

The Impact of Wireless Device Access on Content Delivery Networks

Justin C. Denney and Nicholas J. P. Race

Computing Department
Faculty of Applied Sciences
Lancaster University
Lancaster, United Kingdom
{denneyj, race}@comp.lancs.ac.uk

Abstract—Content Delivery network architectures are initiatives designed to support the effective delivery of continuous and discrete media to end-users. Mobile devices are now capable of exploiting services such as content delivery, but with the protocols governing the Content Delivery Networks designed for wired networked topologies; an assessment of the impact of mobile devices on the network has never been undertaken. Wireless device access causes significant issues in the ability of the media transport system to function efficiently or as intended. This paper will highlight the current Content Delivery architectures available, their level of support for mobile devices and the issues therein.

1. Introduction

Growth in the availability of both domestic and business high bandwidth Internet access [1] has seen the increased popularity of high performance, high quality and reliable content for both entertainment and business-critical applications. Content Distribution Networks (CDNs) represent a recent technology which aims to improve the delivery of content to users, utilising a system of high-speed servers, caches and performance motivated redirections [2]. Recent CDN research has produced a new collection of techniques used to optimise multimedia content delivery, such as patching, dynamic caches and media segmentation.

Despite the deployment of CDN technology, the organisation and distribution of content remains a contentious issue [3]. Recent research in this area has seen overlay networks and peer-to-peer networking emerge as alternative solutions for delivery of content [4], [5]. Overlay networks allow a simpler virtual-network topology to be accessed over the physical network infrastructure. The overlay or peer-to-peer network connects nodes sharing a common interest, such as a specific CDN. Providing a smaller dedicated virtual network allows for more effective discovery of the network topology and sharing of media placement and statistical information (such as the current network conditions).

At the same time that CDN technology is being developed the use of wireless devices such as mobile phones, Personal Digital Assistants (PDA) and web pads is expanding [1], [6]. Mobile phone penetration is at an all time high and telecommunications companies are developing and rolling out faster 3G networks and more complex device software. With every new development mobile users are seeing levels of functionality converging on desktop computers. With improved functionality comes a greater expectation, with users now increasingly expecting to have the same or similar levels of usability from their mobile device as from their networked computers. If they can connect effectively to a CDN on their desktop machine they expect to be able to do similar on their mobile devices running on the latest high bandwidth wireless networks. Competing against this notion, mobile devices have their own set of specific problems including reduced processing power, memory limitations, smaller screens, slower network connections and loss of connectivity while roaming between subnets.

CDNs, overlays and peer-to-peer networks all provide solutions for the delivery of content to static nodes. Critically, however, there is no consideration for the delivery of content (in particular, audio/video streams) to mobile devices or mobile users. This paper studies how current CDNs interact with wireless devices and the impact this has on the content delivery service and the underlying networks efficiency, throughput and performance.

2. Problem Domain

Content Distribution Networks are a subset of technology designed to support the effective transportation of content across a network to an end-device. Content can be anything, but in CDN terms is generally defined as an object that if stored centrally could not be delivered at a satisfactory standard from a single host, given a varying number of requests. Multimedia is an example of such content, with high timeliness and bandwidth requirements. The development of content delivery technology is an ongoing process, but the most common design principals employed in CDNs are described in the following sections.

Section 2.1 describes traditional techniques used in deployed CDNs, Sections 2.2 to 2.4 describe current research initiatives that are more specifically designed to support multimedia streams.

2.1. Traditional Techniques

Traditional techniques used in Content Delivery Networks to achieve efficient content delivery can be categorised into two types, DNS redirection and URL rewriting.

A CDN using DNS redirection maintains the authoritative DNS server for the network. It is termed redirection because when the server receives a request from a client (or another DNS server) it resolves the request to the IP address of a single server (from many servers) capable of handling the request. The choice of server is based upon factors such as availability and network conditions. Redirection allows

the required load to be distributed over a range of servers, with minor informative dynamic changes.

Universal Resource Locator (URL) rewriting is a system whereby the CDN origin server rewrites URL links as part of dynamically generated pages, to redirect clients to different content servers hosting the required object. Examples of objects that would benefit from replication over a number of servers include those with high bandwidth or timeliness requirement. The objects are identified and then their URL can be dynamically modified to redirect clients to the server hosting the object.

