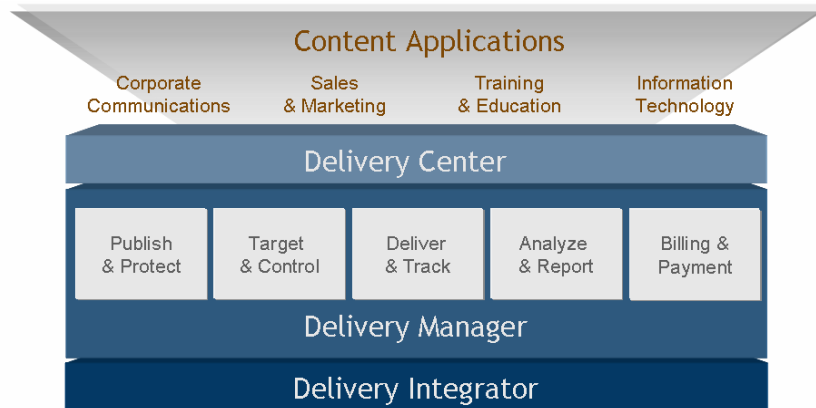


Ignite provides the industry's most secure and scalable Content Delivery Solution, enabling customers to efficiently publish, deliver, and manage digital assets – from rich media content for training and communications to software patches and virus updates – to anyone, anywhere, at any time. Ignite's patented Content Delivery Solution overcomes network and connectivity constraints that have limited the ability to reach online audiences with the highest quality, secure rich media. Ignite's Solution has been deployed around the globe at companies like Accenture, BearingPoint, Sabre, and Procter & Gamble.



The Managed P2P Difference

Peer-to-peer (P2P) networks have long been an effective way to deliver content because of their ability to leverage individual user bandwidth. P2P networks enable content owners to drastically reduce content delivery costs – and those cost savings grow exponentially when delivering long-form video content. However, P2P networks have developed a negative reputation over the past few years due to their use by consumer-oriented file sharing services for downloading content such as music and movies. Content distributed via these services is often infected with malware and viruses, and the services' lack of efficient traffic management practices often overburdens Internet capacity and individual computer resources. By contrast, Ignite's managed P2P technology delivers all the benefits of P2P networks while preserving the rights of content owners. Ignite allows administrators to balance the rate of P2P network usage with the need for secure, reliable content delivery, including high-quality live streaming support.

How It Works

A very common way for today's Intranet LAN administrators to conserve bandwidth on their Internet connection or on connections to other LANs is to install proxy servers that cache frequently-accessed data. This usually carries a high cost and entails setting up dedicated, over-provisioned systems, purchasing, configuring, and maintaining proxy server software. Furthermore, each computer or workstation on the LAN must then be reconfigured in order to "know about" and utilize the proxies.

As far as Ignite traffic is concerned, our managed P2P technology is specifically designed to allow Intranets to reap the benefits of using a proxy server philosophy, but without the need to provision dedicated equipment, and without the need for the LAN administrator or corporate users to perform any installation or reconfiguration. This is accomplished by enabling Ignite Delivery Centers to share files amongst themselves across their local network neighborhood, thereby eliminating the need for a specialized proxy server. Furthermore, the network traffic is not significantly affected since modern switched network architectures are well suited for this type of relay communications.

Terminology

Delivery Center refers to the Ignite Delivery Center application, running on a device (mobile phone, desktop, notebook, etc.).

Server refers to an Ignite Content Delivery Server.

Dedicated Edge Relay (DER) refers to an appliance or application that can be installed on existing end-user computers or file servers. DERs are similar to CDNs, with a few important differences. In smaller locations that cannot justify investing in dedicated caching devices or appliances, or locations that need to support a diverse network topology that may not be permanent, Ignite's DER application can be used to provide localized caching capabilities *exclusively for Ignite traffic*.

Dynamic Edge Relay refers to an Ignite Delivery Center that is acting as a dynamic redistribution point (i.e., to transfer files to other Delivery Centers).

Package refers to the package of one or more files that collectively constitutes one content delivery.

In the managed P2P scheme, every Ignite Delivery Center actually functions as a caching redistribution point. Once an Ignite Delivery Center polls and determines (is told) that it requires a delivery, it queries the other Ignite Delivery Centers on its local network for that content. If no adjacent Delivery Center has the file, the Delivery Center will attempt to retrieve the content from network local caching/proxy servers or dedicated Ignite edge relay servers, and as a last resort, from the Ignite Server. However, if an adjacent Ignite Delivery Center or dedicated edge relay server already has the required content, the requesting Delivery Center will download this file from that Delivery Center rather than from the Ignite Server.

