the RTCP sender report packets associate the sampling clock with the real-time clock. Receivers can use this association in RTCP sender reports to synchronize the playout of audio and video.

For each RTP stream that a sender is transmitting, the sender also creates and transmits source description packets. These packets contain information about the source, such as the e-mail address of the sender, the sender's name, and the application that generates the RTP stream. It also includes the SSRC of the associated RTP stream. These packets provide a mapping between the source identifier (that is, the SSRC) and the user/host name.

RTCP packets are stackable; that is, receiver reception reports, sender reports, and source descriptors can be concatenated into a single packet. The resulting packet is then encapsulated into a UDP segment and forwarded into the multicast tree.

RTCP Bandwidth Scaling

You may have observed that RTCP has a potential scaling problem. Consider, for example, an RTP session that consists of one sender and a large number of receivers. If each of the receivers periodically generates RTCP packets, then the aggregate transmission rate of RTCP packets can greatly exceed the rate of RTP packets sent by the sender. Observe that the amount of RTP traffic sent into the multicast tree does not change as the number of receivers increases, whereas the amount of RTCP traffic grows linearly with the number of receivers. To solve this scaling problem, RTCP modifies the rate at which a participant sends RTCP packets into the multicast tree as a function of the number of participants in the session. Also, since each participant sends control packets to everyone else, each participant can estimate the total number of participants in the session [Friedman 1999].

RTCP attempts to limit its traffic to 5 percent of the session bandwidth. For example, suppose there is one sender, which is sending video at a rate of 2 Mbps. Then RTCP attempts to limit its traffic to 5 percent of 2 Mbps, or 100 kbps, as follows. The protocol gives 75 percent of this rate, or 75 kbps, to the receivers; it gives the remaining 25 percent of the rate, or 25 kbps, to the sender. The 75 kbps devoted to the receivers is equally shared among the receivers. Thus, if there are R receivers, then each receiver gets to send RTCP traffic at a rate of 75/R kbps, and the sender gets to send RTCP traffic at a rate of 25 kbps. A participant (a sender or receiver) determines the RTCP packet transmission period by dynamically calculating the average RTCP packet size (across the entire session) and dividing the average RTCP packet size by its allocated rate. In summary, the period for transmitting RTCP packets for a sender is

$$T = \frac{\text{number of senders}}{.25 \cdot .05 \cdot \text{session bandwidth}}(\text{avg. RTCP packet size})$$

And the period for transmitting RTCP packets for a receiver is

$$T = \frac{\text{number of receivers}}{.75 \cdot .05 \cdot \text{session bandwidth}}(\text{avg. RTCP packet size})$$

7.4.3 SIP

Imagine a world in which, when you are working on your PC, your phone calls arrive over the Internet to your PC. When you get up and start walking around, your new phone calls are automatically routed to your PDA. And when you are driving in your car, your new phone calls are automatically routed to some Internet appliance in your car. In this same world, while participating in a conference call, you can access an address book to call and invite other participants into the conference. The other participants may be at their PCs, or walking with their PDAs, or driving their cars—no matter where they are, your invitation is transparently routed to them. In this same world, when you browse an individual’s homepage, there will be a link “Call Me”; clicking on this link establishes an Internet phone session between your PC and the owner of the homepage (wherever that person might be).

In this world, there is no longer a circuit-switched telephone network. Instead, all calls pass over the Internet—from end to end. In this same world, companies no longer use private branch exchanges (PBXs), that is, local circuit switches for handling intracompany telephone calls. Instead, the intracompany phone traffic flows over the company’s high-speed LAN.

All of this may sound like science fiction. And, of course, today’s circuit-switched networks and PBXs are not going to disappear completely in the near future [Jiang 2001]. Nevertheless, protocols and products exist to turn this vision into a reality. Among the most promising protocols in this direction is the Session