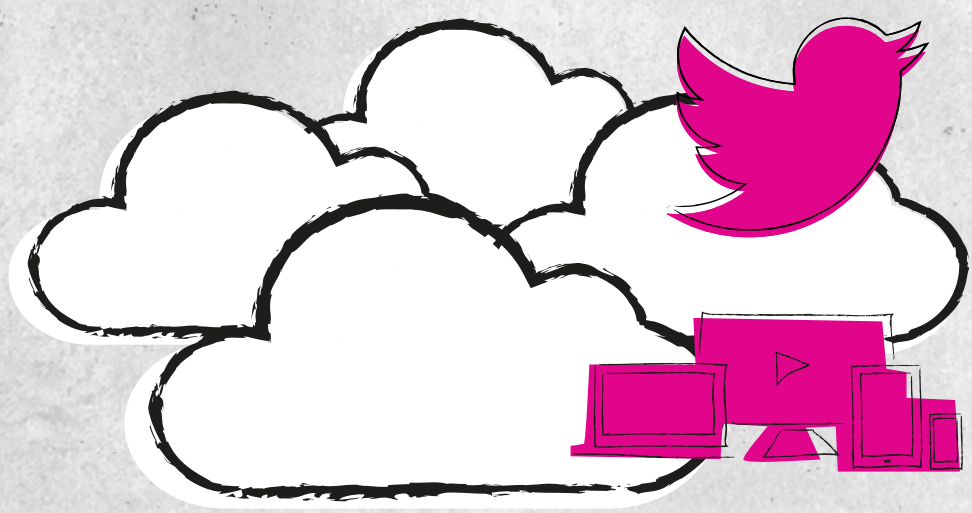
WHITEPAPER

edgeware



TV THAT'S FASTER THAN A TWEET?

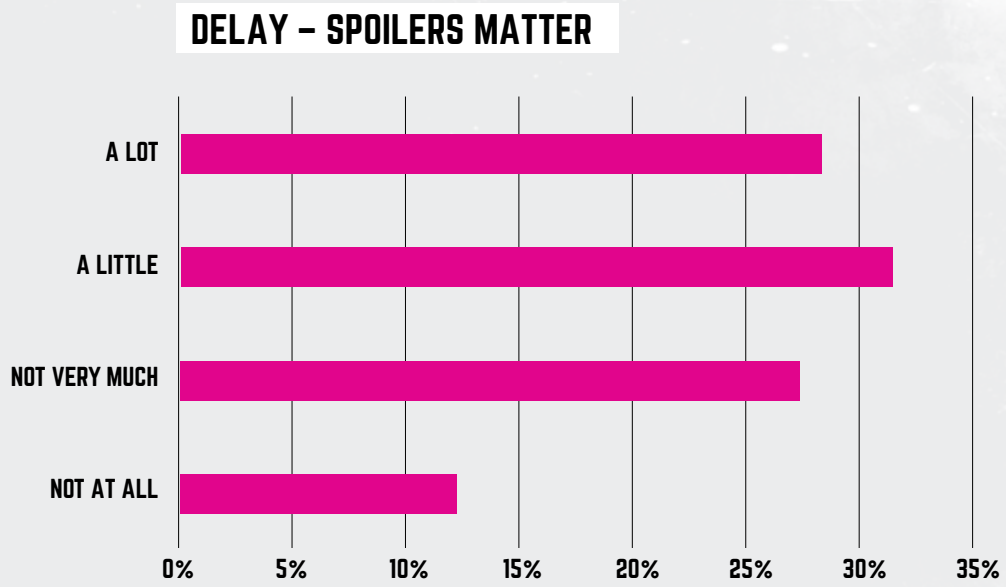
What TV latency is acceptable to avoid social media spoilers?

TABLE OF CONTENTS

WHAT TV LATENCY IS ACCEPTABLE TO AVOID SOCIAL MEDIA SPOILERS?

With many high-value live TV events now being watched online, the additional delays introduced by an IP delivery system can have a significant impact on the viewing experience, particularly for live sports. If latencies are too high, compared to satellite, terrestrial or cable delivery, then viewers are exposed to social media posts, texts and on-line commentaries that can contain spoilers for their game. Learning of a goal scored in the soccer match before you've seen the free-kick awarded is no way to experience content which is often acquired at huge expense by the broadcaster.

In a recent research poll* of over a thousand viewers in the US and the UK, 60% of viewers said that spoilers caused by transmission delays mattered to them ('a lot' or 'a little') and most (52%) wouldn't tolerate delays above 15 secs.



