Content Delivery Network

The customer is a leading Japanese optical/semiconductor based communication company.

Business Objective

Many companies are struggling hard to support growing volumes of streaming media on their intranet and extranet infrastructures. Bandwidth shortages, unregulated and unexpected growth in demand, service degradation, overloads, long response times etc is affecting their business. Content Delivery Networks (CDNs) prevent these problems by managing & delivering streaming-media content over the network in the most effective and efficient manner.

The objective of the project was to deliver multimedia content – data, audio and video over (non real-time) Internet, which is an unreliable network. Delivery of Media Content through an error prone channel is always a bottleneck in Multimedia Streaming Applications. Our state of the art solutions for Content Delivery Network is overcoming this bottleneck by employing Forward Error Correction (FEC) techniques in transmission and reception. CDN systems are preferred in an environment where the number of clients connected to and subscribed to a single program can range from hundreds to thousands. Multicasting is the backbone of this emerging technology.

System Overview – *Challenges and Solution*

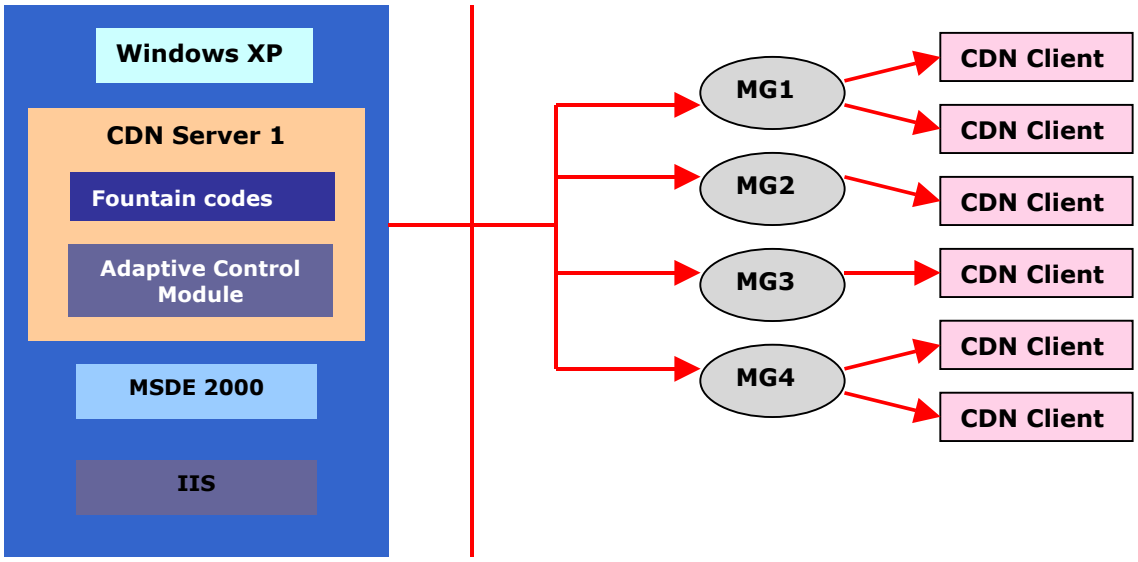
We can choose Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) protocol for this application. The primary difference between UDP and TCP lies in their respective implementations of reliable messaging. TCP includes support for guaranteed delivery, as the recipient automatically acknowledges the sender when a message is received, and the sender waits and retries in cases where the receiver does not respond in a timely way. UDP, on the other hand is connectionless protocol and unreliable, which does not establish a virtual circuit like TCP, nor does it demand an acknowledge. A UDP datagram can get "lost" on the way from sender to receiver, and the protocol itself does nothing to detect or report this condition. UDP is sometimes called an unreliable transport for this reason. Another way in which UDP works unreliably is in the receipt of a burst of multiple datagrams. Unlike TCP, UDP provides no guarantees that the order of delivery is preserved. For example, a client application might send the following four datagrams to a server

D1
D22
D333
D4444

but UDP may present the datagrams to the server-side application in this order instead:

D333
D1
D4444
D22

In practice, UDP datagrams arrive out-of-order relatively infrequently -- generally only under heavy traffic conditions. On the surface, an "unreliable" network protocol may not seem very worthwhile or desirable. But we chose UDP because, it can be very useful in certain situations, and it enjoys one key advantage over TCP -- speed. The reliability features built into TCP can be expensive in terms of overhead at execution time. Also, UDP does not preclude reliable message delivery; it merely defers those details to a higher level of the network stack. It is simpler, faster and cheaper than TCP. UDP is ideal for applications where we are willing to sacrifice some reliability in return for performance.



But for CDN we need reliability and cannot afford to have data loss; the only compromise we can make here is on the realtime aspect. So CDN requires reliable and fast content delivery in non-realtime. But UDP being a connectionless protocol tends to create packet losses depending on the operating conditions. Heterogeneous networks are more prone to packet losses. A combination of congestion control, selective retransmission and forward error correction (FEC) techniques make our solution unique.

Multicast is based on the concept of a group (Virtual Group). An arbitrary group of receivers expresses an interest in receiving a particular data stream. This group does not have any physical or geographical boundaries—the hosts can be located anywhere on the network. The broadcasting company or the Server would make available the file for a specific duration. Hosts that are interested in receiving data flowing to a particular group must join the group. Hosts must be a member of the group to receive the data stream.

Delivering multimedia content to various clients using multicast can load the server and network heavily. If too many clients were joining one multicast group, then that server would be loaded. To balance the load another server also starts this multicast group and the new clients would be switched to this server. This is how the system achieves load balancing using server balancing. Another load balancing technique that can be used Network balancing using a prediction method. This uses a OLAP server, which comprises of a SQL server and an Online Analytical Processing System, which captures the network history, processes it and use the knowledge gained for changing the network bandwidth.

Content Delivery Network

Features & Benefits

1. Recovery of lost packets employing FEC and Negative ACKnowledgement (NACK) based retransmission. Even from a 50% packet loss operating condition recovery is possible.
2. Controlled sending rates and Packet sizes. Sending rates can be varied from 28 kbps to 10mbps.
3. Congestion control system based on information obtained from NACK packets. Round Trip Time and Packet loss are extracted from NACK packets and used for controlling the congestion
4. Caters to 4 Multicast Groups simultaneously. Each group can serve thousands of clients.
5. Creation of Content List at the server side makes it easy to locate the content. The server centrally controls entire manipulation of media files.

Development platform & Tools

Windows XP
Microsoft .NET

Tools:
Visual Studio .NET / VC++

Conclusion

Content Delivery Networks help maximize your Internet performance while relieving congestion on the original server. Content Delivery is for any service provider with heavy concentrations of content, high volumes of downloaded content, or that streams audio and/or video say for eLearning, online demonstrations, media streaming etc

Content Delivery Networks have the power to raise performance and guarantee the availability of content, while at the same time potentially increasing your revenue.