2.2. Overlay Networks

Overlay networks are not specifically designed to support content delivery, but do provide an effective transportation platform for explicit media delivery mechanisms and are therefore the focus of much new research in the CDN area. Overlay networks [4] are virtual networks composed of all nodes interested in a particular service. The overlay is an abstracted, application layer network where every node has overlay specific routing tables and neighbourhood sets. At the overlay level a node is only capable of routing traffic to other overlay nodes. The ability to abstract just the overlay nodes over the complete network simplifies the process of locating of all available services. For this reason overlay networks are being used as infrastructure facilitators for content delivery networks. CDNs require large volumes of control traffic such as frequent media availability advertisements, media and patch requests. If routed on the network layer, advertisements have to be flooded to all nodes on the same IP subnet(s) as the node, irrespective of the number of nodes involved in the overlay. Using the overlay routing tables the message will only be routed to other overlay nodes.

Using overlay networks as the underlying routed network with higher level support for content delivery, in the form of ALM and / or Dynamic Caching systems are seen as the way forward for aggregated Content Delivery architecture solutions [5], [7].

2.3. Application Layer Multicast

Application Layer Multicast (ALM) is a media delivery mechanism that is commonly built on top of peer-to-peer networks (overlays are an example of this). It is an effective way of distributing live broadcasts such as video conferences, television broadcasts and radio. ALM is undoubtedly one of the major areas of research employed in CDN systems [5], [8], [9]. ALM forms multicast trees, where each node in the tree is also obliged to connect N other nodes to the same tree. Multimedia streams start at the root node and flow down the tree from parent to children. This evenly distributes the required bandwidth over all the cooperating nodes. Using this formation the server can theoretically support an infinite number of nodes without any load increase.

ALM is designed to only support non time-constrained media sources. All nodes irrespective of when they joined the tree receive the stream from the same point. This limits the type of media a CDN can support if designed using an ALM structure.

Some initiatives have been developed in order to overcome this limitation, allowing time-constrained media (requests are made to and from a specific point in a media source) on an ALM structure. Cooperative networking (CoopNet), [10] is such an initiative. CoopNet uses a standard ALM model for supporting live streams but adds specific time constrained media extensions. Media servers are required to continually monitor their bandwidth usage, serving all incoming requests until N bandwidth is consumed. Adequate service can then no longer be guaranteed and all subsequent requests for direct connections to the server are denied. In this situation the server responds with a list of other nodes that have downloaded at least part of the requested stream and may be willing to service the request. The client is then required to contact each node in a sequential fashion until a suitable server is located.

2.4. Continuous Media Architectures

A large proportion of media accessed over computer networks is time-constrained, (Streams are requested from a specific point in the media source). New technology such as Tivo are also becoming very popular allowing live broadcast media to be stored as time constrained media with the possibility of serving the stored media across a network. The popularity of time-constrained media has inevitably led to content delivery architectures specifically designed to support it. The architectures are designed around techniques attempting to provide the bandwidth distribution characteristics of ALM into a time-constrained media distribution architecture. Self-Organizing Cooperative Caching (SOCCER) [11] is a content delivery architecture specifically designed to support time constrained media and defines techniques including:

Media segmentation: Media is broken up into a number of discrete units, which can be cached without overloading client resources. Specific segments can be requested and transported around the network to support time constrained media.

Dynamic caches: Proxy servers or cooperating clients distributed around the network, capable of serving both time constrained and live streams to clients on the server's behalf. They reduce the load on the network by being able to serve a single incoming continuous stream to multiple clients, even if secondary streams are requested T (seconds) after the primary request initiated the stream.

Patching: When a new client joins an existing stream from a dynamic cache it may be T (seconds) behind the start of the stream, known as the temporal distance. The client caches the incoming stream whilst requesting the missing data and appending (or patching) the incoming stream with the missing data.

Dynamic caching systems are not classically built onto overlay or peer-to-peer structures, but there have been a limited number of new initiatives to merge the two technologies [10]. Overlay structures simplify some of the problematic issues for dynamic systems, specifically media and server advertisement. Dynamic cache systems such as SOCCER have to define complex, expanding Time To Live (TTL)

schemes for advertisements to ensure that they do not flood entire networks with control traffic whilst trying to locate cooperating nodes. With the use of overlays, cooperating nodes can be located with ease.

3. Content Delivery To Wireless Nodes

As mobile devices become more pervasive, the ability for these devices to access a range of multimedia will become ever important. This trend can already be seen in the latest generation mobile phones and PDAs, which are capable of displaying high quality video streams and video conference calls. As their ability to perform these activities is improved, the usage of mobile devices will also change with a trend towards commercialised media streams such as radio, news 24, movie trailers, video phone calls and even full theatrical movies. There is already a large body of work researching the delivery of multimedia sources to mobile devices [12], [13], [14]. Most of these focus on the mobile device itself and consider how it can be adapted to prevailing conditions (i.e. requesting a lower resolution video when the device CPU begins to labour). Other research places the emphasis on the server, which is responsible for adapting media to suit a mobile device [15]. These initiatives do not consider the system as a whole, the interaction of the mobile device with the servers, the network response and the impact on the service.