** Survey carried out by YouGov in March 2017, and commissioned by Edgeware*

Millions of sports viewers are already participating in social media or are in contact with their friends during a game. During Super Bowl 51, Facebook reported 64 million people posting 240 million interactions, alongside 27.6 million tweets and 150 million game-related postings on Instagram. And recent research from Nielsen found that more than 84% of American tablet and smartphone owners use their devices while watching TV.

THE PROBLEM

Unfortunately, a lot of internet TV is delivered using general-purpose CDN services, which can often introduce delays of more than 30 secs. These services use infrastructure initially designed to distribute software files for PC's and smartphones, where latency wasn't an important factor. But for premium live sports TV content, then low-delays are paramount.

SCOPE

First, we should note that we are just looking at the relative delay between TV delivered over an IP network vs. conventional distribution via satellite, cable or terrestrial broadcast. There are additional delays incurred in the production of the live TV feed after the capture of the image in the camera, but these will be equal in both distribution scenarios. In an extreme case, the viewer may be watching TV within earshot of the sports venue, but for the scope of this paper these relatively short production delays are not discussed here. We will also not consider the TV viewer watching within earshot of a neighbour watching on cable or satellite. This viewer may hear an audible reaction from 'across the fence', but will not be exposed to the absolute spoiler of a post on social media.

There are also variations in delay across the different conventional broadcast methods, as satellite will usually incur longer propagation delays, caused by the longer path to and from the satellite, despite transmitting at the speed of light. An IP -based TV service taking its feed directly from the stadium would actually have a starting advantage over the satellite feed, but for the purposes of this paper we will assume that all delays caused by IP distribution technologies are entirely over and above those incurred by satellite or other broadcast methods.

WHY THE DELAY?

In considering delays to IP based TV, it is important to understand that the continuous TV stream that appears on screen is actually many short segments of video, each its own file, typically between 2 and 10 secs in duration. This approach allows the TV client to adjust picture quality if available bandwidth changes, to reduce the risk of buffering, and saves an entire movie file being transmitted unnecessarily if the viewer decides to change channel. Leading encoder vendor, Bitmovin recommend using DASH or HLS chunk sizes of around 2 to 4 secs, for example, which is a good compromise between encoding efficiency and flexibility for stream adaption to bandwidth changes.

CAUSES OF DELAYS

The causes of delays to on-line TV fall broadly into two categories:

a) Delays in the origin and CDN systems

An encoded live TV stream has to go through several tiers of processing before it reaches the viewer's device. It must be ingested into the origin system, repackaged into the correct format needed by different device types, and then it is usually encrypted. This processing can add some short delay but the main delay introduced is due to segmentation. You cannot start delivering a segment until you know where it ends, therefore you will introduce at least a one segment length of delay. The only way to address this is to use 'chunked encoding', which is supported with DASH/

CMAF. With chunked encoding you can start delivering pieces of a segment before the whole segment is known, but this is still new technology and not yet well supported by clients.

The data then has to cross a physical IP network, where there are both propagation delays caused by the speed of light and some queuing and processing delays in the switches and routers.

As it enters the CDN, each 2-4 second video segment must be written into memory, and then read out again. In a large CDN there will be multiple layers of cache memory, with higher layers holding more of the popular content than the more distributed local caches. In CDN systems that use generic web caches, and haven't been optimized for video or aren't purpose-built for TV, the files can't be read-out of memory until the entire file has been written in. Writing these files in and out of multiple memories can add more delay into the system.

b) Delays in the client

As we've already mentioned, the TV show is sent in a series of short chunks. The client, or player, will usually wait until it has received a few segments of video before it starts to play. The client behaves this way to overcome any short-term variation in bandwidth and reduce the risk of the screen freezing while the client waits for more data to arrive. For example, if the client waits for 3 video segments of 3 secs duration each, then we have 9 secs of delay introduced 'intentionally' by the client, as a trade-off for a more consistent quality of experience by the viewer.

So, is it possible to reduce or eliminate these delays? In a word, yes – but with some important caveats. Let's take each area in turn:

CAN WE ELIMINATE THESE DELAYS?

a) Reducing delays in the origin and CDN systems

Purpose-built TV systems, like those from Edgeware, can perform on-the-fly repackaging and encryption in near real-time, reducing delays in the origin system. Even more significantly, the files can start to be read out from the CDN cache memories as soon as they have started to be written in. This is a benefit of the system being designed for distributing TV, rather than generic files. In practice a purpose-built TV CDN, rather than a generic CDN service, can deliver TV over an IP network with only sub-second latency, even across multiple tiers of caching. This can represent a reduction in latency from at least 30 secs down to less than 1 second.