Implementation

Communication among Ignite Delivery Centers is performed using two basic protocols: a file-exchange protocol and a control protocol. The first is used to securely transfer files between Delivery Centers, and is based on derivative edge relay communications using the HTTP protocol. The latter is issued by Ignite Delivery Centers to listen to and query their neighboring Delivery Centers and to respond to the inquiries. The control protocol enables each Delivery Center to build and maintain a common and consistent “picture” of content that is available across their local network in an efficient way, by exchanging multicast/broadcast messages.

Each Ignite Delivery Center maintains two hash tables that contain information about available content location:

Local-files table – A hash-table of content that resides on that local system.

Network-files table – A hash-table of content that resides on other systems on the local network. For each file, this table holds the address of the machine that possesses it. This hash-table size is limited and dynamically remotely configurable by the administrator so as not to consume too much system memory. Old entries are purged when the size of the table reaches its upper limit.

Each file is identified by its unique fingerprint, which is an MD53 digest of the file’s content.

Control Protocol

The control protocol’s primary goal is to provide each Ignite Delivery Center with knowledge about locations of content across its local network. Control messages are sent and received as broadcast or multicast packets. Managed P2P will use either broadcast or IP multicast according to the configuration set by the network manager. IP multicast is preferable in terms of load on the network’s computers, but it is not supported on all platforms.

Control Protocol Algorithm Description

Step 1 – Delivery Center A determines that it needs to download content, and multicasts (or broadcasts) a request message. A request message contains a protocol identifying version number (PVN) and a list of MD5 digests of the needed file(s).

As an optimization technique, a list of needed files is used rather than a single MD5 digest. This reduces the total number of multicasts on the network, since a package often contains more than one file within it.

Step 2 – All the neighboring Ignite Delivery Centers listen for and receive this request message, then search for the requested file in their local-files hash table. A Delivery Center that does not find the file locally does not reply. Otherwise, it waits a short random time interval and then multicasts a response message, unless another Delivery Center has already responded. A response message contains the identifying PVN, the list of MD5 digests of files that were found and a TCP port number. The port number identifies on which TCP port the responding Delivery Center is waiting for edge redistribution requests.

The Ultimate Enterprise Live Streaming Solution

Ignite’s managed P2P technology gives enterprises the ability to implement a high-quality live streaming solution that is scalable and network-friendly. Because Ignite leverages centrally-controlled and managed P2P relationships to pull content from each other or from designated localized content proxy/caching servers, audiences can receive high-quality streams without adversely affecting enterprise networks. In addition, this technology provides Ignite customers with unparalleled reach and efficiency, enabling audiences in low-bandwidth locations to participate in live streaming events.

Ignite’s live streaming solution not only increases audience participation but also provides stream publishers with extensive real-time and historical data regarding this audience involvement.

For further details, please refer to the *Ignite Enterprise Live Streaming Overview*.

Note: Response messages will also be broadcast for files that are currently in download from the server.

Step 3 – If a response was broadcast, other Delivery Centers with pending responses for the same content will not broadcast a second response. This is done to prevent broadcast storms or network chatter. Therefore, only a single response is likely to be broadcast for each request, since the first Delivery Center to finish the random wait (described in step 2) will be the one to respond, and the others will detect that it has responded and therefore will not respond.

Responses are sent only for content with as yet unknown locations. For example, Delivery Center A requested files W, X, Y and Z. Delivery Center B has files W, X and Y, and was the first to respond. Delivery Center B's reply message indicated it has files W, X and Y. Another Delivery Center, C has files X, Y and Z. Since it replied after Delivery Center B, it will only advertise that it has file Z because this is the only file with yet unknown location.

Step 4 – In principle, at this stage the requesting Delivery Center either receives a reply and receives the files from the Delivery Center that replied, or if a reply is not received within a certain period of time, it proceeds to download these files from a network caching/proxy server, or an Ignite dedicated edge relay, and as a last attempt, directly from Ignite's content server. In reality however, the implementation of this step is not that simple. A naïve implementation such as this would cause multiple Delivery Centers to transfer the same files from the Ignite server simultaneously, because when several Delivery Centers need to download the same files at approximately the same time, none of them will get a response to its request.