The most suitable form of media distribution for mobile devices is server based media distribution. Media is stored on, or streamed from large centralised machines designated specifically for this task. There are peer-to-peer architectures allowing a more distributed media architecture, [16] but they are not suited to mobile devices, with their limited memory and limited bandwidth, hosting media is largely infeasible. The following section describes the ability of server based content delivery networks designed for a wired environment to serve the wireless world.

3.1. Traditional Techniques

The traditional CDN techniques as discussed in section 2.1 are not ideally suited to delivering multimedia sources. They are media insensitive and maintain 1 to 1 relationships between source and client, so they do not offer any other quality of service enhancements than the distribution of multiple servers. The normal characteristics of wireless devices such a connection loss during handoff will cause significant problems with this system. A new stream will have to be requested every time the connection is lost, which may or may not be allowed to start from a specific point, depending on the protocols and applications the system supports. Some form of signalling would also be required in order for the server to detect that a mobile device has moved and may be connected to a different server (which again may or may not be supported). Mobile IP may be considered a solution but in reality presents further problems. The incoming stream will be lost but not terminated, the device will experience a loss of data for a short period, with no mechanism for which to replace the missing data.

Traditional techniques are ill-suited to mobile multimedia environments and are overshadowed by recent initiatives; the remaining sections will focus on the more recent research proposals.

3.2. Live Media and Mobiles

Application Layer Multicast provides an efficient method of distributing live streams across many cooperating nodes. In order to achieve this, the ALM system relies on all contributing nodes having enough bandwidth to both send and receive a stream in equal quantities [5]. Bandwidth is a big problem area in the mobile arena, as wireless bandwidths are still limited and commercial wireless networks are expensive to use. Forming complex ALM trees where mobile nodes are required to re-send incoming data to one or more children is unrealistic and expensive in terms of bandwidth and the processing capabilities of the mobile device. Charging a user for the service will become extremely difficult as it would be unreasonable to expect a user to pay upload costs when they are serving other users of which they will be largely unaware.

Mobile devices will not only experience problems themselves, but may also effect other nodes in the ALM system, any mobile roaming between networks is likely to cause blackouts¹, when switching network, which will affect the devices children (nodes the device is forwarding data to) and children's, children etc. If there are a number of mobile devices on the same tree branch, the aggregation of the blackouts would cause the stream to stop for significant periods of time and generally become intermittent and completely unacceptable. Impact on other nodes also comes in the form of quality loss. When a mobile device is able to join an ALM session, it will only be able to receive / send data at a lower bit rate, again due to bandwidth and processing limitations. This limits its children and their children to this low bit rate, even if local resources would allow a much higher quality stream to be received.

Live media in mobile environment will inevitably become available in the form of direct peer-to-peer connections or ad-hoc connection, such as video calls (set up like a phone call between two devices). The effect that mobility has on this situation largely depends upon the class of wireless network the mobile devices are using. Telecommunications networks are designed to support mobility by assuring that a mobile device never changes address once it has established a primary data transfer, and achieves this by routing all information through the same gateway node (GGSN) no matter where it moves, even abroad. This does remove the problems of handoff, but also reduces the routing flexibility of the network. It does provide an intermediate proxy called a SGSN that is always located topologically close to the device as it moves, but this is not as effective as having a dedicated network such as a CDN. Devices running on wireless LAN are still going to be influenced by the problems of handoff and address changes.

Value added services is another area that will see growth, just as it has seen growth in standard telecommunication services such as caller id and call waiting, allowing a greater flexibility in your call. Users will be looking for similar value added services

¹ The short period of time during which a mobile node moves between subnets and can no longer send or receive information (also known as a handoff).

from their ad-hoc video call system, such as the ability to conference between more than one participant. The “Add user X into current conversation” feature. This and other similar scenarios require a flexible system that is not provided in the standard network. Essentially you are creating a bi-directional ALM tree, with exactly the same problem. A mobile node in a Many-to-Many conversation will need to upload their own stream to $N-1$ nodes as well as receiving $N-1$ streams, these capabilities are not common in mobile devices.