We can't do much about the speed of light or other network delays but these are not usually material and can normally still be accommodated in a sub-second delay across the whole CDN.

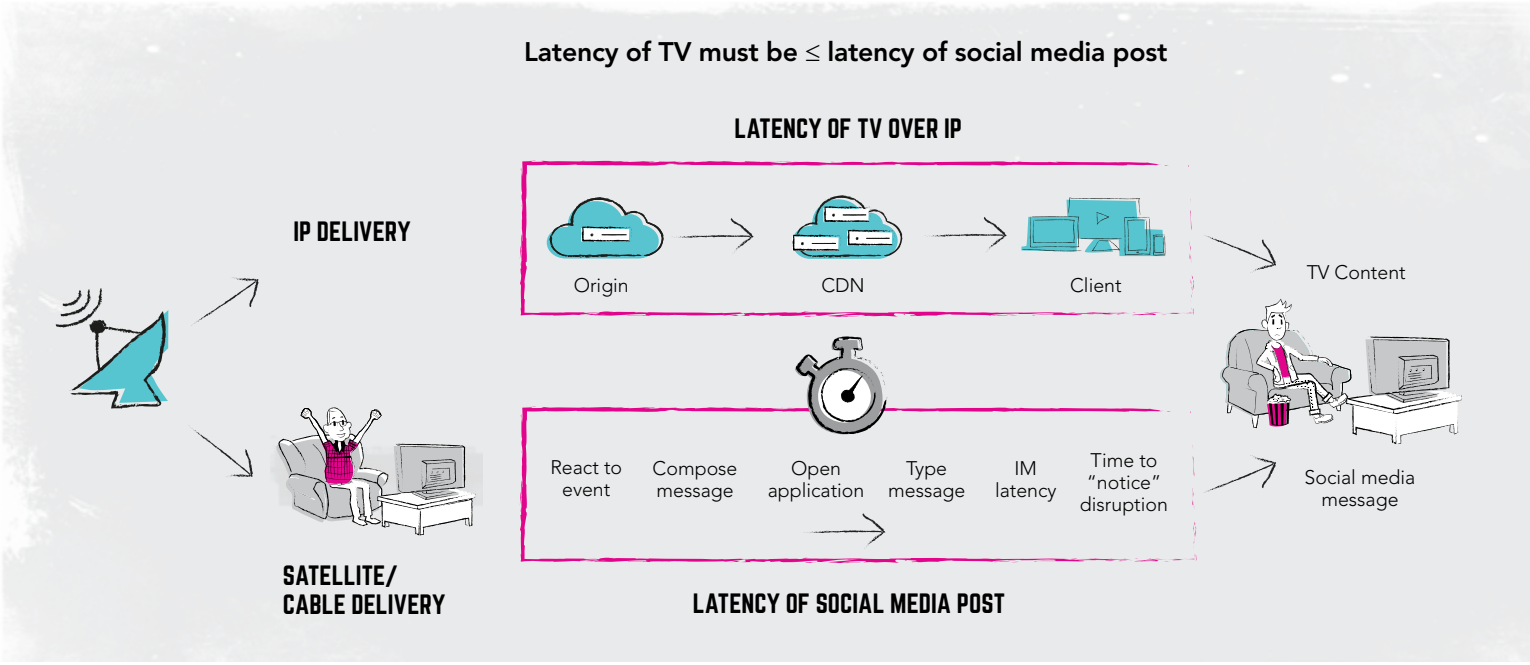
b) Delays in the client

There are also techniques available to sync the client with the transmission, removing the delays introduced by most clients as they wait for one or more segments to download. For niche applications this is worth considering. But it is often not practical to impose a client onto a disparate set of subscribers who are choosing to watch on different types of device, sourced from different manufacturers, all with

different default clients. Most broadcasters and content owners don't want to restrict their service to a specific client, so choose to live with an open client choice as a 'real-world' limitation.

As we've already discussed, in practice, the segment length and buffering within the client may introduce around a 9 secs delay.

So, if we can reduce the CDN latency to less than 1 second, and we must live with an origin delay of around 3 secs and client delay of around 9 secs, the question that we must now ask is, 'will a 13 second delay to live TV result in live sports viewers suffering from social media spoilers?'