Since other Delivery Centers that require the same content may still be downloading the files when the new Delivery Center broadcast its request, none of them will be completely ready to edge relay those files. In order to solve this problem, the first Delivery Center that needs some content files will download them from the server. Other Delivery Centers that need them will download them from the first Delivery Center, even if the first Delivery Center is still in the process of downloading them (i.e. when a group of Delivery Centers need the same files at approximately the same time, a single Delivery Center will download the files from the Ignite server, and all the rest will download those files from that Delivery Center or from intermediary Delivery Centers as described in the previous steps). This behavior is achieved using the following algorithm:

1. The requesting Delivery Center transmits the request and waits for a response. If it does not receive any response to the message it sent within a certain time period, the Ignite Delivery Center will multicast a response message (as if it were replying to its own request) indicating it is handling these files and proceed to acquire them from the Ignite Server.
2. Other Ignite Delivery Centers awaiting timeout for their own requests will detect this broadcast or multicast and not multicast a response for themselves. In addition, they will create an entry in their network files hash table, indicating which Ignite Delivery Center will be serving those content files. This way only a single Delivery Center will access the server for any given list of files. The other Delivery Centers will retrieve the files from the first Delivery Center using the file exchange protocol. If the file transfer is refused because the responding Delivery Center is too busy (see File Exchange section below) the connecting Delivery Center will attempt to re-establish the connection. If the responding Delivery Center is still busy, the algorithm described in this section will be repeated a number of times before the Delivery Center will give up and retrieve the files directly from the server or another Ignite Delivery Center.

Administrative Control of Discovery and Transfers

Network administrators are provided a great deal of flexibility and control with respect to managed P2P protocol settings. Among the configuration options are the ability to modify what protocols (broadcast or multicast) are used for neighbor discovery. The administrator can also configure what specific UDP port is used for that discovery packet. Discovery packets are designed to be as small as possible (typically less than 200 bytes) and will always fit within a typical network MTU packet size. The broadcast or multicast packets are used only for discovery, not for neighbor file transfers and are very lightweight. The discovery protocol also protects against network broadcast storms. This is accomplished through response pre-emption. When one computer responds to a broadcast it responds with another broadcast. Since other computers were able to "hear" the response, they will not respond.

The file transfers between neighbors are unicast, and therefore private in nature. Administrators can control what port is being used by end users' computers for this transfer, as well as employ hard bandwidth and concurrent connection limits for these peer transfers.

Content Download Optimization

Managed P2P attempts to minimize total download time (the time that passes until all the Delivery Centers who need a certain file finish receiving it) while minimizing the load on each Delivery Center as much as possible. This is accomplished via the following methods:

- The exit degree of each Delivery Center is bound. This means that by default each Ignite Delivery Center may only dynamically edge relay to three other Delivery Centers simultaneously. This number is centrally, dynamically configurable by the network administrator. Additional Delivery Centers attempting to download a file from the same Delivery Center will receive a "busy" response message. This feature reduces the overall load on each individual Delivery Center.
- For each needed file, the algorithm attempts to find the Delivery Center best suited to provide the download. For example, if Delivery Center A already downloaded a larger portion than Delivery Center B from a certain file, then Delivery Center A is surely more eligible at that moment to further relay that file. Such Delivery Centers will be given precedence, in turn meaning that they should wait less time before responding, thereby increasing their chance to become the ones to dynamically edge relay the files. For this reason, the Ignite Delivery Center calculates the random delay so it is:
 - Proportional to the percentage of the content files downloaded so far.
 - Inversely proportional to the number of Delivery Centers it is serving at that moment.

This also helps to minimize download time and Ignite Delivery Center load.

- When different Delivery Centers respond (usually at different times) indicating they have a specific file, the rest of the Delivery Centers listening across the network will update the IP address they maintain for this file in their network files table. In a naïve implementation this would always cause all the Delivery Centers to "remember" only the last location of the file that was advertised, and therefore they all may end up going to that address, overloading it. For this reason, the updating of the address is done in a probabilistic manner:
 - New IP address = Where generation is a number indicating how many times this address had been updated.

Assuming Ignite Delivery Center A responds indicating it has content file X, then initially all the neighboring Delivery Centers will mark A as the location of X. If next Delivery Center B broadcasts a response with file X, then each Delivery Center will now change its perception of the location of X in probability .5. This means half the Delivery Centers will now point to A and half will point to B. This way a good load distribution may be achieved. The data flow is likely to resemble a tree-shaped topology rather than a random one, thereby optimizing the average download time as well as the load on the dynamic edge relay Delivery Centers:

- In order to further reduce the load on each Delivery Center, responses indicating Delivery Center A is still downloading a file or is busy are retransmitted as broadcast/multicast messages. This way for each file, only a single Delivery Center B polls Delivery Center A. Other Delivery Centers will listen for and receive the outcome of that polling action through the broadcast/multicast and therefore will not be forced to generate a request for themselves.

The broadcast/multicast traffic is optimized since there is usually no need to broadcast a request (and then a response) for each file needed by each Delivery Center. This is because each Delivery Center listens to all the communication of all the other Delivery Centers and "remembers" the recently seen files in its network-files table.