3.3. Dynamic Caching and Mobile Devices

Supporting time-constrained media is probably the most challenging. The dynamic caching system works exceptionally well for distributing bandwidth across a number of dedicated nodes, reducing both network and server stress. In many ways the dynamic cache system is ideally suited to wireless devices. Patching allows a mobile device to recover missing data lost during handovers and a CDN’s flexibility suits the dynamic nature of mobiles. In reality the introduction of wireless devices into a dynamic caching system causes numerous problems, leading to the eventual collapse of the system.

3.3.1. Quality Adaptation

The varying abilities of mobile devices have led to initiatives designed to provide wider choices of media stored on servers to suit a wide variety of requirements [15]. An informed choice of the most adequate media can then be made that does not overload available resources (bandwidth, processor, display). Transcoding proxies are introduced that take high quality media from a server and transform it into the required format for any given request [17], [18]. This can be done on-the-fly by the proxy or by storing multiple layers of the same media, where each layer represents a different quality. This can improve the mobile devices situation, but only allows 1 to 1 relationships between proxy and client, transcoding is a somewhat older research area than overlays and dynamic caching, so the combination of the two into a single system is yet to be fully realised. Streams in dynamic cache environments are relayed through multiple nodes, each capable of serving this single stream to multiple clients. The relationship of transcoding in a dynamic cache system is not yet clear, each relay node needs to be capable of serving a stream at all the various qualities, irrelevant of the quality being served by its host, which maybe the server or another relay node. The caches need to be able to do this without wasting bandwidth by receiving a primary stream of higher quality than that being served to clients and ensuring that it is feasible in terms of the required bandwidth to serve the request and computationally feasible to transcode the stream to the requested quality, not a meagre task for high quality streams.

3.3.2. Network Component Failure

One of the most common occurrences for a mobile device is a handoff. When a device moves between IP subnets it will have all of its network streams terminated and be unable to reconnect for a brief period until the device is able to obtain a new IP address. When a handoff occurs it is due to the motion of the mobile device, but on the overlay layer, the node will simply observe a loss of incoming packets. The overlay software will interpret this as a scenario caused by the failure of an internal overlay node (dynamic cache). In the event of an observed internal node failure the overlay reacts in different ways. Firstly the failure of the internal node is reported to all other nodes, effectively terminating the use of the internal node in any further operations. The mobile node will then have to request a new stream; this is usually accomplished by connecting directly to the primary server since failure of an internal node is not seen as a frequent event and connecting to the server is a low cost operation.

The internal node failure routines will cause a rapid degradation of the network, when a mobile device enters the dynamic cache system it will function correctly until a handoff occurs. When a handoff occurs, the internal node the device was connected to will be considered failed and removed from all internal-node lists on the network. The result to nodes already receiving a stream from the internal node, now signalled as failed is disputable since the scenario of an internal node, indicated as failed in spite of active traffic has not been investigated and largely depends on the overlay and client software. Possibilities include the removal of the failed node from its routing list but continuing to receive the stream. It may consider it failed and request a new stream causing two identical streams to be received at the same time; possibly leading to corruption of data buffers and certainly reducing network performance. Upon reconnection to the CDN the node will then connect directly to the server increasing its load by one stream, we can therefore state that:

*Per mobile handoff > (Server load increases by one * stream bandwidth & the number of internal nodes is reduced by one)*

A CDN protocol could be designed in such a way as to limit the impact of a handoff by allowing a device to re-enter the dynamic caching system. Whilst this should reduce the load to the server significantly, the approach is not without its problems, the node will be under the jurisdiction of another dynamic cache since it has already determined that the original cache has failed. The new cache may not be serving the same stream, which will require a request for a new one, consequently increasing the load on the network. If the cache is already relaying the stream, it will have a different time signature than the one required. The best scenario in this case would be the stream just happens to be incoming at the exact same time it was lost before the handoff, this is however highly unlikely. In any other case the incoming stream has to be serving the stream at a time position T (*Time Difference*²) greater

² Time difference is the timestamp from the last renderable frame to the first available frame in the incoming stream.

than where it was lost, meaning the incoming data has to be buffered, increasing buffer size and requiring a patch.

This problem could again be alleviated by the redesign of the CDN protocol as to stop the mobile node from assuming an internal node has failed, however this again causes further issues. If the mobile node was able to reconnect to the original cache, the mobile will be lacking *H - Handover Time*³ of the stream, requiring a patch for the missing data. Patching due to handoff would cause significantly increases in the load to the server. The increase of *T* and *H* allows us to state the following about the buffer per handover:

$$Buffer\ Size = (T + H) * Stream\ bps$$

If the old cache is not invalidated the mobile device may fall under the jurisdiction of a new dynamic cache. The new cache may be able to obtain and be able to serve the original stream, forwarded from the previous cache. However after several new caches are acquired, all forwarding the stream, the data path will become seriously degraded. If the new cache can't re-connect to the original stream a new stream is again required.