MENTAL CHRONMETRY OF SOCIAL MEDIA MESSAGING

Or, to put it another way: 'what's the speed of a tweet?'

So, unless we impose a specific client on every viewer, the one factor that remains critical to avoiding spoilers is the time taken for someone to create, send and react to a social media post. If that time is greater than the delay of the TV delivery system, then we don't have a problem in most use cases. Or, put formulaically:

T_s must be $\geq (L_o + L_{CDN} + L_c)$

Where:

T_s is the **T**ime to create, send and react to a social media post

L_o is the segmentation delay in the **o**rigin

L_{CDN} is the Latency across a purpose-built TV **CDN** system and

L_c is the Latency introduced by a **c**lient buffering typical video segments.

We will assume the segment length is 3 secs, and the client is buffering 3 segments and therefore L_o **= 3 secs** and L_c **= 9 secs**

If the TV CDN is built using Edgeware technology we know $L_{CDN} \leq$ **1 second**

So we are trying to ascertain whether $T_s \geq$ **13 secs**

T_s will include the time it takes to react to an event, compose and input the message, the latency of the messaging service itself, and the time taken for the viewer to be disrupted. Calculating this figure takes us into a field called mental chronometry[1] and fortunately there are a number of existing research areas we are able to turn to in order to answer our question. We expand on these in the appendix to this document: CALCULATING THE 'SPEED OF A TWEET'

AND THE ANSWER IS...

Our calculations conclude that the time to create, send and react to a social media message is ≥ **14.75 secs**

No doubt an actively poised sports viewer with something simple to say could potentially shave off a few seconds, but in most cases this represents the minimum time for a post and they will take longer than this figure.

CONCLUSION

So, the good news is that we can conclude that in nearly all cases, and for reasonable video segment lengths, that:

$$T_s \geq (L_o + L_{CDN} + L_c)$$

Which means we can deliver TV over an IP distribution network in less time than it takes to receive a spoiler via social media.

Of course, this still leaves the critical requirement to reduce the latency of the CDN (L_{CDN}) to around 1 second. But with a well-designed TV CDN, using a purpose-built architecture optimized for video, this is perfectly feasible. In fact, Edgeware has already built over 100 such TV CDNs for its customers around the world.

So now there is no excuse for having your premium sports content spoilt by 'anti-social' media.

Share your thoughts on this topic – connect with us on Twitter!
https://twitter.com/edgewaretv

APPENDIX – CALCULATING T_S – THE 'SPEED OF A TWEET'

We can estimate T_s based on a mental chronometry that is the sum of the following components, each of which we can refine individually:

T_r Event reaction time: Time taken for the message sender to **r**eact to event on screen

T_c Message composition time: Time taken to decide what to write and **c**ompose the message

T_a Device access time: Time taken to open the device and **a**ccess the relevant messaging application

T_i Message input time: Time taken to **i**nput the message

T_L Social media application latency: **L**atency of a messaging service

T_d Receivers reaction time to disruption: Time taken for the message receiver to be **d**isrupted and react to the message

So, our formula expands to ask whether, in the worst case, $T_r + T_c + T_a + T_i + T_L + T_r \geq$ **13 secs**

Let's consider each component of T_s in turn:

T_r Event reaction time: Time taken for the message sender to react to event on screen

A simple reaction time measure is usually defined as the time to motion for an observer to respond to the presence of a stimulus. Audio stimuli take slightly less time than visual but as we're concerned with TV we will assume that it is a visual stimulus that triggers the message. There has been extensive research done onto human reaction times, reporting consistent results. The research[2] we will use measured about 50 million individual reaction times with an average result of 280ms and a median of 260ms. So:

$$T_R \geq 0.25 \text{ secs}$$

T_c Message composition time: Time taken to decide what to write and compose the message

Choice Reaction Time (CRT) tasks require distinct responses for each possible class of stimulus. For example, the subject might be asked to press one button if a red light appears and a different button if a yellow light appears. This area of study has led to Hick's Law[3], one of the few laws of experimental psychology. Hick's Law states that the time it takes to make a choice is linearly related to the entropy of the possible alternatives. The results from various reaction-time experiments, using instruments such as the Jensen Box, seem to confirm that this is the case.

