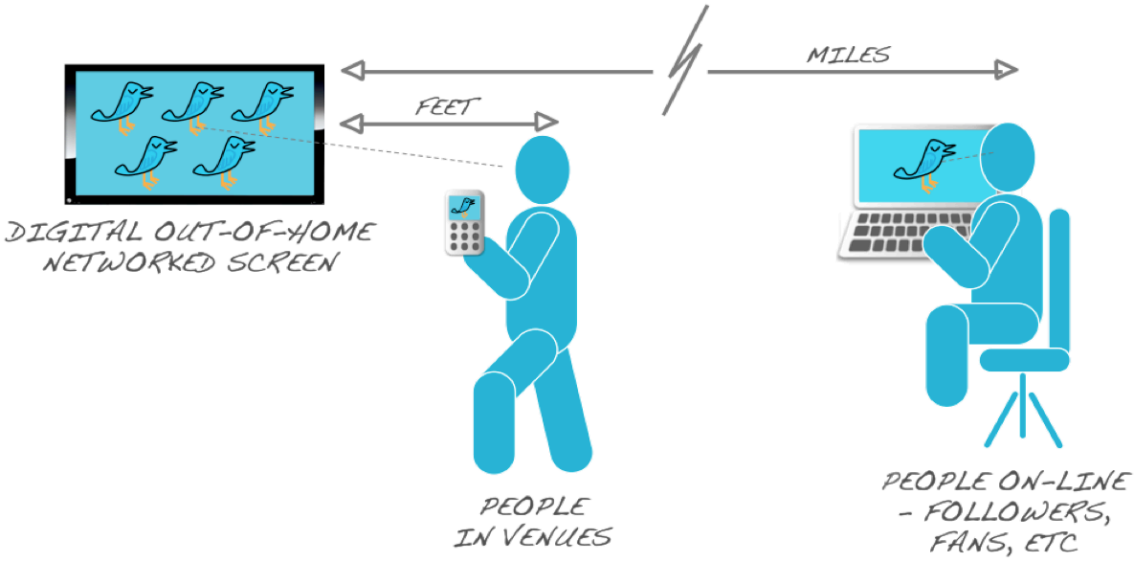
# Twitter on Place Based Screens: Why It's Not So Simple

## Technical Challenges of Real Time Social Media Content Distribution

*Jacob Elder, March 2010*




**Abstract**

*The universal and ubiquitous adoption of Twitter almost overnight has mobilized the whole world into a scrambling gold rush to embrace it as the central tool in every brand's "must have" social media strategy.*

*The Digital Out-of-Home (DOOH) industry stands to benefit enormously from Twitter's explosive growth, but leveraging it for public consumption is not without challenges, as experienced by some of the pioneering campaigns such as the well documented Skittles Twitter experiment.*

*Serving web-based social media in public venues brings a plethora of challenges, from technological hurdles of aggregation and delivery of arbitrary, on-demand social media data streams, to compliance with public decency laws, CAN-SPAM Act and public liability considerations; all while having to facilitate a holistic and engaging user experience to mostly transient, lean-back audiences.*

*The technical challenges range from the well understood battles against Internet spammers and pranksters, to technological hurdles of delivering information in a timely, coherent, and relevant fashion. The biggest challenge of all, however, is the unenviable task of having to moderate in real time voluminous amounts of information that often must be filtered through ad hoc communities topics of interest or venue and context-centric guidelines in order to make it suitable and desirable for public display.*

*This paper examines these issues in great detail.*

A simple Google search reveals a vast array of free-to-use Twitter visualizers. It is tempting to just throw one of these onto a projector and call it a solution for social media. Deeper inspection of these options expose critical challenges for the digital out-of-home (DOOH) implementer.

Twitter makes it possible to build applications and services on top of their user generated content by exposing a rich Application Programming Interface (API). There is a wealth of content available through this interface but reaching it can be problematic.

The most immediately obvious way to get Twitter messages, or "tweets," onto a DOOH screen is to build a Flash application using this API. Flash is the de facto standard platform for interactivity in both DOOH and the broader Internet. Unfortunately, we quickly run into the limitations imposed by the Flash security sandbox.[1] For their own valid reasons, Twitter's cross-domain policy prohibits third-party Flash applications from accessing most of their API.

Thus, DOOH implementers are forced to host a proxy server to relay API requests on their behalf.[2] This proxy host imposes a single point of congestion on the application: users are limited to 150 requests per hour, per IP address. Even a small DOOH network will quickly hit this limit, filling screens with stale messages—or worse, no messages at all. Any requests passing through this proxy will also experience additional network latency and will be susceptible to outages and congestion at the hosting provider.

With Twitter's blessing, it is possible to create special, "whitelisted" user accounts. A whitelisted account is able to make up to 20,000 requests per hour. This is not an SLA—you should still expect to see Twitter's famous "Fail Whale" screens—just a promise not to prohibit access.

Still, polling Twitter for new messages is both inefficient and not likely to produce expected results. The Search API in particular is not meant to be real-time and will frequently return results which are many hours behind what user timelines show. Thus, applications attempting to follow more than a few tags or specific users' tweets will not be able to guarantee when—or if—a given tweet will make it to the screen. A common complaint about applications using this pattern is, "Why is Twitter so slow!?"