3.3.3. Device Buffer Problems

If a device handoffs more than once while receiving the same stream, a new set of problems emerge as the device may already have buffer space from a previous handoff. This will require further insight into buffer organization, because the mobile may currently be using its primary buffer, whilst needing to buffer the new incoming stream into the same buffer, which it is already using. After which a patch is required that needs to be processed before the new stream. In the worst case scenario a mobile device could go through two handoffs in short succession, meaning that the device will have filled its primary buffer and a patch buffer after handoff one, which are then not completely used before a second handoff occurs. It will then have an additional primary stream to buffer and an additional patch to buffer, with a complex playback order see Fig. 1. If the CDN is based on segmentation, then the mobile may have only partially received a segment when you loose connectivity. The device will need to re-request the segment to get the missing data but may not be able to simply replace the buffer, as it may be in-use. The segments will have to be merged in some fashion, a concept that is not common in segmentation techniques.

	Primary Buffer	Patch Buffer
1st Handover	Fourth	Third
2nd Handover	Second	First

Fig. 1. Playback order required in a multi-buffer, multi-handoff scenario.

³ Handoff Time is the duration from the last byte received to the first byte received after a mobile movement event that does not require patching.

The increase in buffer sizes due to handovers and fast multi-handovers will cause serious problems for the wireless world. The limitations and expense of memory in mobile devices mean it is unreasonable to assume that the buffering approach is the most suitable for use with mobile devices. The size of memory is limited and when that memory is filled the accessing program will become unable to function correctly.

3.3.4. Existing mobile solutions

Mobile specific protocols such as Mobile IP [19], HAWAII [20] and Cellular IP [21] have been designed that solve the issue of a mobile device moving and ending all networked connections. Mobile IP is based on the network layer and uses a proxy device known as a Home Agent which can continually forward data from any node to a registered mobile node no matter where it is located. After a movement event the mobile node updates the Home Agent, allowing it to forward data to the new location. Mobile IP and other similar protocols could be used to prevent the handoff disconnection issue. The mobile device would still lose the stream for the duration of the handoff but it will not be necessary to re-request the network stream. Predictably there are still problems if the handover takes too long to complete, the node will still assume the internal node has expired and request another. The device would then be receiving two almost identical streams when the home agent catches up and forwards the stream, causing related buffer problems. The internal node failure routines could be disabled, thereby removing the problem. This would however require a node to be mobile aware, as well as having strict timeout requirements met to ensure genuinely faulty internal nodes are removed, but not removed by accident.

Mobile IP by its nature ensures mobile nodes maintain the same connection for its entire lifespan. In a mobile CDN environment this is undesirable and inefficient as the mobile device may roam into the jurisdiction of a new dynamic cache, which is capable of performing a more efficient service, but because a static connection exists the node will maintain it. This reduces the benefit of having a distributed cache system. The connection to the original server is never released; optimal caches may become available but are never exploited.

3.3.5. Simulations

In order to assess the scale of the problem, a set of simulations was undertaken using Network Simulator [22] ns-2.1b7. The simulations use an aggregation of common practices used in current overlay and CDN architectures [7], [9], [11]. The simulations were performed using varying numbers of clients, requesting an identical high quality streaming media source. The simulation was carried out in both static and mobile environments. In the mobile simulations the multimedia nodes accessing the service handoff only once whilst moving to a different location on the overlay, blackouts were limited to a maximum of 1 second. In a static configuration the multimedia nodes accessing the service never move (remain static); the topology of the network never changes. The simulation uses an overlay network consisting of fifty core nodes, and is accessed by up to ten high-speed clients.