Hick's Law states that **T=b.log2 (n+1)** where there are **n** equally likely message choices and **b** is a constant

CRT has been shown to be around 0.5 secs for a simple binary choice – from which we can derive the constant **b = 0.33**

In our case, the message sender's choices are likely to be more complex than those in simple CRT tests, but we can use Hick's Law as a shortest case scenario. We will assume a simple and conservative set of 10 alternative message choices in reaction to an incident during a live sports event: for example, exclaim, commiserate, gloat, complain, question, compare, reassure, encourage, disparage, hope. So if **n=10, T_c = b.log2 11.** And:

T_c = 3.5 secs

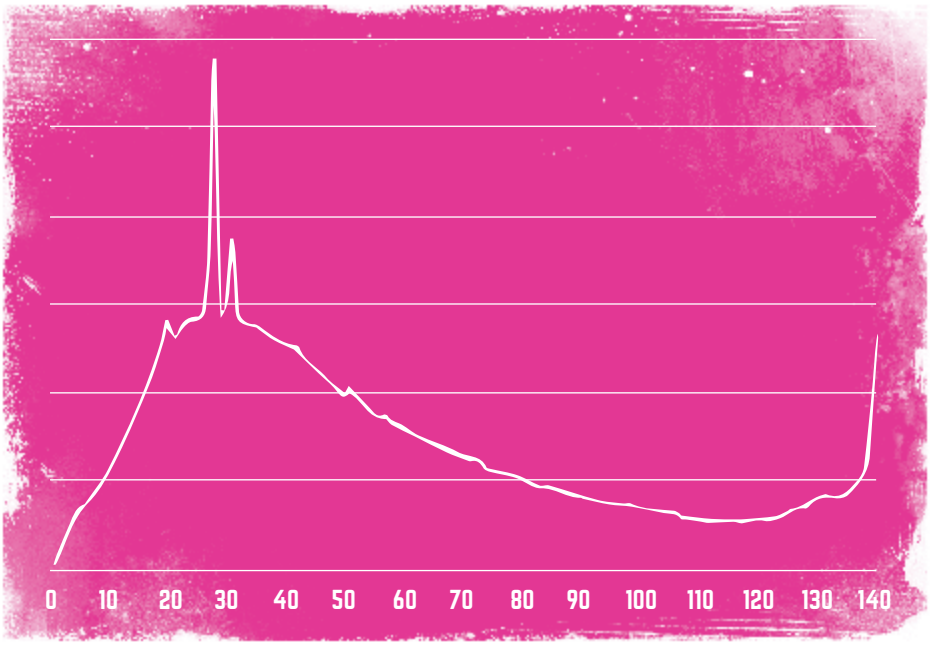
T_a Device access time: Time taken to open the device and relevant messaging application

We have little available research on the typical or fastest times it takes to unlock a device and open the messaging window to compose a social media post. But given that it is possible the viewer is already in dialogue with the message sender, or is a professional on-line commentator, we will assume that the device is already active and the window is open. Therefore, we will assume:

T_a = 0 secs

T_i Message input time: Time taken to type message

Of course, social media posts can vary hugely in length, but most are very short. Twitter employee Isaac Hepworth sampled 1 million tweets[4] and ascertained that the mode (most common) length was only 28 characters.



Number of characters used in twitter posts. Source: thenextweb.com[4]

Note that the spike at the high-end of the sample represents messages that have been forced into the 140 character limit of Twitter, and either truncated at that point or spread over more than one post. To account for the greater majority of messages we will consider the majority of our posts to be only 20 characters long.

Input speeds of 200 CPM (Characters Per Minute) are considered above average[5] on a computer keyboard, which is the likely method of input for sports commentary on-line. Researchers from Google, IBM, Northwestern University, and UC Santa Cruz found that inputting text on a smartphone[6] is 2.74 times slower, at 73 CPM, extending this time for social media posts from friends using a second screen while watching the sports event, who are likely to be using a smartphone or tablet.

Using the faster input figure of 200 CPM, we have the time taken to input a 20 character message:

$$T_i = 20/200 \times 60 = 6.0 \text{ secs}$$

T_L Social media application latency: Latency of a messaging service

Edgeware has measured typical latencies across social media and messaging platforms and found examples to be in the following ranges:

Skype ≈ 1.8 secs

SMS ≈ 2.5 secs

WhatsApp ≈ 1.0 sec

We will take the worst case, so:

$$T_L \geq 1.0 \text{ sec}$$

T_d Message disruption time: Time taken for the message receiver to stop what they are doing and react to the message – 4Secs

A study by Microsoft[7] on instant messaging, and the time taken to react to a message, found little deviation regardless of the type of task being undertaken (drawing, writing, editing etc.). Note the device on which the message arrived (in this case a PC) was already 'to hand', suggesting this is the minimum realistic time. The mean time from receiving to viewing an instant message was 4.0 secs. So:

$$T_d = 4.0 \text{ sec}$$

AND THE ANSWER IS...