Twitter's recently announced[3] Streaming API improves this situation somewhat. With it, developers can make long-lived connections and receive tweets matching certain limited criteria. The default access level allows up to 200 keywords or hashtags and the following of up to 400 users. An application requiring phrase searches (e.g., "digital signage" or "red sox" as opposed to "#weather") would still have to be built on the Search API.

DOOH applications can now theoretically deliver tweets with latency approaching that of text messaging, but our problems are not over. Leveraging the Streaming API requires a sophisticated back-end infrastructure.

An old engineering adage holds true here: "Any tenfold quantitative change is also a qualitative change." Just as a single-screen application is radically easier to execute on than one spanning

ten screens, it would be disastrous to equate success on ten screens and a hundred messages per hour with success on ten thousand screens and thousands messages per hour.

Consider that a successful campaign will by definition generate hundreds of tweets per hour. Many of these are sent by impatient users in sight of the DOOH installation, so these must pass through the system with as little delay as possible. The long-lived nature of the connections involved requires the API proxy to use non-blocking, asynchronous programming techniques in order to remain economical and stable under such a load.

The Streaming API is based on a technique called long-polling, which extends (some would say abuses) traditional HTTP to allow data to move between client and server asynchronously. The principle advantage in this design is broad support—few networks block HTTP—but long-polling doesn't permit altering the search parameters without tearing down and re-establishing the connection. Any tweets which would normally have been delivered while a search term is being added are therefore lost. Twitter does suggest[4] a way around this, but it's not for the faint of heart. Implementing their proposed algorithm while meeting advertisers' requirements for high availability adds yet more complexity.

| | REST API | Search API | Streaming API |
|---|---|---|---|
| **Proxy** | Required | Optional | Required |
| **Authentication** | Required | None | Required |
| **Latency** | Medium | Very High | Very Low |
| **Following Users** | Yes | No | Yes |
| **Phrase Searches** | No | Yes | No |
| **Hashtag Searches** | No | Yes | Yes |
| **Keyword Searches** | No | Yes | Yes |

Each account may create only one standing connection to the Streaming API. Subsequent connections from the same account may cause previously established connections to be disconnected. Excessive connection attempts, regardless of success, will result in an automatic ban by Twitter for appearing to abuse its Terms of Conduct.[5]

Both the Streaming API and the Search API filter out "low-quality" messages. Twitter defines low-quality as having excessive duplicate content such as repeated keywords—what most people would consider spam. On the other hand, Twitter doesn't care if users' tweets may cause offense.[6]

A naïve solution to this problem would be to compare each message to a list of known profanities, but creative users will always find new ways to spell offensive words, even posting offensive words using one letter per message. Campaigns often have more challenging requirements for content curation and restrictive terms of service, such as prohibiting competing brands or diverging from specific brand messaging. To make matters worse, it may not always be legal to display user-generated content in public without users' explicit consent.

DOOH solutions integrating Twitter and other services must include comprehensive safeguards against the wrong message going to the screen. The hypothetical API proxy needs to do much more than relay requests from thousands of DOOH screens to Twitter's servers.

- **Moderation.** An automatic filter can help, but real peace of mind requires live, trusted humans viewing messages in context.
- **Contextual Filtering.** Consider a hypothetical DOOH platform running simultaneous campaigns for Converse and for Nike. These competing brands would have nearly the same criteria for user-generated content, with the one exception that Converse would mandate the exclusion of pro-Nike messages and vice versa. A "first come, first served" strategy can't ensure that all campaigns see an acceptable level of activity. Once a message is received by a given application, it should be available to any and all active campaigns or risk "Ghost Town."[7] The alternative—replicating the entire DOOH platform for each customer—is not a scalable solution and would be prohibitively expensive.
- **Curation.** It is already passé to note that attention is the new currency. When a campaign reaches a critical level of traffic, the moderator's mission shifts from "Is this message profane?" to "Is this message worth the audience's attention?" This is a different kind of problem and requires a different user interface.
- **Multiple Human Agents.** Social media is global, multilingual, and subject to explosive growth. Any DOOH platform using it must enable large teams of moderators to be productive with a minimal learning curve.

The problems presented here are not unique to Twitter. Brands and users are beginning to expect seamless interaction across multiple channels and multiple social networks. Facebook, Foursquare, and Flickr all offer their own APIs. Each will add their own unique challenges for digital out-of-home.

*Jacob Elder is a Systems Architect at LocaModa, a technology company extending interactive media to audiences beyond the web.*
*CONTACT: jelder@locamoda.com*
*TWITTER: http://twitter.com/jelder*

[1] See Twitter's cross-domain policy file at http://api.twitter.com/crossdomain.xml

[2] There are ways around this, such as running an API proxy locally on DOOH screen controllers, but no DOOH platform currently offers this.

[3] http://apiwiki.twitter.com/Streaming-API-Documentation

[4] http://apiwiki.twitter.com/Streaming-API-Documentation#UpdatingFilterPredicates

[5] http://apiwiki.twitter.com/Streaming-API-Documentation#AccessandRateLimiting

[6] "Dealing With F**K" http://bit.ly/DwFk

[7] "Overcoming Ghost Town" http://bit.ly/OcGhT