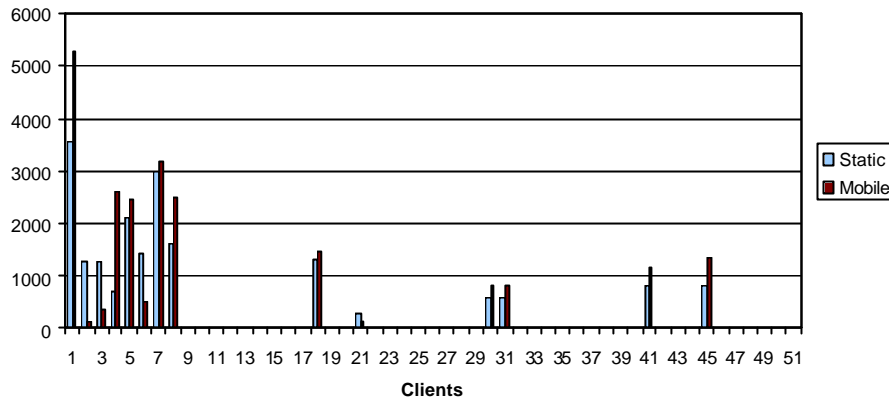


Fig. 2. Volume of traffic on outbound links per node for both multimedia clients that are mobile and non-mobile. Node 1 is the primary server, 2 to 6 are relay nodes and 7 & 8 are primary routers.

Fig. 2 shows the aggregation of a series of simulations run with varying numbers of clients, in both static and mobile configurations. Specifically it shows the volume of data outbound from each node in the core overlay network. In a static configuration it can be seen that the bulk of the bandwidth is distributed evenly over the server, relay nodes and core routing nodes (nodes one to eight). When simulated in the mobile configuration the bandwidth distribution breaks down, the server produces half as much bandwidth again and the relay nodes are either barely utilized or having to serve up to three times as much bandwidth. Outside of the server and relaying nodes, onto the overlay itself the bandwidth requirements are still higher in the mobile scenario, due to patching and wasted bandwidth when handovers occur, until the server recognizes that the stream is no longer being received and terminates it. The mobile bandwidth throughputs are far more irregular, making congestion a problem. This irregularity also stems from the topology of wireless networks, since mobile devices gain access from a gateway or base station, all traffic going to or from those mobile devices will take approximately the same network route and all data will have to pass through the same gateway. This will cause a congestion hotspot at the mobile end of the connection, and the degradation of the dynamic caching system will cause congestion at the CDN side of connection.

It can be seen from Fig. 2 that the bandwidth requirements are higher in a mobile configuration; Fig. 3 shows a more detailed analysis of the overall bandwidth required with a varying number of clients. In the first few tests the mobile configuration shows less bandwidth being consumed than in a static configuration. This is due to the fact that the streams are not being routed through the dynamic caching system for part of their lifetime, having a direct link to the server will consume less bandwidth. This trend however does not last, as patching and the general break down of the system means that the network is having to throughput more data in order to keep up. The mobile configuration is producing 1000Kbps more throughput than a static configuration to serve just five clients.

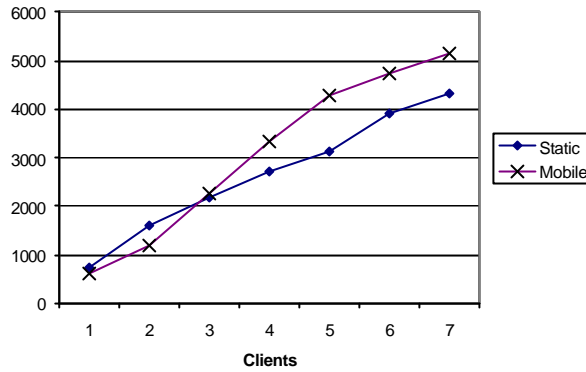


Fig. 3. Overall network Stress for both static and mobile scenarios.

A more practical assessment of the stress of the server is shown in Fig. 4 graphing the number of maximum simultaneous connections the server has to process⁴ and the total number of served connections in both mobile and static configurations. The dynamic caching system in the static configuration performs exceptionally well, after the initial five nodes have joined the stress on the server never increases and remains serving just the relay nodes. The mobile configuration performs very differently as the load to the server increases rapidly in relation to the number of client nodes accessing the service.

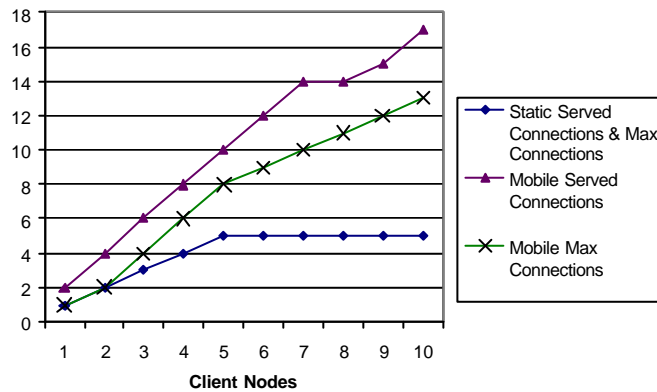


Fig. 4. Server Statistics⁴ showing maximum simultaneous connections and total number of connections for both mobile and non-mobile scenarios.