So, summing up the elements in the composition, creation, transmission and reaction to a social media message we find that in the majority of cases, the total time, T_s, to be:

$$T_r + T_c + T_a + T_i + T_L + T_r$$

$$0.25 + 3.5 + 0 + 6.0 + 1.0 + 4.0 \geq 14.75 \text{ secs}$$

REFERENCES

1. *Mental chronometry, Wikipedia:* https://en.wikipedia.org/wiki/Mental_chronometry

2. *Reaction Time Statistics, Human Benchmark:* http://www.humanbenchmark.com/tests/reactiontime/statistics

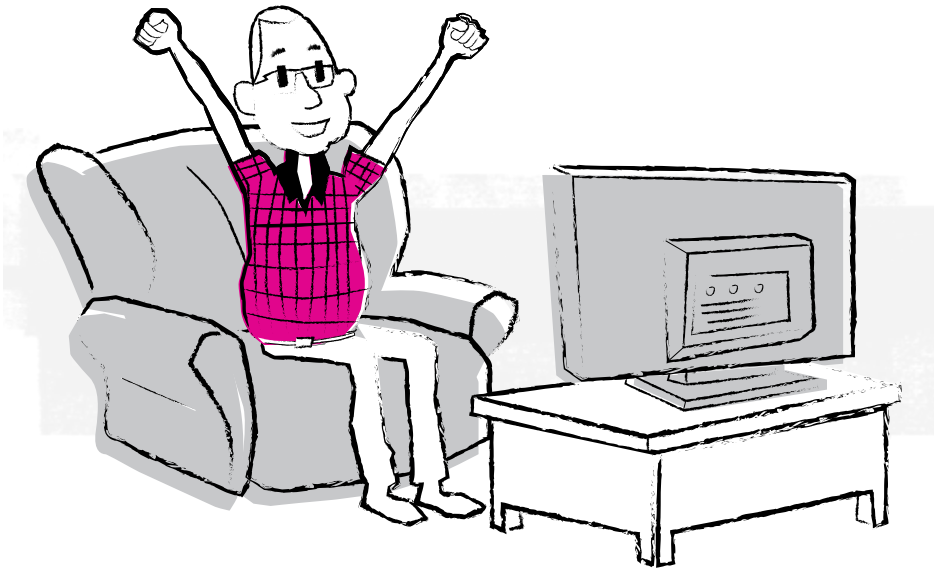
3. *Hick's law, Wikipedia:* https://en.wikipedia.org/wiki/Hick's_law

4. *Interesting fact: more Tweets posted are 28 characters than any other length, TNW:* https://thenextweb.com/twitter/2012/01/07/interesting-fact-most-tweets-posted-are-approximately-30-characters-long/#.tnw_87BvcReB

5. *What Is a Good Typing Speed Per Minute?, Chron:* http://smallbusiness.chron.com/good-typing-speed-per-minute-71789.html

6. *Smartphone Keyboards Vs. Your Productivity, InformationWeek:* http://www.informationweek.com/mobile/mobile-devices/smartphone-keyboards-vs-your-productivity/d/d-id/1102283?

7. *Instant Messaging: Effects of Relevance and Timing, Microsoft Research (Mary Czerwinski, Edward Cutrell and Eric Horvitz).* ftp://ftp.research.microsoft.com/pub/ejh/hci2000.pdf



LET'S MAKE TV AMAZING AGAIN!

WWW.EDGEWARE.TV

STOCKHOLM HQ
Edgeware AB, HQ
Mäster Samuelsg. 42
11th Floor
SE-111 57 Stockholm
Sweden
+46 736 126 840
sales@edgeware.tv

USA
Edgeware, Inc.
200 E. 5th Ave., Ste. 125
Naperville, IL 60563
USA
Toll Free Phone:
+1 888 324-1970
sales_americas@edgeware.tv

MEXICO
Edgeware
Miguel de Cervantes
Saavedra 193-802
Col. Granada
Del. Miguel Hidalgo
Mexico D.F, C.P. 11520
sales_americas@edgeware.tv

HONG KONG
Edgeware
Room 2503, 25/F,
BEA Harbour View Centre
56 Gloucester Road
Wanchai, Hong Kong
Phone: +(852) 3184 0660
sales@edgeware.tv