File Exchange Protocol

This File Exchange Protocol's primary goal is to enable the edge relaying of content from Delivery Center to Delivery Center. The file transfer is accomplished using the standards-based HTTP protocol.

When Delivery Center A requires a specific content file, it connects to a Delivery Center, B, that previously advertised that it has it, using the TCP port advertised by Delivery Center B (see the Control Protocol section above). Delivery Center A sends the requested file's MD5 digest. It is assumed that a file's MD5 digest uniquely identifies it.

Delivery Center B, acting as an HTTP-based dynamic edge relay server, will look for the requested file in its local files hash-table. If it was not found – it will reply "File not found", otherwise it will transmit to Delivery Center A the requested file. Delivery Center B can perform several edge relay transfers simultaneously if needed, within the dynamically configurable limit that exists on the number of simultaneously served Delivery Centers. This limit is set in order to prevent a serving dynamic edge relay Delivery Center from becoming overloaded by serving files to its neighbors. By default this limit is set to 3. If this limit is exceeded, the serving Delivery Center will reply "Server is busy" to the Delivery Center that sends a request. This will cause the requesting Delivery Center to re-multicast a request for the content files that it needs as described in the control protocol section above.

The content file transfer contains mechanisms such as timeout timers, for detection of a disconnected or a very slow file transfer. When the receiving Delivery Center detects that no new data was received for some time, it reconnects to the sending dynamic edge relay Delivery Center (using a new HTTP connection) and queries it as to its current state. If there was a timeout or the serving Delivery Center was terminated and restarted, it will send an indication of that to the inquirer. Obviously, if a connection cannot be made, the inquirer assumes the serving Delivery Center has disconnected, and will restart the process from the beginning (starting with a request multicast).

If a requested file has not yet finished its download, the requesting Delivery Center will receive a message indicating the file is not ready and an indication of the amount of the file downloaded so far. The requesting Delivery Center will continue polling the dynamic edge relay Delivery Center until the file download is complete. In the event a long time period passes and no progress is made on the download, the requesting Delivery Center aborts the wait cycle and restarts the entire procedure. If a serving dynamic edge relay Delivery Center is too busy with requests it will return a "busy" message. The requesting Delivery Centers will reattempt acquisition of the content file, and then, if it still receives a "busy" will restart the process from the beginning.

Additional Features

Auto-detection

When each Ignite Delivery Center is first started, it attempts to detect other neighboring Delivery Centers supporting managed P2P. If none are found, managed P2P is disabled, because carrying out all the actions described above will only prolong the download times since no other Delivery Center is known to be available to perform edge relay tasks. When another eligible Ignite Delivery Center is started, it will be automatically detected and managed P2P will be automatically re-enabled to utilize the network more efficiently for upcoming edge delivery scenarios.

Throttling

It is possible to limit the bandwidth consumed by each Ignite Delivery Center during its dynamic edge relaying of content files to other Delivery Centers, specifically to avoid over-burdening any specific host. This limit is also dynamically and centrally re-configurable by the administrator.

Centralized, Dynamic Reconfiguration

There are many parameters specific to the Ignite Delivery Center which can be dynamically reconfigured to further control or "shape" the network and the behavior of the dynamic edge relays within that network, including connection concurrency, router hops, bandwidth consumption, etc. These parameters are further discussed and illustrated in Ignite's white paper Network Control and Optimization.

Security

Since with managed P2P every Ignite Delivery Center on the network may act as a dynamic edge relay, sending content files from its local disk to requesting neighbor Delivery Centers, security policies must be enforced with great care. The built-in security features of managed P2P are implemented in a very straightforward manner, preventing malicious use of the technology.

The following security-related features are implemented within Ignite's managed P2P:

- Managed P2P ensures that no file(s) outside the Ignite-controlled data directory tree will ever be relayed from the Delivery Center. Therefore relayed files are only files that were received from an Ignite Server. Malicious users cannot invoke managed P2P to obtain "random" files from the Delivery Center's drive.
- Files are always referenced by their signature (a 128-bit MD5 digest), so in order to receive an edge relayed content file from another Delivery Center, the receiving side must know this signature. Simply knowing a name of a file is not enough information to either request or receive it.

Summary

Ignite's managed P2P technology gives users "the best of both worlds" – leveraging all the benefits of P2P networks while providing a more secure and reliable means of content delivery. Managed P2P allows Ignite Delivery Centers to establish secure P2P relationships to pull content from each other or from designated localized proxy/caching servers. This unique content redistribution technology enables Ignite users to efficiently and effectively deliver large quantities of content over large, global networks.

To experience Ignite's Content Delivery Solution firsthand, visit our website at www.ignitetechnology.com and click on the "Experience Ignite" link.