The number of served mobile connections levels-off at seven clients. This is when the system starts to labour, connections are broken, patches not delivered and requests

⁴ Although patches are essentially the same as a regular multimedia stream for the purposes of these statistics, patches are not classed as connections.

lost. The increase to the load of the server, can obviously not sustain a linear increase in its served connections and eventually reaches a level in which it will no longer be able to handle to amount of stream requests and associated bandwidth. At this point the server will collapse and packets will be dropped from the server. Fig. 5 shows the volume of dropped packets compared with the number of clients requesting streams. A standard dynamic caching system in a wired environment is able to transport all streams requested without dropping a single packet. In comparison when mobile nodes are introduced the network declines very rapidly, starting reasonably low but increasing exponentially. The server simply becomes unable to cope with the volume of requests. Eventually reaching a level at which almost all packets are dropped, and the number of dropped packets levels off.

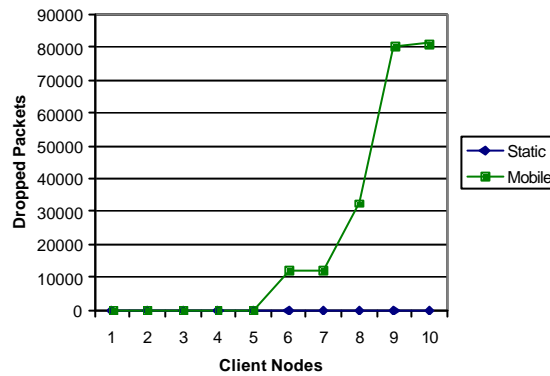


Fig. 5. Packets dropped by the network in comparison to the number of multimedia nodes requesting multimedia streams.

The simulations have shown the extent to which the CDN environment is unable to cope with the dynamic properties of wireless users. The bandwidth is not evenly distributed around the core relaying nodes and causes a general breakdown in the server’s ability to process ever more requests. An overall greater level of bandwidth is consumed on fewer links, leading to massive congestion at this point it inevitably leads to loss of packets as the network is stressed beyond breaking point. The loss of data of course means users are not receiving them, the timeouts or the level of service will fall to a level that becomes unacceptable and the clients will disconnect and request a new stream. The server will eventually drop enough streams to bring the situation back under control, but as soon as the network becomes usable again, the server will receive all of the requests from previously downed clients, leading back to a state of congestion. This system will fluctuate between these states, being unable to provide even an adequate service.

4. Conclusion

This paper has investigated and demonstrated that there is a need for support of mobile devices in CDN architectures. Current CDN architectures provide mechanisms that can conflict with the attributes of a mobile device, degrading the overall performance of the network when exposed to mobile clients. The primary characteristic of a mobile device is that they are highly dynamic, and it is this very characteristic that causes the network to collapse:

- Dynamic in their point of entry onto a network. Mobile nodes move freely between networks, changing the network's topology as they go. The network will have a number of mobile gateways or base stations, but the number of nodes connecting to each is unknown.
- The characteristics of data transmissions. Nodes freely moving between network links will experience highly dynamic changes to their data path, such as loss, latency, quality and throughput.
- Abilities, the abilities of mobile devices are very different from normal nodes. Reduced processing power, memory, screen etc. The differences between different mobile nodes can also be extreme, e.g. between a mono screen PDA and a GHz, large screen notebook.

This is a fundamental change from a standard network, which is very static. Nodes do not move, the topology of the network does not change; the abilities of all the nodes are highly similar.

Current research initiatives attempting to achieve effective multimedia delivery on wireless networks focus on the mobile device or on the servers. They provide mechanisms that adapt the device to prevailing network conditions or try to provide a greater range of media to allow the device a more suitable choice. Whilst none of these initiatives have looked at the network as a whole, overlay technology allows us to do this without being overwhelmed by the complexity of the network. Using overlay and taking CDN technology it should be possible to produce a highly robust, dynamic, state aware content delivery service that can adapt rapidly and pro-actively to the wide range on network dynamics experienced when dealing with wireless access, allowing an entire network to be state-aware and providing key mechanisms that allow the data, network, server and devices to change. Placing this under the umbrella of a single service means the effect each has on the other will not be an unknown quantity. Tying in a unified mechanism for cooperating with existing mobility mechanism or at a minimum an effective way of determining their current state or existence is also preferential. This will not only allow a more effective multimedia delivery service but also allow network operators to change the characteristics of the service depending on requirements.

