

# Chatty Applications



## Challenge

The huge growth in the adoption of smartphones has been accompanied by a similar upsurge in the proliferation of applications. A large number of these applications are free and leverage in-application advertising to create revenue opportunities for developers.

However, many of these applications are not optimised for the mobile environment and are set to send frequent updates to servers in the network, even when the application is in the background. The updates may be to provide location data or simply to poll servers for the latest advertisements.

Users are unaware of the traffic and its effects. The impact of this traffic can be compounded as users add applications to their portfolio. The problem is particularly acute in Android-based smartphones, as

there are more ad-funded applications for this OS, but it is common to all.

This activity generates considerable traffic and leads to strain and congestion in the RAN. In extreme cases, the level of activity can even lead to network collapse. Network and RAN based traffic management solutions can only act once the traffic has left the user device, with the result that, by the time traffic reaches the network, the problem has already occurred.

MNOs have to cope not only with expected RAN traffic and demand but also this unexpected and difficult to forecast activity. They must invest heavily in both CAPEX and OPEX to avoid problems caused by such chatty applications as today's network-based solutions cannot solve the problem.

## Solution

The solution is to use the GoS Agent to extend PCEF functionality to mobile devices and manage signalling traffic from chatty applications. By deploying the GoS Agent, a lightweight mobile client in smartphone devices, to detect this traffic, network based policy solutions such as the PCRF can be given the visibility to enable them to manage such background activity. The mobile client is deployed via OTA updates or natively embedded in the device, and connects to existing policy resources via standard interfaces.

Many Chatty Application solutions and battery saving applications turn off radios and suppress access to the network based on simple and crude policies, for example by blocking all background applications or through 'black' and 'white lists' controlled by operators and users. Unlike these, GoS Networks' solution deploys intelligent, patent- pending algorithms that have been proven to reduce signalling load on average by 30% and increase battery life by up to 50%, but at the same time allow all applications equal and fair access to network resources in order to maintain

background information updates. GoS 360° does not negatively impact users' experience.

Instead of allowing unnecessary unlimited traffic from applications that are in the background and not in use, MNOs can set intelligent policies to limit the frequency with which they contact remote servers. When the application is brought to the foreground, it can be permitted to resume its normal signalling frequency.

GoS Networks' GoS 360° solution for Chatty Applications has been deployed in volume, saving a major tier 1 mobile operator up to \$45 Million in deferred RAN expenditure, that releases capacity and improves user experience. It gets intelligently controlled application access to the network without impacting user experience.

With the GoS Agent and the mobile PCEF solution, MNOs have full visibility and control of traffic in the RAN from smartphone devices, enabling them to gracefully manage network capacity.

## Benefits

- Complements other battery saving techniques such as fast dormancy to assure the best possible user experience without resulting in race conditions between handset FD timers and background applications that generate additional signalling traffic.
- Saves money, not just networks. Through our intelligent algorithms we can suppress and avert signalling storms generated by combinations of poor application and OS design that have led to network outages.
- Provide a framework for additional use cases and benefits

## Examples

- Managing signalling load from chatty applications
- Managing RAN overhead and congestion
- Optimising foreground applications
- Ensuring the best user experience