Proxies. Proxies have been used by many applications to perform various tasks, such as filtering data and connections (e.g., security firewalls) and modifying control data (e.g., network address translators [NATs] change the IP fields). Of particular interest to data adaptation are transcoding proxies. Transcoding is the process of converting data objects from one representation to another. Transcoding proxies can be used to adapt to various situations dynamically, such as the availability of bandwidth and capabilities of the end device. For example, if the end device is not capable of handling full-motion video, a transcoding proxy may convert it to a form that can be displayed on the end device (Han et al., 1998).

Conceptually, a transcoding proxy may be viewed as consisting of three modules: (1) an adaptation-policy module (2) a data (content) analysis module, and (3) a content-transformation module. Figure 1.4 shows the architecture of the transcoding proxy developed at the IBM T. J. Watson Research Center (Han et al., 1998). The adaptation-policy module can take as input information such as the server-to-proxy bandwidth \((B_{sp})\), proxy-to-client bandwidth \((B_{pc})\), client device capabilities (e.g., video display capabilities), user preferences, and content characteristics (provided by content-analysis module). Based on these inputs, it can decide on whether and how to modify the content. The content-transformation module performs the actual modification.

Figure 1.5 presents an example of logic that can be implemented in a transcoding policy module. The transcoding threshold is the document size beyond which transcoding becomes beneficial. To get an understanding of this parameter, consider the following simple cost-benefit analysis. Assume that the proxy uses the store-and-forward mechanism for delivering documents to the client. This is to say, the request from the client first is submitted to the proxy, the proxy then obtains the entire document from the server, and finally, it forwards the document to the client. Further, assume that the goal is to minimize the latency of document retrieval. Following the analysis in Han et al. (1998), the total delay for document retrieval without the proxy \((D_{sc})\) and with the proxy \((D_{spc})\) is

\[
D_{sc} = 2 \times RTT_{pc} + 2 \times RTT_{sp} + S/min(B_{pc}, B_{sp})
\]

\[
D_{spc} = 2 \times RTT_{pc} + 2 \times RTT_{sp} + D_p(S) + S/B_{sp} + S_p(S)/B_{pc}
\]

where \(RTT_{pc}\) is the round-trip delay between the proxy and the client, \(RTT_{sp}\) is the round-trip delay between the server and the proxy, \(S\) is the document size, \(D_p(S)\) is a proxy delay function that relates the proxy processing delay to the document size, and \(S_p(S)\) is an output size function that relates the transcoded document size to the input document size.
Figure 1.4 Dynamic adaptation in IBM's transcoding proxy (Han, Bhagwat, and LaMaire, 1998).
Switch(Client_Type)
    Case Laptop/PC:
        If (Input_Size > TranscodingThreshold)
            Switch(Input_type)
                Case GIF:
                    TranscodedImage = T(Input, GIF, GIF, user preference)
                Else
                TranscodedImage = T(Input, GIF, JPEG, user preference)
            End If
            Case JPEG:
                TranscodedImage = T(Input, JPEG, JPEG, user preference)
            End Switch
        End If
    End Case
Else
        Send Input Image to Client
End If
If (Output size > input Size)
    Send Input Image to Client
Else
    Send TranscodedImage to Client
End If
Case Palm PDA:
    TranscodedImage = T(Input, Any, 2-bit_grayscale, user preferences);
    Send TranscodedImage to Client
End Switch

Figure 1.5 Example of a transcoding policy module (The example is based on the code fragment of a policy module in [Han transcoding proxy 1998]). T (input object, input object type, output object type, user preferences) performs data transformation on the <input object> of type <input object type> into <output object> of type <output object type>.

Obviously, using transcoding is better when $D_{spc} < D_{sc}$. From this, a lower bound (transcoding threshold) on the input document size can be obtained to be

$$S > [D_p (S) + S_p (S)/B_{pc}]/(1/B_{pc} - 1/B_{sp})$$

(1.1)

Note that the server-to-proxy bandwidth and the proxy-to-client bandwidth correspond with the bottleneck bandwidth along the data path from the server to the proxy and the proxy to the client, respectively. These bandwidths can vary dramatically as the user moves and with fluctuations in the available wireless bandwidth. Equation (1.1) can be used to adapt dynamically to changes in available bandwidth. However, for this to work, the transcoding proxy would need a good bandwidth estimator of the bandwidth that might be available in the near future.

Transcoding proxies also can work in the streamed mode. In this mode, the data stream is modified and passed on to the client as it is obtained from the server. For an analysis of when adaptive-streamed transcoding is beneficial and other details about transcoding proxies, see Han et al. (1998).