This paper has highlighted issues with current CDNs and overlay architectures and their support for mobile devices. In doing so has clarified the issues that need to be resolved in order to design the next generation wireless capable content distribution network platforms.

Acknowledgements

The work described in this paper was supported under the Mobile IPv6 System Research Laboratory project (www.mobileipv6.net) – funded by Cisco System, Microsoft and Orange. The author would like to acknowledge their support of this work.

References

1. Balmori, M. "Federal Communications Commission Releases Data on High-Speed Services for Internet Access" (July 2002).
2. Krishnamurthy, B., Wills, C., Zang, Y., "On the Use and Performance of Content Distribution Networks", Proceedings of ACM SIGCOMM Internet Measurement Workshop (November 2001) pp. 169-182.
3. Byers, J., Considine, J., Mitzenmacher, M., Rost, S. "Informed Content Delivery Across Adaptive Overlay Networks", In SIGCOMM (2002) pp. 47-60.
4. Rowstron, A., Druschel, P., "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems". IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany (November 2001) pp. 329-350.
5. Castro, M., Druschel, P., Kermarrec, A-M., Nandi, A., Rowstron, A., Singh, A., "SplitStream: High-bandwidth content distribution in a cooperative environment", IPTPS'03, Berkeley, CA (February 2003).
6. CyberAtlas, "Wanted: One Wireless App Users Simply Cannot Live Without", May 2001.
7. Hau, K., Tran, D., Villafane, R., "Overlay multicast for video on demand on the Internet", University of Central Florida Orlando, U.S.A, SAC (2003).
8. Deshpande, H., Bawa, M., Garvicia-Molina, H., "Streaming live media over a peer-to-peer network", Stanford University, CA, USA (April 2001).
9. Tran, A., Hau, K., Do, T., "ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming", Technical report, CS-UCF (2002).
10. Pandmanabhan, V., Wang, H., Choi, K., Sripanidkulchai, K., "Distributing streaming media content using cooperative networking", in NOSSDAV, Miami Beach, FL, USA (May 2002).
11. Hofmann, M., Ng, E., Guo, K., Paul, S., Zhang, H., "Caching Techniques for Streaming Multimedia over the Internet", Bell Laboratories Technical Report, BL011345-990409-04TM (April 1999).
12. Jeon, Won J., Nahrstedt, K., "QoS-aware Middleware Support for Collaborative Multimedia Streaming and Caching Service", Microprocessors and Microsystems, Special Issue on QoS-enabled Multimedia Provisioning over the Internet, Elsevier Science (December 2002).
13. Yuan, W., Nahrstedt, K., Adve, S., Jones, D., and Kravets, R., "Design and Evaluation of A Cross-Layer Adaptation Framework for Mobile Multimedia Systems", in Proc. of SPIE/ACM Multimedia Computing and Networking Conference (MMCN'03), Santa Clara, CA (January 2003).
14. Chen, k., Shah, Samarth H., Nahrstedt, K., "Cross-Layer Design for Data Accessibility in Mobile Ad hoc Networks", Journal of Wireless Personal Communications, Special Issue on Multimedia Network Protocols and Enabling Radio Technologies, Kluwer Academic Publishers, vol. 21 (2002) pp. 49-75.

15. Fox, A., Gribble, S., Chawathe, Y., and Brewer, E., "Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives," IEEE Personal Communications (1998).
16. Xu, D., Hefeeda, M., Hambrush, S., Bhargava, B., "On Peer-to-Peer Media Streaming", In Proc. of International Conference on Distributed Computing Systems (ICDCS'02), Vienna, Austria (July 2002) .
17. Tang, X., Zhang, F., and Chanson, S. T., "Streaming media caching algorithms for transcoding proxies". In Proceedings of the 31st International Conference on Parallel Processing (ICPP), IEEE Computer Society Press (August 2002) pp. 287-295.
18. Corner, M., Noble, B. D., and Wasserman, K. M., "Fugue: Time Scales of Adaptation in Mobile Video". Proceedings of the SPIE Multimedia Computing and Networking Conference (MMCN '01), San Jose, CA (January 2001).
19. Johnson, D., "Mobility Support in IPv6", draft-ietf-mobileip-ipv6-16, (2002).
20. Lucent Bell Labs, "IP micro-mobility support using HAWAII", draft-ietf-mobileip-hawaii-01, (2000).
21. "Cellular IP", <http://www.comet.columbia.edu/cellularip/>
22. "The Network Simulator - ns-2", <http://www.isi.edu/nsnam/ns/>